The Bumper Book of .muffins
(The cGENIE.muffin user-manual and introduction to Earth system modelling)

Andy Ridgwell

October 5, 2021
1 Installation, configuration, basic usage .......................... 11
  1.1 Before anything else . . . 12
  1.2 Starting (dozing?) off . . . 16
      1.2.1 Logging in! ................................................. 16
      1.2.2 Downloading/installing the model code .................. 17
      1.2.3 Configuring the code .................................... 17
      1.2.4 Testing the model code .................................. 19
      1.2.5 Brief notes on git and code ............................ 21
  1.3 Running the model ............................................. 23
  1.4 Creating new experiments! ................................... 27
  1.5 Model output ................................................ 28
      1.5.1 Time-slice output ....................................... 28
      1.5.2 Time-series output ..................................... 29
      1.5.3 File naming convention ................................ 29
  1.6 Viewing model output ........................................ 30
      1.6.1 Time-series output ....................................... 30
      1.6.2 2- and 3-D time-slice output .......................... 31
  1.7 Submitting experiment ‘jobs’ ................................ 33
  1.8 ‘Restarts’ .................................................. 35

2 Climate dynamics & experimental design .......................... 37
  2.1 Brrrrrrrrrrrrr – it’s chilly on ... snowball Earth! .......... 39
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Further ideas</td>
<td></td>
</tr>
<tr>
<td>2.2.1</td>
<td>Feedback loop analysis</td>
<td>43</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Continental configuration (1)</td>
<td>43</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Continental configuration (2)</td>
<td>44</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Geothermal heat input</td>
<td>44</td>
</tr>
<tr>
<td>2.2.5</td>
<td>Seasonality</td>
<td>45</td>
</tr>
<tr>
<td>3</td>
<td>Ocean circulation</td>
<td>47</td>
</tr>
<tr>
<td>3.1</td>
<td>Tracing ocean circulation</td>
<td>49</td>
</tr>
<tr>
<td>3.2</td>
<td>Poking the climate beast</td>
<td>53</td>
</tr>
<tr>
<td>3.3</td>
<td>Further ideas</td>
<td>56</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Hosing investigations</td>
<td>56</td>
</tr>
<tr>
<td>3.3.2</td>
<td>‘Anti-hosing’ investigations</td>
<td>56</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Response to transient warming</td>
<td>57</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Ventilation age tracers</td>
<td>57</td>
</tr>
<tr>
<td>4</td>
<td>Climates of Past Worlds</td>
<td>59</td>
</tr>
<tr>
<td>4.1</td>
<td>The climate of the Cretaceous</td>
<td>62</td>
</tr>
<tr>
<td>4.2</td>
<td>The climate of the early Eocene</td>
<td>65</td>
</tr>
<tr>
<td>5</td>
<td>Alternative Worlds</td>
<td>67</td>
</tr>
<tr>
<td>5.1</td>
<td>Generating fake Worlds</td>
<td>68</td>
</tr>
<tr>
<td>5.2</td>
<td>Understanding ocean circulation</td>
<td>69</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Model investigation over-view</td>
<td>69</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Deep-water formation (&amp; baroclinic circulation)</td>
<td>70</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Gateways and barotropic flow</td>
<td>71</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Analysis</td>
<td>72</td>
</tr>
<tr>
<td>6</td>
<td>Fossil fuel CO$_2$ and ‘ocean acidification’</td>
<td>77</td>
</tr>
<tr>
<td>6.1</td>
<td>Exploring the consequences of fossil fuel CO$_2$ emissions</td>
<td>79</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Idealized emissions forcing</td>
<td>80</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Historical (real-world!) emissions forcing</td>
<td>83</td>
</tr>
<tr>
<td>6.2</td>
<td>Further ideas</td>
<td>86</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Assessing the importance of emissions rate</td>
<td>86</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Determining thresholds of environmental impact</td>
<td>86</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Future atmospheric CO$_2$ concentration pathways (‘RCPs’)</td>
<td>89</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Isotopic tracing of fossil fuel CO$_2$ uptake</td>
<td>89</td>
</tr>
<tr>
<td>7</td>
<td>Ocean biogeochemical cycles</td>
<td>91</td>
</tr>
<tr>
<td>7.1</td>
<td>The ocean’s biological pump</td>
<td>93</td>
</tr>
</tbody>
</table>
7.2 Single nutrient (PO₄) cycling and basic controls on biological productivity
7.2.1 Further ideas
7.3 The role of temperature in the biological pump
7.4 Iron co-limitation of biological productivity
7.5 Ocean carbon geoengineering
7.5.1 Iron fertilization
7.5.2 Phosphate fertilization
7.5.3 Enhanced weathering
7.5.4 Modifying spatial patterns
7.6 Further ideas
7.6.1 Impacts to look out for
7.6.2 Other thoughts and suggestions
7.6.3 Further modifications of the biological pump in the ocean

8 Marine ecosystems and dynamics
8.1 Getting going with ECOGEM
8.1.1 Running the model
8.1.2 Viewing 2D time-slice output
8.1.3 Comparing to observations
8.2 Ecosystem configuration
8.2.1 Visualising composite data
8.2.2 Iron limitation
8.3 Increasing ecological complexity
8.3.1 Plankton size classes
8.3.2 Viewing 2D time-slice output
8.3.3 Create your own ecosystem
8.4 Build it up, tear it down
8.4.1 A fully size-structured ecosystem
8.4.2 Ecosystem characteristics
8.4.3 Mixotrophy
8.5 Ecology in ... future oceans
8.6 Ecology in ... past oceans
8.7 Ecology in ... fake oceans!
8.8 EcoGENIE 1.1

9 The land surface
9.1 xxx
14.2 Time-slice output
14.2.1 Frequency and timing of time-slice data saving ........................................... 193
14.2.2 Saving at the experiment end ................................................................. 194
14.2.3 Seasonal/monthly data saving ............................................................... 194
14.2.4 More frequent data saving ................................................................. 195
14.2.5 Less(!) frequent data saving ............................................................... 195
14.3 Time-series output
14.3.1 Frequency and timing of time-series data saving ........................................ 196
14.3.2 Saving orbital insolation ........................................................................ 196
14.4 Data field selection ................................................................................. 197
14.5 Re-start files ............................................................................................ 199
14.6 Useful output
14.6.1 Physics ................................................................................................ 200
14.6.2 (Bio)Geochemistry .............................................................................. 203
14.6.3 Biology/Ecology .................................................................................... 206

15 Introduction to model results analysis ......................................................... 207
15.1 Plotting with Excel ................................................................................... 208
15.2 Plotting with Panoply
15.2.1 Issues with Panoply default settings ...................................................... 210
15.2.2 Basic plots – examples ........................................................................ 211
15.2.3 Difference (anomaly) plots .................................................................... 211
15.2.4 Ocean velocity plots .............................................................................. 212
15.3 MATLAB plotting ..................................................................................... 214
15.3.1 MATLAB 101 ....................................................................................... 214
15.3.2 MATLAB and ASCII (time-series) ....................................................... 214
15.3.3 MATLAB ‘Import Data’ ....................................................................... 214
15.3.4 MATLAB and netCDF (time-slice) ...................................................... 215

16 muffinplot ..................................................................................................... 219
16.1 Installation ................................................................................................. 220
16.2 Time-series plotting .................................................................................. 221
16.3 Spatial plotting ......................................................................................... 222
16.3.1 Basic usage .......................................................................................... 226
16.3.2 Example analysis ................................................................................ 229
16.3.3 Further refinements ............................................................................ 230
16.3.4 Time-series plotting ............................................................................ 232
16.3.5 Sediment model output analysis .......................................................... 232

17 muffindata .................................................................................................... 233
17.1 Data re-gridding ....................................................................................... 234
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.2</td>
<td>Miscellaneous data processing</td>
<td>235</td>
</tr>
<tr>
<td>17.3</td>
<td>Model ensembles</td>
<td>236</td>
</tr>
<tr>
<td>18</td>
<td>muffingen</td>
<td>241</td>
</tr>
<tr>
<td>18.1</td>
<td>muffingen basics</td>
<td>242</td>
</tr>
<tr>
<td>18.1.1</td>
<td>overview of the muffingen software</td>
<td>242</td>
</tr>
<tr>
<td>18.1.2</td>
<td>Installing muffingen</td>
<td>242</td>
</tr>
<tr>
<td>18.1.3</td>
<td>Running muffingen</td>
<td>243</td>
</tr>
<tr>
<td>18.2</td>
<td>Configuring muffingen – overview</td>
<td>245</td>
</tr>
<tr>
<td>18.3</td>
<td>Pre-generated worlds</td>
<td>248</td>
</tr>
<tr>
<td>18.3.1</td>
<td>Conceptual worlds</td>
<td>248</td>
</tr>
<tr>
<td>18.3.2</td>
<td>Idealized-continent worlds</td>
<td>248</td>
</tr>
<tr>
<td>18.3.3</td>
<td>Modern worlds</td>
<td>250</td>
</tr>
<tr>
<td>18.4</td>
<td>Configuring muffin experiments</td>
<td>251</td>
</tr>
<tr>
<td>18.5</td>
<td>muffingen parameter settings – details</td>
<td>263</td>
</tr>
<tr>
<td>19</td>
<td>FAQ</td>
<td>267</td>
</tr>
<tr>
<td>19.1</td>
<td>Installation related questions</td>
<td>268</td>
</tr>
<tr>
<td>19.2</td>
<td>Cluster/queue questions</td>
<td>269</td>
</tr>
<tr>
<td>19.3</td>
<td>Help! My experiment has died ... why?</td>
<td>270</td>
</tr>
<tr>
<td>19.3.1</td>
<td>Sources of initialization error</td>
<td>270</td>
</tr>
<tr>
<td>19.3.2</td>
<td>Carbonate chemistry errors</td>
<td>271</td>
</tr>
<tr>
<td>19.3.3</td>
<td>Other possible geochemical errors</td>
<td>277</td>
</tr>
<tr>
<td>19.3.4</td>
<td>Meaning of specific error and warning messages</td>
<td>278</td>
</tr>
<tr>
<td>19.4</td>
<td>General model experiment questions</td>
<td>279</td>
</tr>
<tr>
<td>19.5</td>
<td>Climate-y questions</td>
<td>282</td>
</tr>
<tr>
<td>19.6</td>
<td>Ocean biogeochemistry-y questions</td>
<td>284</td>
</tr>
<tr>
<td>19.7</td>
<td>Geologic carbon cycle questions</td>
<td>288</td>
</tr>
<tr>
<td>19.8</td>
<td>Data saving / model output questions</td>
<td>289</td>
</tr>
<tr>
<td>19.9</td>
<td>Forcings questions</td>
<td>290</td>
</tr>
<tr>
<td>20</td>
<td>HOW-TO (technical)</td>
<td>293</td>
</tr>
<tr>
<td>20.1</td>
<td>HOW-TO ... get started with cGENIE.muffin</td>
<td>294</td>
</tr>
<tr>
<td>20.2</td>
<td>HOW-TO ... linux</td>
<td>295</td>
</tr>
<tr>
<td>20.3</td>
<td>HOW-TO ... git</td>
<td>297</td>
</tr>
<tr>
<td>20.4</td>
<td>HOW-TO ... install Ubuntu muffins</td>
<td>298</td>
</tr>
</tbody>
</table>
20.5 HOW-TO ... install RedHat muffins 302
20.6 HOW-TO ... install macOS muffins 303
20.7 HOW-TO ... install Windows(!) muffins 306
20.8 HOW-TO ... run and submit muffins 310
20.9 HOW-TO ... re-start muffins 312

21 HOW-TO (experiment) ................................................................. 313
21.1 HOW-TO ... diagnose how the system works 314
21.2 HOW-TO ... screw with the climate system 324
21.3 HOW-TO ... iron (and dust) 328
21.4 HOW-TO ... other (bio)geochemical cycle 338
21.5 HOW-TO ... biological production 346
21.6 HOW-TO ... ecology 350
21.7 HOW-TO ... remineralize ... but differently 351
21.8 HOW-TO ... force the system 354
21.9 HOW-TO ... sediments and weathering 371
21.9.1 ... OMEN-SED ................................................................. 400
21.10 HOW-TO ... geoengineer 404
21.11 HOW-TO ... orbit 411

22 Model Code ................................................................. 419
22.1 Testing 420
22.2 Common Modifications 421
22.3 Uncommon (ill-advised) Modifications 423
22.4 Other Modifications! 428

23 Examples ................................................................. 429
23.1 xxx 430

24 Appendix I – Tables ................................................................. 431
24.1 Data save levels 432
Before anything else . . .
Starting (dozing?) off . . .
Logging in!
Downloading/installing the model code
Configuring the code
Testing the model code
Brief notes on git and code
Running the model
Creating new experiments!
Model output
  Time-slice output
  Time-series output
  File naming convention
Viewing model output
  Time-series output
  2- and 3-D time-slice output
Submitting experiment ‘jobs’
'Restarts'

1. Installation, configuration, basic usage

Stuff to keep in mind:

- *(cGENIE.)muffin* is a model. Models ARE NOT the ‘real World’. (Don’t get confused!)
- The low resolution (for a 3-D ocean circulation model) of the *muffin* model limits its applicability for very short time-scale problems. In configurations not incorporating the PLASIM atmospheric model, there is no atmospheric dynamics or inter-annual variability in the ocean-atmosphere coupled system.
- *muffin* is best thought of as a ‘discovery and exploring’ tool for learning how the Earth system (might) work rather than necessarily as a detailed ‘simulation’ tool.
- It is possible to have fun. (Once the installation pain is over ...)

Aragonite saturation (Ω)

Atlantic streamfunction

Aragonite saturation (Ω)

Atlantic streamfunction (Sh)

Land surface and hydrology

3-D climate and biology Earth system model

© 2011 GENIE
Chapter 1. Installation, configuration, basic usage

1.1 Before anything else . . .

ReadMe

Some warnings and reminders in this manual are repeated over and over and over and . . . over again. Some warnings and reminders are repeated over and over and over and . . . over again. This is because you will forget immediately each time! ;)

Software version naming conventions

You will be using the current version of the cGENIE Earth system model, code-named ‘muffin’ (if Apple can have ‘Leopard’, ‘Lion’ etc., I can haz a baked goods version naming convention). The documentation may not be fully consistent in this respect . . . and you may need to translate occurrences of e.g. a directory named ‘cgenie’ to ‘cgenie.muffin’. For brevity, the cGENIE.muffin model will be referenced by just ‘muffin’.

linux . . .

The (cGENIE.)muffin model currently natively compiles and runs only under linux\(^2\) (e.g. distributions such as Ubuntu) and unix-like operating systems (such as macOS). Furthermore, muffin has traditionally been configured and accessed (aka ‘run’) at the ‘command line’ of the linux (or macOS equivalent, which is unix-like) operating system. The command line is a place where you type text and when you press Return, something (hopefully, good!) happens. Typically the stuff you type started with a ‘command’ word, and often followed by one or more options and parameters. The command word and any options / parameters MUST be separated by SPACEs.

The start of each line of the command line is indicated with something like: $ . The $ is called the ‘prompt’ and is ... prompting you to type some input (commands, Tweets, swear words, etc.). See – the computer is just sat there waiting for you to command it to go do something (stupid?). Typically, you will also be informed (reminded) of the user-name, computer name, and current directory, e.g.:

[username@host ~]$ 

which is in this example is user ‘username’ (yours will be different!) on computer name ‘host’\(^3\) and the current directory is the ‘home’ directory – represented by the ~ symbol. If, for example, you were instead currently in the myfolder directory, you would see at the command prompt something like:

[username@host myfolder]$ 

If you are not or not very familiar with the linux/unix command line, such as how to navigate up and down the directory tree and to display the contents of the current directory you are in – for a brief summary of some basic/useful linux commands and usage – see HOW-TO Section 20.2 (page 295).

\(^1\) Also read footnotes please.

\(^2\) For some of you, the mechanics of running the model will be about as much fun as sticking your tongue in an electrical outlet (a popular hobby in England). (However, if you are an experienced linux/unix/tongue-in-electrical-socket user, you can skip onto the next Section and save yourself an entire 15 seconds of reading words.)

\(^3\) Sprout the cat will eventually appear under ‘cat-of-the-day’ on my home-page if you press F5 enough times – all my computing clusters are named after my cats . . .
1.1 Before anything else . . .

NOTE: BE VERY CAREFUL that spaces are not missed out when typing out example lines. Also be careful not to confuse the number one (1) for the letter el (l). Mis-spelling/typing will probably be the primary reason for any wailing and gnashing of teeth ...

NOTE-the-second: If you find yourself terminally bored typing in long long instructions lines, you may be tempted to simply copy-paste from the muffin manual PDF to the command line. This can work, but firstly be aware that trying to copy-paste multiple lines at once is doomed to failure – copy-paste the first line and then following that add the second (or subsequent) line. Also note that the inverted comma symbol in the PDF is not the same inverted comma symbol that linux is expecting ... You should also make liberal use of the up arrow key that bring back the previously entered command (keep pressing the key for progressively older commands). For instance, a mistake in a command line can be corrected by bringing back the offending line and using the left/right arrows to navigate through the characters and correct any mistake.

NOTE-the-third: In places, instructions may be given for specific programs and computer platforms and hence may differ slightly from the software reality in front of you. Use your judgement in translating such instructions. Many other alternative software choices exist for editing files or viewing results, as are other ways of configuring software and file editing/transferring methodologies. Do what suits you best – you can view such instructions where they occur, as more representing an example methodology rather than a literal interpretation of the Constitution.

Required computer hardware

It is possible to install and run the ‘(cGENIE.)muffin’ Earth system model either on a linux box (e.g. Ubuntu) or on a Mac. Otherwise, you will need an account on a linux-based server or cluster. In this case, you can use any platform (linux, macOS, Windoz, as well as iOS and Android (which is based on linux in any case)) to connect to the cluster (but see software requirements, below).

Required computer software

To install and run muffin, at a minimum are needed:

1. A FORTRAN (f90 and f77 combatable) compiler of some sort. This may come with the operating system as standard, possibly gfortran.
2. A git client. If not standard, this is relatively easy to add and install.
3. Compiled netCDF libraries (not so much fun ...).

To edit files and visualize results, you will need some specific software. The exact software will depend on your operating system, but essential are:

1. A viewer for netCDF format spatial data. A Java viewer called Panoply is provided by NCAR for all platforms – http://www.giss.nasa.gov/tools/panoply/ (Note that you will need Java installed!) (Or alternatively: MATLAB.)
2. A simple text editor, except not the rubbish default Windows one – you need one that can display unix ASCII text without screwing it up. Options for Windows users are: notepad++

---

4Sets of detailed installation instructions are available in the HOW-TO section of this manual.
5Note that it is not possible at this time to run muffin under Windows ... at least, not without near infinite pain.
6Only if muffin is being run locally and a cluster account is not provided (but assuming that your cluster account is correctly configured ...).
Chapter 1. Installation, configuration, basic usage

You will need to edit text-based configuration files, possibly in installing and configuring muffin, but definitely in configuring model experiments. So now might be a good time to check that you can use the/an editor! (You will also be using the same editor to view some of the model output.)

You have two alternative options for editing and viewing text files, depending on whether you are a unix nerd with no life, or prefer anything to do with computers to be wrapped in cotton wool and covered with dollops of treacle. EITHER: Use the linux vi (/vim) application (or similar e.g. emacs) if you are familiar with it. I think that this pretty much sucks as a text editor and life is far too short and brutal if you don’t like this sort of thing … OR … use a suitable linux-friendly text editor (NOT Microsoft Notepad) in conjunction with the Secure File Transfer Client. For example: SciTE (http://www.scintilla.org/SciTE.html) is suitable, or Notepad++.

If you fiddle about with the settings under Options/Preferences in the WinSCP program and apply a little common sense, it should be possible to configure things so that you can simply double-click on a file in the remote (right-hand) window panel and it will open like magic (almost)! Saving the file after editing) should then result in the file being saved back to the cluster. Or you can select Edit With (and then SciTE) from right-mouse-button-clicking on the filename. Or … a crude but workable approach is to use an sftp client to drag the file to your local machine (assuming muffin is installed remotely), edit it there, and then drag it back again.8

Model documentation in general

This (the muffin manual), and additional documentation (of varying degrees of up-to-date-ness) can be found:

1. On GitHub.

Here, the latex source for the documentation lives, allowing you to compile the most up-to-date PDF document. And ... make changes yourself and have them incorporated into the official documentation9.

7http://www.seao2.info//cgenie/software/ssh-client.exe
8Note that care still has to be taken to avoid certain Microsoft text editing programs under Windoz.)
9First clone the git repository. Make changes. Commit them locally. Make a 'pull request’ …
2. On my website.

Here, are only compiled, PDF versions of the documentation are provided, and may be somewhat out-of-date (the current versions of everything live on GitHub).

This document in particular

The instructions may not be entirely bug-free – use your judgment.

Go!

OK – now we are ready to start . . .
1.2 Starting (dozing?) off . . .

You are going to be installing the model from scratch – why? Why not? It will be a happy character-building experience for you ... trust me ...

1.2.1 Logging in!

In running muffin locally – log into your linux/macOS box (and skip on to the next section)! 10 Or ... and much more likely to be the case – if you are running muffin remotely (e.g. on a user account on a computing cluster or server), then log into the remote server or cluster account using a 'suitable terminal program'11:

- If logging in via a linux/macOS box, open the terminal/shell window and simply SSH in12, e.g.

  $ ssh username@clustername

  where clustername, the cluster (or remote server) name(!) might, for example, be catname.ucr.edu, and enter your account password (and tell it whether or not you want this password stored, if asked).

  IMPORTANT! – When you type in a password in linux, NOTHING appears on the screen, not even ******** as in common on Windozs. As you type (the password), characters are being entered ... you just cannot see them. Don’t panic – just type in the password (even if you cannot see characters appearing) and hit Enter.

- On a Windoz machine – first start the WinSCP program (an sftp file transfer client). Under Host Name, enter the remote server or cluster name (e.g. catname.ucr.edu): The Port number should be set to 22 (except for sterling.ucr.edu). Enter your computing cluster user-name on the line below this ('User Name') and then the Password. Click on Login. This is your file transfer client.

  You will also need a terminal window. This can be opened by clicking on the ‘Open session in PuTTY’ icon on the top icon row, or pressing Ctrl+P.

  You should now have TWO windows open – a ‘shell” window (lines of text on an otherwise blank screen) and a file manager (transfer) window. Ensure that you have both these before moving on. It is recommended that you maximize both these windows to full screen. (But no-one will die horribly for not doing so. Probably ...).

  You can also log in directly from the shell/terminal window e.g. PuTTY first (rather than first opening an sftp connection first). You will still have to open an sftp connection for file transfer.

  Note that the cluster you access may not use the standard ‘port’ number of 22. For a different port number – in a GUI interface (e.g. WinSCP) simply change the number in the port box, or at the command line, you will need to type:

  $ ssh -p xxxxx username@clustername

  where xxxxx is the port number.

10 If you fail at this step, you’ll have to take up box-modelling instead.
11 It very much depends on what software you are using. Provided are instructions for some examples, but only examples, and your reality may be rather different.
12 If your current directory looks something like this: [username@clustername ~]$ then you are probably already logged in! Otherwise, it will look like: username@localcomputername:~$
1.2 Starting (dozing?) off ...

1.2.2 Downloading/installing the model code

The next step is to download a copy of the source code for muffin.

As of 2018, the branch of the (c)GENIE source code that is cGENIE.muffin, was migrated from a server hosted by the University of Bristol, to GitHub (and hence the versioning system changed form svn to git – a “distributed” version control systems (DVCS))

The address of the new (centralized) home for muffin is:

https://github.com/derpycode/cgenie.muffin

From here, there are 2 ways to get your mitts on the model code:

1. The preferred/advised way is to clone the repository to where you intend to run muffin. While you can also use a GUI based git client, easiest is at the command line (e.g. from your HOME directory), using the command git clone:

   $ git clone https://github.com/derpycode/cgenie.muffin.git

   By doing this, you have created your own code repository (and an identical copy of the one hosted on GitHub). As part of the git clone command, you also automatically check out (from your very own personal repository) a copy of the code.

2. (Less good, but OK if you simply want a copy of the code to run an experiment just once, or simply only want to see the code.) By downloading an archive file, containing all the code etc. For this – click on the green Clone or Download button on GitHub, and select Download ZIP. You then unpack/unzip the files and directory structure where you want it. This [archive download] is a perfectly workable way to proceed ... as long as you neither want to update the code with whatever new developments or bug fixes occur in the future, nor want to have any code changes you might make, become part of the official muffin code (i.e. it becomes a one-off installation that has no connection to the GitHub repository).

1.2.3 Configuring the code

You now need to configure some local environment settings so that muffin can find all the libraries etc. that it needs when the model compiles. The changes you need to make will depend on the platform where you will be running muffin (i.e. the computer where you have just cloned the code repository to). This will be your first real test of editing (and saving) a file on the remote server(!) You’ll need to do this a bunch later and so it is worth developing a working strategy now if you are not already linux-friendly. For instance:

---

13 ALSO IMPORTANT: if you had previously installed a version of muffin under svn, note that the default name of the installation (cgenie.muffin) is the same. If you already have a previous (svn version) installation of muffin, i.e. in a directory cgenie.muffin, you should rename, or simply delete (rm -f -r cgenie.muffin) the original installation so as not to lose files or risk ending up with a mixed-up corrupted installation. For example:

   $ mv cgenie.muffin cgenie.muffin.svn

   will rename the installation directory, adding the extension .svn on the end. You will not longer be able to run the original installation now, but you can fix this by editing some of the directory paths – see FAQ.

14But see later for other/better ways of working.

15This: "... clones a repository into a newly created directory, creates remote-tracking branches for each branch in the cloned repository, and creates and checks out an initial branch that is forked from the cloned repository’s currently active branch."

16Note that the major difference then with the svn system, is that previously, the GENIE code repository existed only on the University of Bristol server, and you checked out the code remotely from there.
Chapter 1. Installation, configuration, basic usage

1. Transfer the file locally (to your laptop/desktop). Edit and save it. Then transfer it back again (hence replacing the original with the edited version). (You can also configure many sftp clients to allow you to edit and re-save the file more directly.)

2. Edit the file *in situ* on the remote account using e.g. *vi* (see linux HOW-TO for brief notes).

3. Pray that the file edits itself.  

The required configuration settings for *muffin* depend on the specific computing platform that you are running on. What follows is a list of the changes associated with some possible platforms that you might be using – find the list name (*bold*) that corresponds to the platform that you are using and ignore all the other options.

- **Ubuntu.**
  This is the default assumed platform. No changes are necessary assuming that the netCDF libraries are installed in their default locations and a relatively recent version of netCDF is used. And you have also jumped the shark with a successful install of other libraries and softwares ... see Section 20.4.

- **sterling** (UCR cluster).
  For running *muffin* on this cluster, there is a single file to edit (which lives in the *genie-main* directory):
  1. In the file *user.mak* – at the end of the file, the netCDF path needs to be changed. First comment out the default setting (with a # symbol) and then un-comment the line under the heading ### domino ###. The resulting section of code at the end of *user.mak* should then look something like:

     ### DEFAULT ###
     #NETCDF_DIR=/usr/local
     ### sterling ###
     NETCDF_DIR=/share/apps
     ### eevee ###
     #NETCDF_DIR=/share/apps/netcdf-c-4.4.1.1-fortran-4.4.4
     ### iwan ###
     #NETCDF_DIR=/share/apps/netcdf/4.0/gnu_fc_4.4.4
     ### sprout ###
     #NETCDF_DIR=/share/apps/netcdf/4.0/gnu_fc_4.4.6
     ### almond ###
     #NETCDF_DIR=/share/apps/netcdf-4.0/gcc-4.4.7
     ### VM ###
     #NETCDF_DIR=/home/mushroom/netcdf/ifort
     ### Ben Mac Pro ###
     #NETCDF_DIR=/opt/local

     where the line:
     
     NETCDF_DIR=/share/apps

     now has no comment symbol before it, and the line:
     
     #NETCDF_DIR=/usr/local

     is now ‘commented out’.

---

179 out of 10 cats have demonstrated this does not work.

18If you are using a platform not listed here, you may be able to simply adapt the instructions for one of the ones listed.

19By ‘relatively recent’ – the default settings assume that the FORTRAN and C netCDF libraries are separate.

20To ‘comment out’ a line – simply add a # symbol to the very start of the line. When *muffin* runs, this line will be ignored.

21To ‘un-comment’ a line – simply remove (delete) the # symbol from the beginning of the line.
1.2 Starting (dozing?) off...

- **eevee** (UCR cluster).
  
  For using this particular cluster, there are 2 files to edit (both of which of which live in the genie-main directory):
  
  1. In `user.mak` – as per for domino (except the `user.mak` line needed is headed `### eevee ###`).
  2. In `makefile` – as a consequence of a currently incomplete netCDF install, missing are the C libraries for netCDF. This affects only a C program that is compiled to complete the tests (by comparing model output with an archived copy of the expected results). For now, this test must be disabled by:
     
     On line 21 – comment out (with a `#`); the word `nccompare` (not the whole line).

- **sprout** (UoB cluster).
  
  For this cluster, there are 2 files to edit:
  
  1. In `user.mak` – at the end of the file, the netCDF path needs to be changed. First comment out the default setting (with a `#` symbol) and then un-comment the line under the heading `### sprout ###`.
  2. In `makefile.arc` – towards the end of the file, under the heading:
     
     `# === NetCDF paths ===`
     
     uncomment the line under:
     `### FOR COMBINED C+FORTRAN NETCDF LIBRARIES #######`
     and comment the 2 lines under:
     `### FOR SEPERATE C AND FORTRAN NETCDF LIBRARIES ###`
     
     (so as to select, combined, rather than sperate, netCDF libraries).

- **almond** (UoB cluster).
  
  As per for sprout (except the `user.mak` line needed is headed `### almond ###`).

- **macOS**.
  
  Please refer to the separate macOS instructions in Section 20.6.

- **Windows**.
  
  Please refer to the separate Windows instructions in Section 20.7.

### 1.2.4 Testing the model code

Finally, you need to test the code to ensure that all the files have been cloned/installed correctly. First, change directory (see: Figure 1.1, and refer to the linux HOW-TO) to:

```
$ make testbiogem
```

This compiles a basic carbon cycle enabled configuration of `muffin` and runs a short test, comparing the results against those of a pre-run experiment (also downloaded alongside the model source).

---

22Note: the model is *always* run from `cgenie.muffin/genie-main`

23Remembering that the `$` is to indicate the command line, and you do not actually type it in.
Chapter 1. Installation, configuration, basic usage

code). It serves to check that you have the software environment correctly configured. There may be some ‘Warnings’ reported (== somepony’s sloppy programming) but these are not detrimental to the ultimate science results (we hope!).

If you see the error message:

make: *** No rule to make target

when you try this, you are probably in the wrong directory (it should be cgenie.muffin/genie-main).

________________________

‘Success’ of this test is indicated by the message:

**TEST OK**

If you see this – you can then be certain that the model you have installed is producing identical (within tolerance) results to everyone else in the World who has ever installed muffin. Note that the model will pause for a l o o o o n g time at the line:

./genie.job -t -k -f configs/eb_go_gs_ac_bg_test.xml -o /home/genie00/cgenie_output -c /home/genie00/cgenie -g ../../cgenie -m "" > testbiogem.out;

This is quite ‘normal’ – the model is thinking! Also – ignore the compiler warnings . . . (reflecting my lack of adequate software engineering skills).

That completes the basic code installation.

If the test doesn’t ’work’ – try issuing the command:

$ make cleanall

and then re-try the test.

Refer to the FAQ section at the end of this book for further clues as why the model installation appears not to be working.

NOTE: Currently, the netCDXF C library is not installed on the eevee cluster, meaning that the test experiment will run, but the comparison against a set of standard results will not. Hence it will appear to ’fail’. 

________________________

GitHub does not host files larger than 100 MB, and the ’lookup table’ for calculating opal dissolution in sediments (see e.g. Ridgwell et al. [2003]) is larger than this. It has hence been committed to the git repo as an archived file.

If (and only if) a (open) silica cycle is to be employed in your muffin experiments, does this file need to be unpacked. A script is provided for this – from genie-main, type:

$ ./installmuffin.sh

and this will unpack the opal lookup table to genie-sedgem/data/input as well as unpacking a copy of the calcium carbonate lookup table.

24Note that if you have disabled the compilation of the C program that compares netCDF files, the test will run but never complete and you’ll not get a **TEST OK** reported, even if the test experiment ran correctly.

25For now, the CaCO$_3$ lookup table also included in the git repo in its expended (unpacked) form.
1.2 Starting (dozing?) off...

1.2.5 Brief notes on git and code

The basic git clone

The simplest workable installation of **muffin**, as described above, is to use **git clone**. You end up with a carbon copy of the muffin code repo (I am too lazy to type out ‘repository’ again) which you have also automatically now ‘checked out’. Changes and developments will occur to the code on the GitHub repo from time-to-time, and if may be that either you might benefit by using a more up-to-date code base, or specific changes may have been made that you absolutely need. To determine the status of your repo, type (from `cgenie.muffin`):

```
$ git status -uno
```

However, git has not actually compared your repo with the one on GitHub, because this is apparently network expensive (as if you don’t spend the rest of your life killing the internet by streaming Rick and Morty). So, use:

```
$ git fetch
```

which will ‘download [new and modified] remote content but not update your local repo’s working state’. If you now type `git status -uno`, **git** can tell you if there is newer content (e.g. ‘Your branch is behind ’origin/master’

```
$ git merge
```

Both these commands are also combined in a single command:

```
$ git pull
```

If you are not developing code, and hence not editing files in the repo (but e.g. only adding new model configuration files), your life should mostly be trouble-free with regards to updating your code via **pull**.

---

26Here, the -uno ensures that files (e.g. experiment configuration files) you have created, but not added and committed, are not listed.

27origin is the origin of the repo – GitHub, and master is the name of the branch, in this case, the default branch name.

28If there are no changes (to fetch and merge), the you get the message: Already up-to-date.

29One exception is, if any of the model installation/configuration files that you might have edited, i.e. one or all of: genie-main/user.mak, genie-main/makefile.arc, and/or genie-main/makefile, have also changed on origin/master, then pull (or merge, after fetch) will fail (with the message: ‘Please, commit your changes or stash them before you can merge. Aborting’). The root issue is a conflict between remote file changes, and local ones that you had not committed to your repo.

You probably want to keep your local configuration changes, otherwise you’ll probably end up re-doing them all over again. One solution is to sneakily ‘hide’ (stash) them out of sight, by: $ git stash You can now pull, updating your repo with respect to origin/master. Then – you want your changes back, so apply the stashed changes: $ git stash apply

If unlucky, there will be conflict as your stashed changed are merged back onto your local repo branch (master). If so, you’ll need to edit the file, deciding which line(s) is correct, and delete the version you do not want, along with the ASCII tags/labels.
Chapter 1. Installation, configuration, basic usage

Figure 1.1: Directory structure of the muffin model. Highlighted in red are directories and sub-directories that you will need to access at some point. Vertical green lines designate directory levels, with example commands shown for moving between them.
1.3 Running the model

The overall sequence of configuring and running muffin (job submission to a cluster queue), is shown in Figure 1.2. Refer to this if in any doubt at any point.

At the command-line ($) in the genie-main directory (not your home directory), you will be entering in a command (.//runmuffin.sh) together with a list of parameters that will be passed to the model, and as if by magic the model will run (or sometimes not). The form of the command you are going to be issuing is:

```
$ ./runmuffin.sh #1 #2 #3 #4 (#5)
```

(don’t type it yet!)

The form of the command requires that you must list at least 4 parameters after ./runmuffin.sh, separated by S P A C E S and on a single continuous line (even if it ‘wraps’ around across 2 lines of the screen). These parameters are:

#1 ... is the name of the required base (or ‘basic’) configuration (‘base-config’) of the model.
#2 ... is the name of the subdirectory (if any) containing the user configuration (‘user-config’) file (i.e., the file containing the specification of a particular experiment).
#3 ... is the name of the experiment itself. There must exist a file in the directory specified by parameter #2 (LABS) with exactly the same name as you enter here for parameter #3 (i.e. parameter #3 points to a file in the directory given by parameter #2).
#4 ... is the run length of the experiment in years – this must be entered as an integer.
(#5) ... There is also one optional (5th) parameter (described later).

As an example of running the muffin Earth system model:

#1 : The base config is: cgenie.eb_go_gs_ac_bg.worbe2.BASE
#2 : The user config directory is: LABS
#3 : The user config file (the experiment name) is: LAB_0.EXAMPLE.
#4 : Run the experiment for ten years: 10
#5 : (There is no restart file, and so no 5th parameter needs to be passed . . . )

The full command for your first example experiment, which you are going to issue from the "/cgenie.muffin/genie-main directory, then looks like:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worbe2.BASE LABS LAB_0.EXAMPLE 10
```

(you can try it now!)

REMEMBER: This must be entered on a single CONTINUOUS LINE. The (single) S P A C E S are vital. Take care not to confuse an el (‘1’) with a one (‘1’) when typing this in ... (it is a ‘one’ here).
Chapter 1. Installation, configuration, basic usage

What should happen is: First, you will end up twiddling your thumbs a while, as all the components of muffin are compiled from the raw source code (FORTRAN). When it has finished doing this, the model will initialize and carry out some brief self-checking. Only then will it start actually ‘running’ and doing something, starting with a header describing the columns of numbers that follow:

```
model year  … guess!
ice(%)  – global sea-ice fraction (%)
<SST>  – global sea surface temperature (’SST’) °C
<SSS>  – global sea surface salinity ’SSS’ (%)
```

The choice of what information to display on screen as the model is running is rather arbitrary, but the chosen metrics do tend to summarize some of the main properties of the climate system and carbon cycle – for my own personal convenience rather than reflecting any fundamental scientific truth ... you may also see columns of information for:

```
pCO₂(uatm) — mean atmospheric CO₂ concentration (in units of μatm)
δ¹³CO₂ — mean δ¹³C value of atmospheric CO₂ (%o) (NOTE: only if ¹³C tracer is selected)
<DIC> — global mean ocean dissolved inorganic carbon (DIC) concentration (μmol kg⁻¹)
<ALK> — global mean ocean alkalinity (ALK) concentration (μeq kg⁻¹) and in experiments with a modern continental configuration, also:
AMO(Sv) — Atlantic meridional overturning circulation (Sv)
```

(assuming that you have a modern continental configuration to even have an Atlantic ocean!)

This information is reported at the same intervals as time-series data (see later and/or refer to the User Manual) is saved and is indicated by:

```
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :
```

Interleaved between these lines are lines reporting the saving of time-slice data (the 2- and 3-D model states – more of which later as well as in the User Manual). These appear as:

```
>>> SAVING BIOGEM TIME-SLICE AVERAGE CENTERED @ year:
```

You can stop the model at any point (all data up to that time will have been saved) by hitting: <Ctrl-C> (CONTROL key + ‘C’ key).

Just from examining the screen output: how close to steady state does the system appear to have come after just 10 years? i.e., do SST and/or sea-ice extents appear to be converging towards stable (constant) values? This will be an important question to think about later on: ‘has the model reached steady-state (and does it matter)?’
1.3 Running the model

In this particular example, the output should look something like the following:

```
*******************************************************
*** Initialisation complete: simulation starting ...***
*******************************************************

| model year | pCO2(uatm) | d13CO2 | AMO(Sv) | ice(%) | <SST> | <SSS> | <DIC>(uM) | <ALK>(uM) | pCO2(uatm) | d13CO2 | AMO(Sv) | ice(%) | <SST> | <SSS> | <DIC>(uM) | <ALK>(uM) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.00 | 278.000 | -6.500 | 0.000 | 0.000 | -0.000 | 34.900 | 2244.000 | 2363.000 |
| 1.00 | 279.960 | -6.598 | 13.613 | 0.744 | 2.509 | 34.901 | 2241.498 | 2363.111 |
| 2.00 | 279.525 | -6.580 | 12.828 | 3.499 | 4.471 | 34.901 | 2240.173 | 2363.135 |
| 3.00 | 279.258 | -6.568 | 11.695 | 5.028 | 5.996 | 34.901 | 2239.169 | 2363.161 |
| 4.00 | 279.044 | -6.558 | 10.444 | 5.929 | 7.209 | 34.901 | 2238.354 | 2363.191 |
| 5.00 | 278.899 | -6.551 | 9.380 | 6.156 | 8.156 | 34.902 | 2237.664 | 2363.220 |
| 6.00 | 278.777 | -6.545 | 8.500 | 6.975 | 8.975 | 34.902 | 2237.069 | 2363.246 |
| 7.00 | 278.680 | -6.541 | 7.922 | 6.623 | 9.637 | 34.903 | 2236.548 | 2363.267 |
| 8.00 | 278.601 | -6.537 | 7.917 | 6.738 | 10.225 | 34.903 | 2236.087 | 2363.285 |
| 9.00 | 278.528 | -6.534 | 7.952 | 6.740 | 10.732 | 34.904 | 2235.682 | 2363.301 |
| 10.00 | 278.466 | -6.531 | 8.025 | 6.694 | 11.176 | 34.904 | 2235.325 | 2363.314 |

*******************************************************
*** Simulation complete: shutdown starting ...***
*******************************************************
```
Figure 1.2: Schematic of the sequence-of-events in configuring and running an experiment.
1.4 Creating new experiments!

The key to creating new experiments is to remember that the name of the user-config file, that contains the parameter settings that define that specific experiment, becomes the name of the experiment and hence name of the model results sub-directory in cgenie_output. Changing the name of the user-config file, hence leads to a new experiment name and a new model results sub-directory. (Conversely, not changing the name of the user-config file and re-running it results in the results of any previous experiment run using that user-config file, being over-written.)

There are two obvious ways to create a new user-config file and hence new experiment (although the simplest and recommended way, is the 2nd one):

1. Create a blank text file and populate the contents with the parameter value assignments you need for your new experiment.
   Inevitably, it is difficult to remember all the names and even values you want to specify, meaning that you’ll end up looking at existing user-config files and copying and pasting lines (or even the entire contents) from the old file(s) into your new user-config file. So then you may was well ...

2. Copy and edit an existing file!
   This is the most practical approach – pick an existing user-config file that is closest to the specific experiment that you want to run – copy and rename it, then edit the contents and save. For the purpose of trying this out, you can literally pick on any existing file in the LABS subdirectory (of genie-userconfigs).
   You can do this by:
   - Using your sftp client (program):
     First – drag the existing user-config file to your local computer. On your local computer, rename the file as per how you would ’normally’ rename any file.
     Now you can edit the parameter values, re-save it, drag it back to the cluster (from your local computer) using the sftp client ... and finally run the experiment.
   - Or, if you are comfortable working at the linux command line; working from the same directory (e.g. LABS) that the file you want to copy lives in, you can copy a file (to a new filename) by:
     $ cp oldfile newfile
     (after which you can edit the new user-config file newfile, re-save, and then run the experiment.)

Remember that the new user-config file needs to be saved in (or copied to) the genie-userconfigs/LABS sub-directory (in the case of experiments carried out as part of the tutorials described in this manual), or anywhere else ... as long as you specify the correct path to the directory where you save the new file to.

30 $ touch file.txt will achieve this at the linux command line.
31 See previous Section and the use of the 2nd parameter passed to runmuffin.sh
1.5 Model output

The first thing to note about output (i.e., saved results files) from **muffin** is that every science module saves its own results in its own sub-directory (and sometimes in very different and difficult-to-fathom ways . . . ) – see Figure 1.1. All the sub-directories of results, plus copies of input parameters and the model executable, are gathered together in a directory that is assigned the same name as the experiment (== **user-config** file name). The experiment results directories all live in:

```
~/cgenie_output
```

and will be assigned a directory name something like:

```
LAB_0.EXAMPLE
```

(this being the results directory name for an experiment called **LAB_0.EXAMPLE**). Within this directory are each module’s results sub-directories. We will primarily consider only results saved by the ocean biogeochemical module **‘BIOGEM’** (subdirectory: biogem). The results files in this example will thus be found in:

```
~/cgenie_output/LAB_0.EXAMPLE/biogem
```

**BIOGEM** has a flexible and powerful facility of saving results by means of spatially explicit ‘time-slices’, and as a semi-continuous ‘time-series’ of a single global (or otherwise representative mean) variable. In contrast, the atmospheric chemistry module **‘ATCHEM’** does not save its own results (**BIOGEM** can save information about atmospheric composition and air-sea gas exchange) while the marine sediment module **SEDGEM** does save its own results, but only at the very end of a model experiment (**BIOGEM** can also save the spatial distribution of sediment composition as time-slices as well as mean composition as a time-series). Furthermore, to attain a common format for both ocean physical properties and biogeochemistry, **BIOGEM** can save a range of ocean results in addition to temperature and salinity, such as: velocities, sea-ice extent, mixed layer depth, convective frequency, etc.

**NOTE:** If in an sftp client window you find that you cannot ‘see’ the **cgenie_output** directory ... or cannot find any of the results sub-directories you are expecting, you will need to refresh the directory listing (e.g. for **WinSCP**, there is a double green-arrow Refresh icon button near the top right of the window that you can click on). sftp client programs generally do not automatically refresh directory listing on the remote computer.

1.5.1 Time-slice output

One of the most informative data sets that can be saved is that of the spatial distribution of properties (such as tracers or physical ocean attributes). However, saving full spatial distributions (e.g., a $36 \times 36 \times 8$ array) for any or all of the tracers each and every time-step is clearly not practical; not only in terms of data storage but also because of the detrimental effect that repeated file access has on model run-time. Instead, **BIOGEM** will save the full spatial distribution of tracer properties only at one or more predefined time points (in units of years). These are termed **time-slices**. At the specified time points, a set of spatially-explicit data fields are saved for all the key tracer, flux, and physical characteristics of the system. However, rather than taking an instantaneous snapshot, the time-slice is constructed as an average over a specified integration interval (the default is set to 1.0 years, i.e. an annual average). **BIOGEM** assumes that the specified time point represents the mid-point of the (annual) average with the results that output years end up being reported as e.g.,
1.5 Model output

0.5
1.5
2.5
4.5
... 

(the mid-points of averages made over the intervals: 0-1, 1-2, 2-3, 4-5 years, etc.).

1.5.2 Time-series output

The second data format for model output is much more closely spaced in time. Model characteristics must then be reducible to a single meaningful variable for this to be practical (i.e., saving the time-varying nature of 3-D ocean tracer distributions is not). Suitable reduced indicators would be the total inventories in the ocean and/or atmosphere of various tracers (or equivalently, the mean global concentrations / partial pressures, respectively). Like the time-slices, the data values saved in the time-series files represent averages over a specified integration interval (the default is set to 1.0 years (annual average) but the results are reported with respect to the mid-point of the average which is where the ‘.5’ bits come in again).

1.5.3 File naming convention

The biogem results directory will contain files with names of the form:

- _restart.nc (is the re-start file created from the run you have just complete, and can be ignored).
- biogem_series_*_res – these are the time-series files (in ASCII / plain text format).
- biogem_year_*_diag_GLOBAL.res – these contain (global diagnostics) summary information and are saved at the same frequency as the time-slices (also as ASCII / plain text).
- fields_biogem_2d.nc – 2-D fields of ocean and atmosphere properties, as NetCDF.
- fields_biogem_3d.nc – 3-D fields of ocean properties, as NetCDF.
1.6 Viewing model output

1.6.1 Time-series output

A descriptive summary of all the time-series (biogem_series_*.res) data files is given in the muffin User Manual if you are really that bored. The files of most immediate use/relevance are:

- biogem_series_atm_humidity.res - mean atmospheric (surface) humidity
- biogem_series_atm_temp.res - mean atmospheric (surface) air temperature
- biogem_series_misc_opsi.res - min/max overturning stream-function values (e.g. AMOC)
- biogem_series_misc_seaice.res - mean ocean sea-ice cover and thickness
- biogem_series_ocn_sal.res - mean ocean surface and whole ocean salinity
- biogem_series_ocn_temp.res - mean ocean surface and whole ocean temperature

One way of viewing the contents of files (in a shell window/terminal) is to change directory to the experiment results directory and opening the file in a file editor at the command line. But that is not so much fun.

Instead – change to the experiment results directory and then to the biogem sub-directory in the Secure File Transfer Client, and try double-clicking (if you have set up the WinSCP preferences correctly) or right-mouse-button-clicking (the then Edit with) on one of the .res files (listed above). For biogem_series_ocn_temp.res, you should see 5 columns:

\%
time (yr) / temperature (C) / _surT (ice-free) (C) / _benT (C) / _surT (C)

for: model time (years) (actually, the mid-point in time of an annual average), annual mean ocean temperature (averaging over the entire ocean volume), annual mean ocean surface temperature (excluding ice-cover areas), annual mean benthic (sea-floor) temperature, annual mean ocean surface temperature (now including ice-cover areas). Other results files may differ in the numbers of columns but all should be identifiable from the header (first line) information.

Remember: WinSCP does not automatically refresh the directory listing. If you cannot see the results sub-directory with the experiment name you have just run, 99 times out of 100, it is because the display of the WinSCP needs to be refreshed – there is an icon at the top of the program window or hit the ‘F5’ key.

For your information and edification (only): Excel, or MUTLAB if you prefer, can be used to graph the time-series results. Either way you will have to deal with the header line(s) that are present at the top of the file (and preceding the rows of data).

In Excel: Choose File then Open. You will want to select Files of Type ‘All Files (*.*)’. In the Text Import Wizard window you can request that Excel skips the first few lines to start the import on the 2nd or 3rd line of the text file. Alternatively: set an appropriate column width manually in Excel to ensure that the columns of data are correctly imported.

MUTLAB will ignore lines starting with a %, which the time-series starts with. However, it may be that the header line wraps-around and there is in effect a 2nd header line but without a %. In this case, extra care (or a quick edit of the header in the ASCII file) will be required to load the data into MUTLAB.
1.6 Viewing model output

1.6.2 2- and 3-D time-slice output

For the time-slice NetCDF (*.nc) files you will be using a program called Panoply. If you want your own (FREE!) copy of this utility, you can get it here (and is available for: Windoz, Mac, and linux operating systems): http://www.giss.nasa.gov/tools/panoply/.

When you open the NetCDF file, you will be presented with a ‘Datasets and Variables’ window (on the left hand side of the application window). This contains a list of all the parameters available that you can display. You will find that the ‘Long Name’ description of the variable will be the most helpful to identify the one you want. Simply double-click on a variable to display.

For the 3-D fields you will be asked first whether you want a ‘Longitude-Latitude’ or ‘Latitude-Vertical’ plot (for the 2-D fields, the plot display will immediately open). For the ‘Longitude-Latitude’ plots – there are multiple levels (depth layers) in the ocean - these data that can be plotted from the surface to the abyssal ocean. For the ‘Latitude-Vertical’ plots – there are multiple possible longitudes at which to plot slices. The default is the global mean meridional distribution. There is also an option for ‘Longitude-Vertical’ plots (which we will not use).

There may be multiple time-slices (i.e., you can plot data saved from different years). By default, only the very first time-slice will be displayed.

You can choose interpolate the data or not (often you may find that it is clearer not to interpolate the data but to leave it as ‘blocky’ colors corresponding to the resolution of the model), change the scale and colors, overlay continental outline, change the projection, etc etc. Grey cells represent ‘dry’ grid points, i.e., continental or oceanic crust.

NOTE: The default settings in Panoply can mislead. Be aware of:

1. Panoply initially displays the very 1st time-slice (often year mid-point 0.5) time-slice rather than the experiment end. This can confuse and look like an experiment has not done anything!
2. By interpolating the data (not always misleading). To remove interpolation, un-tick: ‘Interpolate’ in the ‘Arrays’ tab.
3. By displaying a global zonal mean by default when selecting Latitude-Vertical plots. Then, to further confuse you, by plotting the output up-side-down (to invert: in the ‘Grid’ tab, hit ‘Swap B/T’ (for swap bottom/top).
4. By listing all ‘Plottable variables’ (option at the bottom of the window), when what you ideally want are the shorter and less confusing list of ‘Georeferenced variables’.
5. In Longitude-Latitude plots, by overlaying the modern continental output. (muffin land is marked in grey.)
6. By fitting a scale to the plot when the display window is opened, but not changing the scale when e.g., time or depth is changed. (The point of confusion is that you can quickly move outside the scale and end up with all model points dark blue or red.) Re-fit the scale, or manually set limited, in the ‘Scale’ tab. So be careful when opening a new plot that you are looking at what you *think* you are looking at . . . All the defaults can be changed via the ‘Edit’ drop-down menu and ‘Preferences’.

Explore different data fields and play with different ways of displaying them. Aim for a set of display properties that show the information you are interested in / want to present in the clearest possible manner. Try different years (time-slice number), depth level (for a Latitude-Longitude plot), or longitude (for a vertical section).
To save plots in **Panoply**:

File

Save Image As ...

Then select the location, filename, and graphics format.
1.7 Submitting experiment ‘jobs’

This bit is no particular fun at all, but it is a very handy ‘trick’ for running the model in the background, and maximizes drinking time in the bar vs. sat bored watching a computer screen :) 

Running jobs interactively is all very well, but there are three important limitations:

1. The connection between your terminal and the server computer running the model must remain unbroken. Anything more than a fleeting loss of internet connectivity may result in the experiment terminating.

2. You can only run one experiment at a time . . . unless you want to have thousands of separate terminal open . . . ? I thought not . . .

3. Any cluster or computer you are likely to be accessing using a shell will not have many computing cores itself, either because it is a single machine with only one or two processors, or if a cluster, by using a terminal you are running on the ‘head node’, which will have similar computing core limitations to running on a single machine. The more experiments you run simultaneously, the slower they will all run . . .

The alternative is to submit your experiment as a ‘job’ to a queuing system which then manages what compute resources are used to run the model. Once you have submitted the experiment, that is it – you can go straight to the pub :) 

For example – to run the same experiment as before (LAB_0.EXAMPLE) for maybe 100 years (or even longer if you wish – I am just pulling factors of 10 out of thin air here) but now submit the experiment as a job to the cluster queue, type (again: SINGLE, CONTINUOUS LINE):

```
$ qsub -q dog.q -j y -o cgenie_log -V -S /bin/bash runmuffin.sh
cgenie.eb_go_gs_ac_bg.worbe2.BASE LABS LAB_0.EXAMPLE 100
```

Here, the queue name for this particular cluster is dog.q. The cluster you are actually using will have a different queue name (e.g. refer to any cluster-specific information that you might have).

Note that now you should omit the ‘./’ bit before runmuffin.sh. (If you are interested (I know that you are not): the options following qsub and before runmuffin.sh do things like re-directing screen output and error messaging to a file and specify which linux ‘shell’ to assume. It is even possible to receive an email when the job is done :) ) The status of the cluster queue and how your experiment job is getting on (e.g., “Is it finished yet?”) can be checked by typing:

```
$ qstat -f
```

(qstat -f -u "*") will show all jobs on the cluster.

After submitting an experiment, you receive a job number. This number appears in the first column in the queue status information when you issue a qstat -f command. You should see your job appear on one of a number of compute nodes, perhaps numbered 0-0 through 0-5), although it might briefly reside as a ‘PENDING JOB’. For each node, there are multiple processing cores (depending on the specific cluster and queue), meaning that multiple instances of muffin can run simultaneously on each node. For an 8-level ocean based configuration of muffin, being run for 100 years, the job should remain there in the queue for a few minutes before ‘disappearing’ (your clue that it has finished, or died\(^{32}\) . . .). If you periodically re-issue a qstat -f command you can follow your job’s progress.

\(^{32}\)If your experiment appears on the queue but vanishes after a few seconds, it has most likely died :(

Chapter 1. Installation, configuration, basic usage

A rough rule of thumb is that 8-level ocean muffin @ a horizontal grid resolution of 36x36 will simulate about 1000 years per CPU hour. The 16-level version (which you will use later), runs at about 300-400 years per CPU hour.

NOTE: It may be that the FORTRAN compiler is not accessible by the computer nodes. The implication of this is that the muffin executable must be already compiled BEFORE a job is submitted to the queue. In other words; if you have just changed the model resolution or continental configuration, or number of tracers (i.e. changed the base-config) or issued a make cleanall command you MUST briefly run your desired experiment (or equivalent) interactively (i.e., in the shell window) to ensure that everything is correctly compiled. For instance, either run the experiment for a couple of years or start the experiment for the desired full duration, but 'kill it' (Ctrl-C) once the experiment is running successfully.
1.8 ‘Restarts’

Not much fun here either ... but again... an important and time-saving (== increased drinking time!) modelling technique to learn to use.

By default, model experiments start from ‘cold’, i.e., the ocean is at rest and uniform in temperature and salinity while the atmosphere is uniform in temperature and humidity. All biogeochemical tracers in the ocean have uniform concentrations and/or are zero and there are no biogenic materials in deep-sea sediments. From this state it will take several thousand years (kyr) for the climate system to reach steady-state, and closer to 5 kyr (or more) for ocean biogeochemical cycles and atmosphere $CO_2$ to reach steady-state, and exceeding 100 kyr for sediment composition to re-balance weathering ... Reaching this the equilibrium state is called the ‘spin-up’ phase of the model. There is evidently little point in repeating the spin-up for each and every model experiment that are similar except in a single detail (e.g., testing a variety of different $CO_2$ emissions scenarios all starting from current year 2012 conditions). A facility is thus provided for requesting that a ‘re-start’ is used – starting a new experiment from the end of a previous one, usually a spin-up that has been run explicitly for the purpose of generating a starting point (re-start) of the system at steady-state (equilibrium) for subsequent experiments to continue on from. It is important to note that there is nothing special about a re-start – it is simply an experiment that you have already run. Equally, there is nothing special about the re-starts you will download next – these you could have generated yourself – it simply saves time to have them provided.

To experiment with using a re-start, you will first need to download a file that has been created (a pre-run 10,000 year spin-up). To fetch this: Change to the cgenie_output directory (perhaps by going ‘home’ first (cd <Enter>), and then changing to cgenie_output – refer to linux commands HOW-TO and Figure 1.1), and type:

```bash
$ wget http://www.seao2.info/cgenie_output/LAB_0.SPIN.tar.gz
```

This downloads an archived/compressed copy of the restart from a location on the interweb. Extract the contents of this archive by typing:

```bash
$ tar xfzv LAB_0.SPIN.tar.gz
```

Finally, change directory back to cgenie.muffin and then genie-main so that you are ready to run the model (the model is *always* run from the cgenie.muffin/genie-main directory).

A re-start can be requested for running on a new experiment from the end of a previous one, by providing a 5th (optional) parameter when entering in the runmuffin.sh command. A spin-up of the modern World climate state is provided for you as a restart – LAB_0.SPIN – which you have just unpacked to the cgenie_output results output directory.

To test out the use this restart – create a new (user-config) experiment configuration file in the directory:

```bash
~/cgenie.muffin/genie-userconfigs/LABS
```

taking the file LAB_0.EXAMPLE (which is provided) as a template (no parameter changes need to be made yet). As described earlier – copy this file and give it a new name – here, name it: LAB_0.NEW

You can then specify the use of the restart in your new LAB_0.NEW experiment:

```bash
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worbe2.BASE LABS LAB_0.NEW 100 LAB_0.SPIN
```
The run-time output should now look noticeably different. There should be no (or perhaps just very little) drift in any of the various variable values outputted to the screen – this is because you have (re-)started from the end of a run that had already reached an equilibrium, steady-state.
2. Climate dynamics & experimental design

Stuff to keep in mind:

- Models ARE NOT the ‘real World’ (it is going to be pretty obvious this is the case here).
- Don’t believe what you read in Nature or Science.
Readme

You will need to download a new restart file prior to embarking on the snowball Earth experiments. To fetch this: change directory to the cgenie_output directory, and type (or copy and paste carefully from this PDF ...):

```bash
$ wget --no-check-certificate http://www.seao2.info/cgenie_output/LAB_1.SPIN.tar.gz
```

This downloads an archived/compressed copy of the experiment LAB_1.SPIN – effectively, just an experiment (spin-up) that has already been run for 5,000 years for you. Extract the contents of this archive by typing (also from the cgenie_output directory):

```bash
$ tar xfzv LAB_1.SPIN.tar.gz
```

A new experiment results directly will then appear as if you had just run the entire 5,000 year experiment yourself (and you could in fact have done so). Remember to refresh the WinSCP directory view window if you are using this particular software, or it might appear that nothing has been extracted.

You’ll then need to change directory back to genie-main to run the model.

If ... when you subsequently try and use this restart in an experiment, muffin stops and complains that it cannot find it, check:

1. That you in fact downloaded it to the correct directory, which should be: cgenie_output, and not randomly to e.g. your home directory ...
2. That you unpacked it.
2.1 Brrrrrrrrrrrr – it’s chilly on ... snowball Earth!

To illustrate how ‘easy’ it can be to configure an Earth system / climate model such as muffin and explore the behavior of the Earth system and its response to perturbation – you are going to induce an extreme cooling of the climate system and see what happens. Solar output was weaker during the late Neoproterozoic, a time when the Earth experienced a series (2 ish) of extreme glaciations. Thus, having a mild climate state to start with must have been dependent on sufficient \( CO_2 \) and/or \( CH_4 \) in the atmosphere which presumably must have been highly elevated compared to the modern World ... sort of opposite to the problem we have today ...

You are going to be running experiments in a similar manner to before, and using the re-start experiment that you downloaded. You could then, for example, take the experiment configuration file LAB_1.EXAMPLE (which is provided for you), and run the experiment, which has a new base-config: cgenie.eb_go_gs_ac_bg.woreq1.NONE, by typing:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.woreq1.NONE LABS LAB_1.EXAMPLE 100 LAB_1.SPIN
```

However ... why not get into the habit of creating new and uniquely named experiments and their associated user-config files (no harder than copying it and renaming it – see earlier). If you keep using the same experiment name, the results will be over-written each time you re-run that experiment, whilst having 2 (or more) experiments running simultaneously with exactly the same name causes havoc as they try and over-write each others results files in a somewhat entertaining but ultimately useless way. So – copy and rename the file LAB_1.EXAMPLE to ... LAB_1.EXP1 or whatever (in the subdirectory LABS), and then run your newly created experiment:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.woreq1.NONE LABS LAB_1.EXP1 100 LAB_1.SPIN
```

Remember that because you have changed since the exercises in the previous chapter, running the model first interactively (at the command line) is essential in order for the model code to be re-compiled consistent with the new configuration. i.e. you cannot directly start submitting experiments using the base-config cgenie.eb_go_gs_ac_bg.woreq1.NONE, straight-away after having previous run the model using the cgenie.eb_go_gs_ac_bg.woreq1.BASE base-config. Once you have re-compiled the model with the new base-config and starting running (any experiment), you need not re-compile again and can submit as many jobs to the cluster queue as you like, up until you change base-config once again.

Overall: your task in this exercise will be to determine the radiative forcing (or \( pCO_2 \) equivalent) threshold required to drive the climate system into a full ice-covered ocean (snowball Earth) state. (Read the Hyde et al. [2000] paper.)

Useful 2-D (netCDF — Panoply) variables to view are: surface air temperature and sea-ice extent (and/or thickness). Ocean surface temperature and salinity can be viewed in the 3-D NetCDF results file but are likely to be of less interest.

Time-series (ASCII .res files) are useful for providing simple mean indicators of global climate such as global ocean fractional sea-ice covered.

Note that the model configuration of an idealized super-continent you are using and as defined by the cgenie.eb_go_gs_ac_bg.woreq1.NONE base-config file, positioned symmetrically about the Equator, is pretty unrealistic. But the further you go back in time, the more uncertain it becomes as to exactly where and in what orientation the continents were. Sometimes modelers have to resort to somewhat idealized experiments if the uncertainties are too great. In addition, one can conduct
sensitivity experiments to test whether the continental configuration is important to the results. For instance, *Hoffman and Schrag* [2002] discuss the potential importance of continental configuration, while the entire hypothesis of *Donnadieu et al.* [2004] rests on specific details of the continental configuration being realistic.

For this configuration, the solar constant is set weaker than modern to reflect the fact that the Sun’s output has increased with time and during the Neoproterozoic the solar constant would have been ca. 5% weaker. This is set by the model parameter:

\[ \text{ma\_genie\_solar\_constant} = 1285.92 \]

which is set at the top of the provided *user-config* file `LAB_1.EXAMPLE`. (For reference, the modern value is 1368\(Wm^{-2}\).)

To search for the atmospheric \(CO_2\) concentration (or rather, radiative forcing equivalent) that would lead to a ‘snowball Earth’ state in the Neoproterozoic and answer the question: ‘How low does \(CO_2\) have to be to trigger a ‘snowball’?’ you are going to edit the experiment file that controls the specific details of the experiment – the *user-config* file. From the `genie-userconfigs/LABS` directory, open one of the snowball experiments in your preferred text editor. At the top of the file you should see something like:

```
#
# --- CLIMATE ---------------------------------------------------------
#
# scaling for atmospheric CO2 radiative forcing, relative to 278 ppm
# ea_radfor_scl_co2=20.0
```

Each line that is not commented out (i.e., no #) contains a *parameter* name and assigned value pair, with the format:

```
PARAMETER=VALUE
```

The value of each parameter can be edited to change the experiment. (Additional parameter value specifications can also be added, or existing ones deleted.) In this example, the line:

```
ea_radfor_scl_co2=20.0
```

specifies a radiative forcing of climate by \(CO_2\) equivalent to x20 modern (20x278 = 2560 ppm). If you instead wrote:

```
ea_radfor_scl_co2=1.0
```

this would give you a modern\(^1\) (x1, or 1x278 = 278 ppm) radiative forcing. Note: \(CO_2\) is not being explicitly modeled in this experiment, but the long-wave radiative forcing associated with a specified concentration of \(CO_2\) (in ratio to modern) is being set instead.

Edit the value of `ea_radfor_scl_co2` (lower or higher – your choice) and save the file (with a different filename!). Re-run the experiment until sea-ice extent starts to approach a new steady state. You may want to try longer simulations than suggested (>100 years) if it becomes clear that the model is still far from steady-state by the end of the experiment. You can judge how close to equilibrium things have got by following (and/or plotting) the evolution of e.g., global surface air temperature or sea-ice extent (both time-series files).

\(^1\)Technically: the pre-industrial \(CO_2\) value rather than ‘modern’ *per se*. 
In terms of methodology – or; 'how am I going to answer the question' – you will need to run multiple different model experiments, each with a different value for the radiative forcing parameter, and find out for what value of the parameter, the Earth becomes completely ice-covered. It is your choice whether you change the radiative forcing parameter value, run the experiment, but wait for it to finish before deciding the radiative forcing parameter value to try for the next experiment – i.e. a sequential approach, or try running multiple different 'guesses' simultaneously by submitting multiple different experiments to the queue. Ideally, each experiment will have a different name.

In each experiment, you want to be assessing how far towards the Equator the sea-ice limit encroaches by viewing some of the time-series and time-slice files or even the on-screen summary lines (assuming that you are running interactively rather than via a job submission to the cluster queue). Informative time-series variables include (but are not necessarily be limited to): atmospheric temperature and sea-ice cover. (Sea-ice thickness, on account of the simple physics in the model, low resolution and long time-step, can fluctuate a little. This is also true for sea-ice area. Sea-ice volume is then the most reliable data (column) to keep track of in the time-series output.)

For the time-slice data: atmospheric and ocean surface temperature and particularly sea-ice extent (fractional cover), (2-D biogem NetCDF file) may be informative.

HINT: Be careful with the 'Fit to data' scaling feature in Panoply – at near complete sea-ice cover, you may find Panoply scaling min and max sea-ice between 99.1 and 99.9% or something. Specific fixed scale limits (e.g. 0 and 100) can be set instead.

In answering the question ('How low does CO\textsubscript{2} have to be to trigger a 'snowball'?'), think about what an appropriate degree of accuracy might be for your experiments. Just because computer models generally calculate to around 16 significant places of precision, does not mean you have 16 significant figures of realism. For instance – how many significant figures is the solar constant quoted to and what do you think is the uncertainty in this? Harder to judge is how the assumed (incorrect) continental configuration creates additional uncertainty, or the simple physics assumed in the ocean or sea-ice, or lack of snow on land . . .

Other questions to think about with regards to numerical modeling (as well as this particular experiment) are:

- (Is the model configuration and experimental design ‘realistic’ ... ?)
- What is ‘missing’ in the model and what might the implications for your predictions and conclusions be? For example, there is no land-surface scheme (and hence no concept of ‘snow’) in this particular configuration.
- Are the simulations being run for sufficiently long? Why not if not (i.e., justify your choices of parameter values and experimental assumptions)? How might the results and conclusions be biased (if at all)?
- How would you test model predictions and your overall conclusions?
- How could the experimental design be improved?

Once you are happy about the controls on the snowball threshold try and answer the supplementary question:

*How high does the (CO\textsubscript{2}) radiative forcing have to be in order to escape from a snowball?*

Having determined the appropriate radiative forcing value required to create a snowball state, you can use that experiment as a re-start, and hence carry out a series of experiments with increasing
radiative forcing, all starting from the same within-snowball climate state you have just created. Defining the radiative forcing / climate path going out of a snowball would complete the hysteresis loop of Hyde et al. [2000].

Note that a good re-start is one for which the experiment did not sit too long in the snowball state before finishing (the more sea-ice thickness you create in the first experiment, the more you are going to have to melt in the next and hence the longer this new series of experiments will take to ruin…). To achieve this, you can fine-tune the number of years the experiment is run, i.e. having determined the appropriate radiative forcing value required to create a snowball state, find out when (in terms of number of years) in the experiment the snowball state first occurred, and then run a new experiment that finishes only a decade or so after the snowball state was initiated.²

HINT: If you are having trouble deciding whether or not the snowball is heading in the right direction (i.e. towards an exit!), e.g. because sea-ice is always reported at 100% (or close to), you can keep track of whether there is net melting or net freezing by following mean sea-ice thickness (m) as reported in the biogem_series_misc_seaice.res time-series output file. Your indication of a melting snowball state is a progressive decline in mean thickness (a proxy for global ice volume). Note that you can open and review the results of time-series files at *any* time during the experiment as the lines are written while the data for each time point is generated.

Overall: think critically about the model configuration, the experimental design, and the nature of the scientific question (based on your background reading of snowball Earth). Some of the exploration/testing suggestions (above) may not necessarily give substantially different results. Such a finding would be as valid and interesting as determining an important dependence of a certain assumption, and would for instance indicate that the associated paleo uncertainties are not critical to model assessment of the question.

Always be prepared to justify all your choices for experimental design and model settings, e.g., range of radiative forcing assessed, continental configuration(s), solar forcing, use of re-starts (if any), run duration, etc. etc. etc. etc.

²You cannot select when the re-start is saved – it is always saved at the end of an experiment.
2.2 Further ideas

2.2.1 Feedback loop analysis

To quantify the snowball Earth hysteresis loop in muffin as per Figure 2 in Hyde et al. [2000] you will need to extract from the model ‘meaningful’ measures of climate (e.g., global surface air temperature, fractional sea-ice coverage) as a function of CO\textsubscript{2} multiples, CO\textsubscript{2} concentration, or (better) radiative forcing. For the latter, in muffin, the radiative forcing for a doubling of CO\textsubscript{2} is set at 5.77 W m\textsuperscript{-2}. See: Myhre et al. [1998] (Geophys. Res. Lett. 25, 2715–2718) and/or IPCC [2007] for more on what radiative forcing is and how it is related to a relative change in CO\textsubscript{2} concentration. Also, for making a comparison with Hyde et al. [2000] – for going into the snowball, note that they plot the change in radiative with a ‘cooling’ as positive (a bit daft). Their baseline radiative forcing state (an anomaly of 0 W m\textsuperscript{-2}) you might assume is equivalent to 278 ppm and hence ~130 ppm is an approximately halving of CO\textsubscript{2} and hence creates ~ 5 W m\textsuperscript{-2} of cooling. (You might prefer to plot the radiative forcing change as warming being positive, which makes rather more sense ...)

For coming out of a snowball, because the CO\textsubscript{2} and hence radiative forcing threshold is so high as compared to going in, you may want to be creative in the plotting (assuming attempting to combine both thresholds into a single plot) and, for instance, one might break the scale between the low radiative forcing interval spanning going in and the high one spanning coming out.

Another example is as per Figures 3 and 4 in Stone and Yao [2004] (Clim. Dyn. 22, 815–822) (although here it is the solar constant rather than long-wave radiation forcing that is being varied). So in fact, you could try varying the solar constant as an alternative to radiative forcing and hence be able to come up with a plot directly comparable to Stone and Yao [2004].

2.2.2 Continental configuration (1)

It was mentioned earlier that the position of the continents is an area of modelling uncertainty and might be important. You can test for this. Four alternative base-configs are provided, each defining a different continental configuration:

1. cgenie.eb_go_gs_ac_bg.wopol1.NONE
   – a single polar super-continent, with an ocean resolution of 36x36 with 8 vertical levels. (Note potential ‘l’ and ’1’ confusion in ‘wopol1’.)
2. cgenie.eb_go_gs_ac_bg.wopol2.NONE
   – one continent at each pole, with an ocean resolution of 36x36 with 8 vertical levels.
3. cgenie.eb_go_gs_ac_bg.woreq1.NONE
   – a single Equatorially-centred super-continent, with an ocean resolution of 36x36 with 8 vertical levels. [this is the current configuration (that you used previously)]
4. cgenie.eb_go_gs_ac_bg.woreq2.NONE
   – two continents straddling the Equator, with an ocean resolution of 36x36 with 8 vertical levels.

You can use the given user-config file (LAB_1.EXAMPLE) again as an experiment template file, and any of the alternative configurations can be run very similarly to as per before, i.e.:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.xxxxx.NONE LABS LAB_1.EXAMPLE 100
```

Note that you are using a different base-config file name: cgenie.eb_go_gs_ac_bg.xxxxx.NONE where xxxxx is one of: wopol1, wopol2, woreq1, or woreq2.

Also note that no re-starts are provided for any of these configurations. You may (or may not) want to create some (you will need to judge for yourselves how long to run the re-start experiments
for to achieve as close to steady-state as you think is 'sufficient'). Recall again, that re-starts are just 'normal' experiments that have already been run. Be careful that when changing from one base-config to another, the model re-compiles. Simply running the new configuration briefly (for even just a single year) is sufficient to ensure this. Experiments can then be safely submitted to a cluster queue, i.e. do not try and submit an experiment using a different base-config straight to the cluster queue without having run it (or a short version of the experiment you want) interactively first (to ensure the model is re-compiled). This is also good practice – checking that a new sort of experiment and/or model configuration works as you intend and without hiccups.

2.2.3 Continental configuration (2)

Although much useful can be learned from conceptual configurations and Worlds regarding climate dynamics, it is invariably aesthetically more 'pleasing' to also test ideas in a more paleogeographically realistic configuration. Provided is a set of base-config and user-config files (plus associated boundary conditions) for the position of the continents and climate 635 millions years ago (635 Ma). For this:

The base-config is named: muffin.C.fm0635cb.NONE

The user-config is: muffin.C.fm0635cb.NONE.SPIN

NOTE that the user-config lives in the directory: ∼\cgenie.muffin\user-configs\PALEO so that you either need to specify this different (PALEO) directory when running muffin, or copy the user-config file into LABS.

A re-start experiment is provided called muffin.CB.fm0635cb.NONE.SPIN and which can be downloaded as per before:

$ wget http://www.seao2.info/cgenie_output/muffin.CB.fm0635cb.NONE.SPIN.tar.gz

and unpacked by:

$ tar xfvz muffin.CB.fm0635cb.NONE.SPIN.tar.gz

(Remember: you should be in the cgenie_output directory when you do this downloading and unpacking.)

To run (e.g. for 100 years), following on from its re-start (and leaving the user-config in its PALEO directory):

$ ./runmuffin.sh muffin.C.fm0635cb.NONE PALEO muffin.C.fm0635cb.NONE.SPIN

100 muffin.CB.fm0635cb.NONE.SPIN

(all on one line)³

2.2.4 Geothermal heat input

Finally, muffin will fairly happily build up sea-ice, apparently without limit (with the remaining wet ocean becoming progressively colder and more saline). In the real world, one might expect some sort of limit to the maximum thickness achieved as the heat diffusion across a progressively greater thickness of sea-ice approaches the heat input at the bottom of the ocean from geothermal energy. Different modes of ocean circulation are also possible if one considers heating from the

³NOTE that the base-config and user-config filenames start muffin.C. whereas the re-start filename starts muffin.CB. ... just to try and trip you up ...
2.2 Further ideas

bottom as well as cooling (and brine rejection) from the top and which might affect the entry into or exit from a snowball state.

In the experimental setup you have been given, a geothermal heat input is specified in the ocean circulation module via the following:

\[ \text{bg\_ctrl\_force\_GOLDSTEInTS} = \text{.TRUE.} \]

\[ \text{bg\_par\_Fgeothermal} = 100.0 \times 10^{-3} \]

The parameter \( \text{bg\_par\_Fgeothermal} \) sets the geothermal flux in units of W m\(^{-2}\). (Note that in the Neoproterozoic, the geothermal heat flux could have been somewhat higher than modern. How much higher? A question for Google . . . ?)

An appropriate research question might be to determine in radiative forcing vs. geothermal space (and requiring a 2D grid of parameter combinations to be created and submitted to the cluster), the equilibrium sea-ice thickness and region in which a snowball solution is not possible. However, more simply and suitable to a short exercise: How much of a difference, to the estimated snowball entry and exit thresholds of radiative forcing, does the inclusion of a geothermal input make? E.g., what happens if you set it to zero? What about 10 times modern (or more, although *extreme* seafloor heating can cause numerical instability and the model to crash)?

2.2.5 Seasonality

By default, the idealized model configurations are non-seasonally forced (by solar insolation). You can switch to a seasonally-forced to model by adding the following lines to the user-config file:

\[ \text{ea\_dosc} = \text{.true.} \]

\[ \text{go\_dosc} = \text{.true.} \]

\[ \text{gs\_dosc} = \text{.true.} \]

The scientific question here in trying this would be whether or not taking into account a seasonally-varying climate substantially affects the entry (and/or exit) thresholds for a snowball climate state. (At least, whether it is important in the context of the resolution and physics of the model you are using.)

You can also save the data seasonally if you like – see Section 12.2.3 in the muffin User Manual (this document!).

---

\(^4\)For reference, your configuration has 24 time-steps per year set for the BIOGEM module.
3. Ocean circulation

Stuff to keep in mind:
- Nothing at all – keep your mind completely empty and let the wonderful truths of muffin permeate your entire being.

Background reading (Atlantic circulation and stability in muffin):
  → Simple assessment of the likelihood of AMOC collapse.
- Marsh et al. [2004] (Climate Dynamics, 23 2004, Pages 761 – 777)
  → Characterization of thresholds of AMOC collapse.
  → Role of changing ocean circulation in atmospheric radiocarbon variability during the Younger Dryas.

Background reading (Miscellaneous (model) Atlantic circulation and stability):
  → Provides the background to the Atlantic Meridional Overturning Circulation and hypothesized hysteresis.
- IPCC [2007] (e.g., Section 10.3.4)
  → Future predictions of AMOC strength.
  → Impacts on marine ecosystems and carbon cycling.
- Obata [2007] (J. Clim. 20, 5962–5976)
  → Climate-carbon cycle model response to freshwater discharge.
You will need to download a new *re-start* file prior to embarking on the experiments with modern ocean circulation. To fetch this: change to the *cgenie_output* directory, and type (or copy and paste carefully from the PDF ...):

```
$ wget --no-check-certificate http://www.seao2.info/cgenie_output/LAB_2.SPIN.tar.gz
```

This downloads an archived/compressed copy of the 10,000 year *spin-up* experiment LAB_2.SPIN. Extract the contents of this archive by typing:

```
$ tar xfv LAB_2.SPIN.tar.gz
```

You’ll then need to change directory back to *genie-main* to run the model.
3.1 Tracing ocean circulation

The ocean biogeochemistry module (BIOGEM) in muffin provides a framework for applying time- and spatially-variable ‘forcings’ of the Earth system\(^1\) – fluxes or restored-to boundary conditions that can be prescribed for any gas, dissolved substance (including temperature and salinity), or particulate matter. Examples include freshwater input (== a negative salinity flux forcing) of the North Atlantic to alter ocean circulation, fossil fuel CO\(_2\) emissions to the atmosphere (== a CO\(_2\) gas flux forcing), or aeolian iron supply to the surface ocean (a 2-D dust flux forcing).

For example: view the user-config file: LAB_2.colorinjection – you will see the following lines (under the heading: ‘# -- FORCINGS --’)

```plaintext
bg_par_forcing_name="pyyyyyz_Fred"
bg_par_force_point_i=22
bg_par_force_point_j=33
bg_par_force_point_k=8
bg_par_ocn_force_scale_val_48=0.0
```

The first line points muffin to a directory located in cgenie.muffin/genie-forcings that contains a set of files that define what geochemical property is going to be altered plus information about how the magnitude of the forcing changes with time.

There are then three lines (bg_par_force_point_i=20, ...) that specify the location in the ocean of the geochemical forcing that is going to be applied. The point sources are specified in (i,j,k) coordinates, which in this case is (22,33,08). For the ocean model resolution we are using, the grid is 36x36x16, and in which: longitude (i) is counted from left-to-right (1 to 36); latitude (j) is counted from bottom-to-top (1 to 36); level depth (k) is counted from downwards top-to-bottom (16 down to 1). Thus, (22,33,08) is a release of tracer in the North Atlantic, a little south of Greenland, and intermediate depth (level = 8 out of 16). Refer to the Figures for how the horizontal (Figure 3.1) and vertical (Figure 3.2) grid is specified.

Finally, there is a scaling parameter (bg_par_ocn_force_scale_val_48) which modifies the magnitude of the flux to be applied (in units of \(\text{molyr}^{-1}\)).

You are going to run a brief experiment in which you will be injecting a conservative ‘dye’ tracer into the ocean. The BIOGEM module has two tracers defined for this purpose – ‘blue’ and ‘red’. Open the user-config file: LAB_2.colorinjection and edit the parameter controlling the flux of red dye to read:

```plaintext
bg_par_ocn_force_scale_val_48=1.0E12
```

which specifies a flux of \(1.0 \times 10^{12}\) \(\text{molyr}^{-1}\) rather than zero as given as the default in the example user-config file\(^2\).

The base-config you will be using is different from previously: cgenie.eb_go_gs_ac_bg.worjh2.rb – this specifies a 16 vertical levels ocean and also includes seasonality of solar insolation.

---

\(^1\)Refer to the ‘force the system’ HOW-TO in the muffin manual for further details on forcings.

\(^2\)i.e. don’t leave the value as zero ... otherwise you’ll have no flux forcing applied and will not see anything happen ...
Figure 3.1: The muffin grid for a modern $36 \times 36$ ‘worhj2’ configuration. Light blue numbers are the ‘i’ co-ordinates. Green numbers are the ‘j’ co-ordinates. The depth of the ocean at any location is indicated by its ‘k’ value – a number between 1 and 16, with 16 being the surface layer of the ocean, and 1 the maximum possible depth anywhere. Numbers > 90 (91, 92, 93, 94) and shaded grey are land (and specify the direction of run-off). Location (22,33,08) is highlighted in yellow. The longitude of the western edge of this particular modern ocean grid is at 260W, and the increments are 10 degrees.

Figure 3.2: The muffin ocean vertical level definitions for a modern 16-level ocean grid.
Run the model for ... whatever, 20 years will do. Use the re-start experiment that you have just downloaded to start from:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.rb LABS
   LAB_2.colorinjection 20 LAB_2.SPIN
```

View the results – for instance how the red tracer distribution evolves with time – in the time-slice files (full ocean/atmosphere) properties saved in the netCDF format (.nc) files. You can follow the progress of the dye (and hence diagnose the properties of ocean circulation in the model) by plotting vertical and/or horizontal slices that go through (or near) the cell location in which you inject the dye tracer in the 3D netCDF file. Note that Panoply appears to ‘count’ the ocean layers in the opposite direction to the way in which the ocean model is actually counting them – the correct definition is with ‘1’ being very deepest level possible (and as displayed in the figure).

You can also view the tracer distributions in terms of a water-column integrated tracer inventory (netCDF variable name: `ocn_int_colr`; long name: `colr water-column integrated tracer inventory`) in the 2D netCDF output. (See: Sabine et al. [2004] for the use of water column integrals in the context of the distribution of anthropogenic CO$_2$ uptake and storage.) Changes in tracer inventory with time can be tracked in the time-series file `biogem_series_ocn_colr.res`.

You can also plot the overturning circulation from the 2D netCDF file – variable `phys_opsi` == global overturning streamfunction, `phys_opsia` == overturning in the Atlantic to provide a visualization of the large-scale ocean circulation that drives tracer movement.

Spend a little while altering the flux (`bg_par_ocn_force_scale_val_48`) and/or location (`bg_par_force_point_i`, `bg_par_force_point_j`, `bg_par_force_point_k`) of tracer input. Overall – note how you can use numerical ‘tracers’ to help diagnose (and better understand) the circulation of the ocean.

When you add the numerical dye, particularly early on in the experiment, you may see a ‘front’ of negative tracer concentrations leading the (positive) tracer as it spreads. DON’T PANIC!

The model is ... a model (of the numerical flavor) and not an exact analytical solution. So errors in how it solves ocean transport are to be expected.

Moreover, by default the ocean circulation model employs an isopycnal/diapycnal mixing scheme. This can lead to unwanted negative tracer values when sharp horizontal (or vertical) transitions in concentration occur. (In this example, e.g. by injection dye at a point location into a surrounding ocean of initially zero concentration.)

You can change to a simpler horizontal/vertical (.false.) mixing scheme by adding to the `user-config` file:

```
# turn OFF (=.false.) isopycnal/diapycnal mixing
go_diso=.false.
```

If you try this, you should (hopefully) find much less (or no) negative tracer concentrations occur. However, also note that by changing the physics of ocean mixing, you also slightly alter the large-scale circulation of the ocean (and e.g. the AMOC might change slightly in strength).

Finally, an interesting (honest!) and illustrative exercise is to use the dye tracer to pick out the path taken by Mediterranean Intermediate Water. Despite the low resolution of the muffin ocean
Chapter 3. Ocean circulation

circulation model component and the highly restricted representation of the Mediterranean, the model does predict a salty Mediterranean as a consequence of P-E in this basin (and its catchments) being negative and this water makes its way out in the subsurface into the Atlantic.

Simply specify a dye injection somewhere in the Mediterranean (be careful with the restricted depth of the Mediterranean – if you inject too deeply (into the crust!) then you will not see anything (refer to the figure for the depth level (k) number of the maximum depth of the water column in each location), and it is better to inject it relatively close to the opening of the gateway (try some different locations and see which ones produce a reasonably instructive tracing of Mediterranean outflow). Run for e.g., 20 or 50 years (from the provided spin-up). Then:

1. View the dye-tagged plume of Mediterranean Intermediate Water by plotting a lat-lon slice (from the 3D netCDF file). This will give you the depth of the plume. How does this compare with salinity observations (salinity observations and appropriate global datasets can be found on the web with a little patience)? You can also view the water-column integrated distribution (2D netCDF).

2. Try viewing the plume via a lat-depth slice. Refer to the figure to determine the ‘i’ value up the Atlantic that will just graze the edge of what passes for Spain at this low model resolution. Which direction does it head after exiting the Mediterranean? Is this ‘realistic’?
3.2 Poking the climate beast

Instead of adding a dye tracer, you could add fresh water to the ocean surface to assess the sensitivity of the Atlantic Meridional Overturning Circulation (AMOC) to collapse, in a classic ‘hosing’ experiment.

The user-config file for this is called: LAB.2.hosing. The default (i,j) location of the flux input is the same (as the dye tracer), but now the injection at the surface (level: k=16). Note that the forcing of the salinity tracer is negative (freshwater = negative salinity compared to sea-water)!

To orientate you in freshwater forcing space: bg_par_ocn_force_scale_val_2=-2.0E17 should be sufficient to make ’stuff happen’ and quickly. BUT, this is a pretty extreme flux (see overleaf for a rough conversion between salinity forcing units (mol yr\(^{-1}\)) and fresh water flux (in \(\text{m}^3\text{s}^{-1}\) or Sv). Much more than this and the model may crash or at the very least, you’ll be left with a large freshwater pond in the North Atlantic . . . (See later (Section 1.6e) for some exciting discussion on units!)

To run the model for e.g., 20 years using the same restart:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.rb LABS LAB_2.hosing 20 LAB_2.SPIN
```

20 years should be long enough to see a collapse start to occur, but you might want to run the model for longer (and it can be submitted as a job, of course). Running for longer will also allow you to have a smaller, less extreme (and maybe more realistic) freshwater input flux.

The most obvious property of the Earth system to follow is the Atlantic overturning strength (biogem_series_misc_opsi.res). The AMOC stream-function (in fields_biogem_2d.nc 2-D time-slice netCDF results file, field: phys_opsia) is also illustrative. You can also try and identify the salinity anomaly (see below) due to freshwater input in the 3D salinity tracer field.

There are also important (but not necessarily painfully exciting) impacts on surface air temperatures and maybe sea-ice extent (in fields biogem_2d.nc) (but see below for a better way to visualize these changes). Note the importance (sort of) of the AMOC in transporting heat to the N Atlantic region (the film the Day After Tomorrow was not entirely inaccurate in this particular respect). Be aware of the possibility of climate impacts far from the location of fresh water forcing. Look out for any significant-looking impacts on sea-ice extent, etc.

Note that as the model is running rather slower than in the snowball configuration, you might want to think carefully of making use of cluster queuing possibilities (i.e., running multiple experiments at once in the background).

Also – run a control experiment – an experiment of the same duration of your hosing experiments, but with a zero freshwater flux. The impact of freshwater input, is the difference, at the same model year, between the perturbation experiment and the control. (You only ever need need to run one control, regardless of how many different freshwater flux perturbation experiments you run.)

To more easily assess some of these impacts (and for other sorts of analysis) it is possible to create an anomaly (difference) map in Panoply:

1. First open a dataset, e.g., atm_temp (surface air temperature) in the 2D netCDF file. You can either double-click the variable name, or, with the variable name highlighted, click the ‘Create Plot’ icon.
2. Now, with the atm_temp still selected (and the first plot window still open), click on the ‘Combine Plot’ icon. A dialogue box will appear and ask you to select a plot to combine the
Chapter 3. Ocean circulation

Figure 3.3: Example plot of (normal/default modern) overturning streamfunction (2D netCDF file). (e.g., for Atlantic: netCDF parameter name: phys_opsia, long-name: Atlantic streamfunction). Note that autoscaling has been turned off and the min and max plotting limits set manually. By convention, streamfunctions are plotted with their scale symmetrical around zero, giving red and ‘warm’ colors for positive value and clockwise overturning, and blues and ‘cold’ colors for negative values and anti-clockwise overturning. (The plot has been tart-ed up by overlaying solid contours plus contour labels.) It may be necessary in Panoply to re-orient (invert) the vertical grid.

You can also plot ocean current fields which is sort-of fun and maybe even informative(!):

1. In the 3D netCDF file, the three components of ocean velocity are represented by the variables: ocean velocity – u (Eastwards), ocean velocity – v (Northwards), and ocean velocity – w (upwards). 2. Open up velocity – u. Chose ‘lon-lat’.
2. Select/highlight velocity – v. and click on the ‘Combine Plot’ icon (as per before).
3. Rather than a difference map, which is what you get by default, i.e., ‘Array 1 – Array 2’ – from the drop-down menu (next to the ‘Interpolate’ button) select ‘Vector Magnitude’.

---

new one with. Make sure the name of your first plot window is selected/highlighted. Click ‘Combine’. OR, simply drag a second dataset into the plot window of the first dataset.

3. You now have a plot window that by default it is showing you the difference between two identical (in time) slices. The two different slices are labeled Array 1 (LH side) and Array 2 (RH side).

Keep one array (Array 1) fixed to the initial (year 1 (centered on 0.5)) and vary the year in the second array (Array 2). Note that you can select in Panoply whether Array 1 – Array 2 is plotted, or Array 2 – Array 1, or various proportional or relative differences.

Note that you can switch off the auto-scaling feature (Always fit to data) and center the scale so that no change is white, with positive deviations = red and negative = blue by clicking on Center on 0 (an often used convention in climate field plotting).

Two example plots (using Panoply) are shown in Figure 3.3 for the Atlantic basin, and Figure 3.4 for the Pacific.
3.2 Poking the climate beast

4. You should have a color contoured (or not if you prefer plotting without contouring on) map of ocean current speed, with velocity vectors (direction and magnitude) overlain. You’ll need to re-scale the velocity vectors to properly see them – from the ‘Contours and Vectors’ tab – change the ‘Scale Length’ to e.g., 0.1. (On a Mac, look under the ‘Vectors’ tab for a ‘Reference Value’ to something like 0.1.) When fresh-water hosing – look out for impacts on the N. Atlantic current system associated with the AMOC.

5. You can repeat this for deeper depth levels in the ocean – e.g., between about 1500 and 2000 m is a good place to go looking for the Western boundary current (and AMOC return) in the model (such as it exists at this low resolution) but you’ll need to re-scale the velocity vectors again (e.g., to 0.01 to less).

An example plot (using Panoply for visualizing surface ocean current fields) is given in Figure 3.5.

Finally, a brief note on units ... the freshwater forcing is implemented as negative salinity, just to really screw with your mind. The generic internal muffin model units for the forcing end up as $PSU kg^{-1} yr^{-1}$. Which sort of does not make much sense ...

Start, by thinking of a value of $bg_{-}par_{-}ocn_{-}force_{-}scale_{-}val_{-}2$ of $-34.9$ as equivalent to taking all the salt out of 1 kg of freshwater (since the mean global salinity is 34.9 PSU). Or equivalently, since the ocean volume is fixed, an applied forcing value of $-34.9$ is equivalent to adding 1 kg of freshwater to a (surface) box. So, a value of $bg_{-}par_{-}ocn_{-}force_{-}scale_{-}val_{-}2$ of $-3.49 \times 10^4 (-3.49E04)$ would be a flux of 1 m$^3 yr^{-1}$ (1000 kg m$^{-3}$) of freshwater.

So, in the example earlier ($bg_{-}par_{-}ocn_{-}force_{-}scale_{-}val_{-}2=-1.0E18$), the freshwater flux is $1.0 \times 10^{15} / 3.49 \times 10^4 = 2.8653 \times 10^{11} m^3 yr^{-1}$.

The literature invariably gives freshwater fluxes in units of $Sv$ ($10^6 m^3 s^{-1}$). So in the example, the freshwater flux is: $9.0797 \times 10^5 m^3 s^{-1}$ ($365.25 \times 24 \times 3600 = 31557600 s y r^{-1}$). Or 0.9 $Sv$. Read the literature ... but generally, fluxes of ca. 0.05 $Sv$ and larger (and to quite specific places) are applied in models to induce an AMOC collapse.
Chapter 3. Ocean circulation

Figure 3.5: Example plot of (normal/default modern) ocean current fields (3D netCDF file). Again scaling has been set manually to create an easy-to-interpret axis scale. On the left is the surface field, and on the right an intermediate depth (illustrating what approximates the Deep Western Boundary current in the model in the Atlantic).

3.3 Further ideas

3.3.1 Hosing investigations

What is the largest freshwater flux that can be sustained without ‘collapsing’ the AMOC? Is there a ‘threshold’ (‘ tipping point’) of freshwater input, beyond which the AMOC rapidly decreases in strength? For this – you would run (submit to the cluster) a series of experiments, each with different (increasing) values for the fresh water flux. Remember to include an experiment with zero freshwater flux to act as a ‘control’.

Is the precise location of the freshwater input important (i.e., try tipping it in somewhere else)? For this – you could piece-meal run an experiment, analyse the results, then choose a new location to input the freshwater, or, come up with a systematic search pattern of freshwater input patterns.

Outside of the North Atlantic, are any other major regions of deep water formation (where are they) sensitive to freshwater perturbation and what are the consequences (could it happen in the future)?

3.3.2 ‘Anti-hosing’ investigations

There are questions concerning past changes in the AMOC as to whether it is ‘pushed’ or ‘pulled’. i.e., if the AMOC shoals in depth and/or weakens, is it because its production has weakened, or as Antarctic Bottom water (AABW) strengthened and ‘pushed’ it out of the way (to shallower depths)?

What you might try then is to inject salt in the Southern Ocean as opposed to fresh water in the North Atlantic. All you need do is pick an appropriate grid point (this is worth thinking about carefully and maybe testing different locations) and rather than giving the parameter bg_par_ocn_force_scale_val_2 a negative value, you give it a positive one. (Start by trying similar magnitudes of value as before and see what happens.)

Is the AMOC (for the same magnitude of forcing) more sensitive to being ‘pushed’ or ‘pulled’? (Obviously the answer will very much depend on where the perturbations are being applied.)
3.3 Further ideas

3.3.3 Response to transient warming

A current concern regarding anthropogenic climate change is the ocean circulation (and marine ecological and biogeochemical) response to a strong warming of the surface, as rapid surface warming is assumed (and demonstrated in models) to result in surface stratification of the ocean, likely restricting the nutrient supply to phytoplankton and reducing ventilation of the ocean interior with dissolved oxygen.

You can explore the transient response of ocean circulation to warming by simply adjusting the radiative forcing parameter as per used in the snowball Earth experiments: ea_radfor_scl_co2. By default in the modern continental configuration, this has a value of 1.0, corresponding to 278 ppm atmospheric CO₂. A value of 2.0 would reflect warming equivalent to 556 ppm CO₂. And 3.0 more like an end-of-the-century warming. Note that you are applying the warming instantaneously by manipulating the climate system in this way and hence the changes will be more extreme than those occurring over the time-scale of this century. Also note that a cooling could be applied instead. A user-config — LAB_2.EXAMPLE — is provided as a template for these experiments.

Potentially interesting properties of the Earth system to look at include sea-ice extent and AMOC strength (in the ASCII time-series files), and the overturning stream-function and sea-ice extent in the 2-D netCDF output.

How much radiative forcing is required to collapse the AMOC? What atmospheric CO₂ value does this approximately correspond to?

3.3.4 Ventilation age tracers

muffin has the capability to employ/simulate a ventilation age tracer³ – a numerical tracer in the ocean that tracks the time since a parcel of water last 'saw' the surface. To do this, requires both 'red' and 'blue' numerical tracers to be selected. muffin can then be directly to automatically apply the forcings necessary to each tracer in order to simulate water mass ventilation age.

Provided is an additional user-config file: muffin.worjh2.NONE_colage.EXAMPLE.

A new re-start experiment is provided called muffin.worjh2.NONE_colage.SPIN and which can be downloaded as per before:

```bash
$ wget http://www.seao2.info/cgenie_output/muffin.worjh2.NONE_colage.SPIN.tar.gz
```

and unpacked, as before, by:

```bash
$ tar xfvz muffin.worjh2.NONE_colage.SPIN.tar.gz
```

(Remember: you should be in the cgenie_output directory when you do this.)

To run (e.g. for 10 years), following on from its re-start experiment:

```bash
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.rb LABS muffin.worjh2.NONE_colage.EXAMPLE 10 muffin.worjh2.NONE_colage.SPIN
```

(all on one line)

In the 3D netCDF output file – misc_col_Dage is the output variable that is the calculated ventilation age.

First — explore the distribution of water mass age and think about the physical ocean circulations reasons for this. Are all the modelled distributions reasonable? Are there indicators of facets of the

³Under the 'screw with and/or diagnose the climate system' HOW-DO – see 'Add a water mass age tracer' (and the 'easy'/automatic method described towards the end of that section).


simulated circulation that are not particularly realistic? Try plotting both lon-lat and lat-depth slices through the ocean.

Secondly – you could additionally combine this experiment configuration with fresh water hosing, or a prescribed radiative forcing (and warming), and explore how ocean ventilation responds as ocean circulation changes. (You’ll need to create for yourself a new forcing directory and set of files that now combines both color tracers AND a freshwater flux.)
This Chapter comprises 2 different exercises, both employing ‘realistic’ reconstructions of past continental configuration and climate and designed to explore some of the key controls on global surface climate as well as the spatial pattern of surface temperatures.

The two exercises concern the late Cretaceous and early Eocene climate states, and are somewhat interchangeable.

**Background reading (Cretaceous climate)**

- Hunter, S. J., et al., Modelling Maastrichtian climate: investigating the role of geography,


**Background reading (Eocene climate)**


You will need to download new restart files for this Chapter (both 10,000 year spin-ups). These are:

**Cretaceous:** $ wget http://www.seao2.info/cgenie/labs/AWI.2017/LAB_5.SPIN.tar.gz

**Eocene:** $ wget http://www.seao2.info/cgenie/labs/AWI.2017/LAB_5b.SPIN.tar.gz

Extract the results in the usual way and in the usual place ... and return to genie-main.
4.1 The climate of the Cretaceous

A previously spun-up state of Maastrichtian climate with cGENIE.muffin in a ca. 70 Ma configuration\(^1\) (LAB_5.SPIN) is provided as a starting point. A user-config (LAB_5.EXAMPLE) that continues on this climate state is run:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.p0067f.NONE LABS
   LAB_5.EXAMPLE 10 LAB_5.SPIN
```

Your task now is ... ‘simple’: Account for the Cretaceous reduced Equator-to-pole surface temperature gradient (compared to modern), particularly the apparently much warmer poles. Different data-based time-slices (including the Maastrichtian) are provided in Huber et al. [2002], although the low latitude \(\delta^{18}O\)-based temperatures are now not considered reliable. Jenkyns et al. [2004] contains high latitude (Arctic) data for the Maastrichtian. There are proxy-derived latitudinal temperature gradients and model-data studies for earlier in the Cretaceous – the problem is essentially the same.

The 2-D NetCDF results file contains the surface air temperature field (and sea-ice cover, if any). The 3-D NetCDF results file contains fields for ocean temperatures (and salinity). Both contain continental configuration and ocean bathymetry.

Panoply will plot the zonal average for you (as used in model-data comparisons – e.g., see Bice and Norris [2002]) – in the Array(s) tab, the Plot can be set to Zonal Averages rather than Map. You can get a smooth curve by selecting Interpolate. Remember you can set (and fix) scales rather than let Panoply auto-scale continually.

The following ‘controls’ over the climate system are provided to you in the form of a list of parameters at the bottom of the LAB_5.EXAMPLE user-config file for editing:

```
# --- MISC -----------------------------------------------
#
# === ATMOSPHERE ===
# \(CO_{2}\) radiative forcing scaling factor [DEFAULT = 4.0]
ea_radfor_scl_co2=4.0
# CH4 radiative forcing scaling factor [DEFAULT = 1.0]
ea_radfor_scl_ch4=1.0
# Equator-to-pole different in planetary albedo [DEFAULT = 0.260]
ea_albedop_amp=0.260
# Baseline planetary albedo [DEFAULT = 0.200]
ea_albedop_offs=0.200
# atmospheric diffusivity of temperature (horizontal) [DEFAULT = 5.0e6]
ea_12=5.0e6
# === OCEAN ===
# ocean diffusivity of temperature + salinity (horizontal) [DEFAULT = 1494.4]
go_14=1494.4
# scaling for wind stress (set values of both identical) [DEFAULT = 1.531]
go_13=1.531
ea_11=1.531
```

Most of these parameters are associated with the radiative forcing of climate or atmospheric transports. The most useful ones are likely to be:

\(^1\)Note that for speed, no carbon cycle is selected in this configuration.
4.1 The climate of the Cretaceous

(i) The line:

\[ \text{ea_radfor_scl_co2}=4.0 \]

which specifies a radiative forcing of climate by \( CO_2 \) equivalent to 4 times modern \( CO_2 \) (i.e., \( 4 \times 278 \text{ppm} = 1112 \text{ppm} \)) as per before (e.g. snowball Earth experiments).

The line:

\[ \text{ea_radfor_scl_ch4}=1.0 \]

specifies a radiative forcing of climate by \( CH_4 \) equivalent to 1 times modern \( CH_4 \) (i.e., \( 1 \times 700 \text{ppb} \)).

(ii) cGENIE, as configured here, does not have a land surface scheme (no snow cover) nor clouds nor ice sheets, so a planetary albedo is prescribed (see Figure 4.1). This varies with latitude and is parameterized after a fully coupled GCM simulation. There is a parameter which controls how the albedo varies as a function of latitude, which can be adjusted:

\[ \text{ea_albedop_amp}=0.260 \]

However, if you vary this, why are you doing it? (What is the physical justification for giving the poles a higher or lower albedo relative to the Equator?) There is also a parameter that sets the baseline (minimum) albedo:

\[ \text{ea_albedop_offs}=0.200 \]

i.e., albedo is primarily a sum of the baseline value plus the Equator-to-pole slope (times the latitude).

(iii) The diffusivity of heat in the atmosphere (since it is a simple 2-D atmospheric model, with no atmospheric circulation, the atmosphere is made ‘diffusive’ to help capture heat and moisture transport) is:

\[ \text{ea}_12=5.0e6 \]

Note the maximum value the model can cope with is ca. \( 1.0E7 \).

Run the model for however long you think is ‘necessary’ (/justified). The surface climate will approach equilibrium ‘relatively’ quickly. Deep ocean temperatures will typically take thousands of years to fully adjust ... You can assess how the model approaches equilibrium most easily from the atmospheric temperature time-series results file, and from the ocean temperature time-series results file (allowing to you to contrast surface and whole ocean temperature changes).

You can also try tracing and analyzing the patterns of ocean circulation in the Cretaceous world, in the same way as you did for the modern climate system. And in fact, it would be a useful exercise to directly compare your previous modern results vs. Cretaceous results, or even better, carry out a set of paired experiments (e.g., similar locations and fluxes of dye injection) for both modern and Cretaceous. (Note that if you do this and swap back and forth between modern and Cretaceous configurations, you cannot submit straight away to the cluster but must first briefly run

\[ \text{But effectively, this is going to do exactly the same as changing radiative forcing due to } CO_2. \]
Chapter 4. Climates of Past Worlds

Figure 4.1: Prescribed planetary albedo. The latitudinal (from 90°S (-90°N) on the left, to 90°N on the right) profile of planetary albedo as calculated in a fully coupled GCM is given in green, and the cGENIE ‘fit’ in black.

the experiment at the command line in order that cGENIE is compiled into the new continental configuration.)

A new user-config (LAB_5.colorinjection) that continues on from this climate state, can be run:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.p0067f.rb LABS
LAB_5.colorinjection 10 LAB_5.SPIN
```

(Also note that the base-config is different as it now includes the definition of 2 dye tracers.)

The default locations for the dye release are set different compared to in the modern configuration. BUT, the default location is not necessarily ideal / particularly revealing ... so you’ll need to look count grid cells West-to-East (the i direction) and from South to North (the j direction) in order to determine a more suitable location for tracing circulation.

By means of dye tracing, looking at the (global only) overturning stream-function, and/or temperature and salinity (and density) profiles, see if you can identify where in the Cretaceous ocean deep water forms in the model. AND, more importantly, think about WHY does it form where it does?

Also: what about surface ocean circulation and gyres? Are these located where you would expect (e.g. based on your understanding of modern ocean circulation). Plot velocity vectors to help, and/or refer to the Barotropic streamfunction output in the 2D netCDF file.
4.2 The climate of the early Eocene

A previously spun-up state of early Eocene climate with cGENIE in a ca. 56 Ma configuration (LAB_5b.SPIN) is provided as a starting point. A user-config (LAB_5b.EXAMPLE) that continues on this climate state is run:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.p0055c.NONE LABS LAB_5b.EXAMPLE 10 LAB_5b.SPIN
```

(Note that once again, for speed, no carbon cycle is selected in this configuration.) Again, as per for the late Cretaceous configuration, a configuration that includes the ocean dye (color) tracers is also provided. The corresponding user-config (LAB_5b.colorinjection) that continues on this climate state can be run as follows:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.p0055c.rb LABS LAB_5b.colorinjection 10 LAB_5b.SPIN
```

What to ‘do’ with this? Again – start by tracing and analyzing the patterns of ocean circulation in the Eocene world, as per how you did for the modern and Cretaceous system. Where does deep water form in the Eocene? Is the surface ocean circulation as expected? Can you see any patterns emerging? At least in model world – what seems to be dictating the locations of deep water formation? Does overturning and/or surface circulation appear to significantly vary as a function of continental configuration or radiative forcing? If not, why not?

But how do you know how close, or not, the model climate is to ‘reality’? Available from the website, on the LH side of the page under ‘got data?’ at the bottom (item (7)), are 2 netCDF files containing proxy sea surface temperature (SST) data re-gridded to the cGENIE model grid. One is for pre-PETM SSTs, and one for peak PETM SSTs. (The data are from Dunkley Jones et al. [2013].)

These netCDF files can be opened and visualized in exactly the same way in exactly the same way as per the cGENIE output files. You’ll see a few (there are not many!) data points, in a sea of grey (which stands for ‘no value’). (Note that it is best to turn data interpolation OFF.)

More useful . . . is that the data distribution can be combined with the model ocean temperature output field, in a difference map (see earlier for notes on creating difference maps). You can thus visualize the pattern of model-data mismatch. (Note you might want to hit ‘Fit to data’ to autoscale the plot, or simply pick and manually enter a +/- limit for the data plotting, perhaps ensuring it is symmetrical about zero such that red anomalies could represent proxy SST values higher than in the model, and blue point data being lower than in the model.)

Furthermore, in the Arrays tab, you can switch to a Zonal Average view rather than a Map view (...it is probably easier to visualize the model-data misfit in this way.)

So, starting with the pre-PETM data and given model configuration, make some assessment of how the model fits the data (or not). What might be the reason for the misfit? You might test adding, and adjusting, some of the parameter values controlling surface SST from the Cretaceous climate exercise. Can you reduce the model-data misfit?
Finally, the PETM SST data given in the 2nd netCDF file are from the peak of the PETM. The ocean is rather warmer as compared to just prior to the PETM. Your task is to determine (either using the default model configuration, or your adjusted configuration) much radiative forcing (parameter `ea_radfor_scl_co2`) and hence CO$_2$ is required to explain this warming?
Fake (alternative) Worlds are a useful tool for developing a general understanding of how climate (and ocean circulation) and global biogeochemical cycles operate. Different model resolutions and continental configurations can be generate and run to test specific hypotheses about the role and sensitivity of different elements in Earth system to emergent patterns of climate (and biogeochemical cycling). Ensembles of differing Worlds can also be utilized to more generalize understanding as well as in a fishing expedition of new or unexpected phenomena and system behaviours.

This Chapter will take you through the generation and exploration (analysis) of fake (alternative) Worlds.
5.1 Generating fake Worlds

TO COME ...
5.2 Understanding ocean circulation

What follows, outlines different ways of visualizing and quantifying ocean circulation in muffin, and coming to a model-empirical understanding of ocean circulation controls.

5.2.1 Model investigation overview

In restricting model-based experiments and derived understanding to the modern continental configuration (and climate) in muffin, you’d be unlikely to come to any particularly deep or fundamental insights into ocean circulation controls in general. In any case, the modern and future (and recent glacial past) ocean circulation states and dynamics has been picked at endlessly by 1000s of researchers, writing 100,000s of publications ... and yet still we do not fully understand modern ocean circulation, let alone that at the last glacial.

A better (more fun) approach is to generalise the problem and consider fundamentally different configurations of ocean basins and climate. This is what you will be doing. But first, some potted advice:

- Try and define a hypothesis, or hypotheses, to pursue and test. These need only be very rough at the outset – you’ll likely find that you get new ideas and can either refine your original hypotheses, or come up with new ones, as you start to play with the model and see what is feasible and not feasible in terms of analysis.
- Create a plan of study – what sort of experiments, now many, how are they going to be analysed, how do they all fit together in addressing the overarching hypotheses? Again – it is very likely that your plan will evolve, but try and start out with something to guide you rather than wandering randomly in model world ...
- For creating and an using different worlds – create a list or table of the configurations to be explored, and summarize what you are finding. It might be helpful to plan out on paper first, some of the main continental configurations you need to create and then test in the model. (Again, this will evolve.)
- Once you have some sort of idea what are are going to do – plan the work. This is important because to run the model to steady state is not trivial. Perhaps plan on having to run the model for 5,000 years in order to achieve a fully spun up ocean circulation state. You might get away with shorter runs, but know in advance what sort of error this would induce and is it 'important'? Once you have a list and know how long each might take, you can plan out when they will be run on the cluster (presumably on the queue) and when the results will be analysed. Note that it is a virtual certainty that once you have analysed the results of the first set of experiments, you will want something different (and will then need to revise the plana and list of experiments and analysis etc.).

Overall – scientific investigations with simplified (and relatively fast) Earth system models tends to become a somewhat iterative undertaking and can involve significant trail-and-error. (This is healthy and to be expected and even enjoyed!) Rarely, can you devise at the outset, a single run or set of experiments, and it turns out this is completely sufficient. If you do not see something you do not understand in the results of the initial model experiments, you are either God (e.g. the Spaghetti Monster or Invisible Pink Unicorn), or not doing it properly. Expect to see things that interest you

---

\(^1\)Even so, you should try running the model much longer to be confident that 5,000 years is OK (and sufficiently close to final equilibrium).
and lead you off on a tangent (hopefully – this is how research should be).

Finally – note that in swapping between different continental configurations, muffin currently requires the model executable to be re-compiled. This ... cannot happen on the compute nodes of the cluster if you submit an experiment using a different base-config. When changing to a new base-config: first, run the model briefly interactively (i.e. at the command line). Once it has compiled (and started running), the experiment can be killed (\texttt{Ctrl-C}) and now it is good to submit to the queue as a job. In any case, in utilizing a new configuration that you may never have used before, it is good practice to test it first (e.g. watch it run for a short period).

\subsection*{5.2.2 Deep-water formation (& baroclinic circulation)}

The first investigation is to investigate the controls on the strength and large-scale structure of ocean circulation, particularly in terms of the global meridional overturning circulation (MOC), and associated with this – the primary sites of deep-water formation.

There are presumably 2 key controls to this:

1. **Temperature.**
   
   This is presumably mostly a function of latitude (i.e. the further North or South, the more suitable the site will tend to be for deep-water formation) and to some extent season. There will also be an influence of the prescribed planetary albedo (which includes the effect of clouds) as well as of sea-ice (if it exists).

2. **Salinity.**
   
   This will generally be controlled by P-minus-E – the balance between precipitation to the ocean surface as well as fresh-water run-off from the continents, vs. evaporation from the ocean surface. Sea-ice may also be a key factor, in e.g. leading to the (seasonal) rejection of dense salty brine (and removal of freshwater to the forming ice).

In turn, this suggests two sets of 'knobs' in the model that influence the patterns and magnitude of temperature and salinity across the ocean surface:

1. **Continental Configuration.**
   
   The configuration of the continents and ocean basins will dictate the shape and location of the highest latitude ocean regions – presumably the locations where on average, deep-water formation will occur (and hence forming the downwards/sinking limb of the MOC).

   The relevant model 'knob' is hence how the position and orientation of continental blocks is configured on the surface of the Earth. This (creating or changing the continental configuration used by muffin) can all done using the MATLAB muffingen program.

2. **Global Climate.**
   
   The mean and climate as well as the zonal temperature gradients, together with whether or not sea-ice forms (and how much), will modulate both temperature and salinity patterns at the ocean surface.

   The model knobs here are primarily:

   (a) Atmospheric $pCO_2$, or in the absence of an explicit carbon cycle, a prescribed radiative forcing:

   \[ ea\_radfor\_scl\_co2=1.0 \]

---

\footnote{Note that if the \texttt{base-config} is the same as the previous experiment, but you have changed a parameter value (in the \texttt{user-config}), you do not need to re-compile.}
which here, specifies $\times 1$ $CO_2$ equivalent radiative forcing.

(b) One could also adjust the value of the solar constant (here, given with its modern/present-day default value):

\[
\text{ma\_genie\_solar\_constant}=1368.0
\]

which has a subtly different effect form changing $CO_2$ radiative forcing, as radiative forcing has a relatively spatially uniform impact, whereas changing the solar constant has a disproportionate impact towards the Equator (where the incident solar shortwave radiation is the greatest). So changing the solar constant is likely to impact the pole-to-Equator temperature gradient. e.g. see: Lunt, D. J., A. Ridgwell, P. J. Valdes, and A. Seale, Sunshade World.: a fully coupled GCM evaluation of the climatic impacts of geoengineering, GRL 35, L12710, doi:10.1029/2008GL033674 (2008).

(c) One could also adjust the planetary albedo, particularly in respect of looking to modify the pole-to-Equator temperature. In the muffingen created configurations, there is an explicit 1D file specifying the zonal (with latitude) planetary albedo profile:

\[
xxxxxxxxx.albd.dat
\]

where $xxxxxxxxx$ is the 8-character name of the muffingen created World. It would be a simple matter of editing the values in the file (between the range of 0.0 and 1.0, for perfectly absorbing, and perfectly reflective, respectively).\(^3\) Note that the file data order is North-to-South (going from top to bottom in the file).

\[5.2.3 \text{ Gateways and barotropic flow}\]

The overarching question in the context of ocean gateways and barotropic flow, is a little harder to define succinctly; it has to do with the details of the degree of alignment (or not) of the prevailing wind (stress) with gateways, and the degree to which this give rise to strong zonal ocean flow. A good and obvious modern example is the existence of the Drake Passage, between the northern tip of the Antarctic Peninsular, and the southern tip of Southern America, how this aligns with the prevailing Westerlies in the Southern Hemisphere, and hence the nature and strength of the ACC (Antarctic Circumpolar Current).

So the question naturally arises: how misaligned does a gateway have to be relative to the prevailing wind stress maximum, before the circumpolar (or circum-equatorial) flow ceases. What about having multiple gateways and their relative alignment, including what happens if one gateway is aligned with Westerly wind stress, and a second with Easterly wind stress – who ’wins’)? Also – what about sill depths? For the Drake Passage, the ocean floor, while tectonically messy, is generally relatively deep. What happens to ocean circulation with a progressively shallow sill depth?

The 3 key controls in this exercise are then:

1. **Gateway alignment.**
   The degree of alignment (or latitudinal correspondence) of a gateway with the maximum of the zonal wind stress. (For multiple gateways, what if one aligns with a wind-stress maximum of the opposite sign?)

2. **Gateway width.**
   The width of a gateway (and how important this is in exerting control on ocean circulation).

\[\text{\(^3\)Best to make a copy of the original file before you modify it.}\]
Chapter 5. Alternative Worlds

3. **Sill depth.**
   The sill depth! This could be uniform in depth across the gateway (easiest/best) or could be varying (probably not so easy to learn anything).

   The importance of the position of the gateways is primarily only in the context of the position of the maxima in the wind-stress field. In theory, the shape of the assumed zonal wind-stress could be altered ... but the wind stress needs to be defined on 2 different ocean grids ... and this gets messy ...

   As before, there are a number of changes that can be made in the model to explore the consequences of varying gateway position and sill depth (and knobs, turned):

1. **Land-sea mask.**
   The model knob is hence how the width and location of gateways has been defined in the land-sea mask. There is also the question of how many gateways (and their respective position).

   As before, this (creating or changing the continental configuration used by muffin) is all done using the MATLAB muffingen program. Any of the given example muffingen configurations: drakeworld, eqpasworld, ridgeworld, or waterworld, could be taken as starting points – copying and renaming the configuration .m file, as well as the .dat file in the INPUT directory. Or start from scratch and a the ‘blank’ (initially all ocean) example configuration.

2. **Ocean bathymetry.**
   The bathymetry (ocean floor depth) can be changed to alter the sill depth. The easiest way to do this is probably within the muffingen editor, ensuring the following input parameter setting is set:
   
   ```plaintext
   opt_user=true; % [false/true] enable user input to grid
   ```
   
   in the configuration .m file. (One might also then ‘draw’ in the gateways by hand at the same time.)

   Note that a gap between 2 land masses must be 2 or more cells wide, to count as a gateway and allow barotropic flow in muffingen).

3. **Wind stress strength.**
   Although it is messy to attempt to edit the profile of the applied wind-stress field, it is possible to scale its impact on ocean circulation lower and higher. The muffin parameter for this is:
   
   ```plaintext
   go_13=1.531013488769531300
   ea_11=1.531013488769531300
   ```

   Somewhat bizarrely ... it appears twice ... with different parameter names. Both parameters must be changed to the same value. Place these lines (of the new parameter value assignments) in the user-config file for the experiment. Higher values result in a stronger applied wind-stress on the ocean surface.

   Note that although this is the simplest change to make, it may not have such a clear scientific question associated with it (other than the obvious and trivial).

5.2.4 **Analysis**

How do you ’judge’ the strength and characteristics of deep-water formation, global overturning, and circulation patterns and strength, in general, or their climate (or biogeochemical) impacts?
5.2 Understanding ocean circulation

This is not a trivial question. Often, additional ocean tracers are employed in models to generate a quantitative measure of the age of a parcel of water (mean time since it last saw the surface), of some measure of the efficiency of ventilation, or large-scale transport at the surface or at depth.\(^4\) Or rather involved analysis of ocean physics and transport might be employed.

One starting point is to read some of the literature where the climate properties of various hypothetical worlds have been investigated. Such as:


In addition, the sub-sub-sections that follow outline some simple diagnostics and ways of going about some quantitative analysis.

**Simple/global diagnostics**

- In the biogem output results folder, there is a time-series file named: `biogem_series_misc_opsi.res`
  
  This contains a summary of the evolution with time, of the minimum and maximum (anywhere) global overturning stream-function values.

- Another simple property of the climate system that you might consider, is the pole-to-equator temperature gradient, both in terms of atmospheric temperature, and ocean surface temperature\(^5\) (although they should presumably be closely coupled).

  Why? Because the large scale (overturing) circulation of the ocean should be transporting heat from the high latitude surface to the deep ocean. This presumably would act to reduce latitudinal temperature gradients. In contrast, a strong zonal slow might prevent latitudinal transport of heat.

**Using Panoply**

In a physics-only (T + S tracer) configuration of *muffin* you are somewhat limited in what you can look at. A good starting point is the previous tutorial on ocean circulation and AMOC stability (in the context of the modern world, climate and continental configuration). For instance, you know already how to plot and visualize the MOC using Panoply. The Atlantic basin and hence the existence of an AMOC, is pretty specific and unique to the modern world, so likely you’ll need to focus on the global MOC. (A good starting point is to familiarise yourself with the pattern and intensity of the modern.)

For example – the *drakeworld* configuration gives rise to a global MOC pattern as shown in Figure 5.1.

This is stored as the variable `phys_opsi` in the netCDF file `fields_biogem_2d.nc`. Remember that the first time-slice plotted in Panoply is the first time-slice saved and it may also be up-site-down(!) by default :(.  

\(^4\)We’ll see such tracers employed later in the course.

\(^5\)Panoply has an option for plotting the zonal mean, from which you could read off pole-to-equator temperature gradients.
Figure 5.1: Drakework world overturning circulation.

Figure 5.2: Surface current velocity field (arrows) plus speed (color scale).

You can also plot the barotropic circulation using **Panoply** – the variable is called `phys_psi` – this is shown in Figure 5.3.

Finally in **Panoply**, you might plot the current field (surface or otherwise), as per Figure 5.2.

**Using MATLAB ...**

**TO COME ...**

**Via color tracing**

Numerical color tracers can be used (as you have seen previously) to trace flow-paths. The default `base-config` files provided as part of the **mufflingen** software, define a configuration with only 2 tracers in the ocean – T and S. To add red and blue dye tracers, in the `base-config` file, find the section marked:

```plaintext
# *******************************************************************
# TRACER CONFIGURATION
# *******************************************************************
# the total number of tracers includes T and S
```
5.2 Understanding ocean circulation

Figure 5.3: Drakework world barotropic circulation.

You need to make several (but simple) edits here:

1. Firstly, change the number of tracers from 2 to 4, in line:
   
   ```
   GOLDSTEINNTRACSOPTS='$(DEFINE)GOLDSTEINNTRACS=2'
   
   i.e. to make:
   
   ```
   
   ```
   GOLDSTEINNTRACSOPTS='$(DEFINE)GOLDSTEINNTRACS=4'
   
   2. Secondly, you need to list the additional list selected biogeochemical tracers. For this add the following code in place of the (<<< >>>) line:
      
      ```
      gm_ocn_select_48=.true.
      gm_ocn_select_49=.true.
      
      3. Lastly, under the section list biogeochemical tracer initial values, and in place of the (<<< >>>) line:
         
         ```
         bg_ocn_init_48=0.0
         bg_ocn_init_49=0.0
         
         To apply a 'red' dye tracer to the ocean, you can employ the same forcing as used in the experiments in Chapter 3 – pyyyyz.Fred and implement this with similar user-config settings:
         
      ```
      #
      # --- FORCINGS -----------------------------------------------
      #
      bg_par_forcing_name="pyyyyyz.Fred"
      bg_par_force_point_i=1
      bg_par_force_point_j=1
      bg_par_force_point_k=1
      bg_par_ocn_force_scale_val_48=0.0
      ```
      
      However, the specific i, j, k (longitude, latitude, ocean depth layer) model grid locations will depend on the resolution of the World you generated or had adopted, and it may have e.g. 18 or 36
maximum i, j values, and generally, either 8 or 16 maximum depth levels (k). (The values shown would be for the southern and western corner of the grid at the bottom of the ocean ... but note that this location might be land or seafloor in your particular World!)

Via age tracing
Refer to the 'HOW-TO' in the muffin manual – 'Add a water mass age tracer':

"Water mass age tracing can be configured quite simply, and with the tracer concentration manipulations carried out automatically, as follows. As base-config file with both red and blue color tracers defined is still required, but rather than have to define and use a set of forcings (and associated forcing configuration), a single parameter is added in the user-config file:

bg_ctrl_force_ocn_age=.true.

This will automatically create the age tracer and additional explicitly output (in netCDF) both the total age of a water parcel, as well as the age relative to the surface (ventilation age). In this methodology, in the netCDF output, the concentration ratio of blue/red, should be 'age' – the mean time that a parcel of water was last at the surface."

Note that an experiment spin-up duration of perhaps as much as 10,000 years is required in order to obtain a stable tracing of water mass age.
Exploring the consequences of fossil fuel CO$_2$ emissions

- Idealized emissions forcing
- Historical (real-world!) emissions forcing

Further ideas

- Assessing the importance of emissions rate
- Determining thresholds of environmental impact
- Future atmospheric CO$_2$ concentration pathways ("RCPs")
- Isotopic tracing of fossil fuel CO$_2$ uptake

6. Fossil fuel CO$_2$ and ‘ocean acidification’
Readme

You will need to download a new restart file prior to embarking on the experiments. This pre-industrial spin-up includes a basic ocean (-atmosphere) carbon cycle plus various diagnostic anthropogenic tracers, following Cao et al. [2009].

To fetch this: change to the cgenie_output directory, and type:

$ wget http://www.seao2.info/cgenie_output/LAB_3.SPIN.tar.gz

Extract the contents of this archive by typing:

$ tar xfv LAB_3.SPIN.tar.gz

(and then change directory back to genie-main to run the model)
6.1 Exploring the consequences of fossil fuel CO$_2$ emissions

For the next experiment(s) you can chuck CO$_2$ into the atmosphere, just for the hell of it. As much as you want! Apparently, humans are actually doing this now. Imagine that!

A new user-config for muffin – LAB_3.CO2emissions – is provided and configured with climate being responsive to any changes in atmospheric CO$_2$ (i.e., it takes account of CO$_2$-climate feedbacks):

# set CO2-climate feedback
ea_36=y

Note that although a biological scheme is set up (with the fixation of organic carbon at the surface and sinking into the ocean interior, plus a rate of calcification by plankton at the surface ocean that is responsive to ocean acidification and saturation state (i.e., it takes into account CO$_2$-calcification feedbacks, which will additionally interact with climate – see Ridgwell et al. [2007b, 2009])...), we will concentrate only on the carbon geochemistry. (A 'biological pump' in the ocean is set up simply to create a modern-like ocean surface carbonate chemistry. How the biological pump actually operates, response to changes in e.g. climate and nutrient availability, we'll consider in another chapter.)

In this user-config file, a release of CO$_2$ to the atmosphere is prescribed, which by default is set to a value of just 1 PgC and over an interval of just a single year. (Releasing CO$_2$ just over a single year is obviously rather unrealistic and many impacts will decay away rapidly, but represents a useful idealized experiment for assessing the time-scale(s) of fossil fuel CO$_2$ uptake by the ocean.) Additional (netCDF) output has also been prescribed, via the user-config parameter setting: bg_par_data_save_level=10 (see Section 14.4) so that more information relevant to assessing ocean acidification is saved.

Run the experiment for e.g., 20 (or more if you like) years, starting from the pre-industrial re-start experiment LAB_3.SPIN, i.e.:

$ . ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.BASE LABS
   LAB_3.CO2emissions 20 LAB_3.SPIN

As for what model results variables to consider... think about the climate change and ocean acidification literature and which environmental (physical and geochemical) properties are considered either critical for ecosystems or are simply helpful and illustrative. Refer to Section 14.6.2 for a summary of some of the key ocean acidification (and other) variables that may be saved by the model$^2$. In the 3-D netCDF time-slice file remember, for instance, that ocean surface waters in which aragonite becomes under-saturated (OHMEGA < 1.0) is regarded as a critical threshold for organisms making aragonite shells and skeletons and spells TROUBLE for some poor calcifying marine organism somewhere. (Temperature is also highly relevant to marine ecosystems under future global change.) Note that the calcification response is encoded in the model and described in Ridgwell et al. [2007a,b] and may or may not reflect the Real World.

For climate change... the variables of particular interest should be obvious. Remember that there are both time-series outputs, as well as 2D and 3D fields, any or all of which might be helpful for elucidating impacts.

---

$^1$e.g. obtain from http://www.seao2.info/pubs.html

$^2$(depending on specific data saving configuration)
Chapter 6. Fossil fuel CO$_2$ and ‘ocean acidification’

6.1.1 Idealized emissions forcing

You can easily modify the experimental design to release more/less CO$_2$ very much as you did for the red dye tracer. In the *user-config* file, the line:

```
bg_par_atm_force_scale_val_3=8.3333e+013
```

scales the time-history of the CO$_2$ flux, given in the forcing file:

```
biogem_force_flux_atm_pCO2_sig.dat
```

... which can be found in the directory:

```
cgenie.muffin/genie_forcings/pyyyy.z.FpCO2_Fp13CO2
```

The format of this file is:

```
-START-OF-DATA-
  0.0 1.0
  1.0 1.0
  1.0 0.0
999999.9 0.0
-END-OF-DATA-
```

and defines an emission of 1$molC$ (carbon) per year over the first year (year 1.0) of the model experiment (between year 0.0 and 1.0), but which in the example *user-config* is then scaled by a value of $8.333 \times 10^{13}$ (by the parameter *bg_par_atm_force_scale_val_3*) to give a total of 1$Pg$Cyr$^{-1}$. (Year 999999.9 has no special meaning and is simply just waaaaay into the future . . .

Pause . . . and note briefly how the final CO$_2$ flux is arrived at. *muffin* calculates it by multiplying the value in the forcing file (1.0) by a modifying parameter in the *user-config* file ($8.3333e+13$). The total flux is hence: $1.0 \times 8.333 \times 10^{13} = 8.333 \times 10^{13}molCO2yr^{-1}$. If you set both values as 1.0, you’d get very little carbon released (a single mol!). If you screw up and multiply $8.3333e+013$ and $8.3333e+013$ together as the total flux . . . you’ll soon know it as you cook the Earth . . . But it does not matter which parameter has value 1.0 and which scales the units ($8.3333e+013$). For now, it is simply more convenient to be able to edit the *forcing* file with 'simple' numbers (and leave the large numbers and units conversion in the *user-config* file).

Together, the scaling and forcing value gives a CO$_2$ release of 1$Pg$Cyr$^{-1}$ for just a single year compared to current emissions are about 10$Pg$Cyr$^{-1}$. So, do not expect anything exciting if all you emit to the atmosphere is a single measly 1$PgC$.

(The parameter: *bg_par_atm_force_scale_val_4*=-27.0 specifies the carbon isotopic composition of fossil fuel carbon and can be ignored for now.)

Because ‘accidents can happen’ and the global environmental changes induced by the massive fossil fuel CO$_2$ release can obscure mistakes made in the experiment configuration (parameter values) and/or the re-start used, you are strongly advised to first (or in parallel, as a job submitted to the cluster – refer to Lesson Zero to remind yourself of the commend line syntax needed for this) – and run a control experiment:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.BASE LABS LAB_3.CONTROL 20 LAB_3.SPIN
```

Here – the *user-config* defining the control experiment (LAB_3.CONTROL) is identical to that for the actual experiment itself (LAB_3.CO2emissions) with the exception of the scaling of the CO$_2$ emissions that is set to zero. Note that it is left completely to you to create the experiment configuration file LAB_3.CONTROL ...
If everything is OK with the control experiment, atmospheric CO\(_2\) (and climate) following on from the re-start should be stable and there should be little (or no) drift in any of the output variables (because the spin-up you are re-starting from should have been run to an equilibrium state and you have not changed anything in the control experiment, right?).

It is good practice (i.e., do it!) to always run a control experiment for each different type of experiment – e.g., you only need to run one control experiment for a set CO\(_2\) emissions experiments differing only in total carbon release of the time-history of that release.

When you have run both the real and control experiment, compare the results. View (or plot) both relevant time-series output, and create anomaly maps of key time-slice variables in Panoply or MATLAB, using a corresponding time-slice from the control experiment to create the experiment anomaly with.

OK. You might want to run something a little more exciting now. For instance, rather than

```
-START-OF-DATA-
  0.0  1.0  
  1.0  1.0  
  1.0  0.0  
  999999.9  0.0  
-START-OF-DATA-
```

you might have:

```
-START-OF-DATA-
  0.0  1000.0  
  1.0  1000.0  
  1.0  0.0  
  999999.9  0.0  
-START-OF-DATA-
```

for a total of 1000PgC is released over a single year. Now you should see some policy-relevant impacts occur :o) (Again, contrast control and experiment results to quantify/visualize the impacts of the CO\(_2\) release.)

You can control the shape of the emissions profile as well as its magnitude. Between the start and end ‘tags’ in the text forcing file, the data is arranged into 2 columns: the first contains a series of tie-points for defining the timing of changes in emissions, and the 2nd column contains flux information (units of PgCy\(^{-1}\) when scaled by the parameter parameter bg_par_atm_forc_scale_val_3 in the user-config). At each time-step of the model, the CO\(_2\) flux to be applied to the atmosphere is interpolated between these time points.

For instance, in the forcing (directory) file biogem_force_flux_atm_pCO2_sig.dat, the purpose of:

```
  0.0  1.0  
  1.0  1.0  
  1.0  0.0  
  999999.9  0.0  
```
is to specify a uniform flux of 1.0 \((\text{scaled to } \text{PgCyr}^{-1})\) over the first full year of the model run, followed by a sharp turn-off to zero flux at the end of first year (and remaining zero thereafter). To extend the period of emissions – for example:

\[
\begin{matrix}
0.0 & 1.0 \\
10.0 & 1.0 \\
10.0 & 0.0 \\
999999.9 & 0.0
\end{matrix}
\]

would result in a uniform flux lasting 10 years with a sudden cut-off and zero thereafter (i.e., once scaled by the parameter in the user-config \(-1\text{PgCyr}^{-1}\) over 10 years \(-10\text{PgC}\) total emissions). In contrast:

\[
\begin{matrix}
0.0 & 0.0 \\
10.0 & 1.0 \\
10.0 & 0.0 \\
999999.9 & 0.0
\end{matrix}
\]

would result in a linear ramp, starting from zero at the start of year 0.0, to \(1.0\text{PgCyr}^{-1}\) at year 10.0 and then suddenly ceasing and remaining at zero for the remainder of the experiment (a total \(CO_2\) emission of \(1 \times 1.0 \times 0.5 = 5\text{PgC}\) over 10 years).

To ramp up (over 10 years), and then down again (over 10 years), you would specify:

\[
\begin{matrix}
0.0 & 0.0 \\
10.0 & 1.0 \\
20.0 & 0.0 \\
999999.9 & 0.0
\end{matrix}
\]

Make up a few 'shapes' (and hence different experiments), maybe for the same integrated/total emissions, explore the effect of different rates of rise/fall in the release rate. And/or for the same release rate and/or duration, explore the impact of different total emissions of \(CO_2\). Try and think in terms of hypotheses and formulate questions to guide your experimental design/configuration. (An alternative approach is to create random scenarios and in the analysis, fish for interesting patterns that could lead to knowledge and/or specific questions to be tested further, but it is better to start off with a hypothesis in mind.)

Note that you can either edit and re-use the same forcing directory and name, modifying the file `biogem_force_flux_atm_pCO2_sig.dat` each time but then losing an explicit record of how you might have set the emissions profile previously, or you can copy and rename the entire forcing directory (and then edit `biogem_force_flux_atm_pCO2_sig.dat`). If you copy and rename the entire forcing directory, in the user-config, you then need to specify this new forcing (directory) name, e.g.:

```bash
# specify forcings
bg_par_forcing_name="pyyyy.FpCO2_Fp13CO2.NEW"
```

if you, for instance, called your new forcing directory (in genie-forcings): `pyyyy.FpCO2_Fp13CO2.NEW`.\(^3\)

\(^3\)Refer to the directory map in Figure 1.1 if in doubt here.
6.1 Exploring the consequences of fossil fuel CO\(_2\) emissions

6.1.2 Historical (real-world!) emissions forcing

Historical and future (e.g. IPCC 'SRES') emissions scenarios can also be prescribed explicitly. An example is given as user-config file: LAB_3.historical. In this, a historical emissions forcing (technically: a prescribed concentration profile of pCO\(_2\) and other anthropogenic gases) is specified by the forcing:

```plaintext
bg_par_forcing_name='worjh2.historical2010'
```

In contrast to before, no additional scaling is needed because the forcing specification directly follows the observed change in atmospheric concentration with time (in units of atm CO\(_2\)).

An additional line appears in the user-config because the historical pCO\(_2\) transient starts in the 1700s (for which a nominal date of 1765 is often used) rather than year zero. For example, to start muffin from year 1765 rather than zero, the start year parameter is set thus:

```plaintext
bg_par_misc_t_start=1765.0
```

It is also convenient to specify a set of time save points that are consistent with the historical period. In the example user-config, the addition of the parameter settings:

```plaintext
bg_par_infile_slice_name='save_timeslice_historicalfuture.dat'
bg_par_infile_sig_name='save_timeseries_historicalfuture.dat'
```

specifies a series of time points at which data is saved that aligns with historically relevant years.

Try viewing the contents of these (text) files:

```plaintext
save_timeslice_historicalfuture.dat
save_timeseries_historicalfuture.dat
```

which can be found in the directory: genie-biogem/data/input, to get a sense of how frequently the data will be saved, and how this differs from the default settings, which are defined in the files:

```plaintext
save_timeseries.dat
save_timeslice.dat
```

A suitable experiment would then be one run for 245 years so that it reaches year 2010 (having started from year 1765):

```bash
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.BASE LABS
   LAB_3.historical 245 LAB_3.SPIN
```

(Here, the 245 year run duration is simply calculated to get you from year 1765 to 2010 ...)

WARNING! Ignore the ‘WARNING’ at the start – these are simply telling you that more tracer forcings have been specified than you have selected tracers for in the base-config: (cgenie.eb_go_gs_ac_bg.worjh2.BASE). (A different base-config with additional selected tracers could have been specified to make use of other historical changes in atmospheric composition, such as of radiocarbon (\(^{14}\)C) and CFCs.) Also: from year 1765 onwards, changes in atmospheric CO\(_2\) only rise very slowly initially. Don’t expect to see anything happen in 10 seconds flat because relatively few people and countries in the 1800s could be bothered to burn much more than a little local coal. You could potentially start your experiment at year 1850, changing the value of bg_par_misc_t_start and specifying shorter experiment duration if you are desperate for the End of the World to come.

Don’t forget: you could submit this experiment to the cluster and do more (idealized emissions) ‘playing’. 
Chapter 6. Fossil fuel CO$_2$ and ‘ocean acidification’

Given that there is observationally-based information on the distribution of anthropogenic CO$_2$ taken up by the ocean (e.g., Sabine et al. [2004]) ... and you are running a historical transient experiment with the model driven by observed increases in atmospheric pCO$_2$ ... you are in a position to critically evaluate the models ability (or lack of) to represent the future-critical process of oceanic fossil fuel CO$_2$ uptake and transport by large scale ocean circulation.

In the 2D netCDF output, there is a variable for the water column integrated inventory of DIC – equivalent to the Sabine map except you will need to subtract the preindustrial background of DIC first, i.e., to create a DIC anomaly map representing only the added fossil fuel CO$_2$ component of ocean DIC. The data in the Sabine paper clusters around 1994. A time-slice centered on this year (1994.5) has been configured in the model exactly for this purpose. Your baseline state can either be from prior to CO$_2$ emissions commencing at any significant rate (e.g., 1750.5) or (better), from a control experiment. Note that similar comparisons could be (and are regularly) made with other tracers such as CFCs, which provide additional insights into the patterns and time-scales of trace gas update and ocean circulation. (See: Cao et al. [2009])

Observational data, re-gridded to the muffin grid and in netCDF format can be downloaded from the ‘usual place’ (http://www.seao2.org/mymuffin.html) from the ‘got data?’ box on the left. You could for instance, compare horizontal or vertical slices (3D netCDF) and create difference (anomaly) maps. Somewhat more representative of the entire ocean is to compare (or calculate difference maps) of zonal average profiles. Unfortunately, the observations are not in the form of water column integrals and hence you cannot create difference maps of model as per the Sabine paper ... unless you are MATLAB-friendly and you use the 3D BIOGEM MATLAB plotting scripts (https://github.com/derpycode/muffinplot) whose use is somewhat described in the muffin user-manual. Examples of MATLAB plotting of the model vs. observed anthropogenic anomaly are shown in Figure 6.1 (note the use of plotting un-interpolated model grid data as colors but with an interpolated contour overlain to help guide the eye and pick out features).

Finally, and the closest to being slightly interesting: rather than applying highly idealized pulses of CO$_2$ emissions, the IPCC ‘SRES’ emissions scenarios can be used to make future projections with. An example forcing of this sort is provided and can be selected by changing the name of the forcing selection parameter (bg_par_forcing_name) to one of the following:

- pyyyyz.FpC02_Fp13C02_A1_AIM
- pyyyyz.FpC02_Fp13C02_A1G_MINICAM
- pyyyyz.FpC02_Fp13C02_A1T_MESSAGE
- pyyyyz.FpC02_Fp13C02_A2 ASF
- pyyyyz.FpC02_Fp13C02_B1_IMAGE
- pyyyyz.FpC02_Fp13C02_B2 MESSAGE

These are ‘future’ emissions scenarios, which all start at year 2010, and end at year 2100. They are derived from different socio-economic and future technological assumptions in making their future emissions projections.

Again, as these forcings have units of PgC yr$^{-1}$ in the time-series files, you will need to add a scaling parameter to the user-config file to turn units of PgC yr$^{-1}$ into molC yr$^{-1}$, i.e.,

bg_par_atm_force_scale_val_3=8.3333e+013

For complete ‘realism’, you will need to run this experiment starting from the end of the historical transient experiment you have just run (Section 1.6), e.g.,
6.1 Exploring the consequences of fossil fuel CO\textsubscript{2} emissions

Figure 6.1: Observed (top) vs. Model (bottom) anthropogenic CO\textsubscript{2} inventories. Data and model water column integrals in units of mol CO\textsubscript{2} m\textsuperscript{-2} and are nominally with respect to year 1994.

$./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.BASE LABS
LAB_3.future 90 LAB_3.historical

and with the start from year now set to year 2010 (the end year of the historical transient):

bg_par_misc_t_start=2010.0

Note that the user-config LAB_3.future is not provided for you – you will need to create this (or a file named whatever you like) by copying e.g., LAB_3.EXAMPLE and making the parameter changes described above (forcing specification parameter, emissions scaling parameter, and start year parameter).

You can also easily replace the details of the emissions with other SRES scenarios – simply find the year vs. emissions rate information from the interweb\textsuperscript{4} and edit or copy-and-paste the flux values for each decade into the file biogem_force_flux_atm_pCO2_sig.dat in the forcing directory. muffin will then automatically interpolate between the decadal tie-points to give a continuous change in emissions. Now you are able to make a rather more realistic/plausible assessment of when and where potential ecological impacts (via assumed ocean chemistry criteria) might occur.

Try running (e.g. as jobs submitted to the cluster queue), some other actual or made up SRES emissions scenarios.

(Note that the past few IPCC assessment reports have switched to ’using ’RCP’s – Representative Emissions Pathways, rather than the SRES emissions scenarios – see later Section.)

\textsuperscript{4}e.g., http://sres.ciesin.columbia.edu/final_data.html
6.2 Further ideas

6.2.1 Assessing the importance of emissions rate
By editing the flux and/or timing information, you can control the $CO_2$ emissions trajectory as well as the total of fossil fuel carbon burned. Explore different $CO_2$ release assumptions and note their differing impact on climate and ocean (carbonate) geochemistry. Much more realistic and appropriate to our current global experiment is a lower rate (order of 10 or 20 $PgCy^{-1}$) released over a longer interval (order of 100 years) as compared to a conceptual 1000 $PgC$ near-instantaneous pulse. Because the experiments are getting longer to run in real time . . . remember to make appropriate use of the cluster queuing facility – i.e., think about whether you want to sit around starting at the screen for 15 minutes waiting for a new line of numbers appear – if not: submit to the cluster queue. For instance, one might try and address the question: “For a given total release of fossil fuel $CO_2$, is it safer to burn it slower?” The answer is maybe not completely obvious, as burning carbon resources slower will result in a small global impact, but perhaps one that persists for longer(??). You could conceive of an ensemble (related set) of model experiments, maybe one of 100 $PgCy^{-1}$ for 1 yr, one of 10 $PgCy^{-1}$ for 10 years, and one of 1 $PgCy^{-1}$ for 100 years, and run them all for e.g., 100 years. (As jobs submitted to the queue, all can be run simultaneously (with blank spaces between commands!)). (Don’t forget the control experiment! (configured the same, except with 0 $PgCy^{-1}$ of emissions for 100 years).)

Note note that you should create 3 new forcings based on the original if you are editing the same original forcing and expecting to run different ones at the same time. Really, this is little more than you did in copying and renaming user-config files in order to create new experiments ... except that now it involves copying and renaming entire directories in genie-forcings. Remember that the forcing is specified by the directory name assigned to bg_par_forcing_name (enclosed in ”) and you will need to change this to match the name of your new forcing directory.

6.2.2 Determining thresholds of environmental impact
There are various concerns about the impacts of continuing fossil fuel $CO_2$ emissions and a number of proposed climatic (e.g., the 2 degree C global warming limit often mentioned in policy documents) and ecological ‘tipping points’. You can assess the maximum allowable $CO_2$ emissions to remain within particular global environmental limits in the model. For example:

- What is the maximum total $CO_2$ release that can be made without inducing aragonite undersaturation at the ocean surface anywhere (or any season – see Section 5.2.3 in the User Manual for seasonal time-slice data saving)? How important is the time-scale of emissions in determining this? For total emissions above this: where in the ocean does the surface first become undersaturated and what sort of (calcifying) organisms might be impacted there? How large would the emissions have to be in order to start to induce undersaturation with respect to aragonite at the surface in the tropics (home to socio-economically important reef systems)? These are questions that can be addressed with simple $CO_2$ release experiments in ocean carbon cycle models and everyone seems to get a GRL paper out of it each and every time!
- How important are $CO_2$-climate feedbacks in amplifying or diminishing future climate and

---

5 These all represent rather unrealistically small total $CO_2$ releases and you may want ot consider a total more like 1000 $PgC$ or rather more. You may also want to think about more realistic shapes rather than pulses, such a some sort of ramp up and then down in the emissions rate (but for the same total emissions).
ocean carbonate chemistry changes – e.g., is the same atmospheric pCO₂ value reached with and without climate feedback (and surface warming) – if not, why?

Hint: the solubility of CO₂ in sea-water is a function of temperature and at higher temperatures, CO₂ is less soluble. This means that with climate warming, CO₂ solubility declines, less is taken up by the ocean and more is left in the atmosphere – driving further heating in a positive feedback.

All your CO₂ emissions experiments to date have had this feedback enabled – specified at the top of the *user-config* file by:

```plaintext
# set climate feedback
ea_36=y
```

To quantify the role of the carbon-climate feedback, you need to run an identical emissions experiment, but with the feedback disabled:

```plaintext
# set climate feedback
ea_36=n
```

Note that you also need to run a historical transient experiments with no carbon-climate feedback, if you are starting emissions experiments from the year 2010. i.e. you will have a set of future emissions experiments including the carbon-climate feedback that are run from a historical transient experiments that also includes carbon-climate feedback, vs. a set of future emissions experiments without the carbon-climate feedback that are run from a historical transient experiments that also does not include carbon-climate feedback.

The importance of the feedback is simply the difference between the 2 sets of experiments, at the same year.

- Also: How much CO₂ emission does it take to significantly ‘collapse’ the AMOC and over what time-scale? (Or alternatively: what is the atmospheric pCO₂ threshold for collapse?)

If the AMOC weakens or collapses . . . why in the absence of a prescribed freshwater perturbation does this happen? What physical process are at play in response to rapid CO₂ release too the atmosphere that may act to reduce or shutdown deep-water formation in the ocean model? (Plotting appropriate ocean property anomalies between the CO₂ release experiment and a control experiment might help.)

Related to a previous possible investigation – does the rate of CO₂ increase (for the same total release) matter? If so, why? What is happening in general to the structure and dynamics of the upper ocean when surface warming is very rapid?

Experiments could be hypothetical and consisting of CO₂ pulses or ramps (or exponentials) and run on directly from a pre-industrial spin-up, or more ‘realistic’ and run on from the end of a historical transient experiment (e.g., starting in year 2010).

Also:

- How much carbon can be burned (and how quickly) such that atmospheric pCO₂ does not increase any further and remains at some specific value (you might pick ×2 or ×4 preindustrial pCO₂ (ca. 278 ppm))? This is difficult to determine well because to hold atmospheric pCO₂ constant, continuing emissions are required (as CO₂ continues to be taken up by the ocean).

One possible approach to tackling this, might be:

1. Run one or more (SRES) emissions scenarios and find the year at which your pCO₂ threshold value is crossed.
2. Re-run the same emissions experiment, but only for as many years as the threshold-crossing year you had identified, is reached. This will become your new re-start exper-
3. Create a new experiment, with zero emissions, and run on from your new re-start. You should see atmospheric $pCO_2$ start close to the value you reached at the end of the last experiment, but then decay away as the ocean continues to absorb carbon from the atmosphere. The decline in atmospheric $pCO_2$ each year tells you something about the yearly emissions needed to keep atmospheric $pCO_2$ constant.

An approximate rule-of-thumb is that 1 ppm in atmospheric $pCO_2$ is equivalent to 2 PgC (actually, a more exact conversion is 1 ppm = 2.123 PgC). So you could calculate a yearly time-series of how much (in ppm) atmospheric $pCO_2$ falls by each year ... convert this to PgC, and then create a new forcing, with this as the annual emissions.

4. Create and run a new experiment, from the same re-start experiment you created, and apply the new emissions forcing you created. See if atmospheric $pCO_2$ is approximately maintained constant(?)

5. If not sufficient constant to your satisfaction, you could also carry out a second iteration – calculating from your latest experiment, the ppm change in $pCO_2$ each year, converting to an emissions rate, and running a further experiment ...

Note that if you 'overshoot' and $pCO_2$ rises above the threshold value rather than falls below, you are allowed to have a negative carbon emission to the atmosphere in your forcing.

6. The sum of the carbon emissions up to the threshold being reached, is your answer to how much more carbon can be burned and not cross that threshold ... but you will see that after that, further emissions are allowed without $pCO_2$ exceeding the threshold. So the answer e.g. for year 2100 (or 2200) will be larger.

There are automated ways provided in the model framework of achieving this, and you can for instance simply tell muffin to maintain a specific (or changing) value of atmospheric $pCO_2$, and from this diagnose the emissions rate that was required. For instance, you’ll see in the original re-start user-config for this chapter – LAB_3.SPIN – the forcing used is worh2.preindustrial, and if you go an look in that forcing directory:

biogem_force_restore_atm_pCO2_sig.dat

you will see that this basically requests that atmospheric $pCO_2$ is held at 278 ppm ($0.278 \times 10^{-6} \text{ atm}$). You could then copy and rename this forcing, and change:

```
0.0 278.0E-6
999999.0 278.0E-6
```

to:

```
0.0 556.0E-6
999999.0 556.0E-6
```

(in the case of a $\times 2$ preindustrial $pCO_2$ threshold). muffin will then attempt to maintain $pCO_2$ at this specified value. The equivalent carbon emissions required to do this, are diagnosed and provided as a time-series output (in units of molCyr$^{-1}$) in file:

biogem_series_diag_misc_specified_forcing_pCO2.res

- Similarly – how much more carbon can be burned but still keep global mean surface air temperature from rising beyond the 'Paris' limit of 2°C (as compared to preindustrial)? This is also difficult to determine well because there are significant climate lags in the system with warming continuing even if atmospheric $pCO_2$ was held constant. Trial-and-error would be one approach ...
6.2 Further ideas

6.2.3 Future atmospheric CO$_2$ concentration pathways ('RCPs')

The more recent/current incarnation of IPCC future scenarios revolves not around making projections of future greenhouse gas emissions rates, but rather future greenhouse gas concentration pathways. (The reasoning is partly to cut out the differences in carbon cycle feedback between models, where for the same unit CO$_2$ release, different climate/Earth system models might project a different atmospheric CO$_2$ concentration and hence climate change (in addition to differences between models in climate sensitivity).) In muffin, these work pretty well much like in the historical forcing scenario, where the observed change in atmospheric CO$_2$ concentration in the atmosphere with time, is prescribed and the model ‘forced’ to conform to this trajectory.

A series of RCP scenarios for how atmospheric CO$_2$ concentrations may evolve with time are provided. Each actually starts at year 1765 and hence incorporates the historical transient. They can hence be used with an experiment starting at 1765 and re-starting from a steady-state preindustrial spin-up. Or they can be jumped into at any point, and e.g. experiments started at year 2010, when only the year 2010 onwards part of the CO$_2$ restoring forcing is utilized. The RCP forcings currently provided are:

- pyyyyz.RpCO2_Rp13CO2.RCP3PD
- pyyyyz.RpCO2_Rp13CO2.RCP4p5
- pyyyyz.RpCO2_Rp13CO2.RCP6p0
- pyyyyz.RpCO2_Rp13CO2.RCP8p5

and can be selected simply by changing the name of the forcing in the *user-config*, e.g.:

```python
bg_par_forcing_name='pyyyyz.RpCO2_Rp13CO2.RCP8p5'
```

which would select the RCP8.5 scenario.

**NOTE:** ensure that the scaling parameters are either commented out, e.g.:

```python
#bg_par_atm_force_scale_val_3=8.3333e+013
#bg_par_atm_force_scale_val_4=-27.0
```

or simply deleted in their entirety (because the RCP forcing contains the actual values/correct units and you do not need to modify them any further).

The equivalent carbon emissions required to follow these concentration pathways are diagnosed by *muffin* and provided as a time-series output (in units of molCyr$^{-1}$) in file:

`biogem_series_diag_misc_specified_forcing_pCO2.res`

6.2.4 Isotopic tracing of fossil fuel CO$_2$ uptake

See: 12.2.1
The ocean’s biological pump
Single nutrient (PO₄) cycling and basic controls on biological productivity
Further ideas
The role of temperature in the biological pump
Iron co-limitation of biological productivity
Ocean carbon geoengineering
Iron fertilization
Phosphate fertilization
Enhanced weathering
Modifying spatial patterns
Further ideas
Impacts to look out for
Other thoughts and suggestions
Further modifications of the biological pump in the ocean.

7. Ocean biogeochemical cycles
You will need the following re-starts files prior to embarking on the experiments in the Chapter.

7.2 This is of a basic single ($PO_4$) nutrient biological export scheme.

```sh
```

7.3 ...

7.4 $ wget http://www.seao2.info/cgenie_output/201015.ENS.solscav6x6c.00.tar.gz

7.5 $ wget http://www.seao2.info/cgenie_output/LAB_4.SPIN.tar.gz

Extract the results in the usual way and in the usual place ... and return to genie-main in the usual way ... all ... as usual.
7.1 The ocean's biological pump

In this Chapter we’ll step through some of the facets of the cycle of carbon (and nutrients) in the ocean – the ’biological pump’. (And then in the following section we’ll look at the same processes in a different way via ‘geoengineering’ and hopefully learn something further about how it all ’works’.)

**muffin** incorporates a variety of options for simulating biological export production. To date, these have all been rooted in what is effectively a nutrient mass balance approach to estimating export production, nicely encapsulated by Ernst Maier-Reimer as: “conceptually not a model of biology in the ocean but rather a model of biogenically induced chemical fluxes [from the surface ocean]” [Maier-Reimer, 1993]. Hence in schemes of this nature, which we will term ‘biogenic export’, there is no attempt to explicitly account for changes in cell numbers/biomass and hence, nor zooplankton, which will impact a bias particularly in the seasonal time-dependent response of export. Previous biogenic export schemes utilized in **muffin** considered a single nutrient (P limitation) only and took either restoring to observations [Cameron et al., 2005] or explicit P-limitation [Ridgwell et al., 2007a,b] approaches. More recently, additional limitation by iron (P+Fe) has been implemented, while nitrogen cycling (including N fixation and denitrification) and hence P+N limitation, has been implemented and explored in the context of extreme nutrient and oxygen cycle perturbation associated with the Cretaceous Oceanic Anoxic Events [Monteiro et al., 2012; Naafs et al., 2029]. Work currently in development includes the additional consideration of Si limitation (P+Si limitation) control on production of diatoms vs. non-diatoms following Ridgwell et al. [2002].

Overall, biological production and net export of particulate (POM) and dissolved (DOM) organic matter plus calcium carbonate (CaCO$_3$) are directly determined by the availability of nutrients (phosphate and/or total dissolved iron and/or NO$_2^-$ and/or NH$_4^+$) and/or dissolved silica) together with the degree to which physical conditions, particularly light and temperature, are conducive to growth. DOM is remineralzied (transformed back to inorganic dissolved constituents) relatively rapidly (a ca. annual time-scale) and hence mostly at or close to the ocean surface, while POM is assumed to sink down into the ocean interior. See: Hülse et al. [2017]$^1$ for a comprehensive review (esp. of implementation in models).

Note that a full description of the various biological and ocean interior remineralization schemes (together constituting the biological pump) is not given here as background. Rather, the relevant provided references should be read (rather than e.g. downloaded and forgotten ...).

---

7.2 Single nutrient (PO$_4$) cycling and basic controls on biological productivity

The following exercises will utilize one of the more basic representations of the biological pump – one with only a single nutrient (PO$_4$) potentially limiting to biological export, considered. The scheme is described in full in Ridgwell et al. [2007]$.^2$ We will consider the following scientific questions as a starting point and then devise some model experiments to address them:

1. What would the Earth look like with no biological production in the ocean? Specifically: how important is the biological pump in controlling atmospheric $p$CO$_2$ and how much higher would $p$CO$_2$ be in the absence of a biological pump in the ocean?

2. Conversely, is the biological pump operating at its maximum efficient in the ocean today, and if not, how much more could atmospheric $p$CO$_2$ conceivably be lowered.

3. How does the biological pump affect the distribution and hence availability of dissolved oxygen in the ocean? (How more oxygenated would the ocean be with no export of organic matter into the ocean interior, and how much more severe would oxygen depletion be if plankton were able to utilize all the nutrients at the ocean surface?)

4. If carbon and nutrients are returned back to solution (remineralized) much closer to the ocean surface, does atmospheric $p$CO$_2$ increase or decrease? (The faster return of DIC to the ocean surface will tend to increase $p$CO$_2$ while the return of nutrients will tend to enhance biological productivity and decrease $p$CO$_2$ ...)

5. What about if carbon and nutrients are much more efficiently exported into the very deepest parts of the ocean – does $p$CO$_2$ increase or decrease (and what happens to e.g. $[O_2]$)?

6. What is the role of the calcium carbonate (CaCO$_3$) 'counter pump'? How much higher is $p$CO$_2$ in the presence of calcifying organisms in the ocean compared to just organic matter production and cycling?

7. What about feedback with climate – how does changing global surface temperature (and hence ocean circulation patterns and sea-ice extent) modulate the impact of changes to the biological pump in the ocean? Is the resulting feedback positive or negative (and why)?

The associated experiments would ideally be run for several thousand and maybe 5000 years to obtain a new (quasi) steady state of carbon and nutrient cycling in the ocean. But some of the impacts in the ocean and on atmospheric $p$CO$_2$ will develop relatively rapidly (decades). Hence model experiments could be run for a few 10s of years, or e.g. 100 years, or for a full e.g. 5000 years but periodically downloading and viewing the results as the experiment progresses (and not necessarily waiting until the end of the experiment, or even ever looking at the final state!).

All the experiments will start from the same re-start$^3$ and will be based on the same user-config$^4$ (and note the use of a different base-config to previously):

```
./runmuffin.sh cgenie.eb_go_gs_ac_bg.worbe2.BASES LABS EXP.7.2 5000 EXAMPLE.worbe2.Ridgwelletal1997.SPIN
```

(here a 5000 year experiment run-time is given, but it need not be this – see above). Remember to copy and rename EXP7.2 in order to create new and unique experiments each time.


$^3$As per described in Ridgwell et al. [2007].

$^4$The same user-config as used to generate the re-start (with the exception that the atmospheric CO$_2$ concentration is no longer prescribed).
7.2 Single nutrient (PO$_4$) cycling and basic controls on biological productivity

The first thing to do is to run a control experiment, for however long you have decided to run all the actual experiment experiments for. (You need not wait for this to finish, but can submit it as a *job* to the cluster and get on with running some real experiment experiments.) The control experiment can simply be derived (copied) from \textit{EXP}7.2, with no further alterations needed. In this \textit{user-config}, the value of atmospheric $pCO_2$ is not prescribed (there is no \textit{forcing} defined) and hence $pCO_2$ is free to respond to any change in parameters. In the case of the control – there are no changes in the biological pump parameters compared to the \textit{spin-up}, so you should see no (or little) drift in $pCO_2$ occurring.

Also think about the output you want to view (and how to view it). You are likely to want to look at anomalies vs. the control, as well as absolute patterns in the ocean. In the \textit{user-config} – check that the results/output select ‘level’ is going to give you the output you are expecting. This is parameter:

\begin{verbatim}
bg_par_data_save_level=5
\end{verbatim}

Refer to Section 14.4 and Appendix I, for more details on how different (2D and 3D) \textit{time-slice} variable fields and \textit{time-series} are selected to be included in the model output.

In terms of the specific ’questions’ above (the numbering system is the same), both w.r.t. what parameters and parameter values to change and well as what to ‘look for’:

1. The parameter scaling the rate of nutrient uptake (and hence biological export) is:

\begin{verbatim}
# maximum rate of conversion of dissolved PO4 into
# organic matter by phytoplankton (mol kg$^{-1}$ yr$^{-1}$)
bg_par_bio_k0_PO4=1.9582242E-06
\end{verbatim}

(in units of \textit{mol PO}_4 kg$^{-1}$ yr$^{-1}$). Setting this to zero would ’turn off’ completely the biological pump, leaving you with an abiotic ocean.

What to look for? The value of atmospheric $pCO_2$. You might also confirm that biological export really is zero.

To obtain these values you can refer to the respective time-series results files. Or, in the summary text file, which for the \textit{re-start} is:

\begin{verbatim}
biogem_year_09999_500_diag_GLOBAL_AVERAGE.res
\end{verbatim}

and will differ in your actual experiment only in the value of the year in the filename. This file contains summary information at a specific \textit{time-slice} time, including mean atmospheric composition, mean ocean composition, biological export fluxes.

2. Similarly to above – nutrient uptake and the strength of the biological pump can be enhanced by increasing the value of the parameter \texttt{bg\_par\_bio\_k0\_PO4} (maybe 10 times larger?). The question is then what does the ocean and atmospheric $pCO_2$ look like if the biological pump is operating at its maximum rate and (almost) all nutrients are consumed at the surface. (If you find that there are regions of nutrients that never get fully consumed ... why?)

3. For the question on dissolved oxygen ([O$_2$]) availability in the ocean interior – the parameters and experiments are as above, except you are looking for how the mean and distribution of dissolved oxygen in the ocean changes. Mean ocean [O$_2$] values can be obtained from time-series of the summary file. Spatial distributions are recorded in the netCDF output. You might view horizontal or vertical slices (from 3D), or benthic distributions (2D). Think about where animals tend to live and where they could potentially be impacted if export from the ocean surface was much higher than ’today’ (the \textit{spin-up}).

4. The parameter controlling the depth-scale (and vertical distributed) at which particulate organic matter (POM) is remineralized in the ocean interior is controlled by:
Reducing the value of this parameter forces a greater proportion of sinking POM to be remineralized closer to the surface.

What to look for? Largely as before – the summary file is useful for atmospheric $CO_2$, mean ocean $O_2$, and global (biological) export fluxes (as you might expect nutrients released closer to the ocean surface to result in greater surface ocean nutrient supply and hence fuel higher export). You might also consider the spatial patterns of nutrients and dissolved oxygen.

See Meyer et al. [2016] for an example of a study testing these sort of model changes and its implications (and hence ideas of what else to look for, and for ‘reasonable’ parameter values to test).

5. Conversely, you can also increase the value of the depth scaling parameter in order to force a deeper mean depth of POM remineralization. You can also partition more of the total export into a POM form that is assumed to be resistant to degradation and is transported to the ocean floor completely unchanged (see: Ridgwell et al. [2007]):

$$bg\_par\_bio\_remin\_POC\_frac2=6.4591110E-02$$

Increasing this value forces a greater fraction of the POM exported from the surface ocean to reach the ocean floor.\(^6\)

In both (4) and (5), it may not be obvious which way atmospheric $pCO_2$ responds – e.g. for (4) by returning nutrients more efficiently to the ocean surface, you increase export, and hence increase the fixation and removal of atmospheric $CO_2$ from the ocean surface, drawing down atmospheric $pCO_2$. BUT, at the same time, carbon released from POC (as DIC) through bacterial remineralization, also occurs at a shallower depth and is returned to the surface more efficiency ... tending to increase atmospheric $pCO_2$ ... Just this problem – 2 opposing effects with an uncertain net impact – is exactly why (to find out the answer) you build and run numerical models of the system!

6. The export of $CaCO_3$ from the ocean surface is calculated as a ratio to the export of POC. By default, this ratio varies with the carbonate chemistry (saturation state) of the surface ocean, following Ridgwell et al. [2007a] (and see also Ridgwell et al. [2007b] and Ridgwell et al. [2009]). The scaling parameter controlling the molar ratio of $CaCO_3$ to $POC$ is:

$$bg\_par\_bio\_red\_POC\_CaCO3=0.044372$$

Setting this zero will effectively turn back the clock 200Ma to a world prior to the evolution of planktic calcifiers (e.g. see: Ridgwell [2005]), or conversely, a future world in which they are all driven extinction from ocean acidification ...

What is the impact on atmospheric $pCO_2$ of setting this to zero? From your knowledge of carbonate chemistry ... why does this happen? You might try e.g. doubling the value (and hence doubling the export of $CaCO_3$ from the surface ocean). As well as $pCO_2$, you might also look at fields of $ALK$ (or other carbonate chemistry parameters) in the ocean to see what other geochemical changes occur.

Because (rightly or wrongly) in the model, $CaCO_3$ depends on surface ocean $\Omega$ (w.r.t. calcite) you might explore what happens to $CaCO_3$ production and export under a release of (fossil

---


\(^6\)See Ridgwell et al. 2007] for a description of how this all works.
7.2 Single nutrient (PO$_4^-$) cycling and basic controls on biological productivity

Finally, you might explore how in some (or all) of the above experiments, a changing climate in response to changing atmospheric pCO$_2$, in turn modulates the impacts through carbon-climate feedback. By default in the user-config, climate-CO$_2$ feedback is disabled:

```
ea_36=n
```

i.e. changing atmospheric pCO$_2$ does not influence the climate system (which remains implicitly forced with a preindustrial radiative forcing value). This allows us to avoid complications arising due to carbon-climate feedback and hence obtain a the underlying response of the system to changing the biological pump in the ocean.

You can re-enable climate feedback by setting:

```
ea_36=y
```

Impacts may include (but not be limited to): ocean stratification (rapid warming) and/or changes in convection at high latitudes, increased/decreased sea-ice extent that affects the ocean surface area available for biological export, changing solubility of oxygen in sea-water. Note that in this particular simple biological export scheme, there is no temperature-dependence of biological activity and hence export.

And that pretty much wraps up all the main knobs that it is possible to play with using the very basic biological pump scheme in muffin. (However, a few further ideas follow ...)

7.2.1 Further ideas

You might also play with the ‘half saturation constant’ for nutrient uptake (see Ridgwell et al. [2007]):

```
# [PO4] M-M half-sat value (mol kg-1)
bg_par_bio_c0_PO4=2.1989611E-07
```

As set, this specifies that at a PO$_4^{2-}$ concentration of about 0.2μmol kg$^{-1}$, growth (net export) is restricted to half the maximum (and then declines further at even lower ambient nutrient concentration).

You might, for instance, try setting the value to zero, which assumes maximum growth can continue right up nutrient completely running out. This would be similar to assuming all the primary producers in the ocean had extremely small cell sizes (and hence low half saturation values) and were adapted to oligotrophic conditions.

Or, test an ocean dominated by assumed very large cell sizes and phytoplankton that struggle under anything other than fully eutrophic (high nutrient) conditions. e.g. you might try values of 1.0μmol kg$^{-1}$ (1.0E-06) or even 2.0μmol kg$^{-1}$ (2.0E-06) and see what happens, particularly to the pattern of surface nutrient concentrations and export.

By default, the update and export in organic matter, of carbon, occurs in a fixed ‘Redfield’ ratio, with PO$_4^{2-}$. The classical value is 106 (i.e., for every mole of PO$_4^{2-}$ taken up from the ocean, 106 moles of CO$_2$ are removed (and fixed into organic matter). The parameter determining this is:
Chapter 7. Ocean biogeochemical cycles

bg_par_bio_red_POP_POC=106.0

To change the default value (106.0), simply add a new line at the end of the user-config file specifying the value you want. A larger number means that \( PO_4 \) is being utilized more efficiently and more organic matter is being produced for the same nutrient consumption.

You should see impacts on atmospheric \( pCO_2 \) and the oxygenation of the ocean interior form changing this.

To test the effect of there being more (or less) \( PO_4^{2-} \) in the ocean in the first place, it is possible to increase the inventory of the ocean as a whole following a re-start, by:

\[
\text{bg_ocn_dinit_8}=1.0\times10^{-6}
\]

which will add 1 \( \mu \text{mol kg}^{-1} \) of \( PO_4^{2-} \) uniformly to the ocean. (A larger/smaller number will obviously increase the glacial nutrient inventory by more/less. A negative number will remove \( PO_4^{2-} \).)

There is one more knob controlling organic matter export and cycling in the ocean that you could tweak and explore the effect of the assumption regarding how much organic matter produced is partitioned into particulate form that sinks, vs. dissolved, controlled by:

\[
\text{bg_par_bio_red_DOMfrac}=0.66
\]

which specifies 66\% of organic matter production is diverted into dissolved form. See Ridgwell and Arndt [2014]. And/or, you might adjust its mean lifetime in the ocean, which by default is 0.5 yr:

\[
\text{bg_par_bio_remin_DOMlifetime}=0.5
\]

In addition to the primary environmental properties of atmospheric \( pCO_2 \) (and climate if you have the feedback enabled), biological export production (and hence the patterns and magnitude of of food sources for marine ecosystems), and dissolved oxygen, you might also look at how the patterns of carbon isotopes (\( \delta^{13}C \)) (particularly of DIC) change in the ocean.

How did the different changes in the biological pump that you tested, alter (if at all) the patterns of \( \delta^{13}C \) in the ocean? Can you distinguish between the different biological pump changes, based on \( \delta^{13}C \)? (Carbonate and organic carbon \( \delta^{13}C \) is a key paleoceanographic proxy and one people would ideally like to use in order to reconstruct changes in the past such as in the biological pump.)

---

7.3 The role of temperature in the biological pump
7.4 Iron co-limitation of biological productivity

"Iron (Fe) is a “micronutrient”; essential to the biochemistry of all cells and, in particular, for enzymatic activities associated with photosynthesis and with nitrogen fixation in the marine environment. Yet it is only required in very small quantities, as little as one part in 2000 compared to phosphorus or one in 200,000 compared to carbon!

As with macronutrients such as phosphate (\(PO_4^{3-}\)) and nitrate (\(NO_3^-\)), supply of Fe to the surface ocean occurs through up-welling and mixing of ocean waters from below. However, dissolved Fe has a short lifetime in the oxygenated seawater environment. \(Fe^{II}\), the most soluble state, is rapidly oxidized to \(Fe^{III}\), which is highly insoluble and tends to precipitate out and be removed (scavenged) by particulate matter settling through the water column. The result is that in up-welling water, the ratio of dissolved Fe to that of highly soluble \(PO_4^{3-}\) is lower than the ratio required by phytoplankton cells to grow and divide. Consequently, phytoplankton in surface waters cannot fully utilize the abundant up-welled phosphate (or nitrate) unless more Fe is brought into the system.

Dust is important because mineral aerosols contain iron, primarily in the form of Fe oxides such as hematite and oxide-hydroxides such as goethite (found as coatings on other mineral grains). However, the present-day flux of aeolian Fe is not everywhere sufficient to correct the relative nutrient imbalance. For instance, dust fluxes to the Southern Ocean are among the lowest anywhere on Earth, and the aeolian Fe supply is too small to compensate for the depleted Fe relative to \(PO_4^{3-}\). Consequently, phytoplankton cannot fully utilize available macronutrients, and dissolved \(PO_4^{3-}\) exists year-round at the ocean surface. Similar reasoning applies to the presence of excess \(PO_4^{3-}\) in the Eastern Equatorial Pacific. Although dust supply to the North Pacific often appears moderately high in global model simulations, fertilization experiments in the Northwest Pacific suggest that this region is still iron-limited. Hence, the geography of complete nitrate utilization at the surface will be controlled to a first order by the mass distribution of dust deposition."  

All the experiments in this section will start from the same re-start and will be based on the same user-config:

```
./runmuffin.sh muffin.CB.worjh2.BASESFeTDTL LABS EXP.7.4a 100 201015.ENS.solscav6x6c.00
```

Remember to copy and rename the file EXP7.4a in order to create new and unique experiments each time. (The experiments need not be run for 100 years ... you may want longer or shorter, depending on how you are perturbing the model and what exactly you are 'looking for'.)

The forcing used in both the experiment includes a prescribed dust flux to the ocean surface (the Mahowald part of the directory name string). This is necessary because the model configuration you are using includes a co-limitation of biological productivity by iron (Fe) in addition to phosphate (\(PO_4\)). (The files associated with the dust forcing are: biogem_force_flux_sed_det_sig.dat and biogem_force_flux_sed_det_SUR.dat but you do not need to edit these files.)

---


10 Note the use of a different base-config to previously.
Note that no prescribed value of atmospheric $pCO_2$ is given in the forcing for this user-config (although it was for generating the re-start) – this is so that any change you make to the marine iron cycle and hence to the biological pump in the ocean can be seen as an impact on atmospheric $pCO_2$. This also helps make for a better control experiment, because if the re-start was somehow mis-matched with the user-config (e.g. if you did not have the correct parameter set used to generate the re-start), or you accidently changed something you did not mean to (or notice), you could expect to immediately see atmospheric $pCO_2$ starting to drift (maybe as much as 1 ppm or more over the course of a few decades).

Also to note: in this particular configuration of marine iron cycling, the solubility of iron in dust, is assumed to be the same everywhere. (In reality, it is likely to vary spatially, with remote regions experiencing lower dust fluxes likely to be characterized by higher solubility.)

What to change?
1. Well ... firstly (for user-config EXP7.4a) you might check that biological productivity is in fact limited at all (anywhere). For this:
   (a) Reduce the half-saturation constant for $Fe$ limitation to zero (so phytoplankton growth is never limited by iron, at least, up to the point where it completely runs out):
   
   $$\text{bg\_par\_bio\_c0\_Fe}=0.10E-09$$
   
   and set this parameter to zero (or something very very small).
   
   (b) Flood the ocean with so much iron, that it simply cannot be limiting anywhere. The dust field itself is sort of difficult to do anything with (in terms of manipulations), so instead, you can change the assumed solubility of iron in dust – i.e. what fraction of iron delivered to the surface ocean in dust is assumed to dissolve and hence become bio-available. The parameter controlling solubility is:
   
   $$\text{bg\_par\_det\_Fe\_sol}=0.001$$
   
   Note that this is a fractional solubility, not a % solubility, so a value of 0.001 equates to a 0.1 wt% solubility.
   
   And change to ... ? Maybe try an order of magnitude (or more) higher value.

   The second modification is likely to have more impact and come closer to answering the question: what would export production and atmospheric $pCO_2$ look like if there was no iron limitation on biological productivity? In the first option, you might still completely run out of iron somewhere in the ocean and hence limit biological productivity. (Maybe try this and find out where this occurs (if at all).)

2. A second question/investigation (also utilizing user-config EXP7.4a) might be to explore the internal cycling of iron and hence the importance of dissolved iron supplied ‘from below’ via ocean up-welling and mixing, rather ‘from above’ and directly from dust.

   The dissolved $Fe$ content of the ocean interior is set by: ocean transport (iron from elsewhere), iron released through the remineralization of organic matter, and iron scavenged from solution and removed onto sinking particles. of these, you have direct control over $Fe$ scavenging, via:
   
   $$\text{bg\_par\_scav\_Fe\_sf\_POC}=0.1$$
   
   that scales the rate at which $Fe$ is removed from solution (this rate also depends on the sinking flux of particulate organic matter as well as the concentration of ‘free’ iron (not bound to organic ligands and hence assumed protected from scavenging).
Increasing this value will increase the loss rate of Fe from the ocean interior (and surface), while decreasing it will reduce the rate of loss. Consider an order of magnitude\(^{11}\) change in value (in either direction) to create meaningful changes to the marine iron cycle.

3. In today’s ocean, iron limitation of biological productivity is only regionally important (compared to e.g. in the first exercise you were changing the dissolved iron supply from dust everywhere). So one might legitimately ask whether specific region are iron limited, and how important this is (or not) to atmospheric \(pCO_2\). However, changing the dust field is not trivial, or rather it would be extremely time-consuming to e.g. edit the numbers to double all the dust fluxes in a specific region (e.g. the Southern Ocean).

\(user-config\) EXP7.4b is provided as an alternative configuration of the exact same marine iron cycle, but with the solubility if iron in dust, dictated by the contents of a 2D file rather than via a single number controlling the solubility everywhere. So rather than:

```
# aeolian Fe solubility
bg_par_det_Fe_sol=0.001
```

controlling a uniform solubility everywhere;

```
# Replace internal dust Fe solubility field?
bg_ctrl_force_det_Fe_sol=.true.
# Filename for dust Fe solubility field
bg_par_det_Fe_sol_file='worjh2.det_Fe_sol.MahowaldUNIFORMsol.dat'
```

specifies that a file containing the solubility value(s) is used instead, and that in this case, the file is: `worjh2.det_Fe_sol.MahowaldUNIFORMsol.dat`

This file can be found in the directory: genie-biogem/data/input and can be edited to change (increase or decrease) the solubility of iron in dust in specific regions (hence changing the dissolved iron flux).

It is important to note that the values in this file are \% father than fractional solubility. By default, the file is populated with value 0.1 everywhere (at all ocean grid points) – 0.1\%, or a fractional solubility of 0.001 (as per was the value of the parameter `bg_par_det_Fe_sol` before.

(Also refer to the geoengineering section following for further thoughts on modifying the flux of iron to the surface in specific regions of the ocean.)

What to look for? Obviously, start with \(pCO_2\) (time-series) and POC export (which can be viewed as both time-series and spatially in the 2D `netCDF` output). (All of these are also included in the summary output files.) Also 2D distributions of \(PO_4^{3-}\) and Fe are highly relevant and useful. For more involved outputs, also in the 2D `netCDF` output you can find the pattern of iron solubility (`misc_sur_Fe_sol`) and the flux of dissolved iron to the ocean surface (`misc_sur_fFe_mol`).

Finally, in the 2D `netCDF` output – `misc_sur_PO4Feimbalance` – is a simple attempt to illustrate where \(PO_4^{3-}\) (positive values) rather than Fe limitation of biological productivity occurs. For example, high positive values (on scale up to 1.0) can be found in the Equatorial Atlantic and Indian Oceans, suggesting plenty or dissolved iron but no phosphate. The Southern Ocean is more iron than phosphate limited, while the Pacific gyres (most negative values) are dominantly iron limited, although may in practice be limited by both.

---

\(^{11}\)It need not be any more than this. In fact, you might try a slightly smaller change, e.g. factor-5.
7.5 Ocean carbon geoengineering

Another way to view and understand marine carbon and nutrient cycling and the 'biological pump' in the ocean is through the lens of geoengineering. In this Section you’ll use geoengineering as an ‘excuse’ to perturb and elucidate the response of the global carbon cycle and climate system and hence learn something new/different about Earth system dynamics compared to e.g. its response only under standard future carbon emissions scenarios.

In the following experiments you are going to explore some of the ocean biological controls on atmospheric pCO₂ (plus ocean acidification, and the distributions and intensities of oxygen minimum zones). Really, the ‘geoengineering’ focus is just an excuse to be looking at how the biological pump in the ocean works, how it regulates atmospheric pCO₂, how sensitive it is to perturbation and what the consequences are of any changes in it. So if you are uncomfortable with ideas of large scale manipulating the Earth system, you might instead think about the relevance of the experiments to e.g. understanding why atmospheric pCO₂ was low at the time of the last glacial.

The overall idea of this Chapter is to run future CO₂ emissions scenarios and test whether ocean carbon geoengineering is an effective means for reducing future ocean acidification and marine ecological impacts (but keeping in mind that you are also exploring the basic natural operation of the system in doing so). You will require a pre-industrial spin-up and will first need to create a new historical pCO₂ transient experiment because you are now using a different base-config (cgenie.eb.go_gs_ac_bg.worjh2.BASEFe) that includes additional tracers for the marine iron cycle, i.e. you cannot simply use any of the experiments from previous labs as a re-start. e.g. refer to Section 14.6 for a guide as to what to ‘look for’ in the model results.

To start with, go ahead and run a new historical transient experiment. A user-config is provided for your convenience (LAB_4.historical) ... but maybe check the settings for e.g. start year, as well as note that there are a number of new parameters to control the iron cycle (amongst other differences) as compared to before:

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.BASEFe LABS
   LAB_4.historical 245 LAB_4.SPIN
```

To avoid the 'WARNING's at the start

The provided example user-config – LAB_4.EXAMPLE – includes parameter settings for controlling any one of 3 different possible ocean carbon geoengineering schemes, described below. By default, these are commented out (== ignored by the model) and only the forcing for the A2 emissions scenario (worjh2.FeMahowald2006.FpC02_Fp13C02_A2_02180PgC) with no geoengineering enacted, is enabled by default. You might regard this as a control (reference) experiment for all the with-geoengineering experiments you might run, i.e. the impacts of CO₂ emissions in the absence of any mitigation by geoengineering. To activate any particular geoengineering forcing: simply un-comment (delete the # at the start of) the appropriate pair of lines (the first line being the forcing specification, and the second one the total flux forcing used in the geoengineering scheme). If you have multiple (un-commented) settings of a parameter (e.g. bg_par_forcing_name), then the value specified in the last occurrence of the parameter, is the one that is applied. This can get confusing, so if you un-comment out one set of parameter options, comment out (add a # to) the ones you are not using.
Chapter 7. Ocean biogeochemical cycles

The geoengineering (and control) experiments need to be run starting from the end of your historical transient experiment:

$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.BASEFe LABS
LAB_4.EXAMPLE 90 LAB_4.historical

Because with a modern configuration and additional tracers in the ocean, the model is running rather slower than in some earlier exercises, you may not want to run beyond the end of the century (hence the 90 year experiment duration, starting from year 2010, as suggested above).

Each of the example geoengineering scenarios are delineated by its own specific forcing – a set of files that live in a uniquely named sub-directory within genie-forcings. The three forcings are:

- worjh2.FeMahowald2006.FpCO2_Fp13CO2_A2_02180PgC_FFe
- worjh2.FeMahowald2006.FpCO2_Fp13CO2_A2_02180PgC_FPO4
- worjh2.FeMahowald2006.FpCO2_Fp13CO2_A2_02180PgC_FALK

Each forcing includes the A2 SRES CO\textsubscript{2} emissions scenario, with the annual emissions (CO\textsubscript{2} flux) biogem_force_flux_atm_pCO2_sig.dat in units of PgCy\textsuperscript{-1} (== GtC yr\textsuperscript{-1}), hence requiring a units conversion setting in the user-config (bg_par_atm_force_scale_val_3=8.3333e+013) that is provided for you under the heading # CO\textsubscript{2} emissions scaling. (Ignore the carbon isotope settings.)

Each forcing also includes a prescribed dust flux to the ocean surface (the FeMahowald2006 part of the directory name string). This is necessary because the model configuration you are using includes a co-limitation of biological productivity by iron (Fe) in addition to phosphate (PO\textsubscript{4}). (The files associated with the dust forcing are: biogem_force_flux_sed_det_sig.dat and biogem_force_flux_sed_det_SUR.dat but you do not need to edit these files.) For the role of iron in controlling ocean productivity: possible starting points for background reading are: Ridgwell and Kohfeld [2007] (PDF available form my website) or Jickells et al. [2005] (Science).

The specific details of the 3 different example geoengineering scenarios are as follows:

### 7.5.1 Iron fertilization

Forcing: worjh2.FeMahowald2006.FpCO2_Fp13CO2_A2_02180PgC_FFe

- a constant (with time) flux of dissolved Fe (in addition to whatever Fe dissolves into the surface ocean from the dust flux) is specified in: biogem_force_flux_ocn_Fe_sig.dat. The magnitude of the applied flux is then scaled in the user-config file by the setting:
  bg_par_ocn_force_scale_val_9=1.0e+09

Note that this is simply an example total global flux. You might consider higher or lower fluxes, as well as potentially how ‘practical’ the annual production and supply of such quantities might be.

A spatial pattern of the flux is also defined, in the file:

biogem_force_flux_ocn_Fe_SUR.dat

An example pattern is set (see later for instructions on modifying this) with a row of grid cells marked along the same latitude in the Southern Ocean. You do not need to retain this pattern. In choosing an alternative: think about where in the modern ocean biological productivity is thought to be at least partly limited by the availability of dissolved Fe. Remember that the model may or may not correspond with reality, i.e. it may or may not predict Fe limitation in the correct regions, which may affect your choice of location for iron fertilization.
7.5 Ocean carbon geoengineering

7.5.2 Phosphate fertilization

Forcing: workh2.FeMahowald2006.FpCO2_Fp13CO2_A2_02180PgC_FPO4 (‘macro-nutrient’ addition)

- a constant (with time) flux of dissolved PO$_4$ is specified in: biogem_force_flux_ocr_P04_sig.dat. The magnitude of the applied flux is then scaled in the user-config by the setting:
  bg_par_ocr_force_scale_val_8=2.0e+12

Again, you should consider this as an example total flux. In choosing a total flux to apply, points of comparison include whatever the total weathering flux (via rivers) of phosphate to the global ocean is. Also: global phosphate (fertilizer) production, which produces an interesting potential conflict between geoengineering and food production, although there are proposals for using fertilized ocean regions for enhanced fish production.

A spatial pattern of the flux is also defined, in the file:

biogem_force_flux_ocr_P04_SUR.dat

An example pattern has been set – here, the Equatorial Atlantic. In choosing your regions(s), think about where in the ocean (again – there may be differences between real ocean and model) productivity is currently limited by PO$_4$. Also be aware of possible on-set of Fe limitation if you relieve the PO$_4$ limitation (i.e., you could potentially lose effectiveness if you supply too much PO$_4$ and instead productivity and CO$_2$ draw-down is capped by a second factor). You could potentially consider PO$_4$ and Fe addition at the same time . . . ?

7.5.3 Enhanced weathering

Forcing: workh2.FeMahowald2006.FpCO2_Fp13CO2_A2_02180PgC_FPO4 (alkalinity addition)

- a constant (with time) flux of alkalinity is specified in: biogem_force_flux_ocr_ALK_sig.dat. The magnitude of the applied flux is then scaled in the user-config by the setting:
  bg_par_ocr_force_scale_val_12=5.0e+13

Again, another example total flux. In choosing a total flux to apply, points of comparison include whatever the total weathering flux (via rivers) of alkalinity (often described in terms of the bicarbonate ion flux) to the global ocean is. Also: global cement (lime) production. (Note that in one mole of lime: CaO, you have 2 moles of alkalinity (Ca$^{2+}$).)

A spatial pattern of the flux is also defined, in the file:

biogem_force_flux_ocr_ALK_SUR.dat

An example pattern has been set – here, bordering the major tropical coral reefs locations in the Western Pacific. In choosing your regions(s), you might think about mitigating specific ecosystem impacts of ocean acidification, or about the feasibility of transport and proximity to abundant limestone (CaCO$_3$ – the source of lime) and/or energy.

7.5.4 Modifying spatial patterns

The spatial patterns of an applied flux forcing to the ocean can easily be modified. The pattern is specified in a simple ASCII (plain text) file, in the file in the forcing sub-directory ending ’_SUR.dat’. The file (in this example the default Fe pattern) looks like:
Here: ‘0’ s represent land and cannot have a forcing associated with them. ‘0.0’ s represent a zero flux to the ocean, and ‘1.0’ s the default Southern Ocean forcing pattern. Note that a distinction is made between a ‘0’ and a ‘0.0’ so that you can make out where the continents are and do not necessarily have to count in the i and j grid directions to find a specific location. The grid is the same as you saw previously in tracing ocean circulation, and which numbered the i and j axes if that helps. For the ALK forcing, ‘1.0’ s are set off of the coast of Australia and SE Asia and in the PO4 forcing, in the Atlantic.

There is no more to changing the pattern of the flux forcing than simply marking with a ‘1.0’ where you would like the forcing applied, and a ‘0.0’ where it should not be. Note that there should be a single blank line at the bottom of the file. (If you have problems applying a modified spatial pattern – check that this is present.) Copy the entire forcings directory and rename each time to create a new pattern, remembering to change the name of the forcing in the bg_par_forcing_name parameter. (is best to keep a copy of the original forcing in case you make a mess of the spatial pattern file, but the original can also be recovered from the code server.)
7.6 Further ideas

7.6.1 Impacts to look out for

- Impacts and ecosystems of interest could potentially be ones residing on the ocean floor, such as cold-water (deep water) corals, and are not necessarily planktic (/surface) only.

- Don’t forget that different calcifying organisms employ different mineralogies (calcite vs. aragonite), with different saturation states and hence potentially susceptibility to ocean acidification. Hence thresholds of both aragonite and calcite saturation will be relevant, depending on the organism. Depending on the organism, saturation changes occurring in specific regions may much more relevant than a global mean change. Also, it might be the seasonal minimum value reached, rather than annual averaged minimum, that is critical.

- Some of the arguments against some forms of ocean carbon geoengineering concern the potential for adverse impacts on marine organisms (and positive climate feedbacks) induced by decreases in the degree of oxygenation in the ocean, such as expanding and/or intensifying oxygen minimum zones. *muffin* saves 3D fields of $O_2$ concentrations that can be plotted in slices through the ocean of various orientations.

7.6.2 Other thoughts and suggestions

- If you want to combine forcings, you need to first update the file: `configure_forcings_ocn.dat` – this specifies which ocean flux forcing will be used – simply copy the relevant line from the equivalent file of the forcing to be added. You will also need to copy in the relevant ‘_sig.dat’ and ‘_SUR.dat’ files. Remember that in the user-config file, you will need to set the relevant flux scaling parameter for each different flux in the forcing.

- By default, the $CO_2$-climate feedback is ‘on’:

  ```
  # set climate feedback
ea_36=y
  ```

  Should you want to assess the impacts of geoengineering independently of changes in climate – the option is there. (Note that under some of the high end $CO_2$ emissions scenarios, there may be a degree of collapse of the AMOC that will presumably affect the patterns of ocean acidification and oxygenation etc.)

- If you are having doubts that your experiment is actually ‘doing’ anything (different from the control) – remember to create anomaly maps (plots) to look for specific changes in e.g. saturation state, pH, or the water column inventory of anthropogenic $CO_2$. Even before this – plot anomalies of the flux you think you have applied, looking specifically at the region you think you have applied it to. For this, *muffin* saves the 3D distributions of dissolved Fe and $PO_4$. See Figures below.

- Always be aware of the caveats regarding this specific model (and models in general) – how much does it different form the ‘real world’ for the modern ocean, particularly in terms of patterns of carbonate saturation? Does it even simulate anthropogenic $CO_2$ uptake adequately in the first place?
Further modifications of the biological pump in the ocean

Other manipulations of the biological pump and ocean carbon cycle are possible and potentially instructive and in the following examples may be of rather more relevance to past climates and carbon cycles and e.g. possible reasons for the low atmospheric CO$_2$ concentrations at the last glacial, as opposed to relevant to geoengineering (a good thing!). The first two of these may have profound effects not only atmospheric pCO$_2$ but also on dissolved oxygen concentrations in the ocean (and hence implications for the suitability of animal habitat such as for fish) and this is something that you will want to look at as part of your overall assessment of impacts.

Remineralization depth

In the model configuration that you have been using, the degradation of particulate organic matter sinking in the water column proceeds according to a fixed profile of flux with depth (there is no e.g. temperature control on the rate of bacterial degradation of sinking organic matter) with CO$_2$ and PO$_4$ released back to the seawater as the particulate flux decreases. The parameter that controls the (e-folding) depth scale of particulate organic matter is:

\[ \text{bg\_par\_bio\_remin\_POC\_el1} = 589.9451 \]

Either edit this value (found under the heading: # -- REMINERALIZATION --) or add a new line at the end of the user-config file specifying the value you want. Units are m.

Read Ridgwell et al. [2007] for additional discussion of this parameter. See Figure 2-4 in Ridgwell [2001] (http://www.seao2.org/pubs/ridgwell_thesis.pdf) for an illustration of how the flux of particulate organic matter decreases with depth in the ocean, plus references therein.

There is also an associated parameter: \( \text{bg\_par\_bio\_remin\_POC\_frac2} \), which sets a fraction of organic matter that is assumed to settle through the water column completely un-altered (currently assigned a value of 0.045 == 4.5%), but this is arguably less useful to change than the remineralization length-scale of the more labile fraction (the other 95.5% of particulate organic carbon exported from the ocean surface).

Note that there may well be no simple parallel that can be found in geoengineering to this process. However, there are hypotheses that during the last glacial and as a result of colder ocean temperatures, the depth scale was longer. Conversely, there are ideas about that the warmer temperatures of the e.g. Eocene ocean and hence faster rates of bacterial metabolism led to a much shallower remineralization depth scale. So a remineralization depth scale that is responsive to temperature may have importance in understanding ocean biogeochemical cycles during both past warm and cold climates as well as obviously, future global change. While you are not implementing a temperature-dependent parameterization explicitly, you can at least test for whether changes in temperature might have important impacts by simply changing the remineralization depth to be shallower (smaller depth-scale under a warming climate) or deeper (greater depth-scale in a colder ocean).

Macro nutrient inventory and uptake

Suggestions have been made that nutrients were used more efficiently during the LGM, meaning that for the same nutrient uptake at the surface more carbon was exported to depth in the ocean. See: Omta et al. [2006]. There are also a bunch of (relatively old) hypotheses concerning differences between glacial and modern ocean in how much nitrate (NO$_3^-$) there was. There is no NO$_3^-$ in this version of muffin (just PO$_4$ and Fe), but an analogous change can be made to the phosphorous cycle.

For the nutrient-to-carbon ratio in organic matter, the relevant parameter is:
7.6 Further ideas

To change the default value (106.0), add a new line at the end of the *user-config* file specifying the value you want. A larger number means that $PO_4$ is being utilized more efficiently and more organic matter is being produced for the same nutrient consumption.

To test the effect of there being more $PO_4$ in the ocean, in addition to using the (surface) flux forcing as described earlier, it is also possible to simply increase the inventory of the ocean as a whole in one go:

```
bg_ocn_dinit_8=1.0E-6
```

which will add 1 $\mu$mol kg$^{-1}$ of $PO_4$ uniformly to the ocean. (A larger/smaller number will obviously increase the glacial nutrient inventory by more/less.)

In terms of geoengineering, changing the ‘Redfield’ ocean plankton might be difficult ... but not impossible, although we are presumably talking about releases of genetically modified organisms to the entire ocean to achieve this meaning there are obviously some severe ethical concerns. However, adding macro nutrients such as $PO_4$ (more often, $NO_3^-$ is talked about) may be more feasible.

**CaCO3:POC rain ratio**

Kicked off by a classic 1994 *Nature* paper by Archer and Maier-Reimer (see: Kohfeld and Ridgwell [2009]), one potential means of changing atmospheric $CO_2$ naturally at the last glacial involves changes in the export ratio between $CaCO_3$ (shells) and $POC$ (particulate organic matter). Such a change in ratio could come about through a variety of ways (e.g., via the ‘silica leakage hypothesis’ (see: Kohfeld and Ridgwell [2009]) and also through the direct effect of $Fe$ on diatom physiology (see Watson et al. [2000] in *Nature* and also Supplemental Information). There are also ideas about an opposite ocean acidification effect, whereby the less acidic glacial (compared to modern) ocean led to increased calcification and $CaCO_3$ export. Note that this response (higher saturation == greater rate of calcification) is encoded into your model configuration – see Ridgwell et al. [2007b].

In *muffin*, the $CaCO_3:POC$ rain ratio is controlled (technically: scaled) by the parameter:

```
bg_par_bio_red_P0C_CaCO3=0.0485
```

The pattern of $CaCO_3:POC$ rain ratio is not uniform across the ocean (why? (see: Ridgwell et al. [2007, 2009]), and its pattern can be viewed in the (2D BIOGEM) netCDF variable: *misc_sur_rCaCO3toPOC*.

(Note that it is unlikely that there is any parallel in a geoengineering context to this process.)
Figure 7.1: **Mean annual ocean surface saturation (aragonite) changes.** Top: pre-industrial model ocean surface saturation (aragonite) with ReefBase tropical coral reef locations re-gridded to the **muffin** grid and color-coded with modern observationally-based saturation values. 2nd and 3rd down: Year 1994 and 2010 ocean surface saturation (aragonite) with ReefBase reef locations. Bottom: Year 2010 ocean surface saturation (aragonite) under the A2 CO2 emissions scenario. The thick white line delineates the 3.25 saturation contour (inferred to reflect a limitation on corals). **Examples here produced using muffinplot** but equally do-able in **Panoply** with the exception of achieving a data overlay. These are provided simply to illustrate some of the impacts you might consider and possible ways of visualizing them.
Figure 7.2: **Ocean surface export (particulate organic carbon) and zonal $[O_2]$ anomalies.** Left: anomalies of global mean annual export production, for $Fe$ fertilization (top), $PO_4$ addition (middle), and ocean liming (bottom). Right: Zonal mean anomalies of dissolved $O_2$ concentrations. **Examples here produced using muffinplot but equally do-able in Panoply with the exception of achieving a data overlay. These are provided simply to illustrate some of the impacts you might consider and possible ways of visualizing them.**
In the chapter we address the role and nature of marine (plankton) ecosystems.\(^1\)

**muffin** includes an explicit ecosystem component, including primary and export production as well as plankton biomass – **ECOGEM**\(^2\) – as an alternative option to the ‘biologically induced export flux’ representation of export production. The ecological model takes what is known as a size-structured approach to representing diversity of function in marine ecosystems, and is flexible in being able to be configured to represent any range of size classes of phytoplankton and zooplankton (and/or mixotrophs).

**Stuff to keep in mind...**

- We will be working with highly idealised ecosystems in a relatively idealised (modern) ocean.
- The aim is to explore why the model behaves as it does.
- The assumption is that this will give us some insight into why the real world behaves as it does. Perhaps. (It is up to you to question the validity of this assumption.)

---

\(^1\)Loosely based on original workshop material devised by Ben Ward <b.a.ward@soton.ac.uk>

Read.me

You will need a *re-start* file to carry out the experiments in the Chapter:

$ wget http://www.seao2.info/cgenie_output/wardetal.2018.ECOGEM.SPIN.tar.gz

Extract the results in the usual way and in the usual place ... and return to genie-main in the usual way ... all ... as usual.
Previously, you were running the standard ‘biogeochemical’ version of muffin. In BIOGEM, the biological pump is driven by an implicit (i.e. unresolved) biological community. As in the real ecosystem, the biological uptake of carbon and nutrients (such as phosphorus and iron) is limited by light, temperature and nutrient availability. However, unlike the real ecosystem, any uptake is directly and instantly converted to particulate and dissolved organic matter (POM, DOM) and exported to the ocean interior via (gravitational) settling and the ocean surface, respectively.

POM is then instantaneously remineralized in the ocean interior, returning nutrients and dissolved carbon and consuming oxygen (amongst other transformations and depending on the specific tracers selected in the ocean). DOM, released in at the ocean surface, is transported with the circulation of the ocean but it fast decay and remineralziation time-scale ensures that nutrients are primarily returned back to the surface ocean.

In contrast, in this chapter, we are going get started with the ‘ECOGEM’ ecological modeling package. This will allow us to extend the capabilities of muffin to examine a range of questions relating to the role of physiology and community structure in regulating the biological pump and hence atmospheric CO₂ etc. In ECOGEM, biological uptake is again limited by light, temperature and nutrient availability, but here it must pass through an explicit and dynamic intermediary plankton biomass pool, before being expressed as the production of POM and DOM (whose fate proceeds as per above):

![chemical equations]

8.1.1 Running the model

We will start with the simplest possible configuration of ECOGEM, with just a single (small) phytoplankton class. You can run this model at the command line, by entering the following command (note that this should be one continuous line)...

```
$ ./runmuffin.sh muffin.CBE.world4.BASESFeTDL LABS wardetal.2018.ECOGEM.EXAMPLE 10
wardetal.2018.ECOGEM.SPIN
```

The model will run as before, except that now we have the ECOGEM debug option enabled, and you will get a lot of extra information about how the ecological model is configured.

The additional lines appearing as the experiment runs, e.g.

```bash
>>> SAVING ECOGEM TIME-SLICE AVERAGE CENTERED @ year : 0.500
```

confirm that ECOGEM is writing ecological time-slices at the same time as BIOGEM is writing its own time slices. (Note that ECOGEM is not currently set up to save time-series data in the same way as BIOGEM.)

---

3 e.g. see Ridgwell et al. [2007]
4 Depending on the specific remineralization options chosen. The default is in fact for POM to settle downwards at a finite rate.
5 i.e. as if the ocean was populated with a small species of phytoplankton (photo-autotroph) and nothing else.
8.1.2 Viewing 2D time-slice output

Following the same convention as for BIOGEM, ECOGEM time-slice output is saved in the sub-directory of your experiment results directory, named ECOGEM.6

1. Open the fields_ecogem_2D.nc file by locating it in the correct directory, and double clicking on it in the file transfer window (if you have that software function configured), or transfer locally and then open (e.g. in Panoply).

2. You should now see a list of 2D arrays that were output by ECOGEM. Looking at the Long Name description, simply click on a variable of interest. If a menu window pops up, just click on Create or hit the Return key.

3. Check the Panoply settings to make sure you really know what you are looking at.
   - Which time-slice (i.e. simulation year) are you looking at?
   - What is the data range (i.e. colour scale)

Remember: you can change the default settings in Panoply to avoid changing things every time you open a new file. First click on Panoply → Preferences.... Here you can switch off interpolation and the grid overlay under the General menu. You can disable the spuriously precise coastline under Lon-Lat Plots.

A guide to some key ECOGEM output variables can be found at the end of Chapter ??.

8.1.3 Comparing to observations

Models are usually intended as an approximation of the real world (whatever that is). It might, therefore, be useful to check if our approximation is in anyway realistic. We can do this by comparing the model output7 to observations.

1. You can download a compilation of key biogeochemical variables (as netCDF files) from the mymuffin webpage (‘Observations for ECOGEM’).

2. You can open the GENIE_observations.nc file in Panoply in the same way as you opened the fields_ecogem_2D.nc file.

3. You can now compare the model output to a variety of key biogeochemical variables that have been derived from ocean measurements. The variables in the GENIE_observations.nc netCDF file include:
   - T and S (not so interesting, except to e.g. evaluate how well muffin simulates different temperature regimes (and habitats)).
   - DIC and ALK (ignore).
   - Phosphate is of more interest – how well does muffin simulate surface nutrient availability, and to what degree does ecosystem complexity help explain observations.
   - Observed Chlorophyll is remote-sensed, and can be contrasted with the chlorophyll biomass simulated by ECOGEM.

You might visually compare model with observations, and/or e.g. create difference maps.

4. When doing this, be asking yourself the question: does the model perform well or poorly with respect to reproducing these variables? If not, why not?

---

6 for this particular example experiment, the full path to the ECOGEM results would be:
/cgenie_output/wardetal.2018.ECOGEM.EXAMPLE/ECOGEM/.

7 NOTE: ECOGEM only saves a limited number of surface (2D) data arrays. You can look at other variables (in 2D and 3D) by opening the corresponding BIOGEM netCDF files.
8.2 Ecosystem configuration

In the last section you ran a very simple configuration of the ECOGEM ecosystem model, and compared it to observations. In this section we are going to add a bit more ecological realism, with the aim of improving model performance (i.e. as contrasted against observations). We will start by adding a zooplankton population that should bring a degree of ‘top-down’ control to the phytoplankton population.

Details of the ecosystem are specified in the user-config file: wardetal.2018.ECOGEM.EXAMPLE

1. Locate the user-config file (in ~/cgenie.muffin/genie-userconfigs/LABS), and open it in your preferred text editor.

2. The user-config file can be used to configure the ecological model to your liking.
   One of the most important amendments to note straight away can be seen near the very start: bg_par_bio_prodopt="NONE". This effectively disables the simple biological export scheme in BIOGEM, replacing it with the explicit biology of ECOGEM. This is a necessary step whenever running ECOGEM, because we do not want the implicit and explicit biological schemes to be implemented simultaneously ...

3. We can also see a load of other model parameters. Any that begin with ‘bg_’ correspond to BIOGEM, while ‘eg_’ corresponds to ECOGEM. The ECOGEM parameters begin further down the file.

4. One of the most important parameters specifies the ecosystem configuration file:
   eg_par_ecogem_plankton_file = 'NPD.eco'

   This points to a file (located in ~/cgenie.muffin/genie-ecogem/data/input/) that specifies every plankton population (‘species’, if you like) that is accounted for in the model experiment. If you open that file in a text editor, you will see something akin to the following:

   01 02 03
   \ /  \ /  \ /
   -START-OF-DATA-
   Phytoplankton 10.00 1
   -END-OF-DATA-
   \ /  \ /  \ /
   01 02 03

   DATA FORMAT AND ORDER
   ------------------------
   COLUMN #01: plankton functional type name
   COLUMN #02: plankton diameter (micrometers)
   COLUMN #03: number of randomised replicates

   INFO: TRACER ASSIGNMENT RULES
   -----------------------------
   Plankton functional type one of: Prochlorococcus
   Synechococcus
   Picoeukaryote
   Diatom
   Coccolithophore
   Diazotroph
   Phytoplankton
   Zooplankton
   Mixotroph
5. The first thing to note is that only the lines in between the -START-OF-DATA- and -END-OF-DATA- tags are read by the computer. The rest is there solely for your guidance.

Each line that is entered in the computer-readable area tells the model to create a distinct plankton population in the model. The ‘plankton functional type’ of this population is specified in the first column, while the plankton diameter is specified in the second column. A ‘1’ must always be placed in the third column (it doesn’t ‘do’ anything, but the model still needs it ... why ... ?).

In this ‘NPD’ (single nutrient-plankton-detritus) configuration, we only have a 10 micron generic phytoplankton. The ecological and physiological traits of this population are assigned automatically according to the size and the functional type (here: photo-autotroph).

NOTE: The only PFTs available at the moment are Phytoplankton, Zooplankton and Mixotroph. The other groups currently have no real functionality associated with them.

NOTE: This file format is fussy, and you cannot have any empty lines in between the -START-OF-DATA- and -END-OF-DATA- tags – every line between these tags must have data (3 parameter values).

6. We can increase the ecological complexity of the model by adding another plankton population. Save the ecosystem configuration file under a new and highly intuitive name (such as NPZD.eco), and add another line specifying a 100 micron zooplankton. It is important that the zooplankton is 10 times larger than the phytoplankton in terms of diameter. This is the optimal predator-prey length ratio in the default configuration. (You could maybe think about changing this value later on.)

7. To run the model with this new configuration, change the name of the ecosystem configuration file in the user-config file...

    eg_par_ecogem_plankton_file = 'NPZD.eco'

8. Save the new user-config file under a different name (e.g. BSS.NPZD.SPIN). You can now execute the model at the command line...

    ./muffin.CBE.worlg4.BASESFeTDTL / BSS.NPZD.SPIN 10 wardetal.2018.ECOGEM.SPIN

    ... or submit it to the cluster queue ...

    qsub -q dog.q -j y -o cgenie_log -V -S /bin/bash runmuffin.sh
    muffin.CBE.worlg4.BASESFeTDTL / BSS.NPZD.SPIN 10
    wardetal.2018.ECOGEM.SPIN

9. Once you have completed the new simulation, compare the new results to the old simulation, and also in terms of its ability to reproduce observations. Has the addition of zooplankton to the model improved its behaviour or not? Look also at the global distributions of carbon biomass in the phytoplankton and zooplankton populations (again, a log scale might help).

    How have the zooplankton interacted with the phytoplankton to change the ecological dynamics in the model?

8.2.1 Visualising composite data

We can perhaps get a better handle on this question by looking at the ratio of phytoplankton-to-zooplankton biomass. Such ratios can, however, be difficult to assess simply by eye-balling two maps. Instead we can use Panoply to combine data arrays.

8Don’t forget to change the name of the user-config file here as well ...
8.2 Ecosystem configuration

1. First close all your Panoply plot windows. Then open a new one for C Biomass - Popn. 001 (10.00 micron phytoplankton). Next, select C Biomass - Popn. 002 (100.00 micron zooplankton), and click the Combine Plot icon at the top of the Panoply window.

2. A box will open up asking you in which existing plot should I combine the variable. As you now only have one plot available, this should be a straightforward choice. Click Combine.

3. A new map should appear showing the total zooplankton carbon biomass minus the total phytoplankton carbon biomass (see the label on the colour scale). This is not what we want. Below the map, under the Array(s) tab, there is a drop down menu showing the range of different ways the two arrays can be combined. We want to look at the Z:P biomass ratio, so select Array 2 / Array 1.

4. You now need to make sure that you are looking at the right year (you can time-lock the two arrays by clicking on the chain icon). You may also find it helpful to look at the data on a log scale, with a scale range of 0.1 to 10. You might also like to change the Color Table: option to GMT_polar.cpt.

Questions:

- What does this plot say about the relationship of zooplankton and phytoplankton in different regions of the ocean?
- In what regions do zooplankton or phytoplankton dominate?
- What affect does a high Z:P ratio have on the biomass of the phytoplankton population?9

8.2.2 Iron limitation

Up to this point, we have only considered phosphate as a limiting nutrient. (Iron was included in the model, but it was not limiting to phytoplankton growth.) You can switch on iron limitation by modifying two lines in the user-config file:

```plaintext
eg_useFe=.true.

and

eg_fquota=.true.
```

Give the user-config file a new name (e.g. BSS.NPZD_Fe.SPIN), then re-run the model.

1. Examine the effect of iron limitation in the new model. What has changed?

2. We can get a more exact picture of the nutrient limitation terms via the netCDF output variables: eco2D_xGamma_Fe_001 and eco2D_xGamma_P_001. These two variables take values of between 0 and 1. A 1 indicates that the factor is not limiting to growth. A 0 indicates the factor is completely preventing growth.
   - In what regions are iron and phosphorus more or less limiting to growth?
   - In regions where neither is limiting, what other factors might be important?

3. Plankton stoichiometry plays a critical role in determining which nutrient is most limiting to growth. You can increase the plankton Fe : C ratio by increasing the minimum and maximum iron quotas. Look at the parameters `eg_qminFe_a` and `eg_qmaxFe_a` in the user-config file.

---

9For example, in terms of the chlorophyll concentration.
Chapter 8. Marine ecosystems and dynamics

- What happens to the ecosystem if you increase these parameters by a factor of 2, 5 or 10?
- How does a change in these parameters affect the model behaviour?
- What has changed in terms of the patterns of nutrient limitation?
- What has happened to the concentration of the limiting and non-limiting nutrient?

NOTE: It can be ‘risky’ to just change the parameter value in place, as you might forget what you started with ... Instead; you might copy/paste a new version of the line in question, and comment out the original by placing a ‘#’ at the beginning of the line. For example:

```python
#eg_qminFe_a = 3.0e-6
eq_qminFe_a = 6.0e-6
```

changes the minimum iron quota by a factor of 2, whilst keeping a record of the original setting (inactivated by the #). Or, you might do something like:

```python
eq_qminFe_a = 6.0e-6 # 3.0e-6
```

that reduces the total number of lines you end up with in the user-config file.

4. Nutrient supply ratios are also important in determining the limiting nutrient.

   The bg_par_det_Fe_sol_exp parameter determines the solubility of atmospheric iron inputs in seawater. Decreasing the value of bg_par_det_Fe_sol_exp will therefore decrease the iron-to-phosphorus supply ratio.

   - What happens to the ecosystem if you decrease bg_par_det_Fe_sol_exp by e.g. 10, 20 or 50%?
8.3 Increasing ecological complexity

In the last section, we looked at the results of some simulations based on ‘NPD’ (one nutrient-(phyto)plankton-detritus) and ‘NPZD’ (one nutrient-(phyto)plankton-zooplankton-detritus) type ecosystem models. Here we will begin to incorporate a bit more ecological complexity.

8.3.1 Plankton size classes

We are going to add a few more plankton size classes, so we have small, medium and large phytoplankton and zooplankton.

1. Save the ecosystem configuration file under a new name (e.g. 3P3Z.eco), replacing the existing plankton populations with the ones described in Table 8.1.
2. To run the model with this new configuration, change the name of the ecosystem configuration file in the user-config file:
   ```
   eg_par_ecogem_plankton_file='3P3Z.eco'
   ```
3. Save the new user-config file under a different name (e.g. BSS.3P3Z.SPIN) and then run the new model at the command line (e.g. for 10 years).

<table>
<thead>
<tr>
<th>j</th>
<th>PFT</th>
<th>Diameter (μm)</th>
<th>j</th>
<th>Functional Type</th>
<th>Diameter (μm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Phytoplankton</td>
<td>0.6</td>
<td>4</td>
<td>Zooplankton</td>
<td>6.0</td>
</tr>
<tr>
<td>2</td>
<td>Phytoplankton</td>
<td>6.0</td>
<td>5</td>
<td>Zooplankton</td>
<td>60.0</td>
</tr>
<tr>
<td>3</td>
<td>Phytoplankton</td>
<td>60.0</td>
<td>6</td>
<td>Zooplankton</td>
<td>600.0</td>
</tr>
</tbody>
</table>

8.3.2 Viewing 2D time-slice output

Open up the 2D time-slice data for the new experiment, following the same procedure as in the previous section. You will now see a lot more time-slice variables now appear listed in Panoply. We have all the same diagnostics as before, plus some new ones relating to the new plankton populations you have just added. We also have a load of other arrays describing the size distribution and diversity of the photosynthetic community (non-phototrophic populations are ignored in these metrics). These were not included before, because there was only one phytoplankton population.

Size fractions

Variables beginning “eco2D_Size_Frac_...” give the chlorophyll biomass in the three size fractions:

1. picophytoplankton (diameter ≤ 2 μm)
2. nanophytoplankton (2 < diameter ≤ 20 μm)
3. microphytoplankton (diameter > 20 μm)

Size metrics

Any other variables beginning ‘eco2D_Size_...’ give metrics describing the phytoplankton size distribution.

- eco2D_Size_Mean: Geometric mean\(^\text{10}\) phytoplankton diameter, weighted by carbon biomass

\(^{10}\)We use the geometric mean and standard deviation, because phytoplankton biomass is approximately log-normally distributed across the phytoplankton size range.
– **eco2D_Size_Stdev**: Geometric standard deviation\(^{11}\) of phytoplankton diameter, weighted by carbon biomass.

– **eco2D_Size_Minimum**: Diameter of smallest phytoplankton contributing \(>0.1\%\) of the total phytoplankton carbon biomass.

– **eco2D_Size_Maximum**: Diameter of largest phytoplankton contributing \(>0.1\%\) of the total phytoplankton carbon biomass.

### Diversity metrics

Any variables beginning 'eco2D_Diversity...' give metrics describing the phytoplankton diversity.

– **eco2D_Diversity_Threshold**: the threshold diversity index. The number of species contributing \(>0.1\%\) of the total phytoplankton carbon biomass [Barton et al., 2010]\(^{12}\).

– **eco2D_Diversity_Berger**: the inverse Berger(-Parker) index [Berger and Parker, 1970]\(^{13}\). The proportion of carbon biomass made up by all but the single most dominant population. For example, if the dominant population accounts for 40% of the total carbon biomass, inverse Berger (-Parker) index is 0.6.

– **eco2D_Diversity_Simpson**: the inverse (Gini-)Simpson index [Simpson, 1949]\(^{14}\). This is effectively the probability that two samples taken at random from the community will be from a different species (note that the probability of selecting a population is dependent on carbon biomass, not cell abundance). If we define the proportional biomass of each species as its relative contribution to the total carbon biomass in the community, the inverse Gini(-Simpson) index is calculated as one minus the sum of the squares of the proportional biomasses of each species.

– **eco2D_Diversity_Shannon**: the Shannon(-Wiener or -Weaver) index [Shannon, 1948]\(^{15}\). With the proportional biomass defined as above, the Shannon index is defined as the sum of [the proportional biomass multiplied by the logarithm of the proportional biomass] for each species.

**NOTE:** The threshold index is a fairly crude measure of the total number of species in the community, relative to a small and arbitrary threshold of relative biomass. This index is not very sensitive to the relative biomass of individual species (although one very successful species can raise the absolute value of the threshold, thus lowering the diversity). The other three indices do more to quantify the evenness of the community. The more unequal the proportional abundances, the smaller the value of the index. If almost all the abundance is concentrated into one type and all the other types are very rare, the latter three indices can become very small. A community with fewer species, but with more evenly distributed biomass, may well have higher values for these three diversity indices.

---

\(^{11}\) The geometric standard deviation describes the relative size range of the phytoplankton. For a geometric standard deviation of \(\sigma\), \(\sim 68.2\%\) of the phytoplankton carbon biomass will be in cells no more than \(\sigma\) orders of magnitude smaller or larger than the geometric mean size.


Have a look at some of these metrics, but bear in mind that they summarise the diversity of a phytoplankton community that includes just three species. They are probably not that revealing, so we will come back to them later. Instead, have a look some of the other metrics describing the model ecosystem.

Questions:

- What are the global distributions of the different size classes?
- How do the global biomass distributions compare to variables such as temperature\textsuperscript{16}, or primary production (Uptake Fluxes C)?
- How does nutrient, light and temperature limitation vary between the size classes?
- Can you account for the distribution biomass between the size classes according to the different limiting factors?

### 8.3.3 Create your own ecosystem

Save the ecosystem configuration file under a new name and add some more plankton populations (whatever and as many as you like\textsuperscript{17}). Update the muffin parameter `eg_par_ecogem_plankton_file` in the `user-config` and save this file under a new name. Run the new model.

Questions:

- How many populations can you get to coexist (i.e. each having a non trivial (>0.1%) biomass)?
- What effect do the new populations have on the community as a whole?
- What effect, if any, do your additions have on the strength of biological export? (e.g. look at `bio_fpart_POC` in `fields_biogem_3D.nc`.)

\textsuperscript{16}Ocean temperature is saved in `fields_biogem_3D.nc`

\textsuperscript{17}Just bear in mind that the more populations you put in, the slower the model will run!
8.4 Build it up, tear it down

8.4.1 A fully size-structured ecosystem

We are now going to switch to a more diverse version of the size-structured ecosystem model. This configuration has 8 size classes of phytoplankton, and 8 size classes of zooplankton, as shown in Table 8.2.

1. Save the ecosystem configuration file under a new name, replacing the existing plankton populations with the ones described in Table 8.2.

2. Update the user-config to point to the new ecosystem configuration file, and save again under a new name. (It is generally a good idea to make a note of the name and goal of each experiment as you set it up.)

3. Run the new model for at least 20 years (this will probably take about 10-15 minutes).

Table 8.2: Plankton functional groups and sizes.

<table>
<thead>
<tr>
<th>j</th>
<th>PFT</th>
<th>Diameter (μm)</th>
<th>j</th>
<th>Functional Type</th>
<th>Diameter (μm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Phytoplankton</td>
<td>0.2</td>
<td>9</td>
<td>Zooplankton</td>
<td>0.6</td>
</tr>
<tr>
<td>2</td>
<td>Phytoplankton</td>
<td>0.6</td>
<td>10</td>
<td>Zooplankton</td>
<td>1.9</td>
</tr>
<tr>
<td>3</td>
<td>Phytoplankton</td>
<td>1.9</td>
<td>11</td>
<td>Zooplankton</td>
<td>6.0</td>
</tr>
<tr>
<td>4</td>
<td>Phytoplankton</td>
<td>6.0</td>
<td>12</td>
<td>Zooplankton</td>
<td>19.0</td>
</tr>
<tr>
<td>5</td>
<td>Phytoplankton</td>
<td>19.0</td>
<td>13</td>
<td>Zooplankton</td>
<td>60.0</td>
</tr>
<tr>
<td>6</td>
<td>Phytoplankton</td>
<td>60.0</td>
<td>14</td>
<td>Zooplankton</td>
<td>190.0</td>
</tr>
<tr>
<td>7</td>
<td>Phytoplankton</td>
<td>190.0</td>
<td>15</td>
<td>Zooplankton</td>
<td>600.0</td>
</tr>
<tr>
<td>8</td>
<td>Phytoplankton</td>
<td>600.0</td>
<td>16</td>
<td>Zooplankton</td>
<td>1900.0</td>
</tr>
</tbody>
</table>

8.4.2 Ecosystem characteristics

We can now begin to look at the size and diversity metrics of the phytoplankton community in a meaningful way.

1. Look first at:
   (a) the total carbon biomass
   (b) the carbon uptake flux (i.e. primary production)
   (c) the geometric mean size

Make sure in each case that you are looking at the last year of model output. You may also find it useful to adjust the colour scale, or to change to a logarithmic colour scale (e.g. try a logarithmic scale from 2 to 20 microns for the geometric mean size).

Looking at the maps, we can perhaps pick out three different “biomes” in terms of their community properties:

(a) The low-latitude oligotrophic gyres are relatively unproductive, and support some of the lowest annual mean biomass in the surface ocean. In these regions the mean phytoplankton size is very small.

(b) Subpolar latitudes between 40° and 50° (either N or S) are much more productive, and support very high annual mean biomass. These communities also have the highest mean sizes of any region.
8.4 Build it up, tear it down

(c) The polar oceans are also highly productive (except perhaps the high Arctic), and support relatively high annual mean biomass. These communities are made up (in the model, at least) of slightly smaller phytoplankton than we see in the subpolar regions.

2. What can we find out about the community structure in these regions? Open up some of the other metrics describing the community (standard deviation of size distribution, size fractionation, diversity, limiting factors). What can you find out about the community structure within each region, in terms of coexistence and exclusion?

- Does the community span a broad or narrow size range?
- How many size classes are coexisting in each biome?
- What is the smallest and largest size class in each biome?
- How much biomass is concentrated in each size fraction (picoplankton, nanoplankton and microplankton)?

Overall – what factors do you think are most important in terms of dictating the global distribution of each size class?

To find out the answers to these questions, you are going to pull the model apart, and then put it back together. At each stage the aim is to bring in a different limiting factor, so that you can see its effect on the model behaviour.

**The fundamental niche**

The first step is to find out the impact of abiotic factors on the distribution of different phytoplankton sizes. In other words, we need to find out what the distribution of the phytoplankton would be in the absence of any ecological interactions, such as resource competition and predation. This is effectively their ‘fundamental niche’.

The fundamental niche is fairly abstract, and not something that can be measured in the real world. In model world, however, we can get a useful estimate of the fundamental niche by making a few simple changes to the model.

1. First of all, you can remove all predation, simply by removing the zooplankton from the `ecosystem configuration` file. Once again, you will need to save a new and appropriately named `user-config` file.

2. Next, you also need to remove all competition for nutrients and light. This involves tweaking the model equations so that the phytoplankton are not nutrient limited, and do not attenuate light. To do this, all you need is to add the following line to the `user-config` file.

   ```
   eg_fundamental = .TRUE.
   ```

3. If you now run the model (just 10 years should do it, in this case), you should have a community of eight phytoplankton size classes that are growing solely as a function of the incoming light and the temperature. This growth will be balanced by basal (i.e. non-grazing) mortality. As there is no feedback between the ecosystem and the environment, populations that can survive will grow exponentially and without limit, potentially reaching astronomical abundance in very little time. Populations that cannot survive will rapidly decline to almost nothing.

   The regions in which each plankton shows positive growth defines its fundamental niche. This is a function of abiotic conditions only, and is the absolute limit of its geographical range. Look at the carbon biomass distribution in each size class (set the data range in each case from 0 to 1 mmolC m\(^{-3}\)).

   - How and why does the fundamental niche vary with size?
Chapter 8. Marine ecosystems and dynamics

• Could the limits of the fundamental niche explain some of the patterns seen in the full model?

Resource competition.
The next step is assess the impact of resource competition. We are first going to do this in the absence of any zooplankton grazing.

1. All you need to do at this stage is to re-enable nutrient and light competition. To do this, simply delete ‘eg_fundamental = .TRUE.’ from the user-config file (or comment out the line to disable it), and save under a new name. Leave the ecosystem configuration file as it is.

2. You should have a community of eight phytoplankton size classes that are competing for nutrients and light, again as a function of temperature. This is a much more realistic simulation, as feedbacks between the ecosystem and the environment serve to limit the size of the phytoplankton populations. Examine the model to find out:
   • What size classes are able to persist when resource competition is enabled?
   • Why are different size classes more or less abundant in different areas?
   • How does the distribution of each size class compare to the fundamental niche?
   • What are the reasons for any differences?

Phytoplankton biogeography at this stage begins to approximate the realised niche, which defines the range of conditions that support a population in the presence of ecological interactions. Note that at this stage, however, we have ignored the effects of any predator-prey interactions, as the zooplankton grazers are still missing.

Resource competition + one generalist zooplankton
The previous simulation is clearly unrealistic (although, hopefully, informative). You are now going to add back in just a single zooplankton class, that grazes equally on all plankton (including itself).

1. Add a 100 micron zooplankton into the ecosystem configuration file, and save under a new name. Also update the user-config file to reflect the change, and save under a similar name.

2. You need to modify the model so that the zooplankton eats all prey with equal preference. This can be done by adding the following lines to the ecosystem configuration file.

   eg_ns=1
   eg_pp_sig_a=1.0e99

NOTE: For aficionados, the first parameter disables prey-switching (i.e. predators no longer preferentially attack the most abundant prey). The second parameter increases the width of the grazing kernel (i.e. predators can attack a range of prey across a huge size range with equal preference).

3. The addition of zooplankton to the model community should give a more accurate approximation of the realised niche.
   • Does the addition of a single zooplankton grazer enable more or less coexistence?
   • What factors might be responsible for any shifts in biogeography?

Resource competition + one “switching” zooplankton
You began with a full food-web containing 8 phytoplankton and 8 zooplankton size classes. The diversity of zooplankton clearly has an effect on the phytoplankton community that is not seen in the previous experiment. This effect can be imitated with just one generalist zooplankton if we instruct it to graze preferentially on the most successful prey.
8.4 Build it up, tear it down

Re-enable this ‘prey switching’ effect by changing the following control parameter to a 2:

\[
\begin{align*}
eg_{gns} &= 2 \\
eg_{gpp\_sig\_a} &= 1.0 \times 10^9
\end{align*}
\]

Compare this simulation to the first experiment (8 phytoplankton and 8 zooplankton) to see how the inclusion of prey switching increases coexistence through the ‘kill-the-winner’ mechanism.

- How does nutrient limitation change with phytoplankton size, and how might zooplankton be affecting this?
- Look at the C:P biomass ratio in the community as a whole, and compare to your estimates from the NPZD model (Lesson 1).
- How does the C:P ratio vary with size? How does having a diverse community affect the coupling of carbon and limiting nutrients?

**Further questions to answer**

- What sets the fundamental niche, and how does it change with size?
- How is the fundamental niche modified by resource competition?
- What species are favoured in terms of nutrient competition?
- How is the outcome of competition affected by...
  - Abiotic conditions?
  - Increased mortality (through generalist grazing)?
  - Density-dependent mortality (through specialist grazing)?
- Do these experiments tell you all you need to know? What other modifications can you think of making?

### 8.4.3 Mixotrophy

Try adding some mixotrophs to the phytoplankton and zooplankton already present in the community. These will have exactly half the nutrient uptake traits of phytoplankton of a similar size, and half the prey capture traits of zooplankton if a similar size. To do this:

1. Save the previous ecosystem configuration file under a new name.
2. Edit the new ecosystem configuration file and add an additional line to add a mixotroph:

```
Mixotroph xxx 1
```

where xxx is the class size of the mixotroph. (And remember to save it.)
3. Update the user-config to point to the new ecosystem configuration file, and save again under a new name. (It is generally a good idea to make a note of the name and goal of each experiment as you set it up.)
4. Run the new model for at least 10-20 years.

**Further questions to explore/answer:**

- How does this effect the mean and standard deviation of cell size? (Size and diversity metrics will be calculated for phytoplankton and mixotrophs together)
- How does mixotrophy effect the C:P ratio of organic matter?
- How does the realised niches of mixotrophs compare to the fundamental niches of phytoplankton?
Also: try replacing all the phytoplankton and zooplankton with a range of sizes of mixotrophs. How does the simulation differ from one with the same size range of separate phytoplankton and zooplankton classes?

For instance – you might also try having an ecosystem comprising just a single small (e.g. $10\mu m$) mixotroph (no phytoplankton and no zooplankton), and perhaps an ecosystem with one small mixotroph $10\mu m$ and one additional mixotroph, 10 times larger $100\mu m$. Compare the productivity of such an ocean compared to some of your previous simplified ecosystem configurations (such as a single small $10\mu m$ phytoplankton, and a configuration with a single small phytoplankton and a single larger $100\mu m$ grazer).
8.5 Ecology in ... future oceans

What might happen to primary production and the biological pump as (future) climate continues to change? What might happen to the distribution of species (here: size classes) in the ocean? Does it 'matter' (e.g. for marine bio geochemical cycling and feedbacks on atmospheric \( pCO_2 \))? (And how different is the Earth system response when using a complex ecosystem model rather than the highly simplified representation of biological export you used previously?)

There are all good questions for a model!

The best place to start is with the full complexity ecosystem structure with 8 size classes for both phytoplankton and zooplankton (as described in Ward et al. [2018]), and continuing on from a re-start of the pre-industrial ocean (and ecology) – wardetal.2018.ECOGEM.SPIN.

For the user-config – for which you can either adapt your own based on your earlier ecosystem experimenting, or adapt the published configuration used to create the re-start – wardetal.2018.ECOGEM.SPIN (in: user-configs/MS/wardetal.2018) – NOTE the different user-config directory!!! and remember to rename it\(^{18}\) so you do not overwrite the re-start ...

Then:

1. First, create a control, e.g. based on wardetal.2018.ECOGEM.SPIN, so that you can more easily spot any mis-matched parameter values or other problems – do this, turn off the atmospheric \( pCO_2 \) forcing (so that \( pCO_2 \) can respond to and indicate any significant changes in the marine carbon cycle), but leave on the dust flux forcing (because we still need a supply of dissolved iron to the ocean surface):

   ```
   # specify forcings
   bg_par_forcing_name="worlg4.FeMahowald2006"
   ```

2. Second, you will need either:
   - A user-config with an conceptual emissions scenario (e.g. anything that you have played with previously). This can be run directly on from the pre-industrial spin-up. This will allow you to explore questions concerning the impact on marine ecosystems and carbon cycling of the rate of warming (driven by your assumed rate of carbon emissions), and/or the importance of and sensitivity of marine ecosystems and carbon cycling, to the total (maximum) warming.
   - A template emissions forcing (plus dust) is provided as: worlg4.FpCO2_Fp13CO2.FeMahowald2006

   As per before, you need to re-scale the values in the forcing because the \( CO_2 \) flux is assumed to be in units of \( Pg\text{yr}^{-1} \)\(^{20}\) (and \( \delta^{13}C \) is given a default value of 1.0). i.e. add the lines:

   ```
   bg_par_atm_force_scale_val_3=8.3333e+013
   bg_par_atm_force_scale_val_4=-27.0
   ```

   and remember that the forcing is provided as a pulse of 1 molyr\(^{-1}\) (1PgC yr\(^{-1}\) when scaled) carbon emissions over just a single year, and hence the time-series file for the forcing: biogem_force_flux_atm_pCO2_sig.dat needs to be edited – see Chapter 6.

---

\(^{18}\)The new user-config file need not be saved back into this same directory.

\(^{19}\)The base-config is as per you have been using in this Chapter – muffin.CBE.worlg4.BASESFeTDTL

\(^{20}\)As per notes in the forcing time-series .sig file itself.
Or:

- A historical transient (plus dust) forcing and then run from e.g. 1750 through 2010 (as per before), for which the forcing is:
  
  `worlg4.historical2010.FeMahowald2006`

  Remember that here that you do not re-scale the atmospheric $pCO_2$ (or $\delta^{13}C$) value as was done with the forcing used to generate the re-start (this historical transient forcing includes the actual values of $pCO_2$ and in the correct units).

  Note that with the full complexity ecosystem model included, the 245 years of this experiment will take much longer than before when you ran a historical transient experiment. You could 'cheat' a little and start from year 1850 rather than 1765, on the basis that nothing much happens in between (in terms of rising $pCO_2$ and warming). To do this – simply run the transient experiment only for 160 years (1850 to 2010) and adjust the starting date in the `user-config`:

  ```
  # change start year
  bg_par_misc_t_start=1850.0
  ```

  You will also need `time-series` and `time-slice` saving that aligns with the historical (and future) period:

  ```
  # save frequency
  bg_par_infile_slice_name='save_timeslice_historicalfuture.dat'
  bg_par_infile_sig_name='save_timeseries_historicalfuture.dat'
  ```

  (i.e. just as before when you ran a historical transient).

  Once you have run this experiment, you can use it as a re-start for a future emissions scenario, with a forcing adapted from e.g. `worlg4.FpCO2_Fp13CO2.FeMahowald2006` and remember to add the emissions units scaling:

  ```
  bg_par_atm_force_scale_val_3=8.3333e+013
  bg_par_atm_force_scale_val_4=-27.0
  ```

  Whichever route you chose, will also require (another) control experiment run for the same number of years (and with dust but not $pCO_2$ forcing).

What to look for? The questions (at the start) should help guide you in this, and might include:

- Patterns of primary productivity and export (where do they increase, decrease), plus time-series of global totals (what is the global impact of warming?).
- Patterns of ocean oxygenation (and patterns of dysoxia and anoxia).
- Surface ocean nutrient distributions, and nutrient limitation.
- Spatial patterns of plankton size – either simply for mean size, or look at the patterns of biomass of specific size classes. (You might follow a the preferred habitat/location of a particular size class with latitude to see whether it experiences a 'range shift'.)
- Diversity etc.

You might also contrast with emissions scenarios that you might have run previously employing the simplified biological export scheme in BIOGEM, rather than the complex ecosystem model of ECOGEM. (Make sure you are comparing between the same emissions scenarios). The question to be answered would be: does representing marine ecology 'matter'?
8.6 Ecology in ... past oceans

Here we consider a series of examples\(^{21}\) of the ECOGEM model of marine ecology, applied to published paleo configuration of muffin and used to ask questions of how different could marine carbon cycling (and atmospheric \(pCO_2\)) and oxygenation have been in the (relatively recent) past and how do model projections compare with proxy observations.

The three examples come firstly from the Last Glacial Maximum, with direct comparison being made to post glacial time (here, the late Holocene), with the intention to explore the role of climate cooling, altered ocean circulation, and increased iron supply to the ocean in modifying plankton distributions and size structures. Secondly, from around the time of the Paleocene-Eocene Thermal Maximum (PETM) some 55 Myr ago, now both considering a different paleogeography, a warmer ocean, and then transient warming on top of that. Lastly we turn to marine ecology and carbon cycling in the aftermath of the impact and extinction event at the end of the Cretaceous, some 66 Myr ago, and return to our earlier exploration of what happens if you lose the larger plankton in the ocean.

\(^{21}\)Either published or in the works or existing in some publication fantasy ...
Chapter 8. Marine ecosystems and dynamics

The Last Glacial

The Last Glacial Maximum (LGM) (ca. 19 to 23 ka) was characterized by lower sealevel and higher ocean salinity, colder ocean temperatures, and a reorganized meridional overturning circulation in the Atlantic. The latter 2 changes in particular should have impacted marine ecosystems and indeed, observations suggest range migration (temperature-tracking) and shifts in the zone of highest productivity in the Southern Ocean, amongst other impacts. The aim of this particular investigation, is to assess what these ecological changes are (at least in model world).

Provided is a configuration of LGM ocean circulation that has been tuned to fit observations of benthic carbon isotopes, which provide an observational constrain on large-scale ocean circulation. To this, you’ll an ecosystem (in place of the default BIOGEM ’induced export’ scheme). Also provided is a muffin configuration for late Holocene (‘HOL’) (0-6 ka), created in exactly the same way and also tuned to respective (0-6 ka) benthic carbon isotope data. Configuration HOL provides a point of comparison (or control) for you to compare the ecology in a colder, LGM ocean against.22

The user-configs you need to use, or copy-rename, can be found in the directory:
genie-userconfigs/MS/odalenetal.CP.2019
(and you can run your experiments from here23, or better, copy-edit-rename the user-configs you run experiments with, from the LABS sub-directory).

Read the readme.txt file for instructions for the basic set of command line parameters needed, but remember that you may be running your user-config with a different name and likely from a different sub-directory. Spin up the following experiments (and then experiment with them later)24:

(1) muffin.CB.GIteliiaa.BASESFeTDTL_rb base-config
   + muffin.CB.GIteliiaa.BASESFeTDTL_rb.SPIN user-config
(2) muffin.CB.GIteliiva.BASESFeTDTL_rb base-config
   + muffin.CB.GIteliiva.BASESFeTDTL_rb.SPIN user-config

Submit these to the cluster ... remembering that you will need to recompile muffin between (1) and (2) (and potentially also re-compile before running (1) if you have not used that particular base-config immediately prior). Run the spin-ups for 10,000 years.

Neither of these configurations currently have an ecosystem enabled, but they will provide a baseline against which you can contrast a pair of experiment that include explicit ecology. What you need to do know is to add in/enable ECOGEM. To do this, you need to modify both base- and user-config files of both the HOL and LGM model configurations.25

1. base-config – First, you need to enable the ECOGEM module.
   At the top of the HOL base-config file muffin.CB.GIteliiaa.BASESFeTDTL_Crb26 you will see the line:
   
   ma_flag_ecogem=.FALSE.

   Simply change this to .TRUE.

---

22 i.e. you run pairs of HOL and LGM experiments and contrast between them, rather than necessarily comparing to previous modern configurations.
23 If you use them in this directory, make sure at the command line, you replace LABS with MS/odalenetal.CP.2019.
24 Noting that the long file-names differ only in a single ‘v’ vs. an ‘α’ ...
25 Strongly recommended that that you copy-rename both sets of files and edit the copies.
26 And similarly for the LGM one.
2. **user-config** – Next, some deletions and additions are needed in the **user-config** provided (which only has implicit biological export enabled). In the section of the file under the heading:

```bash
# # --- BIOLOGICAL NEW PRODUCTION -------------------------------------
#
```

You are going to delete everything there (in that section) and replace it with:

```bash
# biological scheme ID string
bg_par_bio_prodopt="NONE"
```

The only other thing you need then is a section of code that defines all the **ECOGEM** ecological parameters.

Go open file wardetal.2018.ECOGEM.SPIN, which you can find in the **user-config** sub-directory MS/wardetal.2018, and you’ll see under the heading:

```bash
# # --- ECOGEM ----------------------------------------------------------------
#
```

A long list of parameter settings (down to the next section headed **DATA SAVING**). Simply copy-paste this entire section (including **ECOGEM** header lines if you like), anywhere in your **user-config** file. At the very end of the file would do just fine.27

And lastly ... under:

```bash
# # --- MISC -----------------------------------------------------------------
#
```

(And the end of the section) add the following lines

```bash
# kraus-turner mixed layer scheme on (1) or off (0)
go_imld = 1
```

This enables a 'mixed layer depth' scheme in the ocean circulation model; that **ECOGEM** needs to calculate light limitation during esp. intervals of high latitude / wintertime deep mixing.

This ... should do-it, i.e. you have added the same tuned ecosystem model as described in Ward et al. [2018] to your LGM / HOL **user-configs**.

Now you are ready to run new LGM and HOL experiments, with a full ecosystem in each28.

Obvious questions to investigate include not only how ecosystems and patterns of biological export may have differed between LGM and HOL, but also how patterns of nutrients (PO4 and Fe) may have differed ... and also distributions of dissolved O2 in the ocean (and in the interior, rather than across the ocean surface). A water mass ventilation age tracer has also been simulated and the results of this will help in understanding how global circulation patterns differ between the 2 time intervals.

You can also compare between with and without **ECOGEM** versions as the only thing that changes between pairs of HOL or pairs of LGM experiments is the biological scheme29.

---

27Try and ensure that there is blank line at the very end of the file just in case **muffin** has any problems reading it in.

28Remembering that presumably both your **base-config** and **user-config** file names are different as compared to running the non **ECOGEM** version described in the **readme**.

29Actually, this is not true, as in the **ECOGEM** enabled experiments, the mixed layer scheme in the ocean circulation model is also activated and which has an impact on ocean circulation. You could then create a 3rd set of HOL+LGM experiments, using the basic **BIOGEM** implicit export biological scheme, but now also setting the **go_imld** parameter to a value of 1.
Chapter 8. Marine ecosystems and dynamics

The PETM

In this practical we are going to look at the ocean as it might have been just over 55 million years ago, at the Paleocene-Eocene Thermal Maximum (PETM) – in short a lot warmer, and with a somewhat different continental configuration and hence ocean circulation. The exercise is based on a recent ‘ECOGENIE’ (muffin configured to include ECOGEM) publication\textsuperscript{30} which you should read first.

We are going to make the rather strong assumption that the ecosystem is structured according to exactly the same rules as in the modern ocean, and simply run the same ecological model configuration but in a new climate and ocean environment. However, because we don’t really have any good (or any!) data constraints on the iron supply to the early Eocene ocean via e.g. dust, the ecological configuration does not include iron as a limiting nutrient. Bear that in mind when thinking about your results.

The user-configs you need to use, or copy-edit-rename, can be found in the directory:
genie-userconfigs/MS/wilsonetal.2018
(and you can run your experiments from here, or better, copy-edit-rename the user-configs you run experiments with, from the LABS sub-directory).

Read the readme.txt file for instructions for the basic set of command line parameters needed, but note that you may be running a user-config with a different name and likely from a different sub-directory. You want to spin up the following experiments first (and then experiment with them later):

(1) Modern
(3) Early Eocene CO2 and Climate

Submit these to the cluster, but remember that you will need to recompile muffin between (1) and (3) (and re-compile before (1) if you have not been using the base-config muffin.CBE.workh2.BASES immediately prior to this). Run the spin-ups for 10,000 years.

When you have completed the pair of spin-ups, see what you can diagnose and learn about how ecosystems and the spatial pattern of ecology and biological export differ between a colder modern ocean and the warmer Eocene ocean.

For example, in the warmer Eocene world:

- What has happened to the mean plankton size in different regions?
- What has happened to the fundamental niches in different size classes?
- What has happened to the realised niches?
- Is the system more or less productive?
- Has carbon export gone up or down?

See what you can find out about the two systems and think about the mechanisms that might be responsible for the differences . . .

Note that to be more comparable with the Eocene, this particular modern configuration also lacks iron co-limitation. We could also try and remove the effect of the Eocene being warmer than the modern so as to concentrate just on the effect of a different continental configuration and hence ocean circulation. The experiment (included in the user-config sub-directory and briefly described in the readme file:

(2) Late Paleocene Early Eocene Paleogeography
does exactly this, i.e. attempts to 'remove' the effect of higher ocean temperatures by running an Eocene experiment at $\times 1CO_2$.

Conversely, we could run modern at $\times 3CO_2$ and then compare to the $\times 3CO_2$ Eocene experiment. This alternative comparative experiment is provided as:

(S1) Modern with $3 \times CO_2$

and will also require a 10,000 year long spin-up … Note that the 2 strategies, while slightly different, are attempting the same thing (i.e. isolating the effect of paleography and ocean circulation from coeval climate change and warming).

Finally – having run some or all of these spin-ups and contrasted the results (focussing on ecological patterns and biological export, but remembering also to assess differences in ocean circulation), you can investigate the impact of geologically rapid warming a-la the PETM, on the system (esp. ecological patterns and biological export plus ocean circulation). There are two immediately obvious possible approaches:

1. You could use a $\times 1CO_2$ spin-up as a re-start – either or both of modern and Eocene continental configurations – and run the $\times 3CO_2$ experiment from this.
   This will give you an instantaneous warming – much faster than occurred associated with PETM onset, and also faster even than modern anthropogenic warming. However, it does provide a nice simple idealized perturbation to investigate and analyse.
   An experiment duration of 100, or even 10 years, might be sufficient.\footnote{But running for a full 10,000 years would enable you to follow not only the initial rapid warming perturbation, but also the long-term recovery and adjustment to a new steady state.}

Conversely and for fun, you could also start from a $\times 3CO_2$ spin-up, and run a $\times 1CO_2$ experiment, to achieve a rapid cooling. Investigate the differential ecological response to rapid cooling vs. rapid warming.

2. Secondly, in the user-configs, you might note near the bottom is a forcing that determines the value of atmospheric $CO_2$:

   # specify forcings
   bg_par_forcing_name="pyyyyyz.RpCO2_Rp13CO2"

   followed by a line that specifies either $\times 1CO_2$ or $\times 3CO_2$, e.g.
   bg_par_atm_force_scale_val_3=278.0E-06

   or
   bg_par_atm_force_scale_val_3=834.0E-06

   (followed by a line specifying the carbon isotopic composition of the atmosphere).

   Similar to as you have seen before for a fossil fuel $CO_2$ emissions forcing, there is a file:
   biogem_force_restore_atm_pCO2_sig.dat
Referring back to the instructions for changing fossil fuel CO₂ emissions forcing, you should be able to modify this (or better, copy-rename a new forcing directory and edit the file biogem_force_restore_atm_pCO2_sig.dat in that) to create a prescribed time-dependent change in atmospheric CO₂. Note that in the format of biogem_force_restore_atm_pCO2_sig.dat, the values in the second column scale the value of the parameter bg_par_atm_force_scale_val_3 in the user-config. Hence, in biogem_force_restore_atm_pCO2_sig.dat, setting a value of 2.0 in place of 1.0, will double the applied CO₂ forcing. Exactly as per in the fossil fuel CO₂ emissions exercises, pulses, ramps, etc etc can be constructed to control the rate and shape of the applied change in atmospheric CO₂.

Either way – assess the important and impact of the rate of CO₂ rise and hence warming. (Equally, you might explore rapid cooling and how that fundamentally differs in impact from warming.)
The Oceans, Post end Cretaceous Impact
8.7 Ecology in ... fake oceans!

We can also consider the question of the causes and consequences of different ecologies in
the ocean in a more general way and return to our hypothetical or ‘fake’ oceans.

For any of your ‘fake’ worlds’ that you generated earlier, you can add carbon and nutrient
cycling (if that was not already included), plus a marine ecology. To do this, you first need to
create a new base-config that includes all the carbon cycling and nutrients needed by ECOGEM (and
BIOGEM), then you need to create/configure a suitable user-config.

1. base-config – First, you need to enable the ECOGEM module.
   At the top of the your base-config file, change the ECOGEM ‘flag’ to true:

   ```
   ma_flag_ecogem=.TRUE.
   ```

   Next, it is likely that you did not add any biogeochemical tracers when you created your fake
world, and the base-config section:

   ```
   # *******************************************************************
   # TRACER CONFIGURATION
   # *******************************************************************
   # the total number of tracers includes T and S
   # T and S do not need to be explicitly selected and initialized
   # *******************************************************************
   # Set number of tracers
   GOLDSTEINNTRACSOPTS='$(DEFINE)GOLDSTEINNTRACS=2'
   # list selected biogeochemical tracers
   # <<< >>>
   # list biogeochemical tracer initial values
   # <<< >>>
   ```

   Go find the base-config file muffin.CBE.p0055c.BASES in the configs sub-directory of genie-
main. Copy all of the section headed

   ```
   # *******************************************************************
   # TRACER CONFIGURATION
   # *******************************************************************
   # Set number of tracers
   PRINT = 0
   ```

   into your base-config, replacing the original contents with only 2 tracers selected.

2. user-config – Next, you need to define some biogeochemical cycling PLUS and ecosystem.
   Go find the user-config file: wilsonetal.p0055c.8P8Z.pal.3x in the genie-userconfigs sub-directory
   MS/wilsonetal.2018.

   Easiest, is to simply re-use (copy-rename) the user-config file: wilsonetal.p0055c.8P8Z.pal.3x.

   The only parameters you might want to adjust\(^ {32}\), other than a different ecological structure
   (and eg_par_ecogem_plankton_file), is atmospheric $CO_2$, which is set to $\times3CO_2$ by:

   ```
   bg_par_forcing_name="pyyyyy.RpCO2_Rp13C02"
   bg_par_atm_force_scale_val_3=834.0E-06
   bg_par_atm_force_scale_val_4=-4.9
   ```

   (and atmospheric $\delta^{13}C$ to $-4.9$).

   Strictly, the scaling for the air-sea gas exchange coefficient, for a fake world, should be:

   ```
   # re-scale gas transfer coefficient ...
   bg_par_gastransfer_a=0.722
   ```

   (changing the value from 0.5196 to 0.722).

---

\(^{32}\)Note that ocean alkalinity is also set for an Eocene world.
Those changes – enabling ECOGEM and adding ocean (and atmosphere) tracers to your base-config, and then taking a paleo ECOGEM user-config as a template to work from, should get you going with an ecology in your fake world.

If you also want to diagnose ocean circulation better and add a ventilation tracer, then in the base-config, increase the number of selected tracers by 2 (under # Set number of tracers) and add the following 2 lines to the list of selected tracers:

```plaintext
gm_ocn_select_48=.true. # r
gm_ocn_select_49=.true. # b
```

and ... in the user-config, add (anywhere, but e.g., in the MISC parameter section):

```plaintext
# add ventilation tracers
bg_ctrl_force_ocn_age=.true.
```
8.8 EcoGEnIE 1.1

As outlined in the Sections: Choosing a template user-config and Configuring muffin experiments, a number of recommended changes to the configuration of ECOGEM as published by Ward et al. [2018] (EcoGEnIE 1.0).

These changes have been incorporated into an example user-config (in the genie-userconfigs sub-directory EXAMPLES: muffin.CB.worlg4.BASESFeTDTL.ECOGEM_NEW.SPIN and paired with the base-config: muffin.CB.worlg4.BASESFeTDTL), and are:

- Under *** REMINERALIZATION ***:
  
  # set 'instantaneous' water column remineralization
  bg_par_bio_remin_sinkingrate_physical=9.9E9
  bg_par_bio_remin_sinkingrate_reaction=125.0

  which instantaneously remineralizes all particulate organic matter throughout the water column according to the remineralization profile and/or reaction rates. This is activated via a 'very large' value for bg_par_bio_remin_sinkingrate_physical. At the same time, reaction rates (including scavenging) are calculated as if the sinking rate was finite and equivalent to the value of: bg_par_bio_remin_sinkingrate_reaction (m d$^{-1}$).

- Under *** MISC ***:
  
  # set mixed layer to be only diagnosed (for ECOGEM)
  go_ctrl_diagmld=.true.
  # add seaice attenuation of PAR
  egCTRL_PARseaicelimit=.true.
  # relative partitioning of C into DOM
  eg_par_beta_POCtoDOC=0.75

  which firstly substitutes a diagnosed mixed layer depth, rather than actually applying mixed layer physics to ocean circulation, then limits light available under sea-ice in proportion to the fractional sea-ice cover in that grid cell, and lastly, re-partitions carbon from POM to DOM and is tuned to produce an approximately Redfield ratio (106) of 104.7 : 1 in C : P of exported POM., and then:
  
  # maximum time-scale to geochemical reaction completion (days)
  bg_par_bio_geochem_tau=90.0
  # extend solubility and geochem constant T range (leave S range as default)
  gm_par_geochem_Tmin = -2.0
  gm_par_geochem_Tmax = 45.0
  gm_par_carbchem_Tmin = -2.0
  gm_par_carbchem_Tmax = 45.0

  which firstly, limits the maximum consumption of any particular reactant by a single reaction, to an imposed lifetime (90 days), with the remaining parameters extending the valid temperature range for solubility and geochem constants to $-2 - 45^\circ C$. Note that the valid salinity range is left unchanged.

For reference, the original (i.e. as per in Ward et al. [2018]) but reformatted, user-config, is provided as: muffin.CB.worlg4.BASESFeTDTL.ECOGEM_NEW.SPIN (also in the genie-userconfigs sub-directory EXAMPLES).
9. The land surface
9.1 xxx

9.1 xxx
10. Atmospheric dynamics
10.1 xxx

10.1 xxx
10.2 Geoengineering
The long tail of CO$_2$ (and other tales from the sediments)
Sediment model output
Quantifying how long is the long tail of CO$_2$
Sediments of the modern Earth
The marine geology of fake worlds
Further ideas

11. Geological carbon processes


**Chapter 11. Geological carbon processes**

**README**

You will need to download a new *re-start* file. To fetch this – change to the cgenie_output directory, and type:

```
$ wget http://www.seao2.info/cgenie_output/EXAMPLE._rwlma.P04_S18x18.SPIN2.tar.gz
```

Extract the contents of this archive by typing:

```
$ tar xfvz EXAMPLE._rwlma.P04_S18x18.SPIN2.tar.gz
```

You’ll then need to change directory back to genie-main directory in order to run muffin.
11.1 The long tail of CO₂ (and other tales from the sediments)

This chapter introduces the marine sediment model component in muffin – SEDGEM (SEDiment GEochemistry Model) plus ROKGEM, the terrestrial weathering module. The model experiments now include, a representation of deep-sea sediments and interaction between the preservation and burial of CaCO₃ and ocean chemistry plus, they account for the balance (or imbalance) between weathering on land and sedimentary burial (of CaCO₃). For an over-view of the sediment model and what time-scales and nature of carbon cycle interaction between ocean and sediment you can expect, read: Ridgwell and Zeebe [2005] and Ridgwell and Hargreaves [2007]. ROKGEM is described in Colbourn et al. [2013].

Take the new model configuration for a test drive by running on from the provided re-start experiment: EXAMPLE_rwlma.PO4_S18x18.SPIN2.tar.gz.

This is a steady-state climate+carbon cycle experiment that includes the deposition of CaCO₃ in deep-sea sediments as well as the balance between weathering (solute input to the ocean) and burial (output), which in this case, are in fact in balance (because the model has been run for 50,000 years to ensure this ...). Try running (‘briefly’, but 100 years would not be too tedious for this faster configuration!):

```
$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg_sg_rg_gl._rwlma.BASES LABS LAB_11.EXAMPLE 100 EXAMPLE_rwlma.PO4_S18x18.SPIN2
```

Note that the base-config file cgenie.eb_go_gs_ac_bg_sg_rg_gl._rwlma.BASE specifies the use of a sediment model – the ‘sg’ in the filename. Also note that we are running at a degraded (lower) 18 × 18 resolution, and requiring fewer (twice as few in fact!) time-steps per year, for an overall approximately order-of-magnitude decrease in run-time as compared to a standard 36 × 36 resolution configuration of muffin. This is extremely helpful in being able to run muffin on geological

Figure 11.1: Schematic of SEDGEM sediment component.
time-scales but within a reasonable about of real time. (The configuration also utilizes a concep-
tual/idealized continental configuration somewhat similar to as in the snowball Earth experiments.)

The user-config LAB_11.EXAMPLE is set up with the global carbon cycle configured as 'open' – that is to say, that there is an input of carbon (and alkalinity) to the ocean from weathering, and a loss due to preservation and burial of \( \text{CaCO}_3 \) in deep-sea sediments. Depending on the state of ocean chemistry (and biology) and weathering, these two fluxes (input vs. output) do not have to balance, and hence ocean carbonate chemistry (and in turn, atmospheric \( \text{pCO}_2 \)) can evolve with time. The spin-up may not necessarily have the two fluxes perfectly balanced and hence before you run any experiments you might want to confirm whether the spin-up provided really is adequately 'spun-up'.

Note that a residual drift can be dealt with if it is relatively small and near linear via the use of a control experiment, because any experiment you carry out will likely also incorporate (or be biased) by the same residual drift. Running a control gives you something to directly contrast your experiment with and calculating the difference (e.g., a difference map or simple subtraction of global numbers) will give you the effect of whatever parameters you changed in the experiment but corrected for any drift. In previous exercises, we were a little lazy, and difference maps were often created with respect to year 1 of an experiment – strictly, they should have been created relative to the same year of a parallel control experiment, i.e., results at year 100 should have been contrasted with the year 100 results of the control.

\[ \text{1i.e. you might plot some of the variables from the results of the spin-up experiment as a function of time and judge whether they are sufficiently converged or not.} \]
11.2 Sediment model output

There is a whole new set of additional outputs from this configuration of muffin, particularly sediment-specific output from the SEDGEM module and which is saved in the sedgem sub-directory of the main experiments results directory. Data saving differs from BIOGEM in that the composition of the sediments (and other diagnostics such as rain and dissolution fluxes) is saved only at the very end of a model experiment (hence unlike BIOGEM, which by default saves a series of time-slices throughout the course of a model experiment). So if you kill a run before the very end (or the run crashes), you will get no (or little) SEDGEM output.

2D (e.g. surface sediment properties, fluxes, etc.) results can be found in the sedgem sub-directory of your experiment directory in a netCDF file called fields_sedgem_2d.nc. (Note that there is some duplication of results saving, because a series of time-slices of sediment composition are also saved in the 2D BIOGEM netCDF file fields_biogem_2d.nc. BIOGEM also saves a selection of time-series of sediment properties – .res files starting biogem_series_sed. For example, the time-series file: biogem_series_sed_CaCO3.res contains information about how the mean CaCO3 content of surface sediments (uppermost sediment layer) evolves with time.

The 2D distribution of wt%CaCO3 – which is the weight fraction of calcium carbonate (CaCO3) in the surface sediments of the deep ocean (i.e., how much plankton carbonate shell material is there compared to other stuff in the mud at the bottom of the ocean) is saved under a variable called: sed_CaCO3.

The model also generates artificial sediment ‘cores’ (e.g. see: Ridgwell [2007]) and hence what one might expect to see of your applied perturbations as recorded in a sediment core sand then recovered from the ocean floor(!) In the SEDGEM results sub-directory, there is a netCDF file which contains all the locations selected (if any) — sedcore.nc. These are not really aligned with latitude as the Panoply display might suggest – the locations are in fact distributed from all over the ocean (Panoply is being fooled in order to display them together). In the SEDGEM 2D netCDF file, these locations are marked in the netCDF variable grid_mask_sedcore. The locations of these cores are stored in a a file containing a little ASCII ‘map’ of the ocean.

- If you are using a ‘paleo’ configuration of muffin, indicated by a parameter section in the base-config file headed by something looking like:

```
# ***************************************************
# GRID & BOUNDARY CONDITION CONFIGURATION
# ***************************************************
# insert the automatically generated muffingen parameter list here
# ***************************************************
### cGENIE .config file parameter lines generated by muffingen v0.84 on: 191119 ###
```

then there will be a parameter line that direct muffin to look for SEDGEM configuration files in the respective genie-paleo directory, e.g.:  

```
sg_par_pindir_name='../../cgenie.muffin/genie-paleo/fkh_pp03/'
```

- Otherwise, the file lives in: cgenie.muffin/genie-sedgem/data/input

The filename is given by the parameter: sg_par_sedcore_save_mask_name

Simply be editing (using the ASCII text editor) a ‘0.0’ to a ‘1.0’, you can get the model to generate and save a sediment ‘core’ at that particular (i,j) location.
Chapter 11. Geological carbon processes

netCDF file `sedcore.nc` variables include:

- `phys_layer` — sediment layer number (counting down).
- `phys_depth` — (cumulative) depth below surface, measured from the sediment surface to the mid-point of each sediment layer (cm).
- `th` (cm) — thickness of each sediment layer (cm).
- `age_CaCO3` — the mean age of CaCO$_3$ particles in a sediment layer.
  
  Note that this will not be defined if there is no CaCO$_3$ preserved.
- `phys_porosity` — sediment layer density (as if you cared!).
- `sed_POC` and `sed_POC_13C` — mean organic matter content of each sediment layer and its $\delta^{13}C$. But note: in this configuration no organic matter is preserved (hence all zeros for POC).
- `sed_CaCO3` and `sed_CaCO3_13C` — mean CaCO$_3$ content (wt%) of each sediment layer and its $\delta^{13}C$.
- `sed_det` and `sed_ash` — the wt% detrital and ‘ash’ contents of a layer (ash is used as a conservative numerical sediment tracer in order to mark the depth of the start of the experiment).

Obviously — you could plot e.g. CaCO$_3$ (or its $\delta^{13}C$) as a function of depth and/or age across and see how your carbon release experiment might be recorded in the marine geological record (e.g. how does this compare with observations across the PETM?).

Note that the sediment cores reflect not only the material which has accumulated (or not, if it has dissolved . . .) during the course of your experiment, but also the material that accumulated during the 50,000 year spin-up. PLUS, whatever material the sediment core was initialized with to start with. For example, the large interval of initially 100% detrital material at the base of the sediment core simply reflects the initialization of the sediment array in the model.

Also note the ash ‘peak’ near the bottom of the stack (filled) sediment layers – this is a tracer to ‘tag’ the start of the model spin-up. If you look at the spin-up results (not your recent perturbation experiment) – the ash peak lies in a sediment layer with age 50,000 years. But why is there any ash deeper than the age corresponding to the start of the spin-up? How can it get there (i.e. what processes could move solid material deeper within the sediment column)?
11.3 Quantifying how long is the long tail of CO$_2$

You are now considering a rather more complex carbon cycle than before (e.g. now including a number of additional, mostly sediments/weathering processes). It is hence worth conducting some idealized perturbations of the global carbon cycle to get a feel for the sensitivity and time-scale of the (now ‘geological’) system response.

For instance – one illustrative experiment, and which has a parallel to experiments you have conducted previously, is to add a pulse CO$_2$ release to the atmosphere and track the consequences for atmospheric pCO$_2$ and ocean chemistry (particularly ‘alkalinity’), and now also e.g. deep sea sediments. To the LAB_11.EXAMPLE user-config, add the lines:

```plaintext
bg_par_forcing_name='pyyyyzy.FpCO2_Fp13CO2'
bg_par_atm_force_scale_val_3=8.3333e+016
bg_par_atm_force_scale_val_4=-27.0
```

in order to achieve a 1000 PgC total release in a single year (or multiply by 5 to be comparable-ish to the 5000 PgC release in Ridgwell and Hargreaves [2007]) with an isotopic composition of $-27^\circ/oo$ – appropriate for a fossil fuel carbon source. For reference – methane derived carbon (e.g. as from hydrates) would be more like $-60^\circ/oo$.

BE CAREFUL – if you have previously edited files of the forcing pyyyyyz.FpCO2_Fp13CO2, you are going to need to return whatever you have edited back to its original state. Or, carry out the following procedure:

1. From genie-main, $\$ make cleanall
2. Then head up a directory level (cd ..) to cgenie.muffin
3. Type: $ git status -uno
   You should see a list of all files that you originally installed (cloned from GitHub) that you have changed. e.g. you might see:
   ```plaintext
   ```
   (Note that file you create are not listed. TO see those, type: $ git status)
4. To restore the state of any file that you might have edited, git checkout that file out, e.g.:
   ```plaintext
   $ git checkout genie-forcings/pyyyyzy.FpCO2_Fp13CO2/biogem_force_flux_atm_pCO2_sig.dat
   ```

Or ... simply keep the units scale parameter set in the user-config:

```plaintext
bg_par_atm_force_scale_val_3=8.3333e+013
```

as you have done earlier, and edit the number of PgC yr$^{-1}$ in the forcing time-series (.sig) file. (You still might want to restore the forcing file to its original state if you have previously edited it.)

Run the model for as long as you dare (or can be bothered) – 1,000 or 2,000 years might be just enough to start to see impacts on deep-sea sediments, but 5,000 or 10,000 years would be much better. (You can always submit this to the cluster queue and get on with something else.) FYI: 10,000 years is going to take something like an hour ... if you are lucky ...

Plot the time-series of e.g. atmospheric pCO$_2$ and compare to the (much shorter experiments) you have carried out before with a simple ocean-atmosphere only system. Compare how quickly atmospheric pCO$_2$ decays compared to previously muffin publications$^{2}$ (e.g. Ridgwell and Hargreaves [2007]) or other models (e.g. Archer et al. [2009]) and how the sediments respond (e.g. the time-series of sediment CaCO$_3$ content).

---

$^{2}$see: http://www.seao2.info/pubs.html
To properly (quantitatively) appreciate the role of ocean-sediment interaction (and weathering) and controlling atmospheric \( pCO_2 \), you need to contrast these experiments with as similar a model configuration as possible – for instance, one that is identical excepting having no sediments (or weathering). You can achieve this quite simply: create (/copy-rename) a new user-config based on LAB_x.sediments and edit the lines:\(^3\):

```markdown
# set an 'OPEN' system
bg_ctrl_force_sed_closedsystem=.false.
```

changing it to:

```markdown
# set a 'CLOSED' system
bg_ctrl_force_sed_closedsystem=.true.
```

What this does is to force the model to always maintain an exact balance between the preservation and burial in marine sediments of \( CaCO_3 \), with the supply of solutes derived from the weathering of \( CaCO_3 \) on land. Because no excess or deficit of weathering vs. sedimentation is allowed to occur, no changes in ocean chemistry (other than by air-sea gas exchange) occur. This configuration hence acts (geochemically and dynamically) exactly the same way as a configuration without any sediments or weathering being present (and as used previously). By comparing the two experiments: can you deduce the effect of the sediments in modulating (accelerating or decelerating) atmospheric \( pCO_2 \) decline?\(^4\)

Also view the sediment distribution (of \( CaCO_3 \)): what are the impacts on sediment composition in the case of an experiment configured with an ‘open’ system (vs. a ‘closed’ system)? Here, the time-series file of mean global sediment composition biogem_series_sed_CaCO3.res (in units of: wt\% \( CaCO_3 \)) may help illustrate what is going on here. Note that the way the ‘closed’ system is constructed; a response of the sediments is predicted and saved in the output, even though it is not allowed to affect chemistry or atmospheric \( pCO_2 \).

To recap – you are aiming to run a \( CO_2 \) emissions experiment using a muffin configuration including variable weathering and sedimentation (LAB_11.EXAMPLE ) with a version/copy that you have edited to create a ‘closed’ ocean-atmosphere system, with no ocean-sediment or weather feedbacks on atmospheric \( CO_2 \). Ideally, you would also run a pair of control experiments – both ‘open’ and ‘closed’ configurations but with no \( CO_2 \) emissions specified. (A total of 4 experiments.)

---

\(^3\)You do not have to edit the comment line (\#) but it will help you remember what you have done.

\(^4\)e.g. you could compare the \( pCO_2 \) time-series of the 2 different experiments.
Important is to critically and quantitatively assess to what degree muffin provides an adequate representation of the interaction between ocean chemistry and sediment composition (e.g., in CaCO3 buffering of CO2 release and ‘carbonate compensation’). Key to this, is contrasting model output against observational-based data. Such an approach is presented in Ridgwell and Hargreaves [2007].

The required base- and user-config files are provided as part of the muffin code release:

- cgenie.eb_go_gs_ac_bg_sg_rg.worbe2.BASE – the base-config file, including SEDGEM and ROKGEM modules.
- EXAMPLE.worbe2.RidgwellHargreaves1997_S36x36.SPIN1 – the user-config for the 1st stage (20 kyr long) spin-up described in Ridgwell and Hargreaves [2007]. This file lives in the genie-userconfigs sub-directory.
- EXAMPLE.worbe2.RidgwellHargreaves1997_S36x36.SPIN2 is the 2nd-stage, 50 kyr spin-up that uses EXAMPLE.worbe2.RidgwellHargreaves1997_S36x36.SPIN1 as a re-start.

E.g. you would run the 2 experiments, one after the other:

```
./runmuffin.sh cgenie.eb_go_gs_ac_bg_sg_rg.worbe2.BASE /EXAMPLE.worbe2.RidgwellHargreaves1997_S36x36.SPIN1 20000
```

```
./runmuffin.sh cgenie.eb_go_gs_ac_bg_sg_rg.worbe2.BASE /EXAMPLE.worbe2.RidgwellHargreaves1997_S36x36.SPIN2 50000
```

Does the model get the broad patterns right (is it more right than wrong, or more wrong than right)? Do you think the model-data misfits might be important?

(Pre-run experiments of the same name are also provided from: http://www.seao2.info/cgenie_output/)
11.5 The marine geology of fake worlds

You can configure any of your previous fake worlds, or generate new ones, to have a full geologic carbon cycle including deposition of CaCO$_3$ in marine sediments and weathering on land.

In order to generate the requisite SEDGEM configuration files, you need the following settings in your muffingen configuration file:

- opt_makeseds=true; – requests that SEDGEM files are generated.
- par_sedsopt=2; – requests that a randomized bathymetry is generated. This is needed because fake worlds have by default, a flat bathymetry.

Instead, if you 'draw' a non-uniform bathymetry, you would set:

- par_sedsopt=0; – requests that the ocean depth levels ('k1' file') are used to inform the ocean floor depth assumed by SEDGEM.
  (Option 1 is most commonly used in conjunction with a GCM-derived continental configuration.)

So, if you define continents in your fake world, but do not change the bathymetry in muffingen, par_sedsopt option 2 will give you a randomized distribution of sediment depths (used in the CaCO$_3$ solubility pressure calculation only). Selecting option 0 will simply translate your chosen muffingen ocean depths into SEDGEM depths. Note that you can still have a flat bottom to the ocean (no variation in ocean floor depth) and choose option 0.

If you define a specific bathymetry by hand (e.g. draw in ocean ridges), you probably do not want it over-written (by a random SEDGEM depth pattern) and hence you should chose option 0.

For speed of running model experiments, it is recommended that you generate your muffingen muffin configurations at the lowest reasonable resolution – 18 × 18 × 8 would be suitable – 18 × 18 resolution in lon vs. lat, and 8 levels in the ocean. (You could try pushing this a little further and more extreme, e.g. 12 × 12 × 8.) Note that when you create your base-config, you should use the template:

CONFIG_template_08lvl_R07.config
(for the 8-level ocean, rather than CONFIG_template_16lvl_R07.config which is designed for 16-ocean level configurations).

Once you have copy-pasted the configuration output of muffingen into the template base-config file (and suitably renamed it), you need to enable the geologic carbon cycle modules (SEDGEM and ROCKGEM). At the top of the base-config file, ensure that the following are set:

ma_flag_sedgem=.TRUE.
ma_flag_rokgem=.TRUE.

And then, further down and under the heading:

# TRACER CONFIGURATION

you need to define how many 'tracers' in the ocean, what they are, and any initial values that differ from modern defaults. Plus, whatever atmospheric and sediment tracers you want (plus default values in the atmosphere). Simplest at this point is to copy-paste from an existing base-config, such as:

cgenie.eb_go_gs_ac_bg_sg_rg.worbe2.BASE

(which is the base-config used in Ridgwell and Hargreaves [2007]).

As for a suitable user-config ...
11.5 The marine geology of fake worlds

... you could take the *user-config* from *Colbourn et al.* [2013] (itself a modification of Ridgwell and Hargreaves [2007]) – available as part of the *muffin* code release, as file:

```
EXAMPLE.worbe2.Colbournetal2013CTRL (genie-userconfigs sub-directory)
```

Then generalize this according to the paleo sediment configuration used in *Ridgwell and Schmidt* [2010] (file: EXAMPLE.p0055c.RidgwellSchmidt2010.SP1 in the genie-userconfigs sub-directory) (i.e. making CaCO3/POC export invariant, adding a fixed detrital flux to the sediments).

Cutting the lines down to the bare minimum (excepting comments pertaining to new/altered lines), a suitable *user-config* then looks like:

```
# --- CLIMATE --------------------------------------------------
# enable CO2 climate feedback
ea_36=y
# --- BIOLOGICAL NEW PRODUCTION --------------------------------
bg_par_bio_k0_PO4=1.9582242E-06
bg_par_bio_c0_PO4=2.1989611E-07
# --- ORGANIC MATTER EXPORT RATIOS -----------------------------
bg_par_bio_red_DOMfrac=0.66
# --- INORGANIC MATTER EXPORT RATIOS ---------------------------
# set fixed export CaCO3 as a proportion of particulate organic matter
bg_par_bio_red_POC_CaCO3=0.200
bg_par_bio_red_POC_CaCO3_pP=0.0
# --- REMINERALIZATION -----------------------------------------
bg_par_bio_remin_DOMlifetime=0.5
bg_par_bio_remin_POC_frac2=6.4591110E-02
bg_par_bio_remin_POC_el1=550.5195
bg_par_bio_remin_POC_el2=1000000.0
bg_par_bio_remin_CaCO3=0.468
bg_par_bio_remin_CaCO3_el1=1083.486
bg_par_bio_remin_CaCO3_el2=1000000.0
# --- SEDIMENTS ------------------------------------------------
# sediment diagenesis options
sg_par_sed_diagen_CaCO3opt="ridgwell2001lookup"
sg_ctrl_sed_bioturb=.true.
sg_ctrl_sed_bioturb_Archer=.false.
sg_par_n_sed_mix=20
sg_par_sed_mix_k_sur_max=0.15
sg_par_sed_mix_k_sur_min=0.15
# additional detrital flux (g cm-2 kyr-1)
sg_par_sed_fdet=0.180
# --- WEATHERING -----------------------------------------------
# set a ’OPEN’ system
bg_ctrl_force_sed_closedsystem=.false.
# set CaCO3_weathering-temperature feedback
rg_opt_weather_T_Ca=.TRUE.
# set CaSiO3_weathering-temperature feedback
rg_opt_weather_T_Si=.TRUE.
# weathering reference mean global land surface temperature (C)
rg_par_ref_T0=8.48
#CO2 outgassing rate (mol C yr-1)
rg_par_outgas_CO2=5.59E+12
# global silicate weathering rate (mol Ca2+ yr-1)
rg_par_weather_CaSiO3=5.59E+12
# global carbonate weathering rate (mol Ca2+ yr-1)
rg_par_weather_CaCO3=5.59E+12
# d13C
rg_par_outgas_CO2_d13C=-6.0
rg_par_weather_CaCO3_d13C=12.8
# --- DATA SAVING ------------------------------------------------
bg_par_data_save_level=4
bg_ctrl_debug_lvl0=.true.
ma_debug_loop=1
# --- END -------------------------------------------------------
```
Chapter 11. Geological carbon processes

Here, the reference temperature against which the rate of weathering is modified for higher/lower temperatures, is set for a modern-like continental configuration:

```plaintext
# weathering reference mean global land surface temperature (C)
rg_par_ref_T0=8.48
```

If left un-changed, then the negative silicate weathering feedback will operate on atmospheric $CO_2$ – letting it accumulate, or removing it, until the reference temperature is achieved. If instead, you want to achieve a specific atmospheric $CO_2$ at steady state, you need to know the relevant mean global land surface temperature. To determine this – first run `muffin` with a prescribed atmospheric $CO_2$ value, e.g. by adding:

```plaintext
# --- FORCINGS -------------------------------------------------
# specify forcings
bg_par_forcing_name="pyyyyz.RpCO2_Rp13CD2"
bg_par_atm_force_scale_val_3=278.0E-06
bg_par_atm_force_scale_val_4=-6.5
```

(or some other value of $CO_2$) to your `user-config`. Running the model for about 1000 years should be more than enough to achieve a near steady-state of surface climate, allowing to read off the mean global surface air temperature over land (BIOGEM file `biogem_series_misc_SLT.res`), and set the reference temperature to this value. You can then run an experiment without a prescribed atmospheric $CO_2$ forcing, now knowing that the silicate weathering feedback will always act to restore atmospheric $CO_2$ back to that value.

Despite the low grid resolution, this is still going to take a long time for particularly much to happen and you could probably do with some (numerical) 'help'. You can accelerate the run-time of `muffin` in calculating the balance between weathering and sedimentation, a-la Lord et al. [2015]. To do this, you need to add a new section of parameter choices to the `user-config` file:

```plaintext
# --- GEOCHEM ACCELERATION -------------------------------------
g1_ctrl_update_pCO2=.true.
m1_gem_notyr_min=10
m1_gem_notyr_max=10
m1_gem_yr_min=90
m1_gem_yr_max=90
m1_gem_dyr=0
m1_gem_adapt_auto=.false.
```

Here, this specifies that you would like to spend only 10 years of full updating of the model for each 90 years spent in an accelerated calculation which treats the entire ocean as if it were a single geochemical reservoir (see: Lord et al. [2015]) – an almost 10 : 1 acceleration and reduction in experiment run-time. (But note that the ocean circulation and ocean carbon and nutrient cycles will appear to be taking longer to come to equilibrium because they are only being fully updated 10 years in every 100.

Note that this acceleration is good for spin-ups and determining equilibrium weathering-sedimentation states, but not so useful for transient $CO_2$ changes (see in Lord et al. [2015] for how a less acceleration and adaptive setup had to be used). Similarly, acceleration is not so useful for varying orbits experiments.
A couple of different possible experimental investigations follow.⁵

1. Firstly, you could simply test what happens if there is no sediment surface depth variability as compared to a configuration with depth (pressure) variability in sediment surface depth (and one that follows a modern-like distribution of depth vs. area). The science question would be something like – how important is a distribution of ocean depths to setting the steady state alkalinity (or saturation) of the ocean? Or: in having an appreciable amount of shallower sea-floor where $\text{CaCO}_3$ will be preserved and buried much more readily – how much less saturated must the ocean as a whole become in order to balance weathering and sedimentation? For this, you’ll need 2 muffingen configurations, each with the same continental configuration, but one generated with random sediment depths, and one with a uniform depth. Simplest, but more tedious, is to run muffingen twice, with your configuration settings files differing only in the value of par_sedsdp. However, faster is to generate one configuration using the configuration template EXAMPLE_BLANK.m, and then generate a 2nd one, using the ‘k1’ file generated by the first muffingen muffin generation. An example of creating a muffin configuration from a ‘k1’ file is: EXAMPLE_K1_permian.m. Basically, the only thing that changes between the first and second muffingen muffin generation, in addition to par_sedsdp, is the value of par_gcm, which changes from ‘(empty) in EXAMPLE_BLANK.m, to ‘k1’ in EXAMPLE_K1_permian.m.

2. As a variant on the above – you might ‘draw’ a large and relatively shallow (e.g. 1 or 2 km depth) sea-floor plateau, run the model, and determine how much this has influenced ocean chemistry, at steady state (remembering to run an experiment with no sea-floor plateau in a second experiment as a control).

3. How important is continental configuration and the position of the continents in controlling atmospheric $\text{CO}_2$ via weathering? You might, for instance, create 2 different continental configurations in muffingen, both with the same cratonic (land surface) area, and explore whether you achieve a different steady state atmospheric $\text{CO}_2$ depending on whether the continent is at the pole, or centered on the equator. Remember that the parameterized silicate weathering feedback acts to restore global mean surface air temperature over land, to the reference value.⁶ You might then guess the sign of the change in atmospheric $\text{CO}_2$ given that the weathering will act to restore the polar and equatorial continental temperatures to the same value and it does that by causing atmospheric $\text{CO}_2$ to change. (But perhaps the magnitude of the effect will be unexpected, which is why you need a model.)

---

⁵Note that in using the given user-config settings, steady state is only reach after several 100 kyr (which is going to take a while, even at low resolution and with acceleration). Regardless of your assumptions regarding sea-floor bathymetry, you should find that atmospheric $\text{CO}_2$ always reaches the same value.

⁶Note that the simple weathering parameterization in muffin does not care how much land there is, only what the mean surface air temperature over that land is.
11.6 Further ideas

You might further explore the role of weathering and sensitivity to the sensitivity of atmospheric $pCO_2$ and climate to the strength of the weathering feedback as well as to the assumed rate of volcanic $CO_2$ outgassing, as well as how the products of weathering on land are accommodated through the burial (and pattern of burial) of $CaCO_3$ in deep-sea sediments. Parameters of 'interest' here, i.e. ones that you might adjust to explore the silicate weathering feedback and long-term controls on atmospheric $pCO_2$, are listed under # -- WEATHERING --) in the user-config, and include:

- $rg\_par\_outgas\_CO2$ — the global $CO_2$ outgassing rate in units of molyr$^{-1}$.
- $rg\_par\_ref\_T0$ — the reference land surface temperature for weathering (units of $^\circ C$).

In muffin, global weathering rates will increase or decrease depending on whether the mean global surface temperature (which is reported and saved in BIOGEM time-series output file: biogem_series_msec_SLT.res). So, if you increase the reference temperature, weathering rates will drop and $CO_2$ will accumulate in the atmosphere until the Earth was warmed sufficient that the mean global land surface temperature again matches the reference temperature. And vice versa for the case of reducing the reference temperature. Maybe try a pair of experiments (plus a control in which you do not adjust the reference temperature) in which you adjust the value of $rg\_par\_ref\_T0$ both up and down, to confirm this.

Conversely, increasing or decreasing the rate of $CO_2$ outgassing should also act to increase or decrease, respectively, global temperatures (at steady state). Another pair of experiments (plus a 3rd, control) would be to try increasing and decreasing the value of $rg\_par\_outgas\_CO2$.

Now ... one problem concerns the time it is going to take the system to re-equilibrate to e.g. a different value of $CO_2$ outgassing. For example, in Greene et al. [2019]®(and in the Supporting Information data table S1 for the paper), you’ll see time-series of the adjustment of atmospheric $pCO_2$ in response to a change in outgassing rate. And then see it is a ca. 2 Myr time-scale to equilibrium ...

Help is at hand! And if you read the earlier fake world section, you can accelerate the time-to-equilibrium for systems such as these. Simply add, to your user-config:

```plaintext
# --- GEOCHEM ACCELERATION ----------------------------------------
gl_ctrl_update_pCO2=.true.
ma_gem_notyr_min=10
ma_gem_notyr_max=10
ma_gem_yr_min=990
ma_gem_yr_max=990
ma_gem_dyr=0
ma_gem_adapt_auto=.false.
```

Following Greene et al. [2019] and as per the example user-config files associated with this which can be found in: genie-userconfigs/MS/greeneetal.2019 – this specifies that you would like to spend only 10 years of full updating of the model for each 990 years spent in an accelerated calculation which treats the entire ocean as if it were a single geochemical reservoir (see: Lord et al. [2015]) – an almost 100 : 1 acceleration and reduction in experiment run-time. (But note that the ocean circulation and ocean carbon and nutrient cycles will appear to be taking longer to come to equilibrium because they are only being fully updated 10 years in every 1000.

Another pair of important controls on the preservation and burial of \( \text{CaCO}_3 \) in marine sediments (but not one that ultimately at steady state, affects \( \text{CO}_2 \) and climate), are the fluxes of \( \text{CaCO}_3 \) and organic carbon (POC) to the sediment surface.

All other things being equal – increasing the export of \( \text{CaCO}_3 \) and hence flux to the sediment surface requires that fraction of \( \text{CaCO}_3 \) that dissolves increases at steady state. In other words – instantaneously increasing the flux to the sediments of \( \text{CaCO}_3 \) will increase burial relative to the weathering flux. The excess sink of \( \text{CaCO}_3 \) will act to lower DIC and ALK (and \( \text{Ca}^{2+} \)), lowering the carbonate saturation state of the ocean, and reducing the preservation of \( \text{CaCO}_3 \). This will continue up to the point where the preservation and burial of \( \text{CaCO}_3 \) once again balances weathering.

An interesting question is what happens to atmospheric \( \text{CO}_2 \). Ultimately it must return to its initial value due to the silicate weathering feedback. However, this occurs on a long (ca. 200 kyr) time-scale, whereas an imbalance between \( \text{CaCO}_3 \) weathering and burial can act to change ocean chemistry on much shorter (1-10 kyr) time-scales. To explore this, you can adjust the parameter that sets the ratio of \( \text{CaCO}_3 \) to POC export – having the effect of changing the \( \text{CaCO}_3 \) flux to the sediments whilst keeping everything else (almost) in the ocean carbon cycle invariant. The parameter is:

\[
\text{bg\_par\_bio\_red\_POC\_CaCO3}=0.200
\]

Adjusting this value higher will increase the global export of \( \text{CaCO}_3 \) from the surface ocean.

It is much more difficult to adjust the POC flux independent of \( \text{CaCO}_3 \) (i.e. keeping the global \( \text{CaCO}_3 \) flux invariant) because of the way muffin parameterizes biological \( \text{CaCO}_3 \) export as a function of POC export. Without perturbing the ocean \( \text{PO}_4 \) and hence productivity and carbon cycling too much, you can shift slightly more or less POC to a form that is assumed to reach the sediment surface without degradation – i.e. increasing this fraction results in a higher proportion of exported POC reaching the sediment surface, and reducing the fraction decreases the POC flux to the sediments and hence organic matter available to help drive \( \text{CaCO}_3 \) dissolution. This parameter is:

\[
\text{bg\_par\_bio\_remin\_POC\_frac2}=6.4591110\times10^{-2}
\]

Here, by default, just 0.065, or 6.5\% of organic matter exported from the surface ocean always reaches the sediment surface unaltered. (In addition, and particularly at shallowed ocean floor depths, some of the other 93.5\% can also reach the sediment surface. See Ridgwell et al. [2007].)

You might ... investigate other facets of the nature of the relationship between ocean and sediments (and weathering) as how climatic (biogeochemical) signals are encoded in the marine geological record. For instance, you could explore the effect/importance of sediment ‘bioturbation’ (e.g. see Ridgwell [2007]). Whether the surface sediment layers are bioturbation or not is set by the parameter:

\[
\text{sg\_ctrl\_sed\_bioturb}=\text{.true.}
\]

Simply change to \text{.false.} in order to ‘turn off’ bioturbation. What happens if you then run a \( \text{CO}_2 \) release experiment? How is the sediment signal different?

Rather than driving an initial dissolution of \( \text{CaCO}_3 \) in deep sea sediments, the opposite response – increased rather than decreased \( \text{CaCO}_3 \) preservation – can be obtained by removing \( \text{CO}_2 \) from the atmosphere. This can be implemented by a negative rather than positive emissions scaling in the
user-config (of in the forcing itself). BE CAREFUL here, as for a pre-industrial atmosphere with 278 ppm $CO_2$, you do not have a lot more than $\sim 600$ PgC in there (the atmosphere) to begin with. So either: remove less than 600 PgC, or remove the carbon over rather little longer than 1 year.

Again – view the time-series of ocean composition (e.g. DIC, ALK, $\delta^{13}C$) as a function of time, plus mean sediment surface composition (wt% $CaCO_3$). Also view the sediment ‘cores’ and hence what in practice has been incorporated into accumulating sediments as a record of what is a very sharp perturbation at the ocean surface (and atmosphere).

How is an event characterized by $CO_2$ removal from the system recorded differently from one characterized by $CO_2$ release? Are there different implications for constructing core age-scales and chronology, e.g. where in (core) ‘time’ does the excursion maximum appear to lie? Do all sediment locations show identical responses (i.e. does it matter what the initial wt% $CaCO_3$ is?).
12. Proxies and Reconstructing the Past

muffin model proxy capability overview
δ^{13}C
- Fossil fuel δ^{13}C
- Past ocean states and δ^{13}C
δ^{14}C (and Δ^{14}C)
I/Ca
Model-data visualization (and statistics)
READ.ME

You will need to download some new re-start files. To fetch these – change to the cgenie_output directory, and type:

$ wget http://www.seao2.info/cgenie_output/muffin.CB.GIteliivbASESFetDTL_rb.SPIN.tar.gz
$ wget http://www.seao2.info/cgenie_output/muffin.CB.GIteliivbASESFetDTL_rb.SPIN.tar.gz

Then extract the contents. Change directory back to genie-main directory to run muffin.
12.1 muffin model proxy capability overview
12.2 $\delta^{13}C$

12.2.1 Fossil fuel $\delta^{13}C$

The earlier experiments on fossil fuel $CO_2$ emissions to the atmosphere included an assumed isotopic composition of the emitted carbon (and a complete ocean-atmosphere $^{13}C$ cycle, even though we did not pay it any attention at the time. So one useful exercise is to go back to those experiments and explore how the isotopic composition of fossil fuel carbon is propagated from the atmosphere into the ocean and hence through the ocean via its large-scale circulation (and also through organic matter produced at the ocean surface and subsequently exported into the ocean interior). In this, it is helpful to also take a control experiment (you did run one, right ... ?) and create a difference map to better visualize how the $\delta^{13}C$ patterns in the ocean evolve through time.

By default, fossil fuel carbon is tagged with a mean fossil fuel isotopic signature of $-27\%o$. This is set by the parameter:

```
bg_par_atm_force_scale_val_4=-27.0
```

which scales the value of 1.0 specified in the file `biogem_force_flux_atm_pCO2_13C_sig.dat` in the `forcing` directory, which in the previous fossil fuel release experiments was the `genie-forcing` subdirectory:

```
pyyyyz.FpCO2_Fp13CO2
```

To create a more pronounced ‘tag’ of fossil fuel carbon, you could, for instance, make the assumed value of the $CO_2$ more negative (e.g. $-60\%o$ would be the signature of methane (natural gas), or you could push the value even more negative if you like (although you run out of natural analogues for what this carbon could ‘be’)).

Note that for organic carbon, in addition to its isotopic composition depending on that of dissolved inorganic carbon (technically, the $CO_2(aq)$ component of DIC), the carbon isotopic fractionation scheme also depends on the $CO_2(aq)$ concentration, which will tend to increase as fossil fuel carbon is taken up by the ocean surface. This scheme (for $^{13}C$ fractionation into marine organic matter) is described in Ridgwell [2001] as well as more briefly in Kirtland Turner and Ridgwell [2016]. So $\delta^{13}C$ or organic matter may potentially change in non-intuitive directions or values depending on the relative carbon and climate perturbation vs. the magnitude of the isotopic perturbation of the Earth system. You could explore this further by setting the isotopic composition of the emitted carbon to that of the atmosphere prior to perturbation, or even of the bulk (or surface) ocean (DIC), such that the applied perturbation becomes one only of surface ocean carbonate chemistry and climate and hence revealing more clearly the influence of changing $CO_2(aq)$ concentrations alone.

---


You can use the same re-start experiment as used in the fossil fuel CO₂ emissions chapter, or this one:

EXAMPLE.worjh2.Caoetal2009.SPIN

which is also isotope-enabled and the specific configuration that was described and evaluated against other models as well as observations, in Cao et al. [2009]³,⁴

Experiments can be run similar to before (Chapter 6)⁵, assuming that you copy and rename the user-config LAB_3.CO2emissions to e.g. LAB_3.d13CO2emissions, by:

$ ./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.ANTH LABS
   LAB_3.d13CO2emissions 20 EXAMPLE.worjh2.Caoetal2009.SPIN

You might start by viewing the δ¹³C patterns in the ocean in the re-start (and try and understand how they arise).

For specific experiments – you can employ idealised pulse or ramp etc emissions profiles and start directly from the pre-industrial re-start, or take your historical transient experiment (first look at how an δ¹³C isotopic signature associated with fossil fuel CO₂ is already propagating to the modern ocean) and then e.g. run an SRES emission scenario (remembering that you can make the isotopic signature of the carbon more 'distinct', if you wish).

12.2.2 Past ocean states and δ¹³C

δ¹³C can be used to constrain both the strength/efficiency of the biological pump of the ocean, as well as its large-scale circulation (in addition to carbon release to the system and potentially also ocean pH changes in deep time).

Two example configurations and spin-ups are given – one for the late Holocene (effectively, modern (preindustrial)) and one for the last Glacial Maximum (LGM), plus the respective observed benthic δ¹³C data-sets (refer to the Section: Model-data visualization (and statistics)). Both model configurations are derived from fully coupled GCM experiments using the HadCM3 model and hence muffin is forced by 2D stress and speed simulated by the GCM, plus a zonal mean albedo profile. The two configurations⁶ are:

1. muffin.CB.Glteeiaa.BASESFeTDL_rb.SPIN
   This experiment is additionally configured with late Holocene atmospheric CO₂ and δ¹³C, modern orbits, and a late Holocene dust flux to the ocean surface.
   A plausible circulation in the Atlantic and good statistical fit to the benthic δ¹³C is achieved with a modern-like flux correction in muffin:
   bg_par_ocn_force_scale_val_2=0.30
   here ~ 0.30Sv.
   The corresponding observed benthic δ¹³C data-set is: Hol_d13c_CLEAN.txt (columns of: latitude, longitude, depth (m), δ¹³C, and a Site label).

³Cao, L., M. Eby, A. Ridgwell, et al., The role of ocean transport in the uptake of anthropogenic CO₂, Biogeosciences 6, 375-390 (2009).
⁴To get this re-start:
   and extract the contents of this archive by typing:
   $ tar xzf EXAMPLE.worjh2.Caoetal2009.SPIN.tar.gz
⁵But noting that we are using a different base-config now: cgenie.eb_go_gs_ac_bg.worjh2.ANTH
⁶Note that they differ only minutely in their filename ...
2. muffin.CB.Giteiiva.BASESFeTDTL_rb.SPIN

This experiment is additionally configured with LGM atmospheric $CO_2$ and $\delta^{13}C$, a 21 ka orbital configuration, and a LGM Holocene dust flux to the ocean surface. Additionally, all the tracers in the ocean have been concentrated by 3% to reflect the decrease in ocean volume due to the removal of freshwater to the ice sheets. A plausible circulation in the Atlantic and good statistical fit to the LGM benthic $\delta^{13}C$ is achieved with a reduced flux correction in muffin:

```
bg_par_ocn_force_scale_val_2=0.05
```

and so – 0.05 Sv.

The corresponding observed benthic $\delta^{13}C$ data-set is: LGM_d13c_CLEAN.txt (columns of: longitude, latitude, depth (m), $\delta^{13}C$, and a Site label).

Note that the user-configs are not in the sub-directory LABS, but instead in MS/odalenetal.CP2020. You can either copy them to LABS and run as normal, or run them from MS/odalenetal.CP2020 by changing the user-config sub-directory you specify at the command line when running an experiment (better to copy). Also note that in that sub-directory is a README file (readme.txt) that describes more about the experiment configurations as well as telling you now to run them, e.g.

```
$ ./runmuffin.sh muffin.CB.Giteiiva.BASESFeTDTL_rb MS/odalenetal.CP.2019
muffin.CB.Giteiiva.BASESFeTDTL_rb.SPIN 10000
$ ./runmuffin.sh muffin.CB.Giteiiva.BASESFeTDTL_rb MS/odalenetal.CP.2019
muffin.CB.Giteiiva.BASESFeTDTL_rb.SPIN 10000
```

if you keep the user-config files in their original directory, or:

```
$ ./runmuffin.sh muffin.CB.Giteiiaa.BASESFeTDTL_rb LABS
muffin.CB.Giteiiaa.BASESFeTDTL_rb.SPIN 10000
$ ./runmuffin.sh muffin.CB.Giteiiva.BASESFeTDTL_rb LABS
muffin.CB.Giteiiva.BASESFeTDTL_rb.SPIN 10000
```

if you move them. These are the commands to run the spin-ups, so make sure that you copy and re-name the user-config files otherwise you’ll over-write the re-starts that you just downloaded ...

Experiments are a little more challenging to really get anything out of because the model steady-state take ca. 5000 or so years to come close to steady-state. Hence it is unlikely that you will see the full impact of any parameter change unless you run an experiment overnight on the queue. 50-100 years would probably the minimum to start to see meaningful adjustments to e.g. ocean circulation and hence patterns of $\delta^{13}C$.

For any experiment that you run, firstly continue on from the corresponding re-start experiment, and secondly allow atmospheric $CO_2$ and $\delta^{13}C$ to respond freely to the perturbation by removing/disabling the atmospheric forcing. TO do this, you can either switch to a different forcing – one that still has dust deposition and a fresh water transport, but no longer prescribes atmospheric composition, e.g.:

GIteiiaa.Fsal_SUR.Albani.0ka

Or, in the file:

configure_forcings_atm.dat

in the forcings directory, remove the 2 lines that specify application of atmospheric $CO_2$ and $\delta^{13}C$ forcing.

\[7\text{And so the reverse order to the late Holocene data-set ... :)}\]
Remember to run an control experiment – restarting from the spin-up provided, and with atmospheric CO$_2$ and $\delta^{13}$C allowed to evolve freely – there should be (almost) no drift in these variables.

There are fewer obvious interesting experiments for the late Holocene configuration and re-start, but you might still explore some perturbations of ocean circulation on the basis that most (almost without exception all) model deglacial/AMOC collapse experiments are run and published based on a modern or slightly modified modern configuration. SO for instance, you might set the flux adjustment that maintains the AMOC to zero:

```
bg_par_ocn_force_scale_val_2=0.0
```

and see what happens. This should look pretty much like the freshwater hosing experiments (i.e. rather than apply fresh water to the North Atlantic region in a model that already has a constant prescribed net salinity transport from North Pacific to North Atlantic, you are simply taking away the underlying prescribed transport).

Obviously, you might start simply by viewing the distribution of $\delta^{13}$C in the re-start or in your control experiment, and visualize the pattern of ocean circulation. You might also note that the experiment is configured with a number ‘age’ tracer that records the mean time since a parcel of water last ‘saw’ the ocean surface – the ventilation age. (But these particular experiments are not configured with $\Delta^{14}$C.)

For the LGM configuration, you might also try completely collapsing the AMOC (reducing the flux correction from 0.055Sv to zero (or even slightly negative). You are thereby simulating an AMOC collapse from a glacial state – aka ‘H0’. Note that as an alternative to simply removing the default flux adjustment, it is possible to create a time-dependent flux of freshwater (as traditionally applied in models, e.g. a pulse or ramp up/down) by editing:

```
biogem_force_flux_ocn_sal_sig.dat
```
in the forcing.

You might also consider what a slightly strong LGM AMOC might look like in terms of e.g. its steam-function, ventilation ages, and/or $\delta^{13}$C patterns. A moderate and reasonable increase in the flux adjustment would be to dial it up to 0.10Sv. (Dialling all the way up to 0.30Sv would enable you to access what a modern-like AMOC would look like in an otherwise glacial world.)

In whatever flux adjustment perturbation experiments you chose, there should be significant consequences for atmospheric CO$_2$ and $\delta^{13}$C that will be of interest.

Further experiments might include swapping around late Holocene and LGM orbital configurations, or perhaps applying LGM dust fields to the late Holocene configuration, or vice versa. Additionally, the prescribe solubility of iron in dust can be modified via the parameter:

```
bg_par_det_Fe_sol=0.002014275
```
12.3 \( \delta^{14}C \) (and \( \Delta^{14}C \))

**muffin** also has the capability to track radiocarbon throughout all the atmosphere and marine carbon reservoirs. \( ^{14}C \) is treated exactly as per \( ^{13}C \), with the exception that the fractionation with \( ^{12}C \) is exactly twice that of \( ^{13}C \) vs. \( ^{12}C \). As such, most of the input and output is in units of \( \delta^{14}C \), and not \( \Delta^{14}C \). Additional output for selected carbon variables is provided as \( \Delta^{14}C \) for convenience, but all input is as \( \delta^{14}C \). (Further additional output is provided as radiocarbon age.)

A re-start experiment is provided – EXAMPLE.worj2.Caoetal2009.SPIN.tar.gz which is isotope-enabled and described and evaluated against other models as well as observations, in Cao et al. [2009][8]. View the end-of-run summary output file (biogem_year_09999_500_diag_GLOBAL_AVERAGE.res) and note how the atmosphere in this pre-industrial spin-up is reported as being positive 37.4‰ in terms of \( \delta^{14}C \) (which equates to approximately 0‰ as \( \Delta^{14}C \) (e.g. as recorded in the time-series file biogem_series_misc_atm_D14C.res).

If you continue on from this spin-up using the provided experiment file EXP.12.3, you’ll find that although atmospheric \( CO_2 \) and its \( \delta^{13}C \) remain at steady state, radiocarbon progressively decays away. This is because in the spin-up, atmospheric \( \delta^{14}C \) was continually re-set to a pre-industrial value, but in EXP.12.3, their is neither continual re-setting of atmospheric \( \delta^{14}C \) nor any explicit production of \( ^{14}C \) in the atmosphere. We can correct this by adding the parameter:

\[ \text{ac_par_atm_F14C} \]

to the user-config file and giving it a value in units of \( \text{mol yr}^{-1} \). How to determine what value you need ... ? Fortunately there is a HOW-TO – ‘Determine the atmospheric radiocarbon flux required to achieve a steady state 14C system’.

If you specify the value of \( \text{ac_par_atm_F14C} \) as per described in the HOW-TO and based on the final state of the spin-up, and then run a continuing experiment, you should find that the radiocarbon state of the system remains approximately constant (e.g. as indicated by atmospheric \( \delta^{14}C \)), with the prescribed production rate balancing the global decay of radiocarbon (across all reservoirs). Hopefully ...

---

The redox chemistry of iodine in the atmosphere and seawater may involve many different iodine species. However, the iodine species most relevant to the I/Ca proxy are iodide (I⁻) and iodate (IO₃⁻), which are thermodynamically most stable in anoxic and oxic seawaters, respectively. Laboratory crystal synthesis experiments, a synchrotron study and modeling show that it is the iodate species that is incorporated in the calcite structure, whereas iodide may be excluded from the mineral lattice. Hence, I/Ca values of carbonate should record changes in iodate concentrations in the water. If the dissolved oxygen level is the dominant control on the iodate concentration, then I/Ca can, to a first-order approximation, be used as a proxy for dissolved oxygen.

Publications involving muffin simulations of ocean iodine cycling (and the I/Ca proxy) include:


Publications involving I/Ca proxy data, but with muffin simulations lacking an explicit iodine cycle (but including oxygen and sulphate), include:


To enable the iodine cycle in muffin – refer to ‘Include an iodine cycle in the ocean’ in the (Bio)geochemical cycles HOW-TO.

For instance – one might take the pair of Holocene and LGM configurations from previously (and the δ¹³C proxy worksheet) and enable an iodine cycle – adding the relevant additional tracers to the base-configs (and updating the total number of defined tracers), and adding the relevant parameters to the user-configs.⁹

The δ¹³C proxy base-configs are:

muffin.CB.GIteiiaa.BASESFeTDTL_rb.config
muffin.CB.GIteiiva.BASESFeTDTL_rb.config

for Holocene and LGM, respectively, and the associated EXAMPLE user-configs are:

muffin.CB.GIteiiaa.BASESFeTDTL_rb.SPIN
muffin.CB.GIteiiva.BASESFeTDTL_rb.SPIN

⁹The additional tracer definitions are given in the muffingen file: TRACERADDITION.iodinecycle.txt for convenience, while the user-config parameter settings needed are described in the HOT-TO.
An interesting further exercise would be to take the Cretaceous and/or late Paleocene configurations used in the publications listed above in which muffin lacked an explicit ocean iodine cycle, add an iodine cycle, re-run, and contrast with the published $I/Ca$ proxy data...
12.5 Model-data visualization (and statistics)

Observed data-sets can be contrasted directly with model fields, visualized, and the statistics of fit calculated, via the muffinplot MATLAB functions.

Panoply will allow you to create difference maps, but not provide the corresponding statistics. Furthermore, Panoply will require the data to be in the form of a netCDF file in order to plot a difference map with a model field. muffindata contains MATLAB functions to create netCDF format files out of raw (location-by-location) or gridded text-format data.

For the late Holocene and LGM $\delta^{13}C$ experiments, observed benthic $\delta^{13}C$ data is provided in the same directory as the HOL and LGM experiment user-config files (genie-userconfigs/MS/odalenetal.CP.2020). Taking these files ... ensuring that the format is (lon,lat,depth,value) by removing the data column for the Site label, and then (e.g. in MATLAB – data=rmmissing(data);) removing the NaN values and corresponding lines ... the data can be re-gridded to the GENIE grid and saved in netCDF format, enabling direct model-data comparison (and anomalies) in Panoply. These (netCDF) files are provided already processed under the got data? box on the left hand side of the mymuffin webpage (http://www.seao2.info/mymuffin.html). Note that they are for the full 3D model grid, and e.g. surface and shallow depth layers will not contain any benthic data.

Having processed the raw data as described above (no NaNs or missing values, and in a strict (lon,lat,depth,value) column format), the muffindata function make_regrid_data2netcdf_GENIE.m can be used in MATLAB, e.g.:

```
>> make_regrid_data2netcdf_GENIE('d13C_HOL.txt','d13C','Holocene benthic d13C','o/oo',36,36,16,-180,'')
>> make_regrid_data2netcdf_GENIE('d13C_LGM.txt','d13C','LGM benthic d13C','o/oo',36,36,16,-180,'');
```
13. Synthesis: Earth system dynamics

Stuff to keep in mind:

"There are known knowns.  
These are things we know that we know.  
There are known unknowns.  
That is to say, there are things that we know we don’t know.  
But there are also unknown unknowns.  
There are things we don’t know we don’t know."

Donald Rumsfeld, former US Secretary of Defense
13.1 README

blah
13.2 How low can you go?

Your task is to explain low glacial atmospheric CO2.

It is up to you whether you aim for 190 ppm, or want to retain as much consistency with other (paleoceanographic) constraints as possible.

In addition to the control and spin-up user-configs, you have been provided with: exp12_glacial which can be used as a template user-config file for you to base (if you want!) your glacial CO2 investigations on (at least initially).

Before embarking upon some glacial experiments (below) – other model outputs/fields you might also keep in mind and for which some data/observational constraints on your glacial CO2 'solution' may be available (e.g. see Kohfeld and Ridgwell [2009]), include:

- The 2D fields\_biogem\_2d.nc field: phys\_seaice (‘sea-ice cover’), for which some glacial sea-ice limit information exists. (CLIMAP)
- The 2D fields\_biogem\_2d.nc field: ocn\_sur\_temp (‘surface-water temp’) (or view the surface ocean layer in the 3D file), for which 2 (one older, one newer) comprehensive datasets exists – it would be reasonable to question whether you achieve an adequate glacial (surface) climate state and if not, whether this impacts any bias (and in which direction) to your CO2 solution. (CLIMAP, MARGO)
- The 2D fields\_biogem\_2d.nc field: sed\_CaCO3 (‘sediment core-top CaCO3’) (also available from the sedgem model output), for which some glacial CaCO3 distribution data/estimates exist. (Old Catubig paper in GBC)
- The 2D fields\_biogem\_2d.nc field: ocn\_D\_DIC\_13C (‘ planktic-benthic difference DIC\_13C’) and also the individual planktic (surface) and benthic (bottom) fields. A significant amount of d13C data exists in the literature for both glacial and interglacial states.
- The 2D fields\_biogem\_2d.nc field: ocn\_ben\_O2 (‘ bottom-water O2’) (and also horizontal slices in the 3D file). Ideally, no-where in the ocean should anoxia (no oxygen) occur. It certainly should not be widespread across one or more ocean basins if your glacial CO2 solution is to get published in Nature ;)
- The 2D fields\_biogem\_2d.nc field: ocn\_ben\_sal (‘bottom-water sal’) (and also in the 3D file as horizontal slices), for which some estimates exists for a few places in the ocean. (Jess Atkins Science paper)
- The 2D fields for surface and deep PO4 (and also in the 3D file as horizontal slices) as some proxy evidence exists for changes in nutrient utilization. (Cd/Ca proxies – e.g. papers by Elderfield and Rickaby)

Also refer to the 3D netCDF files and the time-series where helpful.

Note here that the spin-up provided is ‘modern’ and hence glacial data cannot be directly contrasted – the above suggestions/guidance are intended as a starting point only.

In your glacial CO2 investigations, 2 separate initial modifications of the model will nudge it in the vague direction of a glacial state (e.g., see Kohfeld and Ridgwell [2009]):

- A modification of surface (actually ‘planetary’) albedo to try and take account of some of
the cooling influences of the large (Northern Hemisphere) ice sheets that were present during the last glacial but which are not calculated or explicitly taken into account in the version of cGENIE you are using.

- A modification of greenhouse gas radiative forcing (as per in the snowball Earth experiments) to take into account the lower CO2, CH4, and N2O concentrations in the atmosphere during the last glacial. While you will be attempting to reproduce ca. 190 ppm atmospheric CO2 (and hence deduce the reasons for low glacial CO2), you may not necessarily achieve this, and you have no means of explicitly controlling the other A Hitchhiker’s Guide to the advanced Black Arts (of Earth system modelling) LAB Session VII 3 greenhouse gases, so you may as well get the radiative forcing and hence the glacial climate state as close as possible before trying to adjust the carbon cycle. But it is up to you whether you prefer to not ‘cheat’ and have whatever CO2 cGENIE simulates, directly affect climate (and hence feed back on CO2).

The glacial boundary conditions are implemented as follows:

- To achieve a pseudo-glacial planetary albedo modification, add the following lines to a user-config file (and see Lab V):

```plaintext
# adjusted planetary albedo
ea_albedop_offs=0.200
ea_albedop_amp=0.360
ea_albedop_skew=0.0
ea_albedop_skewp=4
ea_albedop_mod2=-15.000
ea_albedop_mod4=-2.500
ea_albedop_mod6=0.000
```

- For glacial radiative forcing, add the lines:

```plaintext
# glacial CO2 radiative forcing
ea_radfor_scl_co2=0.6835
# glacial CH4 radiative forcing
ea_radfor_scl_ch4=0.5
# glacial N2O radiative forcing
ea_radfor_scl_n2o=0.8
```

If you like – you can carry out separate experiments to test the effect of each of these in turn and hence to learn the effect and impact on atmospheric CO2 of each individually before combining them. Ideally, these lines should ultimately be included in all (glacial) experiments that you require glacial albedo and radiative forcing for.

By all means play around with the albedo and radiative forcing climate boundary conditions, although one might wonder the reasoning behind adjusting radiative forcing below e.g. that appropriate for full glacial conditions. However, the planetary albedo is less certain as implemented in the model. For instance, one might legitimately adjust this to achieve appropriate last glacial sea surface temperatures (SST), or rather: a glacial-interglacial difference in SSTs similar between model and data.

By this point, you should have created a new (mostly) spun-up model state, incorporating: (i) higher planetary albedo, and (ii) lower greenhouse gas forcing. The duration of this new, glacial spin-up needs to be sufficient to bring the system into (a new) equilibrium (of which the slowest
13.2 How low can you go?

adjusting component will be sediment composition (wtwt% CaCO3 will to some extent be reflected in continuing changes in pCO2 (why?!)).

Before carrying on, check that everything is ‘correct’ (or at least: understandable) so-far. In particular: confirm that you have a colder ocean (due to altered albedo and/or greenhouse radiation forcing) ... no, seriously! You never know what might have gone wrong with a simple slip of the keyboard ... Surface ocean temperature also has established proxies for its glacial value and so the model can be contrasted against data.

The file biogem_series_ocn_temp.res is the time-series results file for ocean temperature – the 2nd column is the mean ocean temperature, the 3rd column is mean sea surface temperature (SST), and the 4th mean benthic (deep (> 2 km) ocean floor). How much colder has it become? Is this realistic? Analyze the SST distribution (the surface field of the 3D netCDF time-slice file, or the 'sur_*' variables in the 2D netCDF file) – how does this compare to observations? In the book chapter the data-based difference in SST between the LGM and Holocene is given. However, the map given is for the glacial-interglacial difference, which happily is something you have previously learned to do using Panoply (I hope!).

Also, what is the CO2 impact of lower SSTs? Note that you may not be able to directly compare your CO2 prediction with all previous studies (e.g. summarized in the book chapter) because in your model, sea-ice and ocean circulation will also have been affected to some extent by the climate change. (How much have they been affected? There are also some proxies for sea-ice extent as well as ideas and hypotheses about ocean circulation changes.) Many (but not all) previous model studies have simply estimated CO2 changes due to temperature change with fixed sea-ice and circulation fixed, by prescribing a different ocean surface temperature for the CO2 solubility calculation. (This is a little beyond the scope of what you are expected to do here, but can be done in GENIE.)

Unless you are *extremely* lucky and already have a value of atmospheric CO2 that is 90 ppm lower than pre-industrial (ca. 278 ppm) ... (WTF?!) – you may want to test other changes that might have taken place between glacial and interglacial that affected CO2. Obviously a spot of creating new user-config files will be in order here (perhaps using: exp12_glacial as a template, but it is entirely up to you). Ideally, you would test the impact of each change individually first before combining them, so as to develop a better understanding of the different ways in which CO2 is controlled (and the associated impacts on other elements of the global carbon cycle and climate) before bunging everything in together.

Before diving straight in – note the number of different modifications of the global carbon cycle that might be considered and tested in the model. What is your methodological strategy going to be? Are you going to add all of the modifications into a single run and hope that you can understand what has happened at the end? (I hope not!) Are you going to run all the modifications individually (how?)? Are you going to try combinations to test whether any combine non-linearly? You will want to make use of the cluster queue and submit at least some of the experiments. You will also need to already have a good idea of how long to run them before (hopefully you obtained this knowledge from the idealized perturbation experiments in the previous tutorials).

Some suggestions (i.e., this not an exhaustive list, nor a prescribed one and not everything necessarily has to be done!):
13.2.1 Global weathering rate.
Refer to Ridgwell and Zeebe [2005] for the role of weathering. Also to Kohfeld and Ridgwell [2009] for some references to the changes in weathering that might have taken place between glacial and interglacial. The namelist parameter that controls the annual rate of solute input into the ocean is:

\[
\text{rg\_par\_weather\_CaCO3}=0.9E+13
\]

Either edit this value (under heading: # — WEATHERING —) or add a new line at the end of the user config file specifying the value you want. Units are mol of CaCO3 weathered per year.

This parameter could also be adjusted to implicitly simulate the effect of a change in carbonate deposition in coral reefs and other shallow water carbonates, changes that GENIE cannot simulate explicitly. See Ridgwell et al. [2003] (http://www.seao2.org/pubs/ridgwell_et_al_2003a.pdf) for references and discussion of the sort of change in carbonate deposition on the shelves that might have taken place. A decrease in CaCO3 removal on the continental shelves can be simulated by increasing the weathering flux to the open ocean. In other words, you can look at the parameter rg\_par\_weather\_CaCO3 as representing the residual weathering flux to the open ocean, after some of the weathering flux has been removed in coastal areas. Even if global weathering of the continents did not change, any reduction in CaCO3 precipitation and removal on the continental shelves would result in an increased solute flux to the open ocean.

[HINT: keeping track of how mean sediment wt% CaCO3 changes will be helpful, as may 2D sediment distributions and ultimately, the sediment core records.]

13.2.2 Iron fertilization.
Read up on this first, e.g., see references in Kohfeld and Ridgwell [2009]. The glacial was dustier than present, hence there can only have been increased aeolian iron supply to the ocean surface. However, what is not so clear is how important (relative to Fe being upwelled) aeolian Fe is today, let alone during the last glacial ...

Anyway: one way to increase the aeolian Fe supply to the ocean surface is simply to increase the solubility of the Fe in dust. This is controlled by the parameter:

\[
\text{bg\_par\_det\_Fe\_sol}=0.0015
\]

with the default being a global average dust Fe solubility of 0.15Increasing will increase the Fe input to the ocean surface everywhere (in direct proportion to the modern spatial pattern). The pattern of total aeolian Fe supply is recorded in the (2DBIOGEM) variable:

\[
\text{misc\_sur\_fFetot\_mol}
\]

with the dissolved component under:

\[
\text{misc\_sur\_fFe\_mol} \quad \text{(misc\_sur\_Fe\_sol is the map of solubility, which in GENIE is not uniform in space – any idea what the reason for this assumption might be?). (A glacially-explicit map of dust deposition could also be applied in place of the modern deposition map – if you would like to test this, I can create one, but note that there are very significant ‘errors’ in re-gridding dust maps to this highly simplified continental topography.)}
\]

[HINT: viewing maps of particulate organic carbon (POC) export may be particularly helpful.]

13.2.3 Remineralization depth.
There is no temperature control on the rate of bacterial degradation of sinking organic matter (see: book chapter + references therein) but the effect of lower ocean temperatures and a slower rate of bacterial degradation of organic matter can be simulated by specifying that particulate organic
13.2 How low can you go?

matter reaches greater depth before being remineralized (and CO2 and PO4 released back to the seawater). The namelist parameter that controls the e-folding depth reached by particulate organic matter before remineralization is:

```
bg_par_bio_remin_POC_eL1=589.9451
```

Either edit this value (under heading: # — REMINERALIZATION —) or add a new line at the end of the user config file specifying the value you want. Units are m. Read Ridgwell et al. [2007] for additional discussion of this parameter. See Figure 2-4 in Ridgwell [2001] (http://www.seao2.org/pubs/ridgwell_thesis.pdf) for an illustration of how the flux of particulate organic matter decreases with depth in the ocean, plus references therein.

There is also an associated parameter: bg_par_bio_remin_POC_frac2, which sets a fraction of organic matter that is assumed to settling through the water column completely un-altered (currently assigned a value of 0.025 == 2.5%), but this is arguably less appropriate to change than the remineralization length-scale of the more labile fraction (97.5% of exported particulate organic carbon).

[HINT: viewing distributions of PO4 and/or O2 in the ocean may be helpful. Perhaps also d13C.]

13.2.4 Macro nutrient inventory and uptake.

Suggestions have been made that nutrients were used more efficiently during the LGM, meaning that for the same nutrient uptake at the surface more carbon was exported to depth in the ocean. See: Omta et al. [2006]. There are also a bunch of (relatively old) hypotheses concerning differences between glacial and modern ocean in how much nitrate (NO3 -) there was. There is no NO3 - in this version of GENIE (just PO4 3- and Fe), but an analogous change can be made to the phosphorous cycle.

For the nutrient-to-carbon ratio in organic matter, the relevant parameter is:

```
bg_par_bio_red_POP_POC=106.0
```

To change the default value (106.0), add a new line at the end of the user-config file specifying the value you want. A larger number means that PO4 is being utilized more efficiently and more organic matter is being produced for the same nutrient consumption.

If you would like to test the effect of adding more PO4 to the (glacial) ocean – a forcing is provided, called:

```
p0000b_FeMahowald2006_ADJUST_phosphate
```

Note that adjusting the ocean PO4 inventory should only be done one (and not accidentally in each successive experiment!).

[HINT: viewing distributions of PO4 and/or O2 in the ocean may be helpful. Also ocean sediment CaCO3 distributions.]

13.2.5 CaCO3:POC rain ratio.

Kicked off by a classic 1994 Nature paper by Archer and Maier-Reimer (see: Kohfeld and Ridgwell [2009]), one powerful means of changing atmospheric CO2 that has been proposed involves changes in the export ratio between CaCO3 (shells) and POC (particulate organic matter). Such a change in ratio could come about through a variety of ways (e.g., via the ‘silica leakage hypothesis’ (see: Kohfeld and Ridgwell [2009]) and also through the direct effect of Fe on diatom physiology (see Watson et al. [2000] in Nature and also Supplemental Information). There are also ideas about an opposite ocean acidification effect, whereby the less acidic glacial (compared to modern) ocean
led to increased calcification and CaCO3 export. Note that this response (higher saturation \(\rightarrow\) great calcification) is encoded into your model configuration – see Ridgwell et al. [2007b].

In GENIE, the CaCO3:POC rain ratio is controlled (technically: scaled) by the parameter:

\[ \text{bg\_par\_bio\_red\_POC\_CaCO3} = 0.03 \]

The pattern of CaCO3:POC rain ratio is not uniform across the ocean (why? (see: Ridgwell et al. [2007, 2009]), and its pattern can be viewed in the (2D BIOGEM) netCDF variable:

\[ \text{misc\_sur\_rCaCO3toPOC} \]

[HINT: viewing sediment CaCO3 distribution may be helpful.]

13.2.6 Sea-ice extent.

Changes to sea-ice extent have already taken place due to changes in radiative forcing and planetary albedo (made previously). There is no much you can do to further adjust sea-ice extent, other than via further changes to climate (via radiative forcing and/or albedo).

13.2.7 Atlantic circulation.

There are a variety of ideas and hypotheses about glacial ocean circulation and what influence it had on atmospheric CO2. At least with respect to making tests and experiments in models, a common ploy has been to produce a collapsed AMOC (e.g., see Chikamoto et al. [2008] (JGR 113)). Rather than apply a continuous freshwater forcing to the ocean throughout an extended (sediment interaction) time-scale (why would this not be a good idea?), there is a parameter in the model which creates an adjustment of the salt balance between the different ocean basins (to make the Atlantic more salty compared to the Pacific). (In other words: salt/freshwater is re-partitioned between the ocean basins rather than ’new’ freshwater or salt externally added.) This parameter is:

\[ \text{ea\_28} = 0.726862013339996340 \]

Setting it to e.g., 0.0, will result in a collapsed AMOC. But maybe that is too extreme? (You might read up a little on the glacial ocean circulation literature and chose a value that gives as an appropriate change to Atlantic circulation as you can judge from the data and literature.)

[HINT: viewing distributions of PO4 and/or O2 and/or d13C in the ocean may be helpful. Also ocean sediment CaCO3 distributions.]

13.2.8 Global ocean circulation / 'brine rejection'.

Some recent research has focussed on the possible role of ’brine rejection’ in creating a saltier Antarctic bottom waters (e.g. see Adkins et al. 2002 Science paper) and hence a denser and more stratified deep ocean,, with the idea being this will trap carbon more efficiently. For a very recent study (and references therein), see:


GENIE has the capability to include this effect (at least crudely) and similarly to Bouttes et al. [2010]. For this, three namelist parameter values need to be set:

\[ \text{bg\_ctrl\_force\_GOLDSTEInTS} = \text{.TRUE.} \]
\[ \text{bg\_par\_misc\_brinerejection\_frac} = 0.1 \]
\[ \text{bg\_par\_misc\_brinerejection\_jmax} = 9 \]

The first, simply allows the BIOGEM biogeochem module to directly influence ocean circulation. The second is the fraction of salt, rejected during sea-ice formation (e.g., see Bouttes et al. [2010]) that is transferred directly to the bottom-most (underlying) ocean cell in the model. The
13.2 How low can you go?

first sets a latitude limit (counted in cells) to the effect – a value of 9 will restrict brine rejection to
the Southern Ocean; a value of 18 will allow it to take place in the North Atlantic as well. (Note
that in e.g., Bouttes et al. [2010], the effect is considered only in the Southern Ocean.)

[HINT: viewing distributions of PO4 and/or O2 and/or d13C in the ocean may be helpful. Also
ocean sediment CaCO3 distributions.]

MISC.

There are of course other possibilities for adjusting the model, although you need an a priori reason
for doing so and what about the possible glacial state of global carbon cycling and climate you are
trying to encapsulate. Examples might include wind speed (or air-sea gas exchange).

Even if you achieve atmospheric CO2 of ca. 190 ppm (and actually, with some mechanisms
on their own and also in combination, it is quite easy to achieve this), how do you know if you are
‘right’? Many of the important constraints are summarized in Kohfeld and Ridgwell [2009] and
Archer et al. [2000]. In particular:

• The distribution of the CaCO3 content of deep-sea sediments. e.g., see Figure 6 in Archer et
  al. [2000]. You are not ‘allowed’ to blanket the entire ocean floor with CaCO3 if you want to
  be consistent with the paleoceanographic record of the LGM ;)

  The predicted distribution of the CaCO3 can be used to assess your circulation change – note
  that there is much less CaCO3 in sediments in the North Atlantic at the glacial [Archer et al.,
  2000]. See: Chikamoto et al. [2008] for a model assessment of the impact of AMOC changes
  on deep-sea sediment composition.

• The ocean should not go ‘anoxic’ (i.e., little to no dissolved oxygen left) over large expanses.
  (But you might consider this relative to the modern configuration – i.e., should the mod-
  ern simulation under-estimate oxygen concentrations in the deep ocean, so will the glacial
  simulation, even if you get the mechanisms exactly ’right’.)

• There is a map of estimated changes in the biological flux to the ocean floor in Kohfeld
  and Ridgwell [2009] (also read the original reference). In the 2D netCDF file, the variable
  focused_POC gives you the flux of particulate organic matter (actually, carbon) to the ocean
  floor. By constructing a difference map of your glacial-interglacial predicted changes, you
  could contrast directly to the Kohfeld et al. [2005] reconstruction.

• The GENIE model is set up to predict d13C distributions. See: Curry and Oppo [2005].
  There is also an atmospheric record of d13C (also predicted by GENIE) – see: Smith et al.
  [1999] and a more recent paper in GBC: Lourantou et al. [2010] (‘Constraint of the CO2 rise
  by new atmospheric carbon isotopic measurements during the last deglaciation ’).

• Other proxies offer varying constraints at the global or regional scales. e.g., see: Elderfield
  and Rickaby [2000] (Cd/Ca ratios).

Note that commonly in (glacial CO2) modeling studies, a steady state (or quasi steady state)
simulation is run for the glacial (and compared to pre-industrial). The version of cGENIE you have
is sufficiently fast to do this quite effectively. It is possible to do non-state (glacial-interglacial)
simulations, e.g. Ridgwell [2001], but this is rather more involved.

Also note that in all of the above possible adjustments to the global carbon cycle, the mecha-
nism of carbonate compensation is operating. Hence there will be direct (changes in carbon cycling
within the water column) and indirect (interaction between ocean and deep-sea sediments) pro-
cesses operating that will affect CO2. Carbonate compensation will typically take a few 10s of
thousands of years to fully adjust atmospheric CO2. Not all previous modeling studies include this effect and in some cases it can drastically influence the predicted change in atmospheric CO2.

Finally . . . maybe you have achieved close to 190 ppm and are not unreasonably consistent with various paleoceanographic proxies and are hence feeling rather pleased with yourself . . . O – sorry – I forget to mention a little something:

- There is as approximately 3% (1 PSU) increase in salinity (and other dissolved tracers) due to the presence of large (Northern Hemisphere) ice sheets, and hence loss of freshwater from the ocean and lower sea-level associated with the last glacial.

Ahhhh . . . and I also forgot:

- It is thought that there was a release of carbon stored on land in vegetation and soils during glacial climates. (cGENIE has some capabilities to model changes in terrestrial carbon storage and hence predict this, but you are not using a version with this science module enabled.)

Unfortunately – both these effects act to increase atmospheric pCO2 at the last glacial (or decrease it across the deglacial transition, which ever way around you like to think of it). So you are actually rather further off achieving a net 90 ppm difference than you thought.

### 13.2.9 Salinity

To test the effect of a 3% increase in salinity at the last glacial, you need to create a new userconfig file (copied from your last glacial ‘best guess’ configuration), add the lines:

```
bg_ctrl_force_GOLDSTEInTS=.true.
bg_par_forcing_name='p0000b_FeMahowald2006_ADJUST_salinity'
```

and run using your last glacial best guess as the restart.

If you want to check that you have applied this forcing correctly: Mean ocean salinity should end higher than the preindustrial restart that was originally provided (or compared to your best guess glacial run). The file biogem_series_ocn_sal.res is the time-series results file for ocean salinity – the 2nd column is the mean ocean salinity. Originally it was 34.904 PSU (or o/oo), now it should be about 35.9.

Note how atmospheric pCO2 has responded to the change in ocean volume and sea-level (and tracer concentrations) alone. How does this reduced resolution version of cGENIE compare to published estimates (too much; too little; why? or if

### 13.2.10 Terrestrial carbon storage

For the reduction in terrestrial carbon storage, you need to try a further experiment (on top of and/or following on from the salinity change experiment), with the user-config line:

```
bg_par_forcing_name='p0000b_FeMahowald2006_ADJUST_terrestrialC'
```

These two forcings are effectively one-off changes imposed on the global climate and carbon cycle. By selecting the salinity forcing, you add 1 PSU of salinity to the entire ocean (and concentrate proportionally all dissolved tracers in the ocean) in a single year. Obviously you only want to do this once, not multiple times (other wise you will get an increasingly salty ocean ...). Similarly with the terrestrial carbon change – as specified, this forcing results in 500 PgC of carbon being added to the atmosphere over a period of 500 years (i.e., at a rate of 1 PgC yr-1) as if to simulate a commensurate reduction in carbon stored on land. One strategy might be to implement this as a
13.2 How low can you go?

second phase of (additional) spin-up (after the salinity modification). Note that while the magnitude of the glacial-interglacial salinity and sea-level change is well constrained, that of the terrestrial biosphere is not (e.g., see: Kohfeld and Ridgwell [2009]). In investigating the potential causes of low glacial CO2, do not feel constrained to necessarily run with the default (500 PgC) forcing provided ... (Think for yourselves!)

To confirm that you have correctly added (rather than subtracted!) carbon to the ocean+ atmosphere – thee ocean + atmosphere carbon inventories should start changing from the start of the experiment incorporating the carbon change forcing

(p0000b_FeMahowald2006_ADJUST_terrestrialC)

and the change should be approximately uniform. You can calculate the change in ocean + atmosphere carbon inventory from the atmospheric CO2 time-series file (biogem_series_atm_pCO2.res – column #2 is the global CO2 inventory in mol) and the ocean total dissolved carbon timeseries file (biogem_series_ocn_DIC.res – column #2 is the global DIC (total dissolved inorganic carbon) inventory in mol). Note you will have to convert from mol to gC (or PgC) in order to compare to the amount you requested. If the rate of inventory change turns out to be not quite linear, and particularly if the inventory change should turn to be not quite what you were expecting ... why? (Hint: refer to the mechanisms discussed in the lecture (and papers) relating deep-sea sediments and weathering to changes in total carbon (e.g., fossil fuel CO2 release.)
14. muffin model output

This section covers **muffin** saves data and how to ensure that the variables you want are saved and when you want.

- Overview.
- **Time-series output.**
  - *Time-series* file naming conventions.
  - Specifying frequency and timing of *time-series* data saving.
  - Seasonal/monthly data saving.
- **Time-slice output.**
  - *Time-slice* file naming conventions.
  - Specifying frequency and timing of *time-slice* data saving.
- Specifying which data fields to be saved in the *time-series* and *time-slice* format.
- **Re-start** files.
Chapter 14. muffin model output

14.1 Overview (and types of model output)

The results of experiments are written to the directory: ∼/cgenie_output

For any particular experiment, all saved model results, plus copies of input parameters and the model executable, are gathered together in a sub-directory of ∼/cgenie_output that is assigned the same name as the experiment (== user-config file name), e.g.: EXAMPLE.worbe2.Ridgwelletal2007.SPIN.

Every science module saves its results in its own individual sub-directory within the experiment directory. So for the module that calculates ocean biogeochemical cycles – BIOGEM, the results files will thus be found in: ∼/cgenie_output/EXAMPLE.worbe2.Ridgwelletal2007.SPIN/biogem

Note that the science module ATCHEM does not save its own results (BIOGEM instead saves relevant information about atmospheric composition and air-sea gas exchange) while SEDGEM essentially saves results only at the very end of a model experiment¹. (BIOGEM can also save the spatial distribution of sediment composition as time-slices as well as mean composition as a time-series). Furthermore, in order to attain a common format for both ocean physical properties and biogeochemistry, BIOGEM saves a range of ocean physics properties in addition to temperature and salinity, such as: velocities, sea-ice extent, mixed layer depth, convective frequency, etc.

Saving full spatial distributions for any or all of the tracers at each and every time-step is not practical, not only in terms of data storage but also because of the detrimental effect that repeated disk access has on model performance. Instead, BIOGEM saves the full spatial distribution of whatever tracer, flux, and/or physical properties of the system are required (how what fields are required is specified is discussed later), only at one or more predefined time points (in years). These are called ‘time-slices’. However, rather than taking an instantaneous snapshot, the time-slice is constructed as an average over a specified integration interval.

The second main data format for model output is that of a ‘time-series’ of change in a single (integrated) property of the Earth system. Model characteristics must be reducible to a single meaningful variable for this to be practical (i.e., saving the time-varying nature of 3-D ocean tracer distributions is not). Suitable metrics include: the total inventories in the ocean and/or atmosphere of various tracers (or equivalently, the mean global concentrations / partial pressures, respectively), global sea-ice coverage. Like time-slices, the data values saved in the time-series files represent averages over a specified integration interval (one year by default).

For both time-slices and time-series output, the files themselves are created during model initialization and are periodically updated (appended to) during the experiment. Hence, even before the experiment has finished they may contain data that is useful to view and can be used to check on the progress of an experiment.

ATCHEM

In the ATCHEM results directory, only the following file will be present:

1. *_restart.nc – Re-start file – a snap-shot of the 2D distribution of atmospheric composition at the very end of the experiment. Not intended for user-access, although it can be plotted just like any normal netCDF format file.

BIOGEM

For BIOGEM, some or all of the following files will be present:

¹With the exception of sediment core location environmental properties, which are saved more frequently.
14.1 Overview (and types of model output)

1. 

 restart.nc – Re-start file – a snap-shot of the 3D distribution of biogeochemical properties of the ocean at the very end of the experiment. Not intended for user-access, although it can be plotted just like any normal netCDF format file.

2. 

 fields.biogem_2d.nc – 2-D fields of (mostly) ocean bottom, ocean surface, and sediment surface properties. Also: water-column integrals of certain geochemistry diagnostics, air-sea gas exchange fluxes, atmospheric composition (plus some physical atmospheric properties).

3. 

 fields.biogem_3d.nc – 3-D fields of ocean dissolved and particulate tracer properties (plus some physical ocean properties).

4. 

 biogem_series_* .res – Time-series results files – globally and surface-averaged (and sometimes also benthic (bottom) surface averaged) property values as a function of time in plain text (ASCII) format.

5. 

 biogem_year_*_diag_GLOBAL .res – Miscellaneous global diagnostic information. These files are saved at each requested time-slice with the file-name string containing the mid-point of the time-slice (as years). The diagnostics include:
   - time mid-point and integration interval
   - global ocean surface area and volume
   - mean global sir-sea gas exchange coefficient (for CO2)
   - mean atmospheric tracer concentrations plus total inventory
   - mean ocean tracer concentrations plus total inventory
   - mean plus total global productivity
   - mean plus total global sedimentation

SEDGEM

In the SEDGEM results directory, some or all of the following files will be written:

1. 

 restart.nc – Re-start file – a snap-shot of the 2D distribution of sedimentary properties at the very end of the experiment. Not intended for user-access, although it can be plotted just like any normal netCDF format file.

2. 

 fields.sedgem_2d.nc – Contains 2-D fields of sediment surface and ocean bottom properties.

3. 

 sedcore.nc – netCDF format file containing the stacked records of accumulated deep-sea sediment composition.

   The locations (if any) of sediment cores to be saved is specified in a plain text (ASCII) file pointed to by the string value of the namelist parameter sg_par_sedcore_save_mask_name. In the mask file, a ‘1’ indicates a location to save a sediment core at, and a ‘0’ indicates that no sediment core should be saved at this location. This file must be present, so to save no sediment cores, simply populate the file with all zeros in an xx by yy grid.

4. 

 sedcoreenv_* .res – These files contain pseudo time-series of surface sediment environmental properties at each of the requested sediment core locations (if any are chosen).

---

2The mid-points at which time-slices are saved are specified as described above.

3The mid-points at which time-slices are saved are specified as described above.

4 .res is a useful format for processing in MATLAB; for other programs, other extensions are needed. If using the Mathematica data processing scripts - see genie-docs/cGENIE.AutomationScripts -.dat is needed; this can be set with gm_string_results_ext=".dat"

5This data is saved only at the termination of an experiment (i.e., the netCDF file contains only a single time-slice).

6The location of this file is specified by the SEDGEM data input directory namelist parameter: sg_par_indir_name which by default is "/genie-sedgem/data/input."
5. `seddiag_misc_DATA_GLOBAL.res` – A summary of mean global sedimentation, dissolution, and preservation fluxes, and surface sediment composition.

6. `seddiag_misc_DATA_FULL.res` – Surface sediment and bottom water properties at each and every sediment grid point.

**ROKGEM**

In the **ROKGEM** results directory, some or all of the following files will be written:

1. `fields_rokgem_2d.nc` – 2-D fields of (mostly) land surface, ocean surface, and atmospheric properties related to weathering.

2. `biogem_series_*` – Time-series results files.
Time-slice output

14.2 Time-slice output

14.2.1 Frequency and timing of time-slice data saving

Rather than taking an instantaneous snapshot, the time-slice is averaged over a specified integration interval $\Delta t$ (in years), defined by the parameter: bg_par_data_save_slice_dt\(^7\). The model state is thus integrated from time $t_n - \frac{\Delta t}{2}$ to $t_n + \frac{\Delta t}{2}$. For instance, setting a value of $\Delta t = 1.0$ year results in all seasonal variability being removed from the saved time-slices, and successive time-slices then only reflect long-term (>1 year) trends in system state.

The mid-point years ($t_n$) for which time-slices should be saved are specified in a single column pain text (ASCII) file in the cgenie.muffin/genie-biogem/data/input directory, whose name is specified by the parameter bg_par_infile_slice_name\(^8\). For example, the default time-slice specification file save_timeslice.dat contains the specification\(^9\):

```
-START-OF-DATA-
  0.5
  1.5
  4.5
  9.5
  19.5
  49.5
  99.5
  199.5
  499.5
  999.5
  1999.5
  4999.5
  9999.5
  19999.5
  49999.5
  99999.5
  199999.5
-END-OF-DATA-
```

where -START-OF-DATA- and -END-OF-DATA- are simply tags delineating the start and end of the time point data. Use of this particular specification lends itself to simple experiment run durations to be adopted (e.g., 10, 100, 10000 years). It provides a good generic starting point in that save frequency is faster to begin with (when environmental variables are more likely to be rapidly changing) and less frequently later (when environmental variables are unlikely to be changing rapidly and maybe converging to steady-state).

To change the time points used for time-slice data saving, either direct edit this file (less good), or create a new file (e.g. simply copy and rename save_timeslice.dat) with the required save fre-

---

\(^7\)An empty list is valid - time-slices will then be populated for you at an interval set by the time-slice integration interval. But if you really don’t want any time-slices, just set the first (or only) time point to occur beyond the end year of the run.

\(^8\)The location of this file is specified by the BIOGEM data input directory parameter: bg_par_indir_name which by default is `/genie-biogem/data/input`.

\(^9\)The order in which the time sequence is ordered (i.e., ascending or descending time values) does not actually matter in practice as long as the list of times is ordered sequentially. The list will be internally re-ordered if necessary according to the selection of ‘BP’ (the model running backwards-in-time) or not according to the logical value of the parameter bg_ctrl_misc_t_BP, which is .false. by default.
Chapter 14. muffin model output

Chapter 14. muffin model output

Quency and timing and saved to the cgenie.muffin/genie-biogem/data/input directory, with the parameter bg_par_infile_slice_name pointing to the new filename).

14.2.2 Saving at the experiment end

Just in case an experiment run duration is chosen such that there is no corresponding save point anywhere near the end of the run, a time-slice is automatically saved at the very end of an experiment regardless of whether one has been specified or not and with the same averaging as used for the specified time-slices.

This lends itself to a means of substantially reducing the amount of data saved, because if you specify no time-slices, you will still end up with (just) the final year saved. You can make this happen (i.e. forcing only an end-of-run time-slice to be saved), by adding to the user-config:

```
# force time-slice save at run end only
bg_par_infile_slice_name='save_timeslice_NONE.dat'
```

14.2.3 Seasonal/monthly data saving

Time-slice (but not currently time-series) data can be saved seasonal or even monthly by selecting by setting a single parameter rather than e.g. specifying a monthly or seasonal data save interval and editing the time-slice definition file with a series of min-points for months (or seasons). The way it works is that the overall averaging interval (parameter: bg_par_data_save_slice_dt)\(^\text{10}\) is broken down into sub-intervals of averaging. i.e., breaking down a year interval (the default) into 4 will give seasonal averaging. The parameter: bg_par_data_save_slice_n where \(n\) sets the number of time steps in each sub-interval of data saving and hence determines whether the averaging is e.g. seasonal or monthly. The slightly tricky part is to be sure of how many time steps in each year ;)

By default, cGENIE.muffin employs 96 time-steps per year for a 16-level ocean circulation model (GOLDSTEIN) and 48 for BIOGEM\(^\text{11}\). Hence for a 16-level ocean configuration, seasonal data saving would be obtained with:

```
bg_par_data_save_slice_n=12
```

(12 BIOGEM steps per averaging interval out of a total of 48), and monthly averages with:

```
bg_par_data_save_slice_n=4
```

(i.e. 4 BIOGEM steps for each of the 12 monthly averaging intervals, giving of a total of 48).

For lower resolution configurations of cGENIE.muffin, GOLDSTEIN may be operating on 48 time-steps per year, and BIOGEM on 24 or even 12. As cGENIE.muffin starts up it will report the ocean and biogeochemical time-stepping, such as:

```
>> Configuring ...
Setting time-stepping [GOLDSTEIN, BIOGEM:GOLDSTEIN]: 100 2
```

which specifies 100 time-steps per year for GOLDSTEIN, and 50 per year (100/2) for BIOGEM for the case of a 16 level ocean using the runmuffin.t100.sh run script. Note that for every year mid-point specified in the time-slice specification file, 4 or 12 (for seasonal and monthly, respectively) times as many time-slices will actually be saved.

\(^{10}\)Default value = 999

\(^{11}\)Note that when running using runmuffin.t100.sh, 100 time-steps are taken in the ocean and 50 in BIOGEM for a 16 level ocean model.
14.2 Time-slice output

14.2.4 More frequent data saving

Explicit frequent saving of fields or properties at specific locations can be done by setting a more higher save frequency of the time-slice data. However, because the 2D and 3D fields may contain a variety of unwanted variables in addition to the target one, save frequency is likely to be limited by the maximum netCDF file size that can sanely be manipulated, i.e. there is a limit to how large a file size you want to generate.

Several alternative options for cutting down the size of the saved data exist:

- (trivial) Make do with global or surface (or benthic) means in the time-series output (rather than saving 2 or 3D data).
- Cut down the types of data saved to the absolute minimum (see ‘Data field selection’ below).
- Save only 2D data (rather than 2 and 3D, which is the default). This can be accomplished by setting the parameters:
  
  bg_ctrl_data_save_2d=.true.
  bg_ctrl_data_save_3d=.false.

  (both are .true. by default). This disables the 3D data saving, although an empty 3D netCDF file will still be created.

One option to increase the frequency of 3D time-slice data saving is to align it automatically with the frequency of time-series data saving. This can be done by setting:

```
bg_ctrl_data_save_3d_sig=.true.
```

(by default, .false.).

The recommended/practical maximum for saved 3D time-slices is around 100 (different time-slices), depending on the number of types of data field selected to be saved.

14.2.5 Less(!) frequent data saving

You could e.g. edit the default time-slice specification file save_timeslice.dat or create your own with just a couple or even one single time-point. Or you can specify that no time-slice is saved(!) Actually, you will always get one – at the very end of the model run.

To do this (rather than creating a time-slice specification file with a single entry equal to the last year of the intended model experiment), set:

```
# force time-slice save at run end only
bg_par_infile_slice_name='save_timeslice_NONE.dat'
```

(which points to an empty file).

---

12 This is in addition to normal 3D saving at the time-slice data saving frequency
13 not recommended
14 A safety feature to ensure that however long the experiment runs, you always get data from the very end.
14.3 Time-series output

14.3.1 Frequency and timing of time-series data saving

For results time-series, a file containing a series of model times \( t_n \) at which data points need be saved is defined in the same way as for time-slices, with the filename specified by the parameter \( \text{bg\_par\_infile\_sig\_name} \). Again, the data values saved in the time-series file do not represent discrete values in time but an average, calculated from time \( t_n - \Delta t/2 \) to \( t_n + \Delta t/2 \) as per the construction of time-slices. The averaging interval, \( \Delta t \), is set by the value of the parameter \( \text{bg\_par\_data\_save\_sig\_dt} \). The format is also identical to before (with tags delineating the start and end of the list of mid-points). If there are less than two elements present in the list, a default frequency of data saving will be invoked, set equal the averaging interval, except in the situation that this results in an unreasonably large amount of data, when an order of magnitude (or more than one order of magnitude where necessary) fewer save points are assumed.\(^{15}\)

The default setting:

\[ \text{bg\_par\_infile\_sig\_name='save\_timeseries\_dat'} \]

provides for reasonably generic data saving, with the save frequency faster to begin with and becoming progressively less frequently later.

There is a related facility to \( \text{bg\_par\_infile\_sig\_name} \) for SEDGEM and ROKGEM in the parameter: \( xx\_par\_output\_years\_file\_0d \), where \( xx \) is \( sg \) for SEDGEM and \( rg \) for ROKGEM. These specify files in the \( \text{genie-\*gem/data/input} \) directory and again contain a list of years for 0D (time-series) output to be generated at. However, unlike BIOGEM, the data saved do represent discrete values in time and not e.g. annual averages.

14.3.2 Saving orbital insolation

[see orbits HOW-TO]

\(^{15}\)For historical reasons ... the maximum number of time-series (and time-slice) data points was set to 4096. This is set by the parameter \( \text{n\_data\_max} \) in \( \text{biogem\_lib.f90} \) and can be altered if required.
14.4 Data field selection

Model output – both time-slice and time-series data are saved in blocks or by categories of model variables. For instance, all dissolved tracers in the ocean (3D netCDF time-slice and/or time-series), or all particle flux fields, all carbonate chemistry and associated variables, all surface sediment composition, etc etc. This still requires a multitude of parameters, one for each category and generally also one for each of time-slice and time-series data. In an attempt to simplify this, a single parameter, bg_par_data_save_level, specifying the sort and amount of data to save can be set instead.

The value of the bg_par_data_save_level save level parameter is given as an integer between 0 and 99. The specific classes (groups) of data that are saved (as a mnemonic code) in response to each different selected save level are given in Table 24.1 (for time-slice saving) and Table 24.3 (for time-series saving), and what the mnemonics strand for in terms of actual data saved (in real words), are defined in Tables 24.2 and ??.

An alternative wordy summary to these tables would be:

0 – Save nothing.
1 – Minimum – basic geochemistry only, i.e. ocean and atmosphere tracer fields (omitting e.g. the miscellaneous fields and time-series – see below).
2 – Basic output == basic geochemistry and some physics.
3 – Basic + biology diagnostics, i.e. ocean and atmosphere tracer fields plus particulate flux fields and biological diagnostics such as limitations on export.
4 – Basic + geochemistry diagnostics, including output on air-sea gas exchange, ocean carbonate chemistry, and geochemical diagnostics such as remineralization rates and transformation, ocean pH field (3D), Fe cycle and speciation diagnostics (2D). In conjunction with the ROKGEM module, also: weathering fluxes.\textsuperscript{16}
5 – Basic + biology + geochemistry diagnostics. A combination of 3 and 4.
6 – Basic + tracer + proxy diagnostics. Tracer diagnostics includes: N*, P* etc., water column inventories (2D). Proxy diagnostics includes: ocean surface and benthic (and surface-benthic) tracers (2D). Also trace metals (e.g. C\textsubscript{fd}).
7 – Basic + biology + tracer + proxy diagnostics.
8 – Basic output + biology + tracer + proxy + geochemistry diagnostics (a somewhat large selection of variables).
9 – Basic output + full physics (e.g. all grid specifications and properties).
10 – Ocean acidification option == biology and geochemical output fields plus all carbonate chemistry.
11 – Preformed diagnostics option == BASIC + biology + tracer + proxy + redox diagnostics.
12 – Circulation option == BASIC + tracer + physics diagnostics.
14 – BASIC + FULL (inc. redox) geochem diagnostics.
15 – BASIC + biology + FULL (inc. redox) geochem diagnostics.
16 – BASIC + biology + tracer + proxy + FULL (inc. redox) geochem diagnostics.
99 – Save everything.
>99 – Use explicit user-specified settings for individual save categories. This is the default and is broadly consistent with previous version of the model.

\textsuperscript{16}The most common option.
All of options 2–99 save as *time-slices*:
- atmosphere tracer fields (2D)
- ocean tracer fields (3D)
- various miscellaneous diagnostics, including: ocean velocity (3D), overturning stream-function (2D), sea-ice extent and thickness (2D), incident radiation (2D), convection diagnostics (2D), air-sea gas exchange diagnostics (2D).
- core-top sediment composition fields (2D) (if SEDGEM is selected)

and as *time-series*:
- atmosphere tracer properties (as: atmospheric inventory (in mol) and concentration (mol kg$^{-1}$) or isotopic composition)
- ocean tracer properties (as: ocean inventory, plus mean (whole) ocean, or isotopic composition)
- mean surface and benthic tracer properties
- various miscellaneous diagnostics, including: insolation, sea-ice extent, volume, and thickness; global overturning stream-function, ocean surface pH, land surface temperature, and Fe parameters.
- sediment (core-top) composition data (if SEDGEM is selected)

In addition, further output will be automatically added to the suite of saved data depending on the module selected and also for certain sorts of *forcing*. 

14.5 Re-start files

Re-start files are saved in the results directories of each module. For ATCHEM, BIOGEM, and SEDGEM, these are in netCDF format.

For the climate modules of cGENIE.muffin (GOLDSTEIN, GOLDSTEIN-SEAICE, EMBM), re-start files can be selected to be saved in either plain text (ASCII) or netCDF format. ASCII format is the current default.
14.6 Useful output

What follows is a short summary of some of the output and how it can be used.

Note – depending on the specific model configuration (which model modules are selected) and selected tracers, as well as specific output choice, not all these variables and files will be present in the model output.

14.6.1 Physics

<table>
<thead>
<tr>
<th>Filename</th>
<th>Data</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>atm_humidity</td>
<td>Mean surface humidity. (?)</td>
<td>(rarely used) Climate and climate change, when a simple global diagnostic is needed.</td>
</tr>
<tr>
<td>atm_temp</td>
<td>Mean surface temperature. (degrees C)</td>
<td></td>
</tr>
<tr>
<td>misc_opsi</td>
<td>Global minimum (-ve) and maximum (+ve) overturning stream-function, also reported for Atlantic and Pacific basins. Units of Sv. (For certain modern configurations.)</td>
<td>Simple diagnostic of large-scale ocean circulation. There is some relationship of the maximum (negative and positive) overturning to ocean ventilation.</td>
</tr>
<tr>
<td>misc_seaice</td>
<td>Sea-ice fractional cover (%), thickness (m), and volume (m³).</td>
<td>As a simple climate (change) diagnostic.</td>
</tr>
<tr>
<td>misc_SLT</td>
<td>Mean global surface land temperature. (C)</td>
<td>For calibrating and analysing global weathering rates.</td>
</tr>
<tr>
<td>ocn_sal</td>
<td>Mean global, and typically also surface and benthic, ocean salinity. (PSU)</td>
<td>For characterizing freshwater changes and salinity forcing impacts.</td>
</tr>
<tr>
<td>ocn_temp</td>
<td>Mean global, and typically also surface and benthic, ocean temperature. (degrees C)</td>
<td>Climate and climate change.</td>
</tr>
</tbody>
</table>

Table 14.1: Summary of the main (useful) time-series output for climate-only ('physics') investigations.
<table>
<thead>
<tr>
<th>variable</th>
<th>variable (long name)</th>
<th>Description</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>atm_humidity</td>
<td>specific humidity</td>
<td>surface humidity (?)</td>
<td>(rarely used)</td>
</tr>
<tr>
<td>atm_temp</td>
<td>surface air temperature*</td>
<td>surface air temperature (degrees C)</td>
<td>climate patterns and anomalies, comparison with terrestrial temperature proxies</td>
</tr>
<tr>
<td>grid_mask</td>
<td>land-sea mask</td>
<td>land-sea mask (n/a)</td>
<td>copy-paste-edit to create masks for data analysis</td>
</tr>
<tr>
<td>grid_topo</td>
<td>ocean depth</td>
<td>ocean depth (m)</td>
<td></td>
</tr>
<tr>
<td>ocn_sur_sal</td>
<td>surface-water sal</td>
<td>surface ocean salinity (PSU)</td>
<td>diagnosing freshwater forcing impacts, regions of likely deep-water formation</td>
</tr>
<tr>
<td>ocn_sur_temp</td>
<td>surface-water temp</td>
<td>surface ocean temperature (degrees C)</td>
<td>diagnosing ocean circulation patterns, pole-to-equator temperature gradients, surface ocean temperature proxy comparisons</td>
</tr>
<tr>
<td>ocn_ben_temp</td>
<td>bottom-water temp</td>
<td>benthic temperature (°C)</td>
<td>diagnosing ocean circulation patterns, benthic temperature proxy comparisons</td>
</tr>
<tr>
<td>phys_cost</td>
<td>convective cost</td>
<td>rate of convective adjustments anywhere in the water column (n/a)</td>
<td>diagnosing deep mixed layers (and light limitation of biology) and deep-water formation regions</td>
</tr>
<tr>
<td>phys_opsi</td>
<td>Global streamfunction</td>
<td>global overturning stream-function (Sv)</td>
<td>diagnosing large-scale ocean circulation patterns, sources of deep-water formation, deep ocean ventilation</td>
</tr>
<tr>
<td>phys_psi</td>
<td>Barotropic streamfunction</td>
<td>barotropic stream-function (Sv)</td>
<td>diagnosing wind-driven ocean circulation patterns, effect of gateways</td>
</tr>
<tr>
<td>phys_seaice</td>
<td>sea-ice cover (%)</td>
<td>sea-ice cover (%)</td>
<td>(climate / sea-ice)</td>
</tr>
<tr>
<td>phys_seaice_th</td>
<td>sea-ice thickness</td>
<td>sea-ice thickness (m)</td>
<td>(climate / sea-ice)</td>
</tr>
</tbody>
</table>

Table 14.2: Summary of the main (useful, physics-focussed) 2D time-slice output.
<table>
<thead>
<tr>
<th>variable</th>
<th>variable (long name)</th>
<th>Description</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>ocn_sal</td>
<td>salinity</td>
<td>ocean salinity (PSU)</td>
<td>diagnosing ocean circulation patterns</td>
</tr>
<tr>
<td>ocn_temp</td>
<td>temperature</td>
<td>ocean temperature (C)</td>
<td>diagnosing ocean circulation patterns</td>
</tr>
<tr>
<td>phys_u</td>
<td>ocean velocity - u</td>
<td>Eastwards component of ocean velocity (m/s)</td>
<td>diagnosing ocean circulation patterns and currents</td>
</tr>
<tr>
<td>phys_v</td>
<td>ocean velocity - v</td>
<td>Northwards component of ocean velocity (m/s)</td>
<td>diagnosing ocean circulation patterns and currents</td>
</tr>
<tr>
<td>phys_w</td>
<td>ocean velocity - w</td>
<td>upwards component of ocean velocity (m/s)</td>
<td>diagnosing ocean circulation patterns (note that velocity is measured at the top of an ocean depth layer, hence n/a for the surface layer)</td>
</tr>
</tbody>
</table>

Table 14.3: Summary of the main (useful) 3D time-slice output.
14.6 Useful output

14.6.2 (Bio)Geochemistry
### Table 14.4: Summary of the main (useful, plus notes on a few less used) time-series output for (bio)geochemistry (non ecological) investigations.

<table>
<thead>
<tr>
<th>Filename</th>
<th>Data</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>biogem_series_*.res</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>atm_pCO2</code></td>
<td>Global inventory (mol), mean concentration (atm) of atmospheric CO₂.</td>
<td>Drivers of and feedbacks with climate. Diagnostic of response to carbon emissions (and removal).</td>
</tr>
<tr>
<td><code>atm_pCO2_13C</code></td>
<td>¹³C inventory (mol) and δ¹³C of atmospheric CO₂.</td>
<td>Diagnostic of carbon emissions (and removal). Comparison with (terrestrial) proxy δ¹³C data.</td>
</tr>
<tr>
<td><code>atm_pO2</code></td>
<td>Global inventory (mol), mean concentration (atm) of atmospheric O₂.</td>
<td>Limited use. Checking on restoring forcing of atmospheric O₂. Impacts of C&lt;sub&gt;org&lt;/sub&gt; burial if included in model.</td>
</tr>
<tr>
<td><code>carb_sur_conc_*</code></td>
<td>Carbonate chemistry components (mean surface) (molkg⁻¹).</td>
<td>Not generally useful.</td>
</tr>
<tr>
<td><code>carb_sur_H</code></td>
<td>Surface ocean mean [H⁺] (molkg⁻¹).</td>
<td>More useful is pH – reported under misc (see below).</td>
</tr>
<tr>
<td><code>diag_misc_specified_forcing_*</code></td>
<td>Applied flux forcings (molyr⁻¹).</td>
<td>Whenever a restoring, or flux forcing is specified, the actual flux employed, is saved here. Useful for diagnosing the flux associated with a restoring forcing (e.g. allowing emissions flux associated with RCP (restoring forcing) scenario to be diagnosed.)</td>
</tr>
<tr>
<td><code>fexport_CaCO3</code></td>
<td>Total flux (molyr⁻¹) and flux density (molm⁻²yr⁻¹), of CaCO₃ export from the ocean surface.</td>
<td>Carbonate production. Impacts of ocean acidification.</td>
</tr>
<tr>
<td><code>fexport_POC</code></td>
<td>Total flux (molyr⁻¹) and flux density (molm⁻²yr⁻¹), of POC export from the ocean surface.</td>
<td>Particulate organic matter export. Impacts of changes in nutrient supply and limitation.</td>
</tr>
<tr>
<td><code>misc_surpH</code></td>
<td>Mean surface pH.</td>
<td>Ocean acidification.</td>
</tr>
<tr>
<td><code>ocn_DIC_13C</code></td>
<td>Global inventory (mol), mean global, surface, and benthic δ¹³C.</td>
<td>Carbon release and removal. Surface-benthic – indicator or strength of carbon export and the biological pump, as well ocean ventilation.</td>
</tr>
<tr>
<td><code>ocn_O2</code></td>
<td>Global inventory (mol), mean global, surface, benthic concentrations (molkg⁻¹) of O₂.</td>
<td>Indication of changes in ocean anoxia. (long-term) Imbalances between burial and weathering.</td>
</tr>
<tr>
<td><code>ocn_PO4</code></td>
<td>Global inventory (mol), mean global, surface, benthic concentrations (molkg⁻¹) of PO₄.</td>
<td>Nutrient limitation. (long-term) Imbalances between burial and weathering.</td>
</tr>
<tr>
<td><code>ocn_TDFe</code></td>
<td>Global inventory (mol), mean global, surface, benthic concentrations (molkg⁻¹) of dissolved Fe.</td>
<td>Nutrient limitation. (long-term) Imbalances between burial and weathering.</td>
</tr>
<tr>
<td>variable</td>
<td>variable (long name)</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>atm_*</td>
<td></td>
<td>Distributions of gases and isotopes.</td>
</tr>
<tr>
<td></td>
<td>carb_sur_ohm_cal</td>
<td>Ocean surface aragonite and calcite saturation.</td>
</tr>
<tr>
<td></td>
<td>misc_sur_rCaCO3toPOC</td>
<td>CaCO3 to POC export rain ratio</td>
</tr>
<tr>
<td></td>
<td>ocn_sur_TDFe</td>
<td>surface-water TDFe</td>
</tr>
<tr>
<td></td>
<td>ocn_sur_TDL</td>
<td>surface-water TDL</td>
</tr>
<tr>
<td></td>
<td>ocn_sur_PO4</td>
<td>surface-water PO4</td>
</tr>
<tr>
<td></td>
<td>ocn_ben_PO4</td>
<td>bottom-water PO4</td>
</tr>
<tr>
<td></td>
<td>ocn_ben_DIC_13C</td>
<td>Benthic δ¹³C.</td>
</tr>
<tr>
<td></td>
<td>ocn_intDIC</td>
<td>DIC water-column integrated tracer inventory</td>
</tr>
</tbody>
</table>

Table 14.5: Summary of the main (mostly useful) 2D time-slice output for (bio)geochemistry.
14.6.3 Biology/Ecology
The primary format for saving spatial (2- and 3-D) data is netCDF (network Common Data Form). More information on the netCDF format and the libraries necessary to compile the model can be found here. The writing of netCDF follows roughly the CF1.0 convention (NetCDF Climate and Forecast (CF) Metadata Convention). The netCDF output is written for the BIOGEM and SEDGEM modules separately, and both modules have a flag that suppresses spatial data file saving in ASCII format, with netCDF format being the default.

Under unix/linux, netCDF files can be interrogated with: $ ncdump -h filename.nc which will give you the header information of the file. The command is included in the netCDF library which has to be present to run the model anyway. It’s useful to get the NCO software package helping to concatenate files or extract variables as shell command. A full list of available software to manipulate or graphically illustrate netCDF files can be found here.

This Chapter covers how to visualize muffin output, with a particular emphasis on (netCDF) spatial data, via:

- **Panoply**
  If you really really must insist on using Windozzzz, the recommended viewer for netCDF is Panoply (see below). Panoply can be run under linux and on the Mac (OS X).
  There is also ncBrowse. Again, this will also run under LINUX and on the Mac (OS X).

- **MATLAB**
  You can also view netCDF files using MATLAB, for which a number of plotting functions are provided. An advantage here is that the MATLAB code can be hacked to produce much more powerful and bespoke analysis and plots.
Chapter 15. Introduction to model results analysis

15.1 Plotting with Excel

Just ... don’t do it!¹

¹Obviously, there may be circumstances where it might be helpful to plot time-series output in Excel, although in practice, time-series output is much easier and faster to load in and plot with MATLAB.
15.2 Plotting with Panoply

In the following sub-sections are some pointers and examples of **Panoply** plotting.\(^2\)

When you open a netCDF file, you will be presented with a **Datasets** window (on the left hand side of the application window). This contains a list of all the variables available that you can display. You will find that the 'Long Name' description of the variable will be the most helpful to identify the one you want. Clicking (once) and highlighting an entry will display further information about that variable in the **Variable** window on the left hand side of the application window.

In a drop-down box at the bottom of the application window is an option for shortening the list of displayed variables in the netCDF file:

- **Georeferenced variables** – All the spatial (2 or 3D) variables.
- **Plottable variables** – As above, but now including axis definitions (which can be plotted as 1D lines).
- **All variables** – all of the variables.

To create a plot of a variable – simply double-click anywhere on the line containing that variable. A dialogue box will open with various options (Figure 15.1). For 3-D model output, you have the option of whether you want a ‘**Lat-Vert**’, ‘**Lon-Lat**’, or ‘**Lon-Vert**’ plot (for the 2-D fields, the only choice is ‘**Lon-Lat**’).

- **‘Lat-Vert**’ plots – a common way of visualizing vertical distribution in the ocean. The default is for **Panoply** to display an average of all longitudes in a zonal mean. Un-ticking the Ave box in the **Array** tab will enable you to specify a specific longitude section. Be aware that Panoply likes to plot the depth inverted by default ...
- **‘Lon-Lat**’ plots – the classic 2D, top-down view of the ocean. There are multiple levels (depth layers) in the ocean of data that can be plotted, from the surface to the abyssal ocean.
- **‘Lon-Vert**’ plots – (an uncommon option).

For all three: there may be multiple time-slices (i.e., you can plot data saved from different years).\(^3\)

\(^2\)*WARNING* These instructions are strictly valid for older version of **Panoply** (ca. version 2.9.4), although some updates to the text have been made in light of version 4.6.2 ... so be aware that the plotting control buttons and options may have subtly changed in newer versions and the text no longer reflect the exact (current) options in **Panoply** ...

\(^3\)Remember that the default, first time slice, will be the first once saved in the experiment. The last one saved and displayed, will reflect the end of your experiments.

![Figure 15.1: Panoply Create Plot dialogue window.](image)

You can interpolate the data or not (often you may find that it is clearer not to interpolate the
data but to leave it as 'blocky' colors corresponding to the resolution of the model), change the scale and colors, overlay continental outline, change the projection, etc etc. Gray cells represent 'dry' grid points, i.e., continental or oceanic crust. To save plots in Panoply – from the file menu: File, then Save Image As ... and then select the location, filename, and graphics format.

15.2.1 Issues with Panoply default settings

The default settings in Panoply, i.e. those used when a plot is first created, can often mislead. In particular, note:

- **Year (Array tab)**
  
  The default is for the very 1st time-slice to be displayed rather than the experiment end. The first time-slice is numbered from 1 to however many total time-slices have been saved (displayed to the immediate right of the Year box), and it is this integer number that appears in the Year box – not the year of the data save. Instead, the mid-point year of the time-slice is displayed in a second box (labeled 'Year mid-point'). Different time-slices to be plotted can be selected by either clicking through the saved year count, or by selecting the save year mid-point from the drop-down list.

- **Scale Range (Scale tab)**
  
  The color scale is auto-scaled so that the range always goes from the minimum to maximum displayed value. This can potentially mislead if save years and/or depth/latitude slices are scrolled through as the scale will be automatically adjusted to fit each plot in turn. Confusion can also arise for fields with no variation, e.g. atmospheric trace gas concentrations or air temperature – the auto-scaled plot in these instances has a uniform color but with odd hatching as Panoply dutifully tries to achieve the impossible (creating a scale of multiple colors for a single value).

- **Zonal averaging (Array tab)**
  
  Lat-Vert plots are displayed as a zonal mean by default. This is indicated by the tick in the Ave box (bottom RH corner). Un-ticking the Ave box releases the averaging with the first longitudinal value of the grid now displayed instead. Similar to how Panoply displays years – the longitudinal grid locations are counter from 1 to typically 36 (depending on the resolution of the ocean grid), with the longitudinal mid-point value in degrees East displayed to the right. Different longitudinal sections to be plotted can be selected by either clicking through the grid point number count, or by selecting the longitudinal mid-point from the drop-down list.

- **Scale bar tick marks (Array tab)**
  
  The tick labels on the color scale are displayed by default in the format: x.y. If the typical values of the variable are order e.g. $10^{-6}$ you will end up with value labels ranging from 0.0 to 0.0 ... This can be most easily resolved in one of two ways:
  
  - The format of the label can be changed by selecting a different option from the pull-down Tick Label Format box (default == %.1f). For instance, %.2e would give a display in the format x.xxExyy (or x.xxEx-yy) or %.6f would give x.xxxxxx.
  
  - An alternative is to re-scale the values. This is done in the Scaling Factor box in which you set the scale factor in powers of 10. For example: setting -6 in effect converts units of mol kg$^{-6}$ to μmol kg$^{-6}$.

- **Other things to watch out for include:**
15.2 Plotting with Panoply

– A plot involving depth, being by default, 'up-side-down'!.
This is fixed in the Grid tab, and the click button, Swap B/T.

– In 'Lon-Lat' plots, the modern continental outline being displayed by default.
This can be fixed by changing options in the Overlay tab.

Simply be careful when opening a new plot that you are looking at what you *think* you are looking at (or what you think you are looking at *is* what you are looking at).

Note that default plotting settings in Panoply can be changed (and saved).

15.2.2 Basic plots – examples

TO COME ...

15.2.3 Difference (anomaly) plots

It is possible to create an anomaly (difference) maps in Panoply which are essential when analyzing changes in a variable that may be small compared to the global spatial variability. To do this:

• First, open the netCDF results file.
• Open the variable of interest, e.g., atm_temp (surface air temperature) in the 2D netCDF file.
• From the upper LH corner of the Dataset Browser window, from the drop-down menu, select the name of the plot you have just created (atm_temp in field biogem_2D...).
• From the upper LH corner of the Dataset Browser window, now click on the Combine Plot icon.

You now have a plot window that is displaying a difference map. By default, it is showing you the difference between two identical (in time) slices. The two different slices are labeled Array 1 (LH side) and Array 2 (RH side).

NOTE: Easier than mucking about with Combine Plot, is having open the first dataset, simply drag another variable from the list of variables in the Sources window, into the Plot window. You can drag either a different, or the same variable, into the Plot window.

For instance, you can keep one array (Array 1) fixed to the initial (year 1 (centered on 0.5)) and vary the year in the second array (Array 2). Note that you can select in Panoply whether Array 1 - Array 2 is plotted, or Array 2 - Array 1, or various proportional or relative differences.

If you switch off the auto-scaling feature (Always fit to data) you can center the scale so that no change is white, with positive deviations = red and negative = blue by clicking on Center on 0. This is something of a convention in the scientific literature.

The same variable in two different model experiments can also be opened up and analyzed combined:

• Start by opening up both required netCDF files.
• Open the variable of interest in one (either one) of the two 2D netCDF data-sets.
• From the upper LH corner of the Dataset Browser window, from the drop-down menu, select the name of the plot you have just created.
Chapter 15. Introduction to model results analysis

- Now double-click in the variable in the 2nd netCDF dataset.
  You now have a plot window that is displaying a difference map, but of the same variable between two different experiments, rather than two years of the same experiment.

15.2.4 Ocean velocity plots
By combining the two (horizontal) fields of ocean circulation, rather than a difference plot, Panoply can create a velocity plot. This is a great way of visualizing surface (and deeper) currents and circulation patterns. To do this:

- First, open the 3-D netCDF results file.
- Open either the phys_u ('ocean velocity - u') or phys_v ('ocean velocity - v') field and select a Lon-Lat plot.
- From the upper LH corner of the Sources window, from the drop-down menu, select the name of the plot you have just created.
- Now double-click on the other velocity variable (whichever of the u and v fields you did not open first).
- By default you get a difference map, which is pretty useless really. From the drop-down Plot menu box (which should be displaying 'Array 1 - Array 2' by default) select: Vector Magnitude (bottom of the list).
  You now have a plot window that is displaying the ocean velocity field, with arrows indicating the direction and speed (length of the arrow) together with an interpolated color background of the speed.

You can re-scale the velocity arrows to more clearly display the circulation pattern by altering the Scale Length value (Contours & Vectors tab). A value of 0.1 is a reasonable choice for surface currents. e.g. see Figure 15.2.

If you want to display deeper (in the ocean) current fields and/or different time-slices, take care that the depth level (/time-slice) in both LH and RH sides of the Array(s) panel must be changed to the same value. If displaying deeper current fields, then the velocity vectors will have to be further re-scaled (to a smaller value) in line with the lower velocities at depth compared to the surface.
Figure 15.2: Example (modern) ocean surface velocity (current) map.
Chapter 15. Introduction to model results analysis

15.3 MATLAB plotting

15.3.1 MATLAB 101

If you need a tutorial on MATLAB, either as a refresher, or because you have not used the program before, refer to (and/or work through) the following sections of the matlabananas MATLAB textbook:

- Chapter 1 – this covers the very basics of using MATLAB, what variables are, including scalars (e.g. single numbers), vectors (1D array), matrices (2D array) and higher order arrays, now to index and carry out basic manipulation of arrays, basic data loading and saving and plotting.
- Sections 2.1 and 2.2 of Chapter 2, which cover creating (script) programs – i.e. adding all your MATLAB commands to a file (a .m or ’m-file’) and running the file, and functions – programs (m-files) where one or more parameters might be passed into the function when it is called (e.g. at the command line), and potentially variables returned.
- In Chapter 3 – subsection 3.1.3 deals with reading in netCDF format data, which is the muffin format for spatial data. Section 3.2 deals with more advanced 2D plotting (also of direct relevance to muffin data processing and visualization using MATLAB).

15.3.2 MATLAB and ASCII (time-series)

The time-series (.res) output of BIOGEM are in a simple plain text (ASCII) format. These can be read in very easily in MATLAB using the load command. Note that a brief description of each column of data in the time-series files appears on the first first line of the file, and that prefixing this is a % symbol, that MATLAB ignores. Hence only the columns of data gets read in by default using load.  

For example:

```matlab
>> co2=load('biogem_series_atm_pCO_2.res','-ascii');
>> plot(co2(:,1),1.0E6*co2(:,3));
```

loads in the atmospheric CO$_2$ time-series file (assigning it to the array variable co2), and then plots (as a line graph) the 3rd column (CO$_2$ concentration) vs. the first (time). Because concentrations are saved in units of atmospheres, a factor of 1.0E6 is applied to convert to μatm. Equivalently:

```matlab
>> co2=load('biogem_series_atm_pCO2\).res','-ascii');
>> scatter(co2(:,1),1.0E6*co2(:,3));
```

15.3.3 MATLAB 'Import Data' ...

You can also import the time-series files by clicking on the Import Data icon:

- Navigate to the BIOGEM sub-directory of your experiment results directory. Ensure that All Files is selected and click on the time-series file you want.
- MATLAB ignores the header lines, and it should be safe to simply click on Import Selection for all columns, or select what you want to plot – typically year (the first column) and another one.
- A default variable name – the filename minus the underscore characters – will appear listed in the MATLAB Workspace window. Double-click on the variable name to open up the

---

4There are also other MATLAB commands for reading in text data – refer to to the MATLAB programming text.
imported data in a table view. If you select columns to plot, and then head over to the main PLOTS tab, a range of plotting options are provided and a plot can be generated by clicking on one of the plotting icons.\footnote{Note that if you have not selected any data columns, then all the plotting icons are disabled and greyed out in the PLOTS tab.}

- Labels can be added, scales and markers changed, etc etc, in the Figure window.

Note that as the data has been imported as an array in MATLAB, you can also plot directly from the command line. (And indeed, you could also have loaded the data from the command line.)

### 15.3.4 MATLAB and netCDF (time-slice)

Basically – the only hard part, having opened the netCDF file is in correctly deducing which dimension in an extracted data array is longitude, latitude, (sometimes depth), or time. Mostly this should be pretty obvious from inspecting the MATLAB Workspace window (assuming you have a column for Size selected to be displayed), or using the size command.\footnote{It also turns out that the order of dimensions for a variable read in by MATLAB, is the opposite of the order listed in the Variable window of Panoply.} Once you have done this, you can plot slices, scale data, average or otherwise process data, extract locations at specific locations, etc etc.

As an example – in MATLAB, first either change directory to the biogem directory of a set of muffin results, or from where-ever you are, add a path to the location of the biogem directory. To open the 2D BIOGEM netCDF results file, type:

```matlab
>> ncid = netcdf.open('fields_biogem_2d.nc','nowrite');
```

To extract a variable, you first need to find its ID from its name:

```matlab
>> varid = netcdf.inqVarID(ncid,'NAME');
```

where \texttt{NAME} is a place-holder for the name of he variable (as a string). You might need to use Panoply to display all the different variable names. Or, you can list all the variables and stuff in the netCDF file using \texttt{ncdisp}:

```matlab
>> ncdisp('fields_biogem_2d.nc')
```

Having, by one means or another, identified the name of the variable you are interested in, you can recover its ID and then the data itself, for example, for \texttt{atm_temp} (surface air temperature):

```matlab
>> varid = netcdf.inqVarID(ncid,'atm_temp');
>> data = netcdf.getVar(ncid,varid);
```

In loading in the variable, you end up with a multi-dimensional array – 2 spatial dimensions and if you have more than 1 \textit{time-slice} of data saved, 1 temporal dimension (and if you loaded in the BIOGEM 3D netCDF file, you end up with a 4-dimensional array). MATLAB reports the size of the array in the Workspace window (depending on which column display options you have selected). In the example here, which took the experiment \texttt{EXAMPLE.worj2.Caoetal2009.RCP6p0}, the array for atmospheric temperature is reported as $36 \times 36 \times 8$.\footnote{Although note that in Panoply, it is reported as \texttt{atm_temp(time=8, lat=36, lon=36)}.} so the last of the 8 time-slices would be accessed as:
Panoply reports variables in the the BIOSGEM 3D netCDF file as having dimensions of: \((time, zt, lat, lon)\) (e.g. \((time=13, zt=16, lat=36, lon=36)\)) whereas MATLAB reads it in the reverse order. For instance, if we load in the (3D) ocean temperature field:

\[
\begin{align*}
&\text{\texttt{ncid = netcdf.open('fields_biogem_3d.nc','nowrite')}} \\
&\text{\texttt{varid = netcdf.inqVarID(ncid,'ocn_temp')}} \\
&\text{\texttt{data = netcdf.getVar(ncid,varid)}}; \\
\end{align*}
\]

and then type:

\[
\begin{align*}
&\text{\texttt{size(data)}} \\
&\text{\texttt{ans =}} \\
&\text{\texttt{36 36 16 13}}
\end{align*}
\]

we get an array orientated as \((lon, lat, zt, time)\). The final time-slice would hence then be accessed:

\[
\begin{align*}
&\text{\texttt{data_last = data(:,:,,:,end)}}; \\
\end{align*}
\]

with \texttt{data_last} now becoming a 3D array with dimensions \((lon, lat, zt)\).

There are three key things to remember at this point:

1. Firstly, the depth levels are read such that index 1 is the surface, and 16 is the deepest ocean depth level (in this case, otherwise it is 8).
2. For the lon-lat part – \((lon, lat)\) equates to \textit{rows vs. columns} in MATLAB, and hence if you were to plot e.g. the surface ocean slice:

\[
\begin{align*}
&\text{\texttt{imagesc(data_last(:,:,1));}} \\
\end{align*}
\]

you will end up with the plot on its side, with latitude along the \(x\)-axis and longitude on the \(y\)-axis.

You can exchange rows and columns in MATLAB with the \textit{transpose operator} (see programming/MATLAB book):

\[
\begin{align*}
&\text{\texttt{imagesc(data_last(:,:,1)')}}; \\
\end{align*}
\]

but ... this still leaves you with an up-side-down plot, because MATLAB reads from the first row down, whereas in latitude, you are expecting to read from -90 degrees up (towards the N pole). \texttt{flipud} accomplishes the final transformation:

\[
\begin{align*}
&\text{\texttt{imagesc(flipud(data_last(:,:,1)'))}];}
\end{align*}
\]

---

\(^8_{zt}\) \(zt\) is the depth level in the ocean.
3. The final complication is that **muffin** netCDF output uses a special value to represent an invalid or null number, e.g. for ocean temperature where the grid point is land (and an ocean temperature value would have no meaning). In the netCDF definition, **Panoply** is told what the special number is, hence it shows land in grey when plotting ocean variables.

Panoply reports this as: \texttt{missing\_value} = 9.969209968386869E36.

There is no easy way (that I can see!) to get **MATLAB** to deal with this for you, so you need to search and replace this value, e.g.:

\begin{verbatim}
>> null=9.969209968386869E36;
>> data_last(find(data_last==null))=NaN;
\end{verbatim}

which searches for this null value and replaces it with a NaN (so that now it can be simply plotted).

Once you are done accessing data, it is good practice to close the netCDF file after you are done with it:

\begin{verbatim}
>> netcdf.close(ncid);
\end{verbatim}

The question then remains: what do you actually 'do' with it in **MATLAB**?

**Plotting sections**

For a quick look-see, \texttt{imagesc} (as per above) is handy.

For more advanced plotting (and for presentation) – refer to the **matlabananas** MATLAB programming text.

As per above, horizontal (lon-lat) fields can be extracted from 2D \texttt{netCDF} output by: \texttt{data(:,:,time)} and from 3D by: \texttt{data(:,:,zt,time)}. Obviously, you could also extract lon-depth via: \texttt{data(:,:,lat,:,time)} and lat-depth via: \texttt{data(lon,:,,:,time)}.

Anomalies (with time) are created: \texttt{data(:,:,,:,time2)-data(:,:,,:,time2)}.

Typically, except a little trial-and-error in extracting the dimensions you want, and also in correcting the orientation of the matrix or resulting plot.

**Calculating inventories**

Often, the inventory (total mass or number of moles) of the ocean or atmosphere is useful to know, particularly as a function of time. For the ocean (or atmosphere) as a whole, the \texttt{time-series} output files report this (alongside the mean global concentration).

You can also calculate this with **MATLAB** from the netCDF output. Fields of ocean concentration are saved in the 3D output (and atmospheric concentrations in the 2D output). To convert concentration in the ocean, in units of \texttt{molkg^{-1}} you’ll need to know the mass of each ocean cell, in units of \texttt{kg}.

When saving using save option 9, or 99\footnote{Parameter \texttt{bg\_par\_data\_save\_level} – see earlier}, you save the 'physics' of the ocean model, which is actually mostly just the grid information, such as cell area, thickness, latitude and longitude edges and midpoints, depth edges and mid point. Also saved are the masses and volumes of the grid of cells. So to derive an array of cell tracer inventories from an array of concentrations, and the array of cell masses (variable \texttt{phys\_ocn\_M}), you’d write:
Chapter 15. Introduction to model results analysis

varid = netcdf.inqVarID(ncid,'ocn_temp');
data = netcdf.getVar(ncid,varid);
varid = netcdf.inqVarID(ncid,'phys_ocn_M');
mass = netcdf.getVar(ncid,varid);
inventory = data(:,:,,:,time).*mass(:,:,,:,time);

Before summing inventory to determine the global total inventory, you will, as before, have to deal with the null values (converting them to NaN) and then deal with the presence of NaNs in the array when summing ...

The advantage of doing the calculations in MATLAB (despite being provided with the global mean and inventory in the time-series files) is that you could calculate the inventory of a tracer (substance) for just the ocean surface, or just a specific region of band of latitude. Or you could calculate the mean concentration for just a specific region of the ocean.10

10In calculating mean concentrations, you’ll need to volume or mass weight the concentrations, and hence still need to use one of the physics variables.
The **muffinplot** suite of MATLAB functions provides a means of plotting a variety of output reproducibly (by means of a saved parameter file) and with the potential for automation (i.e. automatically generating the same analysis for a large number of different experiments).

The functions comprising this software suite include:

- `plot_fields_biogem_2d`– lon-lat plots from the 2D **biogem** output.
- `plot_fields_biogem_3d_j`– lat-depth plots from the 3D **biogem** output.
- `plot_fields_biogem_3d_k`– lon-lat plots from the 3D **biogem** output.
- `plot_fields_ccd`– analysis of the ‘CCD’ (from both **biogem** 2D and **sedgem** 2D output).
- `plot_fields_sedgem_2d`– lon-lat plots from the 2D **sedgem** output.
- `plot_histc_2d`– a generic color-coded histogram function.
- `plot_sedcore`– down-core plots from **sedgem** sedcore output.
- `plot_timeseries_biogem`– time-series plots from **biogem** time-series output.

Note that at this current time, there is no facility for lon-depth plotting (unlike in **Panoply**).

Most of the plots also perform additional functions (which can be generally disabled if not wanted), such as plotting and saving zonal or depth profiles, plotting difference maps, plotting and labelling data on maps and carrying out model-data fit statistics and plotting, extracting model values at data locations.

The following sections provide an overview and examples of such plotting and analysis.
16.1 Installation

muffinplot can be obtained from github. If you do not have a git client on your computer (and hence cannot clone the repository locally), then simply download the archive file of all the code (from clone or download – pick Download ZIP).

When you unpack (or clone) muffinplot (note that it is likely to unpack by default into its own directory), you should see 3 directories – EXAMPLES, and MASKS, source, a series of .m files, and a single lonely .ps graphics file (colorscales.ps). The .m files are split into filenames with or without the label SETTINGS – the ones without are code files (functions), and the ones with the word SETTINGS in their filename, contain parameter settings for plotting.

By default (the parameters can be changed if you wish), wherever you run the muffinplot plotting function from, requires that you also have a subdirectory called cgenie_output present, and that the (complete) muffin experiment output directories of anything you want to plot should be in subdirectories of that – i.e. the contents of cgenie_output should look like the contents of cgenie_output on your cluster account. Mask files reside in the MASKS subdirectory of the muffinplot installation, or in your current directory, or anywhere in the MATLAB path.

The simplest option is to unpack/clone muffinplot to a directory containing cgenie_output and hence all your experimental results directories. In other words:

1. Create some local directory e.g. called RESULTS.
2. Create a subdirectory in RESULTS called cgenie_output.
3. Drag your experiment results folders across into the subdirectory cgenie_output, as per the directory structure on the cluster. (Or better – drag the archived files to cgenie_output and unpack them.)
4. Install muffinplot in the RESULTS directory.
5. Change the MATLAB working directory to RESULTS.

The plotting functions are run simply by typing their name and passing a list of parameters (comma-separated, with the complete list enclosed in parentheses). By default, the SETTINGS files need to be in the same directory as you are running the functions from, or in one of the MATLAB paths. Results are saved to a subdirectory that by default is called PLOTS, which will be created for you if it does not already exist.

All the plotting functions provide some manner of 'help', that can be obtained by typing at the command line:

```matlab
>> help FUNCTIONNAME
```

where FUNCTIONNAME is the function name (as per listed above).

---

1 Although you do not need to copy all the results over ... just the experiments that you wish to plot up.

2 However, the muffinplot functions do not have to be run from the same directory that you are in – you can install them somewhere convenient, and then with MATLAB set to a directory containing a cgenie_output experiment results directory, you can:

```matlab
>> addpath(PATH)
```

where PATH is the path to the directory where muffinplot is installed.
The `muffinplot` function `plot_timeseries_biogem.m` provides a facility to plot BIOGEM time-series (.res) output. You can use MATLAB help on the function name to detail the parameters that need to be passed (and examples).

The `plot_timeseries_biogem` plotting function plots a basic set of time-series variables by default. It then, enables a setup up to 3 additional variables to be plotted. It is also associated with a file of parameter values (plot_timeseries_SETTINGS.m by default) for fine-tuning plots.

The plotting function requires a list of parameters to be passed in the argument list, i.e.:

```matlab
>> plot_timeseries_biogem(PAR1,PAR2,PAR3,... PARn)
```

These are, in order:
1. `PEXP1 – string →` the (first) experiment name.
2. `PEXP2 – string →` is the name of the 2nd (optional) experiment. If no second experiment is selected, then a null string value must be passed, i.e., "".
3. `PTMIN – real →` minimum plotted time (x-axis).
4. `PTMAX – real →` maximum plotted time (x-axis).
5. `PDATA1 – string →` time-series variable name for additional data to plot.
   Omit the ’biogem_series_’ and ’.res’ parts of the filename.
   Leave blank, i.e., "", for no additional data panel.
6. `PDATA1N – integer →` the column number of the data in the time-series file.
7. `PDATA2 – string →` time-series variable name for additional data to plot.
8. `PDATA2N – integer →` the column number of the data in the time-series file.
9. `PDATA3 – string →` time-series variable name for additional data to plot.
10. `PDATA3N – integer →` the column number of the data in the time-series file.
11. `POPT – string →` The string for an alternative plotting parameter set.
    If an empty (’’) value is passed as this parameter, then the default parameter set file is used.
12. `PNAME – string →` The string for an alternative filename.

Note that if an empty value is passed as this parameter, then a filename is automatically generated.

A simple example usage would be:

```matlab
>> plot_timeseries_biogem('myexperiment','',0.0,10000.0,'',0,'',0,'',0,'','')
```

where `myexperiment` is the name of the 1st experiment, followed by and empty string (’’') indicating no second experiment. The results are to be plotted from 0.0 to 10000.0 years (the 2 following parameters; 0.0,10000.0). Then, no additional (maximum 3) optional parameters are requested, and hence the next parameters passed are: ’’’,0,’’’0,’’’0,’’’0.’’’. Finally, the default plotting parameter set is required, and no specific alternative filename is ot be used, accounting for the final 2 empty strings passed.

By default, `plot_timeseries_biogem` plots 2 panels of data, both with 2 (LH and RH) axes:

1. Atmospheric \( CO_2 \). Note that if the experiment was not \( CO_2 \)-enabled (i.e. not run with a global carbon cycle), a warning is given and ‘fake’ data (actually, random numbers) is plotted.

3Obviously – there are lots of different and easy ways of plotting plain text output in the form of a simple column format.
2. Atmospheric $\delta^{13}CO_2$. Note that if the experiment was not $\delta^{13}CO_2$-enabled (i.e. not run with a global carbon cycle), a warning is given and 'fake' data is plotted.

3. Atmospheric temperature (as a global mean, annual average).

4. Fractional (percentage) sea-ice extent.

Additional model outputs can then be added by listing them in the function call. For example, to also plot the global overturning strength, which is contained in the file biogem_series_misc_opsi.res, you would add 'misc_opsi',3, where the 3 indicates the 3rd column of data in the file is to be plotted, which in this case is the global maximum overturning value (and the 2nd column is the minimum value). The complete line looks like:

```matlab
>> plot_timeseries_biogem('myexperiment','',0.0,10000.0,'misc_opsi',3,'',0,'',0,'','')
```

By default, the plotted variables are all auto-scaled. To specify the y-axes, you will need to edit the plotting settings parameter file: plot_timeseries_SETTINGS.m. For the default plotted 4 results variables, the minimum and maximum y-axis limits are specified in the section:

```matlab
axis_pCO2min = 0.0;
axis_pCO2max = 0.0;
axis_d13Cmin = 0.0;
axis_d13Cmin = 0.0;
axis_Tatmmin = 0.0;
axis_Tatmmin = 0.0;
axis_icemin = 0.0;
axis_icemin = 0.0;

The default zero values here, tell the plotting function to create an auto-scale for the y-axis.5 Following this in the parameter file, are the settings for the optional variable plotting:

```matlab
axis_data1_min = 0.0;
axis_data1_max = 0.0;
axis_data2_min = 0.0;
axis_data2_max = 0.0;
axis_data2_min = 0.0;
axis_data2_max = 0.0;
```

Note that if you want instead to copy and rename and then edit this settings file, you will need to pass the new (non-default) filename when calling the plotting function. For example, if you created a new parameter settings file: settings_NEW.m, then the function call would look like:

```matlab
>> plot_timeseries_biogem('myexperiment','',0.0,10000.0,'','','settings_NEW','')
```

### 16.3 Spatial plotting

#### Overview

4 of the muffinplot plotting functions provide spatial (2D) plotting capabilities:

- **plot_fields_biogem_2d**
  
  Plot a 2-D field from: fields_biogem_2d.nc.

---

4 Don’t forget that you omit the 'biogem_series_' and '.res' parts of the filename.

5 Note the units of atmospheric $CO_2$ as $\mu atm$. 
16.3 Spatial plotting

- **plot_fields_biogem_3d_i**
  Plot a vertical-meridional (2-D) slice through the ocean (i.e., all cells have the same \( i \) (longitudinal) coordinate value) from: `fields_biogem_3d.nc`.
  Options are provided for averaging longitudinally over a supplied mask, which may be the entire ocean and hence giving a global meridional cross-sectional mean, of a specific ocean basin, or may be a single cell 'wide' longitudinally and take a meandering path hence simulating an ocean transect. An option is also provided to overlay an ocean circulation stream-function.

- **plot_fields_biogem_3d_k.m**
  Plot a horizontal slice through the ocean from: `fields_biogem_3d.nc`.
  An option is provided for overlaying ocean circulation (velocity fields). Water column integrals can also be calculated and displayed, as well as benthic surfaces, and the function can also determine the spatial distribution of the maximum or minimum value occurring anywhere in the water column (or portion of the water column).

- **plot_fields_sedgem_2d**
  Plot a 2-D field from: `fields_sedgem_2d.nc`.

All 4 plotting functions can also overlay observed data and create difference (anomaly) maps – either between different experiments, time-slices, or variables, or between model and data and provide summary statistics regarding the difference.

**Argument (parameter) list**

All 4 plotting functions share exactly the same format of parameters\(^6\) passed in the argument list:

```
>> FUNCTIONNAME(PAR1,PAR2,PAR3, ... PARn)
```

i.e. take a (long!) list of parameters. These are (in order):

Firstly, a series of parameters for defining experiment, variable, and year:

1. \( \text{PEXP1} – \text{string} \) – is the name of the 1st (main) experiment. A results directory with the same name must exist in the directory `cgenie_output`\(^7\).  
2. \( \text{PEXP2} – \text{string} \) – is the name of the 2nd (optional) experiment. If no second experiment is selected, then a null string value must be passed, i.e., `''.  
3. \( \text{PVAR1} – \text{string} \) – is the name of the 1st (main) variable. If no valid variable value is given, a list of valid variable names will be printed out.\(^8\)  
4. \( \text{PVAR2} – \text{string} \) – is the name of the 2nd (optional) variable. If no second variable is selected, then a null string value must be passed, i.e., `''.  
5. \( \text{PT1} – \text{real (or integer)} \) – is the value of the 1st (main) time-slice. If no valid variable value is given, a list of valid variable names will be printed out.\(^9\)  
6. \( \text{PT2} – \text{real (or integer)} \) – is the value of the 2nd (optional) time-slice. If no second time-slice is selected, then enter `-1`.\(^{10}\)

Then there are 2 parameters for plotting sub-sets of the 2D or 3D data (essential for 3D data which cannot be usefully visualized in raw form):

---

\(^6\)Parameters can be in for form of strings, in which case they must be given as a series of characters enclosed in inverted commas `'`; as real numbers, e.g. `999.5` or `9.995E2`; or integers, e.g. `2`, `10`.

\(^7\)Or alternative directory if the default file path settings have been changed.

\(^8\)As a string, the value must be encased in inverted commas `''.

\(^9\)As `sedgem` does not save multiple and/or time-specific data, a dummy value (anything) is entered here.

\(^{10}\)As `sedgem` does not save multiple and/or time-specific data, a dummy value (anything) is entered here.
1. **PIK** – *integer* – varies in its interpretation and is discussed below.
2. **PMASK** – *string* – is the name of an optional (2D) mask. A null string (""') must be passed if no mask is requested. A file with the same name (plus an extension .dat) must exist in the directory MASKS\(^{11}\). The interpretation of this parameter differs slightly between functions (below).

Next come options for plotting scale control:
1. **PCSCALE** – *real* (or *integer*) – is the scale factor for the plot. For example, to plot in micromolar (umol kg\(^{-1}\)) units, enter; 1e-6. The plot is auto-scaled if a value of zero (0.0) is entered.
2. **PCMIN** – *real* (or *integer*) – is the minimum scale value.
3. **PCMAX** – *real* (or *integer*) – is the maximum scale value.
4. **PCN** – *integer* – is the number of (contour) intervals between minimum and maximum scale values.

Finally, there are 3 parameters for: specifying discrete (observed) data to be plotted (and analyzed against model projections), for specifying the plotting parameter file to be used, and for substituting an alternative filename for all the output:
1. **PDATA** – *string* – is the filename containing the an overlay data set, which must be formatted as separated columns. The precise number and type of columns varies between different functions and also the plotting options chosen, and are hence discussed later. The full filename of this file must be give, including any extensions (e.g. .dat, .txt). This parameter must be passed as a *string*; leave blank, i.e., "", for no overlay data.
2. **POPT** – *string* – is the *m*-file filename (excluding the .m extension) containing the plotting options (*SETTINGS*). This parameter must be passed as a string; leave blank, i.e., "", in order to load the default file (*plot_fields_SETTNGS*).
3. **PNAME** – *string* – is the string for an alternative series of output filenames. This parameter must be passed as a string, e.g., 'experiment2'. If an empty (i.e., '') value is passed to this parameter then the output filenames will be automatically generated.

The basic parameter list for all 4 plotting functions\(^{12}\) is hence:

\[
>> \text{FUNCTIONNAME(PEXP1,PEXP2,PVAR1,PVAR2,PT1,PT2,PIK,PMASK,PCSCALE,PCMIN,PCMAX,PCN,PDATA,POPT,PNAME);}\]

**Function specific interpretation of PIK and PMASK**

A note on the different behaviour of 2 of the passed parameters, depending on which plotting function is used – PIK, and to some extent, PMASK, have quite different interpretations depending on the particular plotting function used:

1. **plot_fields_biogem_2d**
   (a) **PIK** – is the maximum depth (\(k\)) level that will be plotted, i.e. all depth levels deeper than PIK will be excluded. This is useful for plotting a variable only for the 'deep' ocean (rather than the ocean overlaying all ocean depths) for example. This value also provides an alternative way of creating a mask, and only values of \(k\) less than or equal to the passed value will be plotted.

\(^{11}\)Or alternative directory if the default file path settings have been changed.

\(^{12}\)Note that for *plot_fields_sedgem_2d* several of the parameters are redundant but must still be included (typically as zeros). This is in order to retain a common parameter list format between all the different plotting functions.
(b) PMASK – is the name of an optional (2D) mask. A null string (""") must be passed if no mask is requested. (Shallow depths could also be excluded from the plot by means of a mask rather than setting PIK.)

2. plot_fields_biogem_3d_k
(a) PIK – the depth (k) level to be plotted. Note that the levels are numbered from a maximum value designating the surface, to 1 for the deepest ocean level. Typically, maximum values for the number of ocean levels are 8 (e.g. Ridgwell et al. [2007]) or 16 (e.g. Cao et al. [2009]).
Non ocean level k values have special meanings here:
   i. 0
      A zero will result in a water column integral being plotted. With data, the model-data is carried out on the grid as a whole.
   ii. -1
      Will result in the benthic surface being plotted.
(b) MASK – is the name of an optional (2D) mask. A null string (""") must be passed if no mask is requested.

3. plot_fields_biogem_3d_i
(a) PIK – the longitude-depth (i) slice through the ocean to be plotted.
Non longitude grid point i values have special meanings here:
   i. 0
      A zero will result in a zonal mean being plotted. With data, model-data comparison is conducted at the specific data locations, rather than vs. a zonal mean model value.
   ii. -1
      I have forgotten what this does ...
(b) MASK – is the name of an optional (2D) mask. A null string (""") must be passed if no mask is requested.
For example: if the mask is of the entire ocean (mask_worbe2_ALL.dat), the result is a global meridional cross-sectional mean.
If the mask is just of a single basin such as the Atlantic (mask_worjh2_Atlantic.dat), the result is the Atlantic meridional cross-sectional mean.
Masks can also be constructed that are only a single cell wide longitudinally, but which take a meandering path following an ocean transect\(^\text{13}\).
The trivial usage would be to construct a mask consisting of a vertical line of 1s – the result is equivalent to setting an appropriate i value in PIK.

4. plot_fields_sedgem_2d.m is an exception as it does not (currently) use either parameter. PIK must be entered as 0 (any integer will do in fact), and PMASK as "".

The mask itself (if PMASK contains a mask name) is a 2-D array of model grid points (on the BIOGEM grid) in the form of a simple ASCII file. A value of '1' represents a vertical column of ocean cells to include, whereas a value '0' will exclude all cells in the water column at that particular grid point. Examples of some masks can be found in the MASKS subdirectory of muffinplot.

\(^{13}\text{e.g., as in: mask_worjh2_GEOSECS_WATL.dat}\)
16.3.1 Basic usage

What follows are some basic and quasi random examples, just to illustrate a simple use of the three main plotting functions.

1. Surface ocean temperature

Surface ocean temperature can be plotted in 2 ways – via the 2d plotting function (but only if the surface tracer properties fields have been saved, as these are optional), or via the 3d plotting function.

![Surface temperature plot](image)

Figure 16.1: Example basic (default) surface temperature plot.

For example:

```matlab
>> plot_fields_biogem_2d ...
('EXP1','','ocn_sur_temp','',9999.5,-1,16,'',1.0,0.0,30.0,30,'','','example1a');
```

plots from the experiment EXP1, the variable ocn_sur_temp for time-slice 9999.5 (the mid-point time of the final year of a 10,000 year experiment). The color scale is from 0.0 to 30.0, with no re-scaling (1.0), and 30 color intervals in the scale. The default SETTINGS parameter file is used, and the default filename string replaced with example1a. The only other thing to note, is for parameter PIK, a value of 16 is set – corresponding to the ocean surface. See Figure 16.1.

Note that in this example, the variable ocn_sur_temp is assumed. However, the model results variable ocn_sur_temp is not always saved in BIOGEM 2D netCDF output. If the requested variable, such as ocn_sur_temp does not exist, or is mis-spelt, MATLAB will pause and provide a warning. It will then list all the variables in the netCDF file that it can find and wait for a new variable name to be inputted. For instance, a variable that is always saved in BIOGEM 2D netCDF output is atm_temp (surface air temperature) and could be substituted in the plot. Also note that time (the time-slice year to be plotted) is similarly handled – if the specific time-slice value does not exist, a list of all possible time-slice years are provided and a substitute value requested as MATLAB pauses and wait for your input.

To add contours, in the default SETTINGS parameter file (or a copied and re-named version thereof), adjust the following line:

```matlab
contour_plot = 'y'; % [ 'y'] OVERLAY CONTOUR PLOT?
```

The results of this are shown in Figure 16.2.
Refinements to the contouring can be done by changing the lines:

```matlab
contour_mod = 1; % [ 1] NUMBER OF COLOR INTERVALS PER CONTOR
contour_mod_label = 5; % [ 5] NUMBER OF LABELED CONTOURS PER CONTOR
contour_label = 'y'; % [ 'y'] LABEL CONTOURS?
contour_dashneg = 'n'; % [ 'n'] PLOT NEGATIVE CONTOURS DASHED?
```

(these are the more commonly used refinements).

Here: `contour_mod` determines how many color intervals per contour interval, which in the previous SST plot example, was 30. So a value of 1 will give you 30 contours – one every 1 degree C. And a value of 5 will give you 6 contours – one each 5 degrees C.

`contour_mod_label` then determines whether you want the contours labelled or not. The answer (parameter value) to be the character `y` or `n`, as a string (i.e. in inverted commas).

If you elect to have contour labels, `contour_mod_label` determines how frequently to label the contours. A value of 1 labels every single contour. A value of 2 labels every other contour.

So for instance, if you set `contour_mod=5` and `contour_mod_label=2` in the previous SST example, you get a contour every 5 degrees C, and a temperature label every 10 degrees C.

Alternatively, using 3d plotting, you could plot the ocean surface temperature field as follows:

```matlab
>> plot_fields_biogem_3d_k ...
(’EXP1’,’’,’ocn_temp’,’’,9999.5,-1,16,’’,1.0,0.0,30.0,30,’’,’’,’example1c’);
```

The main things that change here are firstly the variable name – now `ocn_temp`, and secondly because this is a 3D ocean field, we need to specify what ocean model level we want to plot – this is where the integer 16 comes in and corresponds to the input PIK, as discussed earlier. The resulting plot will be identical to Figure 16.1.

2. **Global zonal average temperature profile**

To keep with ocean temperature, we can use the `plot_fields_biogem_3d_i` function to plot the global zonal mean (lat-depth) profile (rather than horizontal, surface slice):

```matlab
>> plot_fields_biogem_3d_i ...
(’EXP1’,’’,’ocn_temp’,’’,9999.5,-1,0,’’,1.0,0.0,30.0,30,’’,’’,’example2a’);
```

The only significant change as compared to before, is setting a 0 for input parameter PIK (again – see earlier). The results is shown in Figure 16.3.

3. **Pacific dissolved oxygen profile**

As per choosing ocean levels (k-values) in the lon-lat plotting, you can also specify a specific longitude for creating a lat-depth section rather than calculating and plotting a global zonal
mean. e.g. Figure 16.4 was created by\textsuperscript{14}:
\begin{verbatim}
>> plot_fields_biogem_3d_i ...
('EXP1','','ocn_O2','',9999.5,-1,10,'',1.0E-6,0.0,300.0,30,'','','example3a');
\end{verbatim}

The chosen section is somewhere in the Pacific, along a line of longitude (whatever corresponds to $i=10$ on this \textbf{muffin} model grid ... I guess about 165W ...). Here, the variable to be plotted has also been changed – $ocn\_O2$\textsuperscript{15}. Because the units of dissolved oxygen are much smaller than for temperature (in degrees C), the plotted scale has also been changed – from 0 to 300 $\mu$molkg\textsuperscript{-1} rather than the netCDF variable units of molkg\textsuperscript{-1}. To achieve this re-scaling, a units scaling value of $1.0E-6$ is specified for parameter \texttt{PCSCALE}.\textsuperscript{16}

\begin{verbatim}
>> plot_fields_biogem_3d_i ...
('EXP1','','ocn_O2','',9999.5,-1,10,'',1.0E-6,0.0,300.0,30,'','','example3a');
\end{verbatim}

4. \textbf{Atlantic zonal mean dissolved oxygen profile}

So far, with the exception of plotting a gridded color field and a contoured field at the same time, all these examples can also be done in \textbf{Panopy}. One difference, is the ability in the \textbf{muffinplot} suite of \texttt{MATLAB} functions to apply masks – isolating geographical regions or even single points. In the \texttt{MASKS} directory, are a series of example ASCII mask files, mostly for the 2 (8- and 16-level ocean) modern published configurations of \textbf{muffin}. For instance, \texttt{mask_worj2\_AtlanticALL.dat} has all the grid points in the entire Atlantic basin assigned a value of 1, with 0 everywhere else. If we apply this first to the surface ocean dissolved oxygen field:

\textsuperscript{14}Also turning on the contour plotting.
\textsuperscript{15}You’ll need a biogeochemistry enabled base-config
\textsuperscript{16}Note that the scaling specified is as the new units relative to the old units – here, $\mu$molkg\textsuperscript{-1} relative to molkg\textsuperscript{-1} and hence $10^{-6}$. ALSO NOTE that \textbf{Panopy} does it the other way around ... :(
16.3 Spatial plotting

we obtain Figure 16.5.

Figure 16.5: Distribution of surface ocean dissolved oxygen in the Atlantic.

Here, it is clear how the masked has been applied and all the ocean falling outside of the mask is plotted as white (no data).

We can also apply a mask field to the zonal average plot:

we obtain Figure 16.6.

Figure 16.6: Mean zonal ocean oxygen profile in the Atlantic.

16.3.2 Example analysis

An set of MATLAB m-file functions are provided that define a series of different generic and basic experiment analysis and plottings, with make_analysis_ALL.m provided as a template for carrying them all out in one go. Obviously, the individual aggregate plotting functions can be edited, added to, or with unwanted or irrelevant plots, commented out or deleted – treat these all simply as templates for developing your own analysis strategy (as well as viewing the associated configuration files as illustrations of the function/use of some of the different further plotting options\textsuperscript{17}).

The aggregate plotting functions are as follows:

\textsuperscript{17}Also covered in a subsequent sub-sub-section.
1. **fun_make_analysis_phys.m**
   This encompasses a basic set of analyses of ocean circulation and climatology.
   The script is written as a function, and requires just two parameters to be passed as input:
   (a) The experiment name.
   (b) The (mid-point of the) year of the time-slice to plot.
   In the example of an experiment called 'EXP1', and plotting the last annual time-slice (9999.5) from a 10,000 year model run, the function is hence called:
   ```matlab
   fun_make_analysis_phys('EXP1',9999.5);
   ```

2. **fun_make_analysis_geo.m**
   This encompasses a basic set of analyses of ocean (abiotic) geochemistry and ocean acidification related variables and metrics.

3. **fun_make_analysis_bio.m**
   This encompasses a basic set of analyses of marine biological fluxes and biologically related properties.

4. **fun_make_analysis_ALL.m**
   Aggregates all the above functions (i.e., calls all 3).

To run these example analysis – either copy all the files contained in the EXAMPLES subdirectory, to your working directory (e.g. RESULTS as per the previous example). Then type e.g.,
```matlab
fun_make_analysis_ALL('EXP1',9999.5);
```
OR, add the path to the EXAMPLES subdirectory, e.g.
```matlab
addpath('Y:\_git\muffinplot\EXAMPLES');
```
but obviously depending on quite where you installed muffinplot.

### 16.3.3 Further refinements

A number of additional options for exerting finer control over the plotting are provided as a block of parameters and (default) values in the m-file itself, in a section immediately after the commented help and change-log at the start of the m-file. Not all the options are relevant to all the plotting functions, but the full list (and then defaults in brackets []) is as follows:

1. **lon_min = -180; [-180]** STARTING LONGITUDE FOR X-AXIS
   Sets the longitude of the left-hand edge of the plot.
2. **delta_lon = 90; [90]** INCREMENT OF LONGITUDE ON X-AXIS
   Sets the longitude tick increment.
3. **contour_plot = 'n'; ['n']** OVERLAY CONTOL PLOT?
   Overlay line contours on the color block plot?
4. **contour_mod = 2; [2]** NUMBER OF COLOR INTERVALS PER CONTOR
   Number of color graduations per line contour.
5. **contour_mod_label = 4; [4]** NUMBER OF LABELED CONTOURS PER CONTOUR
   Number of color graduations per labeled line contour.
6. **contour_label = 'y'; ['y']** LABEL CONTOURS?
   Label the line contours (frequency of labeled contours set by contour_label).

---

18Note that if you did not run with ocean biogeochemistry (but rather climate-only), not all the plotting functions will run and you will have to restrict this default analysis to: fun_make_analysis_phys('EXP1',9999.5);
19See 'help' on a specific plotting function for details of the relevant options in the parameter block.
16.3 Spatial plotting

7. contour_noneg = 'n'; [ 'n'] RESTRICT DATA PLOTTED TO > 0.0?
   Restrict the plotted values to non-negative? (Can be useful if slightly negative values exist as can
   occur during tracer transport associated with large concentration gradients.)
8. plot_log10 = 'n'; [ 'n'] PLOT LOG10 OF THE DATA
   Plot data values as log10(value)?
9. contour_zero = 'y'; [ 'y'] PLOT ZERO CONTOUR
   Plot the zero contour?
10. colorbar_old = 'n'; [ 'n'] PLOT 'OLD' COLORBAR
    Plot old style colorbar.
11. data_offset = 0.0; [ 0.0] data offset (273.15 for K -> C)
    Introduce a data offset? This is useful for example for converting K to degrees C (removing the K
    value of 0 degrees C).
12. data_i_j = 'n'; [ 'n'] DATA as (i,j)?
    Overlay data in the form of (i,j) locations rather than longitude,latitude?
13. data_i_j_k = 'n'; [ 'n'] DATA as (i,j,k)?
    Overlay data in the form of (i,j,k) locations rather than longitude, latitude, depth?
14. data_i_j_mean = 'n'; [ 'n'] average DATA by cell?
    Average overlay data per cGENIE grid cell rather than plotting raw locations.
15. data_i_j_k_mean = 'n'; [ 'n'] average DATA by cell?
    Average overlay data per cGENIE grid cell rather than plotting raw locations.
16. data_size = 25.0; [25.0] SIZE OF OVERLAY DATA POINTS
    Size of the overlay data points.
17. data_anomaly = 'n'; [ 'n'] PLOT AS MODEL-DATA ANOMOLY ONLY?
    Plot data locations with the model-data anomaly rather than data value?
18. data_only = 'n'; [ 'n'] PLOT ONLY DATA (no model values)?
    Plot only the overlay data locations (and not any model data)?
19. data_site = 'n'; [ 'n'] PLOT DATA AS SITES (no data values)?
    Plot labeled site locations (no data value fill).
20. plot_land = 'n'; [ 'n'] PLOT DATA OVER LAND?
    Plot data locations lying over land on the cGENIE grid (rather than screen out)?
21. data_uv = 'n'; [ 'n'] overlay (u,v) velocity data?
    Overlay ocean current fields.
22. data_uv_scale = 1.0; [1.0] scaling factor for vector length
    Scaling factor for velocity vectors.
23. plot_opsi = ''; [ '' ] PLOT OVERTURNING STREAMFUNCTION (basin)?
    Plot overturning streamfunction overlay?
24. plot_opsi_min = -15; [ -15 ]; plot_opsi_max = +15; [ +15 ]; plot_opsi_dminor = 1;
    Controls on min, max and (major and minor) contor intervals.
25. dscresz = 0.60; [0.60] FRACTIONAL FIGURE WINDOW SIZE
    Adjustment factor of the fractional size (compared to the screen) of the figure window.

Further refinements: Examples

Examples:

1. To plot the positions (and labels) of data locations:
   plot_fields_biogem_3d_k('cgenie_output','120926.SPIN',",49999.5,-1,'ocn_temp',",",
   16,1.0,10.0,40.0,30.0",'sites.dat')
   where the experiment name is 120926.SPIN, the mapped variable is ocn_temp (although no model field need be plotted – set by an option in the plotting function itself, and the file of data locations is sites.dat.
Figure 16.7: Paleocene-Eocene deep-sea sediment drill locations together with a contour-overlain map of surface temperature.

16.3.4 Time-series plotting
16.3.5 Sediment model output analysis
17.1 Data re-gridding

A number of MATLAB functions are available in migraine for transforming data:

- **make_regrid_data2ASCII_GENIE.m**
  Re-grids plain text (ASCII) point location data to a 2D muffin grid and saves as a 2D ASCII format file.

- **make_regrid_ASCII2netcdf_GENIE.m**
  Converts 2D muffin grid ASCII format data to netCDF format.

- **make_regrid_data2netcdf_GENIE.m**
  Re-grids plain text (ASCII) point location data to a 3D muffin grid and saves as netCDF format.

- **make_regrid_data2netcdf_WOA.m**
  Re-grids plain text (ASCII) point location data to the standard 3D World Ocean Atlas grid and saves as netCDF format.

- **make_regrid_WOA2GENIE.m**
  Converts a 3D World Ocean Atlas grid netCDF variable to the muffin grid and saves as netCDF format (as well as ASCII).
17.2 Miscellaneous data processing
17.3 Model ensembles

The muffindata MATLAB file: fun_make_ensemble_2d.m provides a simple/convenient way of generating ensembles of model experiments. By default, the function is designed for generating 2D ensembles of a first varying parameter vs. a second varying parameter. However, by specifying only a single value of the 2nd (or first) parameter, you’ll obtain a 1D ensemble of just a single parameter varying.

Information about how to configure and generate an ensemble is provided as MATLAB help on this function:

```matlab
>> help fun_make_ensemble_2d
```

The function fun_make_ensemble_2d.m requires 2 parameters (strings) to be passed in the argument list, i.e.:

```matlab
>> fun_make_ensemble_2d.m(PAR1,PAR2)
```

These are, in order:

1. STR_TEMPLATE – string → is the name of the template user-config file (within the muffindata directory) used to generate the ensemble.
   
   Note that the user-config may already contain a value for the varying parameter(s) – the automatically-generated parameter values will be appended at the very end of the file, hence over-writing any previous values.

2. STR_PARAMS – string → is the name of the parameter configuration file (omitting the `.dat` extension). The format of this file is plain text and takes a `.dat` extension. Example contents for this file are provided in the MATLAB help for this function.

An example example usage would be:

```matlab
>> fun_make_ensemble_2d.m('user_config_template','ensemble_params')
```

which creates an ensemble based on the user-config file: `user_config_template` (example file name) and with varying parameter values specified in the file: `ensemble_params.dat` (example file name).

The user-config (and hence experiment names) that are automatically generated end in `.xy`, where `x` represents the `x + 1th` parameter value on the first axis, and `y` represents the `y + 1th` parameter value on the second axis (the `.xy` notation starts counting at `.00`). For instance, `.00` would be an experiment taking the first parameter values specified in `ensemble_params.dat` for both axes. `.10` would have the 2nd lister parameter value on the 1st axis, and the 1st lister on the 2nd axis. `.27` would have the 3rd parameter value on the 1st axis, and 8th listed parameter value on the 2nd axis. etc etc etc

Also in muffindata you can find a BASH shell script for submitting the entire ensemble\(^1\) at the command line (within genie-main) – `sub_ens.muffin.sh`. NOTE: you have to specify the experiment parameters of your ensemble (e.g. base-config and user-config filenames, run duration, etc.) in the BASH file and you should be sure to have compiled the model (by running a brief experiment) with the required base-config prior to using `sub_ens.muffin.sh`.

The lines requiring editing in `sub_ens.muffin.sh` are clearly marked and these instructions are not repeated here.

---

\(^1\)Having first transferred all the user-config files to the cluster to an appropriate genie-userconfigs folder
17.3 Model ensembles

Note that you may need to re-assigned an 'executable' permission after editing and saving sub_ens.muffin.sh. To do this, from genie-main:

```plaintext
>> chmod a+x sub_ens.muffin.sh
```

What do you then 'do' with a whole bunch of experiments with names ending in .xy ... ?? (Note that you can take any series of experiments, and rename such that they all have exactly the same filename with the exception of an .xy format extension. i.e. a series of 5 experiments with whatever varying filenames could become: FILENAME.00, FILENAME.01, FILENAME.02, FILENAME.03, FILENAME.04, which would create a 1-D (1 × 5) ensemble.)

You could of course, pick out individual ensemble member to plot and analyse, referring to the ensemble_params.dat file where you specified the sequence of parameter values on both axes in order to identify the ensemble member you want. And/or ...

The muffindata MATLAB file: fun_process_ensemble_2d.m provides a simple/convenient way of processing and visualizing ensembles of model experiments.

Information about how to configure and generate an ensemble is provided as MATLAB help on this function:

```plaintext
>> help fun_process_ensemble_2d
```

The function fun_make_ensemble_2d.m requires 3 parameters to be passed in the argument list, i.e.:

```plaintext
>> fun_process_ensemble_2d.m(PAR1,PAR2,PAR3)
```

These are, in order:

1. PAR1 – string → is simply the ensemble name, but omitting the .xy numerical code of individual ensemble members).
2. PAR2 – number → is the year of model data to extract. e.g. 9999.5 for the last year of a 10,000 year long run.
3. PAR3 – string → is a name to assign to the output files (e.g. giving you a potentially more meaningful set of filenames than simply re-using PAR1.

Ensure that you either run fun_make_ensemble_2d.m (or rather, the renamed copy of it) from the muffindata directory, or add a path in MATLAB to it. Also add a path to muffinplot if you are going to process netCDF variables.

Within the code of fun_process_ensemble_2d.m are instructions/directions as to how to extract specific variables. More advanced usage can statistically compare to data and calculate the goodness of fit of all the ensemble members, and even automatically identify and further process the 'best' (depending on the specific criteria for 'best').

Unlike creating ensembles, fun_process_ensemble_2d.m needs to be directly edited (or rather: create a copy, rename the file and the function name at the top of the file, and edit that). The basic set-up is as follows and requires 4 steps (which are clearly marked in the code):

1. Define the x-axis of the ensemble, under:

```plaintext
% --- STEP #1 ----------------------------------------------------------- %
% define ensemble x axis
```

Here, the number of strings defined in the array (structure) defines how many different parameter values are represented in the x-axis of the ensemble. This may be just one (for a 1-D
parameter ensemble).

The strings are categorical and need not be numbers and are used to label the axis ticks (with one label corresponding to each different parameter in the 1st dimension of the ensemble). This is also an x-axis label to set.

2. Define the y-axis of the ensemble, under:

   ```
   % --- STEP #2 --------------------------------------------- %
   % define ensemble y axis
   As per above.
   Note that both x-axis and y-axis structures can contain a single string, in which case the ensemble is 0-D (1 x 1) and consists of a single experiment.
   ```

3. Define any time-series variables to be extracted and plotted, under:

   ```
   % --- STEP #3 --------------------------------------------- %
   % define time-series variables to extract and plot
   Here, any time-series variables to be extracted are defined.
   The format is:
   ```
   m = m+1;
   data(m).dataname = 'VARIABLE';
   data(m).datacol = COLUMN;
   data(m).scale = SCALE;
   data(m).dataunit = 'LABEL';
   data(m).minmax = [MINIMUM MAXIMUM];
   ```
   ```
   (The m = m+1; bit it important should should not be left out.)
   ```
   These settings are:
   ```
   - VARIABLE – Is the name of the time-series variable following the biogem_series_ part of the filename and excluding the .res bit at the end.
     For example, atmospheric \( p\text{CO}_2 \) would be specified by:
     ```
     data(m).dataname = 'atm_pO2';
     ```
   - COLUMN – Is the column number in the time-series file to be analysed.
     For atmospheric \( p\text{CO}_2 \), a value of 2 would analyse the total inventory in units of \( \text{mol} \), while column 3 would give the concentration in units of \( \text{atm} \).
   - SCALE – Scale the variable (by this factor)?
     For example:
     ```
     data(m).scale = 1.0E6;
     ```
     would multiply the time-series value by \( 10^6 \) and in the example of atmospheric \( p\text{CO}_2 \), produce the data in units of \( \mu\text{atm} \) (or \( \text{ppm} \)).
     The scaling need not be a simple multiple of orders-of-magnitude, and e.g.
     ```
     data(m).scale = 12.0E-15;
     ```
     will convert \( \text{molC} \) into \( \text{PgC} \).
   - LABEL – What you could like to appear alongside the scale bar ... traditionally the variable name (property displayed) and units.
   - [MINIMUM MAXIMUM] – The scale for the data range.
     For example, having scaled atmospheric \( p\text{CO}_2 \) to units of \( \text{ppm} \), this might be (depending upon the experiment):
     ```
     data(m).minmax = [200 1000];
     ```
17.3 Model ensembles

A grid plot, along with a plain text file of the values, will be created for each defined variable. Missing experiments, variables, or time-points, will appear as a white square in the grid plot for that particular ensemble member.\footnote{So the ensembles need not be complete, or all the runs finished successfully.}

The format of the output filename is: OUTPUTNAME.VARIABLENAME.COLUMN.ps, where OUTPUTNAME is the string (3rd parameter) passed to the function (and which need not be the full experiment name), VARIABLENAME is the name of the analysed variable as defined by data(m).dataname (see above), and COLUMN is the column of the time-series file analysed, as defined by data(m).datacol (also see above).

Be careful – the plain text file (OUTPUTNAME.VARIABLENAME.COLUMN.dat) of values is up-side-down as compared to the orientation of the plot.

4. Define any time-slice (netCDF) variables to be extracted and plotted.

This requires 2 parts:

\%
\% --- STEP #4a ---------------------------------------------------------- %
\% define netCDF variables to extract and plot/analyse

Here, the format is identical to step #3 (don’t forget the m = m+1; bit!), with the exception of the variable name definition:

\texttt{data(m).dataname} = 'mean\_Pacific\_oxygen';

which is not used except in defining the output filename, and the column:

\texttt{data(m).datacol} = 0;

which is also irrelevant and can be assigned a value of e.g. zero.

The second part:

\%
\% --- STEP #4b ---------------------------------------------------------- %
\% extract and plot/analyse netCDF variables

is a little more complicated, and very much depends on exactly what you ’want’ from the 2- or 3-D netCDF that can be represented as a single value.

Common is to conduct some sort of statistical goodness-of-fit analysis of a model variable against observations, or perhaps the same variable in a different model experiment. One might also be interested in statistical or otherwise calculated summary properties of a 3-D field (or fields), such as mean, minimum or maximum values, or even over-turning circulation strength – anything passed back as a single value from the \texttt{muffinplot} plotting functions can be visualized across the ensemble. Code can also be added as and when required, for further analysis of combinations of outputs.

Some examples are given in \texttt{help} for this function.

Missing experiments will be represented a white square in the output plot. However, the \texttt{muffinplot} functions are less forgiving when it comes to missing or incorrect variable names and time-slice times, and everything will stop until you provide a value input at the command-line.
18.1 muffingen basics

18.1.1 overview of the muffingen software

**muffingen** is a collection of MATLAB functions that can create all the configuration files (excluding optional climatic and biogeochemical forcing fields) required by the cGENIE (**muffin**) Earth system model. **muffingen** is designed to take the output from a fully coupled GCM, particularly of past climates with different continental configurations, and re-grid the output needed by the **muffin** model, in the form of files of boundary conditions, all saved in their respective correct format. However, **muffingen** can also be used to draw conceptual alternative Earths (in terms of continental configuration). Again, saving the required **muffin** files in their appropriate formats.

As a MATLAB code, (only) very basic familiarity with using MATLAB at the command line is required. And a copy/license of the MATLAB software … A sufficient grasp of MATLAB can be gained by going through the following sections/subsections of the *matlabananas* textbook, which can be found on GitHub (look for the PDF file compiled form the latex source – BANANAS.pdf):

- Section 1.1 – The MATLAB interface and command line.
- Section 1.6 – Changing directories and file search paths.
- Section 2.2 – Using functions ... of which muffingen is one.

18.1.2 Installing muffingen

The **muffingen** code is hosted on GitHub:

```
https://github.com/derpycode/muffingen
```

There are 2 ways to get your mitts on the **muffingen** code:

1. By downloading an archive file, containing all the code etc. For this – click on the green Clone or Download button, and select Download ZIP.

You then unpack/unzip the files and directory structure where you want it. This [archive download] is a perfectly workable way to proceed¹ ...

2. Or you can clone the repository to where you intend to run **muffingen**. Note that you will need a git client installed on your computer. There are GUI clients for git, or this can be done at the command line:

```
$ git clone https://github.com/derpycode/muffingen.git
```

By doing this, you have created your own code repository (and an identical copy of the one hosted on GitHub). As part of the git clone command, you also automatically check out (from your very own personal repository) a copy of the code.

Note that you download or clone **muffingen** to the computer that you have MATLAB installed on and will use to run **muffingen** (i.e. not necessarily to the computer where **muffin** itself is run).

---

¹Note that this way, you will be unable to easily update the code with whatever new developments or bug fixes occur in the future, nor can propagate back any code changes that you might have made and might want to become part of the official **muffingen** code (i.e. downloading the ZIP file becomes a one-off installation that loses its formal connection to the GitHub repository).
18.1.3 Running muffingen

To use the cGENIE.muffin model configuration generator muffingen – at the command line in MATLAB, simply type:

```matlab
>> muffingen('FILENAME')
```

where FILENAME is the name of an ASCII format configuration file (having a .m filename extension) that specifies the required settings (see subsequent section).

This file (FILENAME.m), must either be present in the directory where you installed muffingen (the directory where e.g. where the file muffingen.m is located) e.g. C:\muffingen. In this case, your MATLAB working directory must also be set to C:\muffingen. Or, the MATLAB working directory and the file FILENAME.m can reside elsewhere (both directory locations must be the same however). In this case, you will then 'add' the location of muffingen to MATLAB's search path by the MATLAB function addpath.

The muffingen model configuration generator then starts, and depending on the specific settings in the configuration file, may require no user input, or may require user input, either because this option was requested, or because a re-gridding issue arose that requires manual intervention to resolve. A series of plots are created (and saved) as the configuration generation progresses together with the muffin model configuration files themselves. All the various steps plus details of how the contents of the configuration files are generated are reported at the command line, and saved in a n ASCII format *.log file for future reference.

Depending on the specific configuration file settings (see later), muffingen has 5 main modes of operation, which are summarized as follows (and described in more detail, along with specific examples, below):

1. **Configuration derivation based on re-gridding climate output from a GCM.**
   The most common usage of muffingen, and enabling a new (typically paleo) configuration to be derived from the output of a GCM experiment. Currently options for utilizing 4 different GCMs are provided: HadCM3(l) ('hadcm3' or 'hadcm3l'), FOAM ('foam'), CESM ('cesm'), and ROCKEE-3D ('rockee').

2. **Derivation based on re-gridding a high resolution input topography.**
   In this option – 'mat' – a land-sea mask and ocean bathymetry are re-gridded from a high(er) resolution .mat (MATLAB) topography file. Zonal wind-stress and speed, and zonal planetary albedo, are prescribed.

3. **Derivation based on an existing model topography ' .k1' file (or simplified .k2) format variant).**
   This option – 'k1' or 'k2' – allows a topography to be re-created, or adapted/ altered, all directly from an existing model ‘k1’ file (and hence existing model configuration). Zonal wind-stress and speed, and zonal planetary albedo, are prescribed.

4. **Derivation based on a prescribed land-sea mask.**
   This option – 'mask' – will create a new configuration from any specified land-sea mask, whether 'real' or completely hypothetical. Zonal wind-stress and speed, and zonal planetary albedo, are prescribed.

5. **From a blank (all ocean) initial template.**
   Finally, this option – '' – enables a topography to be drawn by hand within muffingen and

---

2Refer to the MATLAB 'bananas' text for the syntax for using addpath.
hence represents an interactive alternative to (4) (rather than editing a land-sea mask in a text editor). Zonal wind-stress and speed, and zonal planetary albedo, are prescribed.

Also depending on the specific options, and particularly whether or not \texttt{opt\_user} is set to \texttt{true}, and editor window will open, firstly for changes to the land-sea mask to be made, then later for the ocean bathymetry to be edited. Finally, a window will open allowing 'island paths' to be edited. The latter may be needed if \texttt{muffingen} cannot unambiguously determine them unaided.

Once you have successfully run \texttt{muffingen}, copy/transfer the entire configuration subdirectory (the directory with the same name as whatever you called your 'world') and its contents, from the \texttt{muffingen} output directory (e.g. \texttt{muffingen/OUTPUT.EXAMPLES}) on your local computer, to: \texttt{cgenie.muffin/genie-paleo}. 
18.2 Configuring muffingen – overview

When muffingen is run, a configuration file with a specified filename\(^3\) is loaded. The configuration file is in a simple plain text (ASCII) format, but is given a .m extension, enabling the values of a number of controlling parameter values to be set directly in MATLAB.\(^4\) The configuration file parameters control facets of muffingen behavior such as the primary model of operation, input and output filenames, what types of configuration files you want to generate, as well as there being a number of parameters controlling the finer details of re-gridding and configuration file generation, including whether to enable user-input or not.

The configuration files can be edited with the MATLAB editor (indeed, as a .m file, they are inherently a MATLAB format file), or any plain text (ASCII) editor. They must have a .m filename extension. The configuration file is divided up into a series of sections of different parameter options. The main (most commonly used) parameters are summarized as follows (and are more fully described later):

\[
% *** CONFIG NAME AND MAIN INPUT SETTINGS ****************************
par_wor_name
Defines the name of the model configuration. This must be a string, 8 characters long.
\textit{e.g.} par_wor_name='my_world'
par_gcm
Defines the format of the input. The string is blank for an interactive user-defined world, i.e. par_gcm=''
par_expid
Defines the name of the GCM experiment if a GCM input is selected, or the '.k1' or '.dat' files if those respective input formats are selected by the above option. By default, these files live in the INPUT.EXAMPLES sub-directory.
\]

\[
% *** FILE PATHS ****************************************************************
% *** GCM netCDF FILENAMES *****************************************************
% *** GRID RESOLUTION *********************************************************

- par_max_i
  Defines the number of grid points in the longitude ('i') direction.
  This is typically 36 or 18, \textit{e.g.} par_max_i=36
- par_max_j
  Defines the number of grid points in the latitude ('j') direction.
  This is typically 36 or 18 and is typically the same number as for the i direction, \textit{e.g.} par_max_j=36
- par_max_k
  Defines the number of layers in the ocean circulation model.
  This is almost always either 16 or 8, \textit{e.g.} par_max_k=16
- opt_equalarea [default: true]
  This specifies whether or not an equal area grid is used/assumed. (A value of false results in the latitude grid being defined in equal increments of latitude).
\]

---

\(^3\)i.e. the single parameter passed to muffingen when it is invoked at the command line.

\(^4\)Note that the parameter filename is passed as a string without the .m extension (which is implicitly assumed). An error message will be generated if the file does not exist or has the incorrect extension.
% *** REGRIDDING SETTINGS ************************************************************
par_max_D [default: 5000.0]
Sets the maximum ocean (scale) depth (in m).
See subsequent sub-sub-section on 'Ocean depth (and maximum levels)' for further
details and usage.
par_sed_Dmin [default: 1000.0]
par_sed_Dmax [default: 5000.0]
Together, set the minimum and maximum depths used in generating a random seafloor
topography for SEDGEM (when par_sedsopt=2 is selected).
Note that while 5000m corresponds to the default deepest depth of the ocean circulation
model, about 15% of the modern seafloor lies below this (and mostly shallower than
6000m). So selecting 6000.0 would arguably enable a better simulation of e.g. the
CCD.
% *** CONFIG NAME AND MAIN INPUT SETTINGS ***************************************
opt_user [default: true]
Determines whether muffingen pauses and allows user-modification of land-sea mask
and ocean depth (and island paths).

18.2.1 muffingen configuration examples
A series of example configurations are provided in the main muffingen directory. These illustrate
parameter settings and the primary ways of using muffingen. They also serve as useful template
parameter setting files. They are:

- EXAMPLE_BLANK – Starts from a blank (‘water-world’) all-ocean land-sea mask template.
  User input (allowing continents and seafloor topography to be ‘drawn’) is automatically acti-
vated whether you want it or not!\(^5\)
- EXAMPLE_BLANK_deep – As above, except with a deeper ocean – 18 levels with the first 16
  spanning 5000.0 m.\(^7\)
- EXAMPLE_MASK_draceworld – The first of a series of 4 conceptual worlds (loosely following
  the literature). This starts from a land-sea mask (wordrake.dat) which is stored in the muffin-
gen subdirectory INPUT.EXAMPLES. The land fraction is minimal (as it is trying to reproduce
the sort of zero-area numerical barrier used in the literature). This configuration has a barrier
to ocean circulation, from the N pole down, with a high southern latitude gateway (a Drake
Passage like feature).
- EXAMPLE_MASK_eqpasworld – As above, except with an equatorial (only) gateway.
- EXAMPLE_MASK_ridgeworld – As above, except with no gateway.
- EXAMPLE_MASK_waterworld – As above, but with no barriers, i.e. a ‘water world’.
- EXAMPLE_GCMFOAM_modern – Takes the continental configuration and climate simulation
  output from a fully coupled GCM experiment (files in directory GCMFOAM in the muffingen
  subdirectory INPUT.EXAMPLES ) and derives full muffin model boundary conditions. User
  input (allowing continents and seafloor topography to be ‘edited’) is activated by default.

\(^5\)Because it assumes that you are going to go on and draw something ...
\(^6\)Edit the muffingen code to be able to turn off.
\(^7\)Refer to subsequence explanation re. the details of these parameters.
18.2 Configuring muffingen – overview

- **EXAMPLE_GCMHadCM3L_modern** – Similar to the above – takes the continental configuration and climate simulation output from a fully coupled GCM experiment (files in directory GCMHadCM3L in the muffingen subdirectory INPUT.EXAMPLES) and derives full muffin model boundary conditions. User input (allowing continents and seafloor topography to be 'edited') is activated by default.

- **EXAMPLE_K1_permian** – Starts from a muffin model configuration 'k1' topography defining file: p0251b.k1, which is stored in the muffingen subdirectory INPUT.EXAMPLES. The k1 file defines a late Permian continental configuration (and bathymetry). User modification is enabled by default (so MATLAB pauses for modifications to be made), but this can be disabled (opt_user=false).

- **EXAMPLE_K2_supercontinent** – Starts from a simplified version of the muffin model configuration 'k1' topography defining file – here given the extension 'k2'. This (wppcont1.k2) is stored in the muffingen subdirectory INPUT.EXAMPLES. The k2 file defines an idealized pole-to-pole super-continent. User modification is enabled by default (so MATLAB pauses for modifications to be made), but this can be disabled (opt_user=false).

- **EXAMPLE_K1_SEDGEMonly** – Similar to **EXAMPLE_K1_permian** – starts from a muffin model configuration 'k1' topography defining file: p0251b.k1, but only generates the SEDGEM sediment model files (and nothing else).

- **EXAMPLE_MAT_present_bathymetry** – Takes as input a high resolution topography in a MATLAB .mat file format. In this specific example, the file (subdirectory INPUT.EXAMPLES) inputted is present_bathymetry (.mat) which describes at 1° resolution, the modern plate configurations plus seafloor topography (as negative height). If you want anything other than a 1° resolution (or a different longitude origin) then you need to edit the code:

```matlab
% NOTE: assume equal Dlon, Dlat grid (non equal area) and 1 degree
% also that the input grid starts at -180E
% >>> EDIT ME >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

gi_lonce = [-180.0:1.0:180.0];
gi_latce = [-90.0:1.0:90.0];
% <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

Non-rectilinear grids are not supported. If you have 3 poles in your grid ... too bad.

---

8The specific grid resolution is defined, and the file loaded and the grid orientation assumed, in the LOAD GRID (AXES) DATA and LOAD TOPO & MASK DATA sections. Different resolutions and orientations (and file formats) can be dealt with here by editing the code.
18.3 Pre-generated worlds

A series of conceptual (‘fake’) worlds have been generated and are provided for you convenience:

1. Conceptual worlds
   These include ‘water world’, ‘ridge world’, ‘drake world’, etc. and associated variants, e.g. as published in fully coupled GCMs. They are designed to test controls on large-scale ocean circulation and do conform to an Earth-appropriate ocean volume or total cratonic area.

2. Idealized-continent worlds
   These involve idealized super-continental configurations but aim to retain modern ocean volume and total cratonic area characteristics.

Both genie-paleo configuration file sets (sub-directories), and example base-configs are provided as part of the muffin code base. The details of these two series of fake worlds are as follows. Remember that the .m file configurations that create each of the fake world configurations are saved in the respective genie-paleo sub-directory, and hence both the specific details of how the world was created are available, as well as the ability to exactly recreate the world.

18.3.1 Conceptual worlds

18.3.2 Idealized-continent worlds

The idealized continents come both without (i.e. the configuration is either land, or deep ocean) or with, continental shelves, and in the latter case, come with varying number of ‘steps’ down to deeper water (and hence to the abyssal plain).

The fake world series ‘k1’ configuration names start with one of two, 4 character strings:

- fkh_ == ‘high’ resolution ($36 \times 36 \times 16$)
- fkl_ == ‘low’ resolution ($18 \times 18 \times 16$)

(No low vertical resolution, e.g. 8 level version is currently provided.)

The next 2 characters delineate the shape/position of the world:

- 1e == a single Equatorial-centered super-continent
- np == a single North polar-centered super-continent
- pp == a pole-to-pole super-continent

The 7th character is the ‘series’ represents:

---

Example of aquaplanets and drake worlds in the MITgcm with simplified atmosphere:
18.3 Pre-generated worlds

- 0 == An earlier generation of worlds, with shelves stepping down in increments of only 1 ocean level.

  Note that in generating the wind-stress in this series, the presence/absence of high latitude gateways is automatically identified (meaning that the zonal wind-stress profiles differ between \( np \) and \( pp \) in the magnitude of wind-stress in the Southern Hemisphere) – \( \text{par\_tauopt}=0 \).

- 1 == A subsequent generation, with shelves stepping down in increments of 2 ocean levels.

  Note: An intermediate wind-stress magnitude is applied to both hemispheres in all configurations (i.e. no distinction is made between hemisphere with or without high latitude gateways) – \( \text{par\_tauopt}=3 \).

- 2 == A modification of #1, generated with a newer version of \textit{mufflingen}, and with an imposed limit of 1000 – 5000m for the range of \textit{SEDGEM} open ocean depths (the previous version included depths extending to the shallow sub-surface).

  Note that \( \text{par\_tauopt}=0 \) is used.

  Also note that in \( pp \), shelves are only created on the Eastern margin of the basin so as to retain consistency in total shelf area with \( np \).

The 8th character (digit) specifies the number of shelves, which for the initial series is:

0. (none)
1. level 15
2. level 15 + 14
3. level 15 + 14 + 13
4. level 15 + 14 + 13 + 12
5. level 15 + 14 + 13 + 12 + 11

and for the newer series:

0. (none)
1. level 15
2. level 15 + 13
3. level 15 + 13 + 11
4. level 15 + 13 + 11 + 9

All the oceans are ca. 3500m deep, i.e. truncated at ocean level 3, in order to produce approximately the present-day ocean volume.\(^{10}\)

Example \textit{base-configs} are provide for some, but not all of the idealized worlds, and follow the same naming convention and have full filenames of the form: muffin.CB.********.BASES.config, where "********" is the 8-character fake world ‘k1’ configuration name (see above). A basic set of tracers is provided (BASES).

Other than the rate at which the shelves step down\(^{11}\), the main difference between the different generations of worlds, is in the applied zonal wind field strength. In the earlier fake world series, the ‘zonal wind-stress generation option’ (\text{par\_tauopt}) was 0. This attempts to automatically determine the presence of high latitude ocean gateways and if they exist, apply a stronger zonal wind field in that hemisphere (otherwise, a wind stress appropriate for a modern northern-hemisphere configuration with no zonal gateway is applied). To reduce the number of boundary conditions that change and co-vary between different fake worlds, in the second generation of idealized continent worlds, an intermediate strength zonal wind field is applied to all, regardless of the presence of

\(^{10}\)To change this, simply search-and-replace the 3 in the .k1 file with e.g. 1.

\(^{11}\)(and that the earlier generation was for a pole-to-pole super-continent only)
absence of high latitude circumpolar ocean gateways. This is 'zonal wind-stress generation option' 3. Note that for the same resolution, wind-fields can be substituted/replaced, and using the saved .m configuration file, worlds can be re-generated with different wind-stress options if desired. The depth range of the ocean floor in SEDGEM (which informs the pressure used in the CaCO$_3$ stability calculation) is slightly restricted in #3 (the upper limit increased to 1000m).

18.3.3 Modern worlds

To both help illustrate how the configuration files are organized and accessed for paleo worlds, and to help make modifying older (modern) configurations easier, the 2 basic modern configurations are provided in a genie-paleo format:

- p_worbe2 == The 8-level 36 × 36 ocean configuration of Ridgwell et al. [2007] and including the geological carbon cycle (SEDGEM+ROKGEM) configuration files of Ridgwell and Hargreaves [2007] (and Lord et al. [2015]).
- p_worjh2 == The 16-level 36 × 36 ocean configuration of Cao et al. [2009] and including the geological carbon cycle (SEDGEM+ROKGEM) configuration files of Archer et al. [2009] (and Winkelmann et al. [2015]).

Corresponding example base- and user-config files are provided:

- muffin.CBSR.p_worbe2.BASES.config plus: muffin.CBSR.p_worbe2.BASES.SPIN1 (in: genie-userconfig/EXAMPLES), which runs the first-stage spin-up described in Ridgwell and Hargreaves [2007] (and also used in Lord et al. [2015]).
- muffin.CBSR.p_worjh2.BASES.config plus: muffin.CBSR.p_worjh2.BASES.SPIN1 (in: genie-userconfig/EXAMPLES), which runs the first-stage spin-up used in Archer et al. [2009] and Winkelmann et al. [2015].

To make modified (e.g. wind fields) versions of either of the 2 modern 'paleo' format configurations:

1. Copy and rename the (8- or 16-level ocean configuration) genie-paleo sub-directory, giving it an 8 (or 6) character string name.
2. In this new sub-directory, edit the files (wind stress, wind velocity, wind speed, albedo, and/or topography) you want to modify. (These files do not need to be renamed.)
3. Copy and rename the corresponding base-config file (disabling the parameter settings for SEDGEM and ROCKGEM, and/or enabling ECOGEM, as required).
   Then, in the base-config file, edit all instances of the genie-paleo sub-directory parameters to match the new genie-paleo sub-directory you assigned in step (1). (Parameters, e.g.: ea_1=‘../../cgenie.muffin/genie-paleo/p_worbe2’.
4. Copy-rename the example user-config, or modify your own existing one.
   Note that when employing the geological carbon cycle, the resolution of the sediment (and weathering) grids and associated boundary condition files are now defined in the base-config rather than the user-config (as originally done).
18.4 Configuring muffin experiments

Having created a new muffin configuration using the MATLAB function muffingen.m, you are now going to need to create 2 files before you can run any experiments:

1. A *base-config*, which includes the parameter settings of your new world.
   
   There will be 4 steps involved in this:
   
   (a) Choosing a template *base-config* file, which specifies appropriate physics parameters.
   (b) Copy-pasting a list of parameter settings that has been automatically-generated by **muffingen** into the template *base-config* file.
   (c) Defining the number of dissolved and particulate tracers in the ocean (and gases in the atmosphere) that you want and entering this into the *base-config*.
   (d) Ensure that the appropriate science modules are enabled in the *base-config*.

2. A *user-config*, whether simply to test the configuration or to provide the basis of a series of experiments.

   Templates are provided and more detailed instructions follow. However, note that if you have similar *base-config* and *user-config* files already, then you can simply adapt these. Another option is to utilize *base-config* and *user-config* files associated with any of the published experiments listed in genie-userconfigs/MS, or any of the described EXAMPLE.* example configurations.

   Note that steps 1c, 1d, and 2 are not independent of each other and need to be consistent – e.g. if you define Fe co-limitation of biological productivity in step 2, you need to have selected the appropriate tracers in step 1c. And if you specify biological productivity calculated by ECOGEM in step 2, you need to have selected the ECOGEM module in step 1d.

   Finally...

   3. You will need to ensure that all file are in their correct location in the directory structure.

---

1a. Choosing a template *base-config*

If you do not want to adapt an existing *base-config* (see suggestions above) or want to start ‘fresh’, a pair of alternative template files are provided with **muffingen** (copy and rename the file, and then edit). The template files are missing the definitions of the world settings and tracers, which will be rectified in steps 1b and 1c, respectively. They also only specify a basic set of science modules (addressed in step 1d). Otherwise, they differ only in the tunable physics parameter values, under the heading:

```plaintext
# *******************************************************************
# PHYSICAL CLIMATE CALIBRATION
# *******************************************************************
```

The files are:

- **BASECONFIG.08lvl.config**
  
  This is based on the physics calibration as used in e.g. *Ridgwell et al. [2007]* and *Ridgwell and Hargreaves [2007]* with the parameters tuned for an 8-level ocean modern world (specifically: *worbe2*). For fake and paleo worlds with an 8-level ocean, this is the recommended template.\(^\text{12}\)

\(^{12}\)If you want exactly the same physics as per e.g. *Ridgwell et al. [2007]; Ridgwell and Hargreaves [2007]*, use: cgenie.eb_go gs_ac bg.worbe2.BASES.config as a template.
The differences are:

- The ocean starts warmer (5°C)\(^{13}\) than the default (0°C). This helps the ocean circulation come to an equilibrium state more rapidly, particularly for greenhouse climates:

  ```
  # temp0 -- start with a warm ocean
  go_10=5.0
  # temp1 -- start with a warm ocean
  go_11=5.0
  ``

  Note that both parameters should be changed together (one sets the Northern Hemisphere temperature, and one the Southern ... or something ...).

- There is no (salt / freshwater) flux adjustment, enacted by setting the scaling parameter to zero:

  ```
  # SclFWF -- scale for zero freshwater re-balancing
e_a_28=0.0
  ```

  Note that the built-in (salt / freshwater) flux adjustment is designed to maintain a strong AMOC and the code is specific to the original modern continental configurations (e.g. worbe2, worjh2). A (salt / freshwater) flux adjustment using this parameter and associated code should never be used for paleo or fake worlds.\(^{14}\)

- The sea-ice diffusivity parameters are adjusted for improved stability\(^{15}\):

  ```
  # reduced sea-ice eddy diffusivity
gs_11=1000.000
  # set a fractional sea-ice coverage threshold for preventing advection
gs_par_sica_thresh=0.99
  ```

- Insolation forcing is seasonal:

  ```
  # set seasonal cycle
e_a_dosc=.true.
go_dosc=.true.
gs_dosc=.true.
  ```

  Note that all the respective parameter for all three physical climate components (ocean circulation model, sea-ice model, EMBM atmosphere) needs to be set.

  Note that the original tuned 8-level ocean circulation model configuration had no seasonal cycle (.false.).

- An additional option is provided for toggling between isopycnal/diapycnal (.true.) and horizontal/vertical (.false.) mixing schemes:

  ```
  # it is recommended that it is turned OFF (=.false.) for 'fake' worlds
go_diso=.true.
  ```

  In all modern and 'realistic' paleo configurations, isopycnal/diapycnal mixing is used. However, this scheme can lead to unwanted negative tracer values. In extreme biogeochemical configurations, particularly at low atmospheric \(pO_2\) and when sharp horizontal transitions in redox state in the ocean occur, the magnitude of negative values can become unacceptable.

  Work with fake worlds, where no ocean like that may have ever existed on Earth and

---
\(^{13}\)These parameters could potentially be increased further (to say ... 10°C).

\(^{14}\)Instead, a salt forcing in the desired deep-water formation location, balanced by a freshwater flux elsewhere, can be defined.

\(^{15}\)The original parameter setting was:

\(gs_11=6200.000\)
hence there is no specific ocean circulation pattern to try and reproduce, the recommendation is to use the horizontal/vertical (.false.) mixing scheme.

• **BASECONFIG.16lvl.config**

  This is based on the physics calibration as used in e.g. Cao et al. [2009] with the parameters tuned for a 16-level ocean modern world (specifically: worjh2). For fake and paleo worlds with an 16-level ocean, this is the recommended template.\(^\text{16}\)

  All the same changes as per for the 8-level ocean circulation model based configuration – above – are set:
  
  - The ocean starts warmer (5°C).
  - There is no (salt / freshwater) flux adjustment.
  - The sea-ice diffusivity parameters are adjusted for improved stability\(^\text{17}\).
  - Again as per for the 8-level ocean template *base-config*, an option is highlighted for toggling between isopycnal/diapycnal (.true.) and horizontal/vertical (.false.) mixing schemes:
    
    # it is recommended that it is turned OFF (=.false.) for 'fake' worlds
    go_diso=.true.

  As per for the 8-level ocean configuration, it is recommended to implement simple horizontal/vertical (.false.) mixing.

  Finally, it should be noted that compared to the Cao et al. [2009] configuration, there is no modification of atmospheric diffusivity over the Southern Ocean and Antarctica, which was added to improve the seasonal sea-ice properties in the Southern Ocean in the worjh2 modern configuration.\(^\text{18}\)

\(^\text{16}\)If you want exactly the same physics as per e.g. Cao et al. [2009], use:

cgenie.eb_go_gs_ac_bg.worjh2.BASES.config as a template.

\(^\text{17}\)The original parameter setting was:

gs_11=3573.718

\(^\text{18}\)The parameters for this were:

# diffusivity scaling factor
ea_diffa_scl=0.25

# grid point distance over which scalar is applied (j direction)
ea_diffa_len=3

but by default, reduced atmospheric diffusivity is disabled.

1b. Defining your 'world' in the *base-config*

You are now going to copy-paste in the block of parameters that defines the *muffingen*-generated world.

In the template *base-config* there is a highlighted (<<< >>>) line in the file:

```
# ****************************************
# GRID & BOUNDARY CONDITION CONFIGURATION
# ****************************************
# insert the automatically generated muffingen parameter list here
# ****************************************
# <<< >>>
# ****************************************
```

\(^\text{16}\)
Copy and paste the contents of the muffingen output file: config_yymmdd.txt into the template file where indicated — immediately above, immediately below, or simply replacing the entire <<< >>> line.

1c. Defining tracers in the base-config

As per for defining the specific world, there is an explicit section of the template base-config that needs to be edited:

```plaintext
# *******************************************************************
# TRACER CONFIGURATION
# *******************************************************************
# the total number of tracers includes T and S
# T and S do not need to be explicitly selected and initialized
# *******************************************************************
# Set number of tracers
GOLDSTEINNTRACSOPTS='$(DEFINE)GOLDSTEINNTRACS=2'
# list selected biogeochemical tracers
# <<< >>>
# list biogeochemical tracer initial values
# <<< >>>
# *******************************************************************
```

The template base-config files come with just 2 ocean tracers assumed — temperature and salinity (i.e., there is no carbon cycle or ocean nutrients etc. enabled at this point). These are implicit and essential to ocean circulation and hence are not explicitly listed. However, the count of total number of tracers in the ocean (which determines the compiled tracer and tracer-related array size) includes them, i.e.

```plaintext
# Set number of tracers
GOLDSTEINNTRACSOPTS='$(DEFINE)GOLDSTEINNTRACS=2'
```

So how to fill out the list of tracers? A complete list of all gaseous (in the atmosphere), dissolved (ocean only), and particulate (ocean and also sediments) tracers can be found in the files: tracer_define.atm, tracer_define.ocn, and tracer_define.sed, respectively (in directory genie-main/data/input). However, starting from scratch and knowing which ones to add to create a consistent and sufficient set is not trivial. Instead, and as before, you could take a published configuration from one of the genie-useroncigs/MS subdirectories (refer to the base-config employed in the README file) or an EXAMPLE.

Either copy-paste the entire # TRACER CONFIGURATION section, or individually update the three subsections (or copy-paste the entire block, then edit):

1. # list selected biogeochemical tracers
   First list (set equal to .true. all the gaseous (in the atmosphere), dissolved (ocean only), and particulate (ocean and also sediments) tracers.

2. # Set number of tracers
   Count up how many ocean tracers there are, and add 2 for temperature and salinity (which you do not need to explicitly list), and update (the value of 2) in the line:
   ```plaintext
   GOLDSTEINNTRACSOPTS='$(DEFINE)GOLDSTEINNTRACS=2'
   ```

---

19In the filename, yymmdd is the date of the configuration creation.
3. # list biogeochemical tracer initial values

Then, if you wish for any of the gaseous and dissolved (there is no initialization option for particulate/solid tracers) not to be initialized at zero, list their initial values.

Units of gas partial pressure are atmospheres \((\text{atm})\). Units of dissolved tracers in the ocean are \(\text{mol kg}^{-1}\).

Note that if you employ a re-start, then these values are over-written.

A series of template tracer lists, taken from published configurations, are also provided as part of the muffingen release, and are as described below:

- **TRACERCONFIG.ABIOTIC.txt**

  This is rather trivially ... the same as in the base-config templates and defines an ocean (and atmosphere) with no biogeochemical tracers, just T and S in the ocean.

  You might use this if you were only interested in questions of ocean circulation and climate, although even then, you might want to add an age tracer to the ocean (see HOW-TO on diagnosing how the model works).

- **TRACERCONFIG.Ridgwelletal.2007.txt**

  This is the set of tracers used in Ridgwell et al. [2007] and defines a \(PO_4\)-based biological pump, with \(^{13}C\) in all the carbon pools, and the capability for accounting for sulphate-reduction with dissolved \(O_2\) is depleted. Alternative (non-modern) \(Mg/Ca\) ratios are also allowed for.

  This is one of the simplest but most versatile tracer sets, particularly for paleo, when simulating \(^{13}C\) may be important. It is usable with any \(PO_4\)-only based biological scheme, including ECOGEM with \(Fe\) cell quotas disabled (e.g. as in Wilson et al. [2018]). Other past (published) applications include:

  - Ridgwell and Schmidt [2010]
  - Ridgwell and Hargreaves [2007]
  - Crichton et al. [2020]

- **TRACERCONFIG.Caoetal.2009.txt**

  This is the set of tracers used in Cao et al. [2009] and defines a \(PO_4\)-based biological pump, with \(^{13}C\) in all the carbon pools.

  Unlike TRACERCONFIG.Ridgwelletal.2007.txt, there are no \(SO_4\) or \(H_2S\) tracers defined in the ocean and hence no potential for sulphate-reduction. Excess oxygen consumption then results in the creation and transport (and subsequent destruction) of negative concentrations of \(O_2\). Kinetically (and in terms of large-scale patterns of oxygenation), this is very similar to creating and then re-oxidizing hydrogen sulphide – see Meyer et al. [2016].

  This set of tracers also includes radiocarbon \(^{14}C\) in all the carbon pools, for diagnosing ocean circulation (and water mass ages). Additionally, both \(CFC - 11\) and \(CFC - 12\) are included for tracing deep-water formation. See Cao et al. [2009] for how these tracers are used.

- **TRACERCONFIG.Odalenetal.2019.txt**

  This is the set of tracers used in Odalen et al. [2019] and defines a biological pump with both \(PO_4\) and \(Fe\) co-limitation. There are no \(SO_4\) or \(H_2S\) tracers defined in the ocean and hence no potential for sulphate-reduction, but \(^{13}C\) is included in all the carbon pools.

  This set of tracers includes a variety of numerical/color tracers for diagnosing pre-formed properties – here of \(DIC\), \(ALK\), \(O_2\), \(PO_4\), and \(\delta^{13}C\) (see: 'diagnose how the model works'
The iron cycle is the original configuration, where 3 separate tracers are included: \( \text{Fe} \) (free dissolved iron), \( L \) (free ligands), and \( \text{Fe}L \) (iron bound to ligands). Because the equilibrium between these tracers is re-solved every time-step (and used to determine the \( \text{Fe} \) concentration for scavenging), there is redundancy, and in practice, only 2 tracers are needed. Subsequent iron cycle scheme hence use only 2 tracers (e.g. see: TRACERCONFIG.Wardetal.2018.txt). The 3-tracer scheme was also used in Tagliabue et al. [2016].

- TRACERCONFIG.Wardetal.2018.txt

This is the set of tracers used in Ward et al. [2018] and defines a biological pump with both \( \text{PO}_4 \) and \( \text{Fe} \) co-limitation of biological productivity (in ECOGEM). \( \text{SO}_4 \) or \( \text{H}_2\text{S} \) tracers as well as \( ^{13}\text{C} \) are included. There are no circulation diagnostic and/or numerical/color tracers. The configuration of the marine iron cycle differs from that of Odalen et al. [2019] and Tagliabue et al. [2016] in that only two tracers are now explicitly represented and transported in the ocean – \( \text{TDFe} \) (total dissolved iron, including iron bound to ligands) and \( \text{TL} \) – total dissolved ligand concentration (both free and bound to iron). At each time-step, the concentration of ‘free’ iron (and iron bound to ligands) is solved for and used to calculate iron removal though scavenging.

1d. Enabling science modules in the base-config

The final step in configuring your base-config file, is to ensure that all the science modules you wish to include are included (and those you do not want, are not). The template base-config files specify the basic/simply climate system combination of GOLDSTEIN ocean circulation model, GOLDSTEIN sea-ice model, and the EMBM atmospheric model. In addition, the atmospheric geochemistry module (ATCHEM) and the ocean biogeochemistry module (BIOGEM) are selected. These settings look like:

```
# *******************************************************************
# GENIE COMPONENT SELECTION
# *******************************************************************
# make .TRUE. the cGENIE modules to be included
# *******************************************************************
# GENIE COMPONENT SELECTION
ma_flag_ebatmos=.TRUE.
ma_flag_goldsteinoceangocean=.TRUE.
ma_flag_goldsteinseaiseaice=.TRUE.
ma_flag_biom=.TRUE.
ma_flag_atchem=.TRUE.
ma_flag_sedgem=.FALSE.
ma_flag_rokgem=.FALSE.
ma_flag_gemlite=.FALSE.
ma_flag_ecogem=.FALSE.
```

Note that ATCHEM and BIOGEM have to be selected together, as do SEDGEM and ROKGEM (if you want an ‘open system because ROKGEM provides the weathering input). ECOGEM is a replacement for the implicit biological export scheme in BIOGEM and hence requires BIOGEM to be selected (and hence also ATCHEM).

Remember – if you want the ECOGEM marine ecosystem model, to set ma_flag_ecogem=.TRUE.
Configuring muffin experiments

2. Choosing a template user-config

The final file creation step is for/of a user-config file. Once again – you could take a published configuration from one of the genie-userconfigs/MS sub-directories or an EXAMPLE, copy and rename it and then edit in any changes you need.

A series of template user-config files, some directly derived from published configurations (esp. for the MODERN user-configs), are also provided as part of the muffingen release. Where alternative parameter choices exist in the user-config, these are highlighted and commented out (###).

Also in the MODERN template user-configs, are some suggestions to parameter changes that align the MODERN user-configs with the corresponding PALEO user-configs (where a correspondence exists). These suggested alternative parameter choices are also included commented out (###) and are:

- Under *** REMINERALIZATION ***:
  
  #### set 'instantaneous' water column remineralization
  ###bg_par_bio_remin_sinkingrate_physical=9.9E9
  ###bg_par_bio_remin_sinkingrate_reaction=125.0
  
  which enables instantaneous water-column remineralization of POM – see USERCONFIG.PALEO.BIOGEM.PO4.SPIN for a full description.

- Under *** MISC ***:

  #### maximum time-scale to geochemical reaction completion (days)
  ###bg_par_bio_geochem_tau=90.0

  #### extend solubility and geochem constant T range (leave S range as default)
  ###gm_par_geochem_Tmin = -2.0
  ###gm_par_geochem_Tmax = 45.0
  ###gm_par_carbchem_Tmin = -2.0
  ###gm_par_carbchem_Tmax = 45.0

  The first of these imposes a maximum reactant uptake time-scale. The second extends the temperature limits imposed on gas solubility and carbonate chemistry dissociation parameters. See: USERCONFIG.PALEO.BIOGEM.PO4.SPIN for a full description.

In all the PALEO templates, a series of recommendations are made that differ from some of the defaults in the MODERN template user-configs. These include changes to:

1. biological scheme
2. inorganic matter export ratios
3. water column remineralization
4. reactant consumption limitation
5. extended temperature ranges of solubility and geochemistry constants

These are all described in detail for USERCONFIG.PALEO.BIOGEM.PO4.SPIN.

As mentioned above – the same/similar settings also appear in the MODERN templates (but commented out) should one which to align all the model assumptions across model and paleo applications.

The template user-configs in full, are:

- USERCONFIG.ABIOTIC.TRACER.SPIN

  This is as very basic template for an ocean (and atmosphere) with no carbon cycle.
  You might use or adapt this if you were only interested in questions of ocean circulation and
climate, although even then, you might want to add an age tracer to the ocean\(^{20}\) (see \textit{HOW-TO} on diagnosing how the model works).

While the template \textit{user-config} could be used with any degree of complexity of tracers, it is designed for use with \texttt{TRACERCONFIG.ABIOTIC.txt}, with or without the 3rd tracer in the \textit{base-config} needed for diagnosing ventilation age: \texttt{gm_ocn_select_4=.true.}.

- \textbf{USERCONFIG.MODERN.BIOGEM.PO4.SPIN}

  This provides the \textit{user-config} parameter settings for the pre-industrial \textit{spin-up} experiment described in \textit{Cao et al.} [2009], as well as, commented-out, the settings for the configuration of \textit{Ridgwell et al.} [2007]. (The differences in parameters primarily relate to the two difference vertical resolutions and the calibrations performed on them.) Refer to the EXAMPLES for e.g. the parameter changes needed to the \textit{Cao et al.} [2009] historical transient experiment. Suitable template tracer sets include: \texttt{TRACERCONFIG.Ridgwelletal.2007.txt} and \texttt{TRACERCONFIG.Caoetal.2009.txt}, depending on whether you intend to make use of the additional diagnostic ocean circulation tracers or not.

- \textbf{USERCONFIG.MODERN.BIOGEM.PO4Fe.SPIN}

  This provides the \textit{user-config} parameter settings for a modern (pre-industrial) marine iron cycle (biological productivity limited by both \textit{PO\textsubscript{4}} and \textit{Fe}). It is based on a configuration used by \textit{Odalen et al.} [2019] as well as in \textit{Tagliabue et al.} [2016] and was calibrated for the \textit{worj4h2} world configuration. An alternative set of biological uptake and \textit{Fe} cycle parameters are provided (commented out) which are also in use [unpublished work] and was calibrated for the \textit{worl4g} world configuration.

  Suitable template tracer sets include: \texttt{TRACERCONFIG.Odalenetal.2019.txt} and \texttt{TRACERCONFIG.Wardetal.2018.txt}, depending on the \textit{Fe} tracer scheme selected.

- \textbf{USERCONFIG.MODERN.ECOGEM.PO4Fe.SPIN}

  This is basically, the \textit{user-config} for the modern \textit{ECOGEM} experiment following \textit{Ward et al.} [2018] and bar some comments and tidying up is the same as \texttt{wardetal.2018.ECOGEM.SPIN} that can be found in \text{genie-userconfigs/MS/wardetal.2018}.

  The only difference are a couple of optional suggestions that will align the configuration with the corresponding recommended paleo configurations (in addition to the POM remineralization and geochemical constant temperature range suggestions listed earlier):

  - Under *** \textbf{REMINERALIZATION} ***:
    
    #### DOC lifetime (yrs) -- following \textit{Doney et al.} [2006]
    
    \texttt{bg_par_bio_remin_DOMlifetime=0.5}

    which aligns the lifetime of DOM with the \textit{Doney et al.} [2006] value adopted in the original \textbf{BIOGEM} configuration.

  - Under *** \textbf{MISC} ***:

    #### add seaiice attenuation of PAR
    
    \texttt{eg_ctrl_PARseaiice limit=.true.}

    #### relative partitioning of C into DOM
    
    \texttt{eg_par_beta_P0CtoDOC=0.70}

    which firstly limits light available under sea-ice in proportion to the fractional sea-ice cover in that grid cell, and secondly, re-partitions carbon from POM to DOM and is

\(^{20}\text{If so, un-comment the "optional" parameter setting.}\)
18.4 Configuring muffin experiments

...tuned to produce an approximately Redfield ratio (106) of 104.7 : 1 in C : P of exported POM.

The intended corresponding template tracer is: TRACERCONFIG.Wardeetal.2018.txt.

- USERCONFIG.PALEO.BIOGEM.PO4.SPIN

This is a recommended template user-config for paleo applications. Several parameters have been changed or added compared to published paleo configurations (these recommendations need not be adopted and alternatives are given commented out in the file):

1. **biological scheme**

   ```
   # biological scheme ID string
   bg_par_bio_prodopt="bio_P"
   # biological uptake time-scale
   bg_par_bio_tau=63.3827
   # [PO4] M-M half-sat value (mol kg-1)
   bg_par_bio_c0_PO4=0.10E-06
   ```

   This differs from most of the published paleo applications in that it has a biological export scheme that is temperature-dependent and more responsive to changes in ocean PO4 inventory (a similar configuration was used by Meyer et al. [2016]).

2. **inorganic matter export ratios**

   While it would be perfectly acceptable to utilize the default carbonate saturation-dependent scheme for the ratio of exported CaCO3 compared to POC (see Ridgwell et al. [2007, 2009]), published paleo implementations of muffin have tended to utilize a simple spatially uniform and invariant with time, CaCO3 : POC ratio. This is enacted via:

   ```
   # fixed CaCO3:POC
   bg_opt_bio_CaCO3toPOCrainratio='FIXED'
   # underlying export CaCO3 as a proportion of particulate organic matter
   bg_par_bio_red_POC_CaCO3=0.200
   ```

   with the first parameter specifying a fixed and invariant ratio, and the second parameter specifying what the ratio actually is. As an alternative to the first parameter, one can set the power in the carbonate saturation parameterization, to zero, i.e.:

   ```
   # exponent for modifier of CaCO3:POC export ratio
   bg_par_bio_red_POC_CaCO3_pP=0.0
   ```

   The default/recommended paleo value of 0.2 derives from the model-data analysis of Panchuk et al. [2008], and as such is both subject to the specific caveats of that study, and may not necessarily be appropriate for later in the Cenozoic (or earlier in the Mesozoic).

   For deeper time – prior to ca. the mid Mesozoic – the recommended parameter value is 0.0, on the basis that pelagic calcifiers had not yet evolved and the surface ocean export of biogenic CaCO3 is effectively zero – e.g. see Ridgwell [2005].

3. **water column remineralization** In the default and previously published (in all applications) remineralization scheme, particulate matter only sinks a finite distance each time-step. At the next time-step, it starts from the ocean layer it reaches and sinks further (determined by the default sinking rate of 125 m d⁻¹)²¹. This can make it difficult to e.g. check flux mass balances because there is a lag between export and particulate matter reaching the sediment surface.

   Alternatively, a 'cleaner' approach is to instantaneously remineralize all particulate

---

²¹It always has to travel at least 1 ocean layer downwards on each time-step – more for faster sinking rates and/or thinner layers nearer the surface.)
4. **reactant consumption limitation**

It is good practice not to allow complete consumption of any particular reactant by a single reaction. This is because multiple processes may be competing for the same reactant, whilst at the same time, ocean circulation is transporting the reactant away. The result can be (small) negative tracer values.

```plaintext
# maximum time-scale to geochemical reaction completion (days)
bg_par_bio_geochem_tau=90.0
```

By setting the value of `bg_par_bio_geochem_tau`\(^{23}\), the maximum consumption of any particular reactant by a single reaction is limited by an imposed lifetime (days). Here, the value is set to 90\(d\).

5. **extended temperature ranges of solubility and geochemistry constants**

As described in Ödalen et al. [2018]: "The dissociation constants used in the cGENIE calculations of solubility for \(\text{CO}_2\) in seawater follow Mehrbach et al. [1973], which are only defined for waters between 2 and 35\(^\circ\)C. Hence, the expression for \(\text{CO}_2\) solubility in the model is restricted so that all water below 2\(^\circ\)C has the same \(\text{CO}_2\) solubility (similarly for all water above 35\(^\circ\)C)."

The recommended parameter changes:

```plaintext
# extend solubility and geochem constant T range (leave S range as default)
gm_par_geochem_Tmin = -2.0
gm_par_geochem_Tmax = 45.0
gm_par_carbchem_Tmin = -2.0
gm_par_carbchem_Tmax = 45.0
```

extend the valid range to \(-2 - 45\(^\circ\)C\). Note that the valid salinity range is left unchanged.

Finally, it should be noted that the parameter settings for activating temperature-dependent remineralization (as per John et al. [2014] and Crichton et al. [2020]) are given (and commented out (###)):

```plaintext
# *** Crichton et al. [2020] temperature-dependent remin **************
###bg_ctrl_bio_remin_POC_fixed=.false.
###bg_par_bio_remin_POC_K1=9.0E11
###bg_par_bio_remin_POC_Ea1=54000.0
###bg_par_bio_remin_POC_K2=1.0E14
###bg_par_bio_remin_POC_Ea2=80000.0
###bg_par_bio_remin_POC_frac2=0.008
```

Note that in Crichton et al. [2020], a different biological scheme is used, with a faster timescale of nutrient uptake (but different light limitation) – also given commented out in the `user-config`\(^{24}\).

\(^{22}\)Sufficient that the deepest ocean model layer can be reached within a single time-step.

\(^{23}\)By default it is set to zero which is interpreted as allowing complete reactant consumption as per previously.

\(^{24}\)There is no strict necessity to use the alternative biological export scheme.
18.4 Configuring muffin experiments

- **USERCONFIG.PALEO.BIOGEM.PO4Fe.SPIN**
  This is very similar to **USERCONFIG.PALEO.BIOGEM.PO4.SPIN**, with the exception that it adds iron co-limitation of biological productivity and a marine iron cycle to the $PO_4$-only paleo settings, including the recommend changes to sinking and water column remineralization, reaction completion time-scales, and gas solubility.
  A simplified 2-tracer ($TDFe$, $TL$) iron system is assumed:
  
  ```
  # iron tracer scheme
  # NOTE: the base-config requires TFe and TL tracers
  bg_opt_geochem_Fe='hybrid'
  ```
  
  while the solubility and iron scavenging rates come from the worjh2 calibrated configuration (see: **USERCONFIG.MODERN.BIOGEM.PO4Fe.SPIN**). The only particularly novel addition, is that of a fake and globally uniform dust field in order to supply dissolved iron to the ocean surface:
  
  ```
  bg_par_forcing_name="pyyyyy.RpC02_Rp13C02.DUST"
  ```
  
  This is configured to give the same total dust supply to the ocean surface as per in the dust field used in *Ward et al.* [2018], but is supplied evenly across the global ocean surface.
  Note that there is no (user-config) scaling parameter for sediment flux forcings and hence the total flux forcing appear explicitly in the time-series file: biogem_force_flux_sed_det_sig.dat.
  Also note that for a different total ocean area, the flux per $m^{-2}$ will change as only the global total flux to the ocean surface is conserved.

- **USERCONFIG.PALEO.ECOGEM.PO4.SPIN**
  This is based on the configurations published by *Wilson et al.* [2019], with iron quotas and usage disabled in **ECOGEM**.
  In its modern tuning, **ECOGEM** has a high carbon export and high $C/P$ ratio. This leads to a less well oxygenated modern ocean than observations. In removing iron and configuring $PO_4$ as the sole limiting nutrient for paleo experiments where the dust flux field is not *a priori* known, the $C/P$ ratio increases still further and with it, a further depletion of ocean oxygen. This situation is rectified for paleo applications (under 'recommended') by:
  
  1. **diagnosing, not applying a mixed layer**
     
     ```
     # set mixed layer to be only diagnosed (for ECOGEM)
     go_ctrl_diagml= .true.
     ```
     
     and
  
  2. **re-partitioning carbon from POM to DOM**
     
     (leaving nutrients etc. unchanged):
     
     ```
     # relative partitioning of C into DOM
     eg_par_beta_P0CtoDOC= 0.70
     ```
     
     The alternative calibrated value is given should $Fe$ limitation is included. For dust and aeolian iron supply, simply follow the instructions and parameter values for **USERCONFIG.PALEO.BIOGEM.PO4Fe.SPIN**.

  In addition:

  3. **sea-ice light limitation** The original **ECOGEM** model did not include any limitation of biological productivity by sea-ice cover. This is included now by:
     
     ```
     # add seaice attenuation of PAR
     eg_ctrl_PARseaicelimit= .true.
     ```
     
     which reduces light proportionally to the local fractional sea-ice cover.
Additional recommend changes for sinking and water column remineralization, reaction completion time-scales, and gas solubility, are the same as per for the BIOGEM PALEO template user-configs, described previously.

3. Ensuring files are in their correct location and parameter settings are consistent

1. First, ensure that you have remembered to copy/transfer the entire configuration subdirectory to: cgenie.muffin/genie-paleo.

2. Next, your new base-config will need to be transferred to: cgenie.muffin/genie-main/configs. Ensure that it has a .config file extension. Configuration subdirectory to: cgenie.muffin/genie-paleo.

3. Your user-config can go ’anywhere’ ... ish. Putting it directly in genie-userconfigs may eventually make things over-crowded. Better is to create a sub-directory of genie-userconfigs (or even a sub-directory of this) and place it there.

4. Checking the consistency between the tracers defined in the base-config and the experiment parameter requested in the user-config is the trickiest part. Refer to some of the published configurations (best) and/or EXAMPLES for guidance. Sometimes it is best simply to start the experiment running and see what happens – there are checks (not comprehensive) in the code to see if you have the necessary tracers selected. Read any warnings that are reported as the experiment initializes, but often you can ignore these. Ensure that the total number of defined tracers equals the number of selected tracers you have listed (plus temperature and salinity) in the base-config.

5. Finally, if you have previously run an experiment using a different base-config, you will need to briefly interactively run an experiment with the new base-config in order that muffin knows to re-compile. (Remember that you cannot go directly to submitting jobs to the cluster when changing base-configs.)

Good luck!

25 The directory parameter passed to runmuffin.sh is then just /
26 If you place the user-config file into a sub-directory myexperiments, the directory parameter passed to runmuffin.sh is then myexperiments
Zonal wind-stress

For non GCM-based configurations, no prior wind fields exist. **muffingen** hence creates and configures an idealized zonal wind-stress field (from which wind velocity and wind speed is derived). The zonal wind-stress can take alternative strengths, depending on whether a high latitude gateway (in either hemisphere) exists. This can be prescribed directly, or **muffingen** can be enabled to 'choose' whether or not a high latitude gateway exists and hence whether or not to apply a strong or weak zonal flow. The parameter options (in the **muffingen** configuration file) are:

- **par_taulopt=0;**
  
  **muffingen** chooses whether or not to apply a strong or weak zonal flow, and in which hemisphere, depending on whether it thinks a high-latitude gateway exists or not.

- **par_taulopt=1;**
  
  A weak zonal flow is applied in both hemispheres (e.g. assuming land prevents the existence of an ocean-only circumpolar high latitude pathway in both hemispheres). The presence of a pole-to-pole super-continent (or 'ridge-world') would be an example of when this option might be selected\(^{27}\).

- **par_taulopt=2;**
  
  A strong zonal flow is applied in both hemispheres (e.g. assuming there is no land to prevent the existence of an ocean-only circumpolar high latitude pathway in both hemispheres). The presence of an Equatorial-only super-continent (or 'water-world') would be an example of when this option might be selected\(^{28}\).

- **par_taulopt=3;**
  
  'grey' world – an intermediate strength zonal flow is applied in both hemispheres. For hedging your bets of for use across a wide range of different continental configuration where you do not want to manually chose and prescribe either strong (2) or weak (1) and don’t dare trust **muffingen** to make the choice for you (option 0).

For reference – the modern world has a mix of strong (southern) and weak (northern) hemisphere zonal flows.

The different EXAMPLE (\(m\)) configuration files have a mixture of default choices, for example (not exhaustively listing all the different example configurations ...):

- **EXAMPLE_MASK_drakeworld**, **EXAMPLE_MASK_eqpasworld**, etc.: **par_taulopt=0;**\(^{29}\)
- **EXAMPLE_K1_permian**, **EXAMPLE_K2_supercontinent**: **par_taulopt=1**
- (none of the current examples prescribe strong zonal winds, but could (see footnotes)): **par_taulopt=2**
- **EXAMPLE_BLANK**: **par_taulopt=3**\(^{30}\)

---

\(^{27}\)But note that in the ridgeworld EXAMPLE provided, the automatic assignment option (\(\#0\)) is instead set to be consistent with the series of fake .dat based configurations.)

\(^{28}\)But note that in the waterworld EXAMPLE provided, the automatic assignment option (\(\#0\)) is instead set to be consistent with the series of fake .dat based configurations.)

\(^{29}\)Note that some of these could be pre-determined – e.g. water-world could be set with option \(\#2\) (strong zonal winds), and ridge-world with \(\#1\) (weak zonal winds). For configurations with a circumpolar passage only in one hemisphere such as drake-world, option \(\#0\) will configure this with a weak northern zonal wind field and a strong southern one.

\(^{30}\)By default, a water world is created, which would normally be wind stress option \(\#2\). However, assuming some
Ocean depth (and maximum levels)
The standard configurations of muffin assume a maximum ocean depth of 5000 m. (In fact, until recent code changes, the scale-depth in the GOLDSTEIN ocean model component was hard-coded as 5000.0 m.) This depth is then divided up into 16 (typically) or 8 ocean levels with a logarithmic distribution of ocean layer thicknesses.

The number of layers in the ocean is set by the parameter:

\[ \text{par\_max\_k} \]

and the maximum (scale) depth is:

\[ \text{par\_max\_D} \]

You need not have all \( \text{par\_max\_k} \) layers spread over depth \( \text{par\_max\_D} \). You can specify the minimum (deepest) depth level used in the muffin depth grid via the parameter:

\[ \text{par\_min\_k} \]

By default this has a value of 1. Setting a high value truncates the ocean floor. e.g. for a flat-bottom ocean, a common value is 3, meaning that the ocean model consists only of layers 16 down through 3, giving it a maximum depth of about 3500 m. This, in conjunction with a modern-like land fraction, gives a modern-like ocean volume (given that the modern ocean has an average seafloor depth of ca. 3500 m).

If you require a deeper (maximum depth) ocean than 5000 m, be aware that retaining the same (e.g. 16) value of \( \text{par\_max\_k} \) means that the same number of ocean layers are distributed over the greater ocean depth, and hence, your surface layer will be deeper, potentially changing biological productivity and biogeochemical cycling. There is hence a facility provided for maintaining the thickness depth distribution in the upper ocean while adding additional layers to the bottom of the ocean to create a greater (than 5000.0) maximum ocean depth. This is implemented by setting the parameter:

\[ \text{par\_add\_Dk} \]

(the additional number of depth (k) levels) to a non-zero value. The effect of this is assume that the maximum ocean depth parameter (\( \text{par\_max\_D} \)) corresponds to the specified maximum number of ocean layers (\( \text{par\_max\_k} \)) minus the number of additional layers (\( \text{par\_add\_Dk} \))^\(^{32}\). The maximum ocean depth is then re-calculated based on \( \text{par\_max\_k} \) minus \( \text{par\_add\_Dk} \) number of levels, corresponding to depth \( \text{par\_max\_D} \) (with the same logarithmic distribution of layer thicknesses with ocean depth applied to the additional layers deeper than \( \text{par\_max\_D} \)).

For example, setting:

\[ \text{par\_add\_Dk}=2 \]

in conjunction with a total number of ocean levels:

\[ \text{par\_max\_k}=18 \]

and:

\[ \text{par\_max\_D}=5000.0 \]

unknown continental configuration may be drawn, the 'grey world' intermediate wind stress option is set (#3).

\(^{31}\)(so any value can be set as it is ignored)

\(^{32}\)(rather than the maximum number of ocean levels as usually applied)
together configures an 18-level ocean model with the first 16 (18-2=16) levels spanning 5000.0 m, with an automatically re-calculated maximum ocean depth (par_max_D value) of 6922.2705 m and the additional 2 levels spanning this additional depth (1922.2705 m).  

When muffingen writes out the parameter settings (file: config_???????.txt), the scale depth of the ocean is also written out. By default this will be equal to the value of par_max_D (5000.0 m), unless, the value of par_add_Dk is non-zero, when the above described re-calculation of the maximum ocean depth is made (and written out in the parameter file).

---

33Note that there is no way, unless the logarithmic function for the distribution of layer thicknesses with ocean depth is changed, to retain the same modern/default layer thickness distribution for the upper ocean layers and have a maximum ocean depth that is different to 5000.0 m unless you carry out the calculation yourself for the depth occupied by e.g. 18 levels. (You cannot calculate how many layers occupies e.g. 6000.0 m and still retain the upper ocean depth distribution as the depth profile does not fit neatly into 6000.0 m.)
aka: ’Has this dumb question been asked before?’
19.1 Installation related questions

Stack space
You may encounter issues with regards to the ifort Intel FORTRAN compiler (an maybe others), particularly when using SEDGEM because of the size of the arrays holding sediment information:

"The Intel® Fortran Compilers 8.0 or higher allocate more temporaries on the stack than previous Intel Fortran compilers. Temporaries include automatic arrays and array sub-sections corresponding to actual arguments. If the program is not afforded adequate stack space at runtime relative to the total size of the temporaries, the program will terminate with a segmentation fault."

The (a?) solution is to increase the CPU stack space, Try:

```
$ ulimit -s unlimited
```

Compiler errors
Recent versions (ca. 10) of gcc (GNU C Compiler) may through up errors when compiling the muffin netCDF library routines – ‘Error: Rank mismatch ...’.

This can be resolved by adding to makefile.arc (in genie-main), just before:

```
# === NetCDF paths ===
```

the line:

```
FFLAGS += -fallow-argument-mismatch
```
19.2 Cluster/queue questions

Do I have to submit experiments to the queue rather than running interactively?  
Yes! Except for developing the model and debugging, testing new experimental designs, and forcing a re-compile. The number of instances of the model that can be run simultaneously interactively is limited by the number of processing cores on the head node. The more experiments that are run interactively, the slower everything will go. Additionally, if you even temporarily lose your Internet connection, an interactively-run experiment will die. The queue is there for your convenience, believe it or not ...

Can I leave all my experiment results on the cluster for ever?  
No! Nothing is backed up on the cluster, and space is not infinite. So, periodically, transfer archived (.tar.gz) results off of the cluster and delete both the archive file and the results directory.
19.3 Help! My experiment has died ... why?

19.3.1 Sources of initialization error

If, when using the runmuffin.sh shell script to run a cgenie.muffin experiment, it all goes horribly pear-shaped ...

1. The experiment dies absolutely immediately.
   Check that the runmuffin.sh shell script has executable permissions. Also check that the directory you are trying to run the model from is the genie-main directory.
2. The experiment does not quite die immediately, but does not manage to stagger even as far as the line:

   ```
   >> Here we go ...
   ```

   before dropping dead. If so, there should be an error message telling you that a particular file or directory cannot be found. Check:
   - All the files and directories you have specified exist.
   - You have not omitted spaces where you should not have, nor added spaces where a `_` separator was required.
   - You have not misspelt anything – a common cause of problems is in reading the number one (`'1'`) for the letter el (`'l'`), or vice versa in the computer font (Courier).

   These first two sorts of pain and suffering are due to mis-configuration of the runmuffin.sh shell script. Also refer back to the overall sequence of configuring and running the cGENIE.muffin model shown in Figure 1.2.

   Other sources of error are due to the configuration of cGENIE.muffin (or more rarely, due to the model itself):

   1. As cGENIE.muffin initializes, files may be reported as not being found. One possible cause of this is that `~` may not necessarily get expanded into the path of your home directory (e.g., `/home/mushroom`. In this situation, `~` can simply be replaced with `$HOME`. Note that as well as making this substitution at the command line, the user-config file may also contain instances of `~` (such as in specifying particular forcings).

   2. A missing/not found error can also arise with some compilers if one of the various ASCII input files to BIOGEM (or SEDGEM) does not have a blank line at the bottom (some vague quirk of the unformatted read used in the FORTRAN code). Check: the user-config file, and also any boundary condition files being requested.

   3. Further trouble can occasionally arise when using Windoz and editing files (e.g., the user-config file) and it is possible to corrupt the format of the file. For what file(s) you have edited, use the command dos2unix to strip off Windoz formatting characters (which are invariably invisible in most editors). The syntax for this (or see the linux `man` pages, or even Google it) is `$ dos2unix FILENAME`. 
19.3 Help! My experiment has died ... why?

19.3.2 Carbonate chemistry errors

The most common error as *muffin* starts to run, or during the course of a run relates to the solution of the aqueous carbonate chemistry system and specifically, for \( pH \). There are 2 different errors that can be reported in this respect:

1. **Numerical instability at step;** ...
   
   Data: dum\_DIC,dum\_ALK,dum\_Ca,dum\_SO4tot,dum\_H2Stot,dum\_NH4tot
   
   \( pH(SWS) \), \( pH \) (OLD), \( pH \) (guess #1), \( pH \) (guess #2)
   
   CARBONATE CHEMISTRY COULD NOT BE UPDATED :

   followed by a series of values (corresponding to the listed tracers), and then a block of text summarizing possible meanings of the error (and tracer values).

2. **Number of steps taken without successfully solving for \( pH \) = ...**
   
   Data: dum\_DIC,dum\_ALK,dum\_Ca,dum\_SO4tot,dum\_H2Stot,dum\_NH4tot
   
   \( pH(SWS) \), \( pH \) (OLD), \( pH \) (guess #1), \( pH \) (guess #2)
   
   CARBONATE CHEMISTRY COULD NOT BE UPDATED :

   also followed by tracer values and possible explanations.

Both errors relate to the implicit solution for the \( pH \) (technically: \([H^+]\)) value at the current time-step. The scheme takes an assumed value at model initialization or from a re-start, or during an experiment – the last calculated value at that location in the ocean, and uses this to make an updated estimate of \( pH \). The calculation is carried out iteratively, with the new assumed value of \( pH \) being taken from the previous iteration, until the solution converges, or not.

The first error – instability – is triggered by a negative value of \( pH \) being calculated. The second (much more common) error occurs if the maximum number of iterations has been reached – i.e. if the carbonate chemistry system is not converging within a reasonable number of iterations.¹

The default behavior in response to either situations occurring is for muffin to forego updating \( pH \) in the troublesome ocean grid call and continue to the end of the current time-step, save data snapshot, and then exit the run.

Why does the error arise, and what can be done about it?

1. **Running *muffin* with array dimensions which do not match the number of tracers selected is a common cause of failures to solve the aqueous carbonate system, as often calcium ion or other tracer concentrations become corrupted and get assigned nutty values.**

As per the error message text:

(1) Check the FIRST TWO lines of the ERROR DATA (ocean DIC and ALK):
   
   -> These should be ... reasonable ... of order a few thousand umol kg\(^{-1}\) (units reported are mol kg\(^{-1}\))’
   
   -> NaNs, negative values, obsenely low (<< a few hundred umol kg\(^{-1}\)) or, high (>> tens of thousands) are indicative of array bounds problems. Array bounds problems are mostly commonly due to:
   
   (A) incorrect ocean dimension compared to the compiled executable,
   
   (B) incorrect number of biogeochem tracers in the ocean compared to the compiled executable.

   => Carry out a *** make cleanall *** (from ~/genie/genie-main).

(2) ...

(3) Extreme values of Ca and SO4 (#3, #4) are only vanishingly possible except due to incorrect compiled array dimension or extreme salinity.

¹The default number of iterations is set as 100 and is specified by the parameter: `gm_par_carbchem_pH_iterationmax=100`
Try forcing a re-compile ($ make cleanall) to make this go away.

Also, if a re-start is used which was generated with a different land-sea mask (and base-config) to the current experiment and associated base-config, spurious tracer values can occur and crash the carbonate chemistry scheme. Try running without the re-start and see if that solves it (identifies the source of the problem).

The appearance of all (or almost all) NaNs also hints at a ocean circulation/climate model component problem and only causes an immediate crash in the ocean carbonate chemistry. One not infrequent cause is due to the sea-ice model that is not unconditionally stable and when it fails, can lead to NaN tracer concentrations. In particular, the sea-ice model can fail when there is very little sea-ice cover – either retreating or increasing – and in topographically restricted regions (e.g. small embayments or semi-enclosed seas at high latitudes). If the sea-ice model fails, this should be recorded in a series of error messages notifying a failure to solve. Check the experiment log file if running submitted jobs.

If the root of the issues is indeed the sea-ice model, one or more of the following can be tried:

- Simply run the model ‘warmer’ (higher pCO2 or radiative forcing) so that there is no sea-ice cover anywhere, or ‘colder’ so that there is plenty. (It seems to be the appearance or disappearance of an ephemeral sea-ice cover that is the main issues.) If not running from a restart, you can also try specifying a ‘warmer’ or ‘colder’ ocean start than the default. The initial ocean temperature of a spin-up is determined by the parameters (set in the base-config although you can add new values in the user-config):

  ```
  # temp0 -- ocean temperature start
  go_10=5.0
  # temp1 -- start temperature start
  go_11=5.0
  ```

  (Aim to set both parameters to the same value because I have forgotten what the difference between them is ... (vertical zones of temperature perhaps).)

- Check that you have the revised sea-ice diffusivity value and limits imposed on sea-ice advection, set by (this also appears in the base-config but you can adjust the values in the user-config):

  ```
  # reduced sea-ice eddy diffusivity
  gs_11=1000.000
  # set a fractional sea-ice coverage threshold for preventing advection
  gs_par_sica_thresh=0.99
  ```

- The nuclear option(!) is to disable the formation of sea-ice entirely, which can be done via:

  ```
  eb_opt_freeze=.false.
  ```

  (This simply sets the freezing point of seawater to -999C and hence preserves the water and energy balance of the system.)

2. Failure to converge (the second error) is unlikely to be due to mis-compiled tracer selections or incomparable restart ocean grids, but rather as a consequence of extreme geochemical conditions occurring somewhere in the ocean. This can be due to excessively intense export production and remineralization, as well as at excessively high dissolved silica concentrations – all may be plausible early Earth states, but ones that require an adjustment of how aqueous carbonate chemistry is solved:

---

2Interactive jobs should echo the error messages to the screen as the failure occurs.
19.3 Help! My experiment has died ... why?

- At very high nutrient \((PO_4)\) concentrations, and/or at high ocean temperatures with a temperature-dependent remineralization scheme, and/or with a slow prescribed sinking speed\(^3\), oxygen consumption can be complete and subsequently, sulphate reduction intense. Sulphate reduction results in the release of two mol alkalinitities per mol \(SO_4^{2-}\) consumed, impacting carbonate chemistry in that ocean cell (in addition to the release of \(DIC\) from the remineralized organic matter). Where the resulting \(H_2S\) is re-oxidized, alkalinity is removed and local carbonate chemistry impacted in the opposite sense.

The problems are likely to occur in the ocean sub-surface, although difficulties at the surface and associated with air-sea gas exchange are also possible. Plotting out surface or sub-surface (and ideally; 2D surfaces of the distribution of minimum and maximum values using \texttt{muffinplot}) can help in the identification of the nature of the underlying problem(s).

There are a couple of possible courses of action to take (as an alternative to modifying the experimental design and reducing the intensity of biological export and sub-surface recycling):

(a) Increase the number of iterations allowed in solving for seawater \(pH\). \(^4\) The parameter controlling this is:

\[
gm\_par\_carbchem\_pH\_iterationmax=100
\]

and could, for instance, be increased to a value of 1000 (iterations).

(b) In addition to the 100\(^5\) iterations that \texttt{muffin} is allowed in trying to converge on a solution for \(pH\), \texttt{muffin} can also try a different initial (seed) \(pH\) value if it encounters one of the error conditions. The iteration counter is re-set and the iterative convergence calculation is carried out once again.

The controlling parameter is:

\[
gm\_ctrl\_carbchem\_pH\_seed\_retry=.false.
\]

In a situation by which \(pH\) never ever converges, it may be that the initial value is too far from the (true) solution value. Setting the parameter \texttt{true} and re-seeding with randomly varying values then help find a sufficiently close initial guess that convergence occurs.

(c) Whilst the carbonate chemistry and \(pH\) at the ocean surface is solved at every time-step (in order to calculate air-sea gas exchange of \(CO_2\), carbonate chemistry is not needed for any reaction, with the exception of when one or more of the following tracers are selected:

- \texttt{is\_FeCO3}
- \texttt{is\_Fe3Si2O4}
- \texttt{io\_CH4}

(Carbonate chemistry is also needed for calculating the preservation of \(CaCO_3\), but this is calculated separately in \texttt{SEDGEM}.)

---

\(^3\)I.e. much slower than the default of 125\(md^{-1}\)

\(^4\)Conversely, one could decrease the \(pH\) tolerance criteria for convergence: \texttt{gm\_par\_carbchem\_pH\_tolerance}, which by default has a value of 0.001 – increasing will result in a faster solution and a greater change of passing the convergence criteria, but at the expense of accuracy.

\(^5\)The default value unless it has been changed.
The error then arises most likely when an attempt to solve for \( pH \) is made when preparing 3D fields for *time-slice* saving. This calculation is made regardless of whether or not you want carbonate chemistry *time-slice* data saved. An option is available that disables the updating of \( pH \) associated with *time-slice* saving:

```plaintext
# disable carbonate chemistry updating for time-slices
bg_ctrl_data_save_slice_carb_update=.false.
```

(by default, this parameter is `.true.`). Aside from at the surface every time-step, employing this parameter change results in the ocean interior \( pH \) only being solved for at the very start of the experiment. Carbonate chemistry *time-slice* data is still saved ... but will not change from the initialised values (except at the surface).

This *should* avoid most/all carbonate chemistry crash issues(?)

(d) You could also try forcing \( pH \) to be solved for, throughout the ocean interior, and at every single time-step. This might ‘work’, because the \( pH \) solution is seeded with the previous \( pH \) value, meaning that very frequently updating the value of \( pH \) might allow you to track the transition into extreme redox states with no issues.

The parameter setting to make this happen is:

```plaintext
# force full carbonate chemistry updating
bg_ctrl_carbchemupdate_full = .true.
```

Note that if you have a methane cycle, ocean interior \( pH \) is frequently solved for anyway.

Option (c) is arguably the better one to test first, unless you have a methane cycle. Then (d) is worth trying (but even if it works, may slow things down). Otherwise, (a) and (b) may help but will probably only offer limited improvement in stability.

There is a fifth and slightly different, possibility, requiring the marine carbon cycle itself is altered:

(e) Assuming that the underlying issue is ‘nutrient trapping’ – the rapid biological removal of \((PO_4)\) from the surface, following by shallow remineralization of the organic matter, re-release of the \((PO_4)\) and then efficient return back to the surface to fuel further biological productivity – adding \(Fe\) co-limitation of biological productivity may help, because loss of dissolved \(Fe\) from the water column will tend to scale with the flux of organic matter, hence acting as a negative feedback on the nutrient trapping.

- Very high\(^6\) high dissolved silica \((H_4SiO_4)\) concentrations in the ocean can cause issues. This is because \(H_3SiO_4^-\) appears in the calculation of carbonate alkalinity from total alkalinity.

A parameter is provided to remove \(H_3SiO_4^-\) from the alkalinity carbonate calculation – by setting:

```plaintext
gm_ctrl_carbchem_noH3SiO4=.true.
```

---

You may also be trying to spin up the model with fabulously high \(pCO_2\) (i.e. many 10s or even 100s of thousands of \(ppm\)). What typically happens here, is that mean ocean \(DIC\) and \(ALK\) are invariably initialized to modern (or shallow time) values, in equilibrium with modern (or up to 1000\(ppm\)) \(pCO_2\). Restoring the atmosphere to an excessively high \(pCO_2\) will lead to extreme imbalances between \(DIC\) and \(ALK\) at the ocean surface as \(CO_2\) invades the ocean from the atmosphere.

\(^6\)Orders of magnitude higher than modern
If you do not need to simulate pH or carbonate saturation state, or are not using processes that depend on these (e.g. variable saturation-dependent CaCO₃:POC rain ratios, or the preservation of CaCO₃ in deep-sea sediments, then you might consider setting a radiative forcing in the atmosphere equivalent to the required pCO₂ value, and just let pCO₂ naturally equilibrate with ocean DIC and ALK (and the climate and biological pump state).

If you really *must* have an explicitly high value of pCO₂, then you can try one of the follow-
ing, with the aim to set ocean ALK to a more appropriate value:

1. Firstly you can estimate the initial mean ocean ALK value you need from an off-line carbonate chemistry code.
   Typical would be to assume a modern ocean surface saturation state with an Ω of calcite of around 5.0 (mean ocean saturation is 5.06 in the modern configuration of muffin). Take a reasonable mean ocean surface temperature and salinity ... and an atmospheric partial pressure equal to the value you want to restore to, and then solve for the complete (mean surface) system. Initialize your experiment with the resulting DIC and ALK values.

2. Read suitable DIC and ALK values off of Figure 19.1.
   This is far from a particularly great figure and is poorly resolved in terms of pCO₂ (black) and saturation (Ω) (yellow) contours. You are looking to follow the pCO₂ contour you want (or close to), to higher ocean ALK, until the ocean surface is reasonably over-saturated (i.e. 1.0 < Ω < 10.0). e.g. for ×10 pCO₂, this would occur for an ALK value of ca. 5 mmol kg⁻¹.
   You also want to read off a corresponding DIC value, which you could simply take to be the x-axis value (but noting that this is really DIC plus pCO₂).

3. Restore ocean carbonate chemistry (e.g. surface ocean saturation) at the same time as you restore atmospheric pCO₂.
   Refer to the Forcing HOW-TO, and specifically: 'Set a specific ocean chemistry or saturation state’. The 3rd listed method is the one you want.
   In this method, you might have to ramp up pCO₂ to its eventual, required value, rather than implementing an instantaneously high pCO₂ value.
Figure 19.1: Carbonate chemistry fan diagram (at 15°C).
Note that the total carbon inventory is the sum of both DIC and $pCO_2$. The circle represents mean modern ocean chemistry.
19.3 Help! My experiment has died ... why?

19.3.3 Other possible geochemical errors

It is good practice not to allow complete consumption of any particular reactant by a single reaction. This is because multiple processes may be competing for the same reactant, whilst at the same time, ocean circulation is transporting the reactant away. The result can be (small) negative tracer values (and if you are unlucky, large negative errors). Negative tracer value, in turn, have the potential to cause crashes.

A parameter is hence provided to limit the maximum consumption of any particular reactant by a single reaction via an imposed lifetime (days):

```plaintext
# maximum time-scale to geochemical reaction completion (days)
bg_par_bio_geochem_tau=90.0
```

In this example, the value is set to $90d$.\(^7\)

Negative tracer values can also arise as a result of the default ocean mixing scheme. In all modern and ‘realistic’ paleo configurations, isopycnal/diapycnal mixing is used. However, this scheme can lead to unwanted negative tracer values. In extreme biogeochemical configurations, particularly at low atmospheric $pO_2$ and when sharp horizontal transitions in redox state in the ocean occur, the magnitude of negative values can become unacceptable.

An option is provided for toggling between isopycnal/diapycnal (.true.) and horizontal/vertical (.false.) mixing schemes. For instance, adding:

```plaintext
# it is recommended that it is turned OFF (=.false.) for 'fake' worlds
go_diso=.false.
```

would request the horizontal/vertical mixing scheme and substantially reduce the possibility of negative tracer values arising.

---

\(^7\)By default it is set to zero which is interpreted as allowing complete reactant consumption as per previously.
19.3.4 Meaning of specific error and warning messages

'ERROR: path integral around island too long'
Such an error is possible when developing new or modifying existing continental configurations (and associated 'island' and 'path' definition files), but not in normal running of the model. First try a make cleanall and then try re-running. If the problem persists, it is possible that a key configuration file has accidently/somehow been changed.

'ERROR MESSAGE: Particulate tracer CaCO3 ...'
I have been told 'ERROR MESSAGE: Particulate tracer CaCO3 does does not have the corresponding ocean tracer Ca selected' – is this a problem ... ?

No! You are simply being reminded that you have calcium carbon (CaCO₃) selected as a particulate tracer in the model, but although when it dissolves it releases Ca²⁺ (and removes Ca²⁺ when CaCO₃ is precipitated), you do not have Ca²⁺ selected as an explicit dissolved tracer in the ocean. This is not a problem as by far the most important effect on the carbon cycle of adding/subtracting Ca²⁺ is a change in alkalinity, which is implicitly account for. Only on very long time-scales, or in deep-time situations when the Ca²⁺/Mg²⁺ ratio was very different from today, might you need to select Ca²⁺ (and Mg²⁺) as an ocean tracer.
19.4 General model experiment questions

When does the model need to be recompiled?

cGENIE.muffin will need to recompile in the following situations:

- You have just carried out one of the cGENIE.muffin tests, e.g., make test or make testbiogem.
- You have changed the dimension of the climate model grid (which also means an automatic change in the biogeochemistry modules), either horizontally (e.g., going from $36 \times 36$ to $18 \times 18$) or vertically (e.g., going from 8 levels in the ocean to 16).
- You have changed the number of selected ocean biogeochemical tracers in the base-config and hence changed the value of:

\[ \text{GOLDSTEINTRACSOPTS} = \text{'$(DEFINE)GOLDSTEINNTRACS=2'$} \]

(The latter two involve a change in compiled array dimension.)

In all three situations, the base-config is being changed (or should be\(^8\)). In running at the command line (i.e. interactively), the runmuffin.sh script detects the change in base-config, and automatically forces a re-compile for you. However, the compute nodes of the cluster do not have access to the FORTRAN compiler. As a sad and unfortunate consequence, submitted jobs cannot recompile modules and all science modules must be already compiled when a job is submitted.\(^9\)

To recompile (and re link) the science modules – first, start an interactive run of the experiment you want to conduct. This will ensure that it is correctly compiled. This also serves as a visual check that you have requested a user-config, restart, etc that actually exists. Start the run for the length of time you intend to use when submitting the experiment as a job to the queue, but kill it (keyboard command: Ctrl-C) once it is compiled and you are happy that it is running OK (say, after 10 years). You can now be reasonably confident that the experiment is safe to submit the job to the cluster (and all files and inputs are as they should be).

If you have multiple experiments, all with the same resolution and number of tracers, you DO NOT need to re-run interactively or attempt to recompile. Also, you can add ‘modules’ and not recompile. i.e., you can interactively run an ocean-only carbon cycle. And then submit it. And then submit an experiment using SEDGEM as well. (Because when the model is compiled, ALL sciences modules are compiled, meaning that all there is to do is just link them, which does not require the (ifort) FORTRAN compiler.)

Refer to Figure 1.2 for the sequence of steps associated with configuring and running model experiments.

In the naming of different forcing specifications: what does ‘yyyyz’ mean?

A. The naming convention for forcings is that the (sub)directory name starts with the code for the continental configuration, if the forcing is tied to a specific continental configuration. For example: forcings with the string ‘FeMahowald2006’ relate to the prescription of a dust (Fe flux) field re-gridded from Mahowald et al. [2006]. When this has been re-gridded to the worjh2 continental configuration, worjh2 appears at the start of the name. If the forcing is independent of a specific continental configuration, such as restoring atmospheric $CO_2$ to a prescribed value (uniformly throughout the atmosphere), the string is 'yyyyz', as in e.g.: pyyyyz_RpCO2_Rp13CO2.

---

\(^8\)One could edit a base-config and re-ruin, but it is better to create a new base-config file if editing any of the settings, particularly those affecting array dimensions

\(^9\)It is OK to change the flavor of GENIE as linking is done by the C compiler.
Can I make the model run faster?

*sign* You speed freak. Is this all you care about? What about the quality of the simulation - does that mean absolutely nothing to you? Oh well ... There is a bunch of stuff that slows cGENIE.muffin down that may not be absolutely essential to a particular model experiment. These include:

- **The number of tracers** - if you don’t need ’em, then don’t select ’em! Selected tracers are automatically passed to GOLDSTEIN and advected/convected/diffused with ocean circulation. Similarly, BIOGEM does a whole bunch of stuff with tracers, particularly those which can be biologically transformed. All this is numerically wasteful if you aren’t interested in them. Equally importantly, the more tracers you have selected the more careful you have to be in configuring the model. Superfluous tracers therefore cost more configuration time and/or increase the change of a model crash.

- **Tracer auditing** - the continuous updating and checking global tracer inventories to ensure that there is no spurious loss or gain of any tracer (i.e., a bug) has computational overheads associated with it. Whether this checking is carried out or not is set by the value of the flag `bg_ctrl_audit`.

- **Time-series results saving**. Model tracer (plus some physical) properties are bring continuously averaged in constructing time-series results files. Cutting down on time-series that you don’t need will help minimize model run-time. The various categories of time-series that will be saved are specified by a series of namelist parameter flags. However, within each category (such as ocean tracers - `bg_ctrl_data_save_sig_ocn`) all properties will be saved - you are not given the option to save a defined sub-set (for example, DIC and PO4 in the ocean but not ALK).

- **Time-slice results saving**. If you have relatively few requested time-slices over the course of the model integration then this is unlikely to significantly impact the overall run-time (even will all possible data category save namelist flags set to `true`.). However, note that if you have accidently triggered the default time-slice saving interval (by having no data items in the time-slice specification file (`bg_par_infile_slice_name`)) you may end up with the model running about as fast as a 2-legged dog super-glued to a 10-tonne biscuit.

Note that time-series saving of data that is a 2-D average, such as atmospheric composition at the ocean-atmosphere interface, sediment composition at the ocean-sediment interface, or just ocean surface conditions, is less numerically demanding than mean values that have to be derived from a 3-D data field.

- **Alter the degree of synchronicity between climate and biogeochemistry** (see HOW-TO guide).

As a very rough guide, the impact on total run-time of making various changes to the model configuration are listed as follows. Numbers are given as a percentage increase in total model run-time (using the `/usr/bin/time` linux command). Tracers selected in the ocean are DIC, ALK, PO4, O2, DOM_C, DOM_P, DOM_O2, as well as 13C isotopic components (DIC_13C and DOM_C_13C) (+ T and S). The corresponding tracers are present in the atmosphere and as particulates. The model is run for 10 years as a new run (i.e., not loading in a restart file):

- ADD auditing ⇒ +15%
- ADD time-slice saving ⇒ +20%\(^{11}\)

\(^{10}\)It is `.false.` by default.

\(^{11}\)Because only a 10 year integration has been carried out with a time-slice saved at 10 years, the computational cost of
19.4 General model experiment questions

- ADD time-series saving ⇒ +15%
- REMOVE $^{13}$C isotopic species (= DIC and DOC ocean tracers) ⇒ -10\%\(^{12}\)

You can also run at lower resolution. The basic configuration for a faster 'lego box' **cGE-NIE.muffin** configuration consists of a $18 \times 18$ model grid and an 8 level ocean. The continents are in a zonally-averaged configuration and there is no topography in the oceans.

The model is accelerated by:
1. it’s low resolution
2. taking 48 instead of 96 ocean time-steps per year in the ocean
3. **BIOGEM** being only being updated every 4 rather than every 2 ocean time-steps.

In this configuration 100 years take about 40 seconds, 10 kyr would just take over and hour, and 100 kyr could be run overnight!

---

\(^{12}\)The speed gained by removing two tracers is not proportional to the fractional decrease in number of tracers (in this example reducing from 11 to 9 the number of tracers in the ocean gives only a ca. 10\% improvement in overall speed).
19.5 Climate-y questions

Can I disable climate feedback with $CO_2$?

Yes, for instance when you might want to compare the fate of $CO_2$ released to the atmosphere with climate (and ocean circulation and temperatures) not responding, vs. $CO_2$ in the atmosphere also driving changes in climate (and hence affecting the pathways and transformations of $CO_2$, particularly in the ocean).

To specify no climate feedback, add to the user-config file:

```
# set no CO2 climate feedback
ea_36=n
```
as well as a radiative forcing scaling:

```
# scaling for atmospheric CO2 radiative forcing, relative to 278 ppm
ea_radfor_scl_co2=1.0
```
(a value of 1.0 giving no change in climate relative to the default).

How does the fresh water flux adjustment work?

Without a dynamical atmosphere and sufficient resolution e.g. North American topography, the EMBM atmospheric component component produces an overly weak net transport of fresh water from the North Atlantic into the North Pacific – too weak to maintain an AMOC (how sad).

A fix is provided in the form of a prescribed transport of fresh water from the Atlantic to Pacific (or salt from the Pacific to Atlantic). The transport is split up into 3 zones\textsuperscript{13} – approximately:

1. South Atlantic into the (South) Pacific [latitude band $j=9,12$ ($36\times36$ grid)]
2. Equatorial and southern subtropical Atlantic into the Pacific [latitude band $j=13,25$ ($36\times36$ grid)]
3. North Atlantic (and Arctic) into the North Pacific [latitude band $j=26,36$ ($36\times36$ grid)]

See Figure 1 in Marsh et al. [2004] (Climate Dynamics) for an illustration. By default [Edwards and Marsh, 2004], the fluxes combine for a total net transfer of 0.32 Sv following Oort [1983] and individually are (with parameter name):

1. (xml name: extra1a) $ea_25=-0.03$
2. (xml name: extra1b) $ea_26=0.17$
3. (xml name: extra1c) $ea_27=0.18$

and giving a total net flux adjustment ($extra1a+extra1b+extra1c$) of 0.32 Sv.

These parameters were subsequently tuned, and in the worbe2 (e.g. Ridgwell et al. [2007]) configuration, become:

1. $ea_25=-2.1228021E-02$
2. $ea_26=0.1202921$
3. $ea_27=0.1273681$

giving a lower total net flux adjustment of 0.226 Sv.

Now ... subsequent tunings ... rather than modifying each of the 3 individual fluxes, instead introduced an overall scaling factor that is applied to the default values. For instance, in the worjh2 tuning, this parameter was:

\textsuperscript{13}Also see notes in the xml parameter definition in addition to description below and references.
and whose value is applied to each default individual flux region value (i.e. -0.03, 0.17, 0.18 Sv),
giving a total flux of $0.7268 \times 0.32 = 0.233\text{Sv}$ (or $0.7268 \times (-0.03 + 0.17 + 0.18) = 0.233\text{Sv}$).\(^{14}\)

There are other ways to prescribe a flux adjustment – for instance by creating a salinity flux
forcing, with positive values in the North Atlantic, and negative ones in the North Pacific. The
total/net flux forcing in Sv can then be specified more directly and unambiguously.

**How do I change the orbital configuration of cGENIE.muffin?**
[see Orbits HOW-TO]

**Can I do solar geoengineering (SRM) experiments?**
No! Because you might destroy the planet.
No wait ... in the model (world) ... yes! But because of the absence of a dynamical atmosphere,
options here are limited. However, modification of the solar constant, *a-la* ‘giant mirrors in space’
is possible. Sea-ice (surface) albedo can also be adjusted.

\(^{14}\)Note that if values for the 3 individual flux parameters are specified, then the scaling factor applies to these values
(which replace the defaults.)
19.6 Ocean biogeochemistry-y questions

Can solubility related changes be separated from stratification and circulation changes?

With BIOGEM coupled to the climate model core of cGENIE.muffin\(^{15}\), a change in atmospheric \( CO_2 \) will induce a change in SSTs, which in turn affect the carbon cycle and feedback on \( CO_2 \) via changes in solubility and via changes in circulation (stratification) and thus biological productivity. There are times when it is helpful to separate out solubility related changes from circulation related changes. This equally applies to dissolved \( O_2 \) and \( CO_2 \). The problem is that you need a change in climate and surface temperatures in the climate model in order to induce a change in circulation.

There is a way of having an altered climate and circulation, which then affects the marine carbon cycle, yet specify the SSTs that are actually seen by BIOGEM (and thus used in solubility calculations).

First of all, control the radiative forcing of climate internally in the EMBM rather than externally by the atmospheric \( CO_2 \) concentration calculated by ATCHEM. Turn off explicit \( CO_2 \) forcing of climate by setting: \( ea_36='n' \). The namelist parameter \( ea_20 \) will then dictate the EMBM radiative forcing: a value of 1.0 (default) gives no change in radiative forcing \( (CO_2 = 278 \text{ ppm}) \), a value of 2.0 corresponds to the effect of doubling \( CO_2 \), \( \times 4 \, CO_2 \), etc. Altering the value of \( ea_20 \) thus lets you control climate (and circulation) without having to adjust \( CO_2 \) and the carbon cycle.

Next, SSTs in BIOGEM can be specified independently of the climate model. You achieve this by setting up a restoring forcing of ocean temperatures at the surface. Note that by default, prescribing SSTs (or SSSs) in BIOGEM does not propagate through to the climate model which does its own independent climate thing based on the value of \( ea_20 \). This allows you to retain the surface temperatures and thus solubility associated with a \( \times 1 \, CO_2 \) World, but have a warmer more stratified ocean (appropriate for a much warmer World).

What actually happens is that BIOGEM receives both the altered circulation field and the altered SSTs due to \( \times 4 \, CO_2 \), but sets its own SSTs internally rather than use those calculated by the climate model. Setting up the SST restoring is in principal just like for PO4. The values for the SST field you can simply copy and paste out of a prior \( \times 1 \, CO_2 \) experiment.

The converse experiment, is to have circulation and biological productivity not change, but explore the effect of changes in SST-driven solubility. i.e., to separate the solubility pump from circulation change effects on glacial \( CO_2 \).

What is the difference between the different Fe configurations and schemes?

There is a parameter – \( bg\_opt\_geochem\_Fe \) – that specifies the Fe scheme (although note that you have to have the appropriate/required tracers selected):

1. \( bg\_opt\_geochem\_Fe \) = ’OLD’
   This is the very original scheme, based on 3 tracers – Fe (free dissolved Fe), FeL (ligand-bound Fe), and L (free ligand).

   During scavenging, a new equilibrium partitioning between the 3 species is calculated and imposed on tracer concentrations via the remin array, but only at the surface.

   Entrain some completely unnecessary accounting for Fe addition at the surface as part of the equilibrium calculation, including a call to \( \text{sub}\_calc\_geochem\_Fe \) as part of the main \((i,j)\) biogem subroutine array. All other schemes are free from this.

2. \( bg\_opt\_geochem\_Fe \) = ’ALT’

\(^{15}\)Namelist: \( ea_36='y' \)
This is also based on 3 tracers – Fe (free dissolved Fe), FeL (ligand-bound Fe), and L (free ligand).

During scavenging, a new equilibrium partitioning between the 3 species is calculated and imposed on tracer concentrations via the remin array. This is carried out throughout the water column.

3. bg_opt_geochem_Fe = 'FeFe2LFeL'

Based on the same 3 tracers – Fe (free dissolved Fe), FeL (ligand-bound Fe), and L (free ligand), but also, Fe2, which is ignored for the purposes of calculating scavenging and Fe-FeL-L partitioning. Here, the tracer Fe is explicitly taken to be $Fe^{3+}$.

Scavenging and tracer concentration re-partitioning as per (2).

4. bg_opt_geochem_Fe = 'hybrid'

This is based on the minimum number of tracers required to describe the system – TFe (total dissolved Fe) and TL (total dissolved ligand). The equilibrium partitioning between Fe, FeL, and L, is derived when required.

5. bg_opt_geochem_Fe = 'lookup_4D'

Also based on TFe (total dissolved Fe) and TL (total dissolved ligand), but with Fe calculated via a lookup table.

In schemes 1+2+3, scavenged Fe is removed from the Fe tracer pool. For 4+5, scavenged Fe is removed from the TFe pool.

What is 'tracer auditing' – should I have it switched on?

When developing a new model parameterization, it is of fundamental importance that careful track is kept of the total tracer inventory of the system in the face of internal mass transfer and any inputs (e.g., prescribed restoring or flux boundary conditions) or outputs (e.g., sedimentation). No spurious gain or loss of tracer mass must occur as a result of bugs introduced to the code. The tracer inventories of the ocean can be periodically calculated and compared to that predicted to have occurred on the basis of any net input (or output) occurring in the intervening time to help catch bugs. The simplest implementation would be an audit carried out at system start-up (before any transformation of tracer mass has taken place), and at the very end (after the last transformation of the tracer fields). However, integrating over over an extended time period can lead to the excessive accumulation of numerical (truncation) errors. Instead, the audits are carried out periodically during the model run. The periodicity of tracer auditing follows the times specified for time-series data saving (i.e., at time listed in the file specified by bg_par_infile_sig_name).

The entire audit procedure is as follows:

1. First, an initial inventory is calculated, achieved by summing the product of the concentration of each (selected) tracer with the mass of each cell, across all wet cells.
2. During the model run, the net spatially- and time-integrated transfer of tracer mass arising from all transfers across the external reservoir boundaries is calculated.
3. At a periodic pre-defined time, the inventories are re-calculated. The difference between old and new inventories should be equal to the integrated net flux. If the relative difference between re-calculated inventory and estimated (on the basis of net flux) differs by more than a predefined threshold then an error message is raised (and the model halted if requested)
4. The integrated net flux variable is re-set to zero and steps (2-4) repeated.

In short – if you are not modifying the code then you can take it on trust(!) that the model distribution is free of (major) bugs and that spurious gain or loss of tracers does not occur. It you don’t trust me ... then switch the auditing feature on.
Auditing is inactivated by default. To activate it:

bg_ctrl_audit = .true.

To adjust the threshold (relative) tolerance\textsuperscript{16}:

bg_par_misc_audit_relerr = value

To halt the model\textsuperscript{17} if it fails the tracer drift tolerance:

bg_ctrl_audit_fatal = .true.

A secondary benefit of tracer auditing when running the model interactively, is that it reports back to you the maximum and minimum value of all the tracers (and locations of where this occurs), as follows:

```
>>> SAVING BIOGEM TIME-SERIES @ year 0.50 278.069 -6.501 16.522 3.843 18.543 ...
   temp    / min = 0.2713E+03 (18,36, 8) / max = 0.3030E+03 ( 4,18, 8)
   sal     / min = 0.3337E+02 (10,35, 8) / max = 0.3891E+02 (30,29, 8)
   DIC     / min = 0.1878E-02 (35,24, 8) / max = 0.2581E-02 (33,21, 1)
   DIC_{13C}/ min = -0.4225E+00 ( 3,16, 3) / max = 0.2792E+01 (25,13, 8)
   DIC_{14C}/ min = -0.1779E+03 (33,21, 1) / max = 0.2197E+02 (30,29, 8)
   PO4     / min = 0.7071E-07 (29,28, 8) / max = 0.3030E-05 ( 3,16, 3)
   O2      / min = -0.4521E-04 (27,30, 5) / max = 0.3363E-03 (24,35, 8)
   ALK     / min = 0.2212E-02 (10,35, 8) / max = 0.2724E-02 (33,21, 1)
   DOM_{C} / min = -0.4159E-05 (21,34, 3) / max = 0.1517E-04 (32,25, 8)
   DOM_{C_{13C}}/ min = -0.1000E+20 ( 1, 3, 2) / max = 0.5817E+01 (29,36, 8)
   DOM_{C_{14C}}/ min = -0.1000E+20 ( 1, 3, 2) / max = 0.2236E+04 (29,36, 8)
   DOM_{P}  / min = -0.3924E-07 (21,34, 3) / max = 0.1431E-06 (32,25, 8)
   Ca      / min = 0.9769E-02 (10,35, 8) / max = 0.1136E-01 (30,29, 8)
   CFC11    / min = 0.0000E+00 ( 1, 3, 2) / max = 0.0000E+00 ( 1, 3, 2)
   CFC12    / min = 0.0000E+00 ( 1, 3, 2) / max = 0.0000E+00 ( 1, 3, 2)
   Mg      / min = 0.5505E-01 (10,35, 8) / max = 0.5505E-01 (30,29, 8)
```

\textbf{How do I do an ocean CO$_2$ injection experiment?}

There is a hard way (but maximum flexibility), a less hard way, ... and an easy way. To cut the shit – what follows is the easy way!

First, you want to use the updated tracer forcing format:

bg_ctrl_force_oldformat=.false.

Put this line in the \textit{user-config} file if it is not already there, perhaps under \texttt{FORCINGS} section.

You will need a forcing template for the CO$_2$ injection – \texttt{pyyyyyz_FCO$_2$\_UNIFORM}. This is provided on mygenie.seao2.org. Download and unpack (\texttt{tar xzf pyyyyyz_FCO$_2$\_UNIFORM.tar.gz}) from the directory: \texttt{/genie_forcings}. As it stands, this is configured to stuff 1 PgC yr$^{-1}$ of CO$_2$ into the ocean over the course of one year. The location of the CO$_2$ injection is some random default place that probably does not exist, which is not very good. So, you need to specify your ocean location. For this, add the following lines to a \textit{user config} file:

\begin{verbatim}
bg_par_force_point_i=22
bg_par_force_point_j=33
bg_par_force_point_k=5
\end{verbatim}

which corresponds to a cell in the N. Atlantic (i,j, = 22,33) at an intermediate depth (k=5).

The \textit{i,j,k} coordinates are counted from left-to-right with longitude: \textit{i}, from bottom to top with latitude: \textit{j}, and form top to bottom with depth for ocean level, \textit{k}. The land-sea mask and maximum depth (lowest \textit{k} integer) you are allowed can be got from the BIOGEM 2D netCDF, variable

\textsuperscript{16}By default, this is set to 1.0E-08.
\textsuperscript{17}By default the model will continue running, even if there is an apparent spurious drift in tracer inventories occurring.
19.6 Ocean biogeochemistry questions

grid_level. This is a map of the 'k' values. >90 means land, for the 8-level ocean the ocean depths will be between 1 and 8. 8 being the surface. So the map is of the depth of the ocean and thus lowest k value you are allowed to use.

By default, using the CO₂ injection forcing template you will get 1 PgC emitted to the ocean, in the location you specify. You can scale the amount of carbon up via the namelist parameter:

\[ \text{bg}_\text{par}_\text{ocn}_\text{force}_\text{scale}_\text{val}_3 = xxx \]

where xxx is the multiple of 1 PgC you want to inject. NOT your favorite movie viewer rating. e.g., 100 PgC:

\[ \text{bg}_\text{par}_\text{ocn}_\text{force}_\text{scale}_\text{val}_3 = 100.0 \]

Note that 100.0 PgC is quite a lot of carbon to be injecting into a single location (cell) in the ocean model! By default, the time-scale of injection is set as 1 year. To increase the time over which the CO₂ injection takes place use the namelist parameter \[ \text{bg}_\text{par}_\text{ocn}_\text{force}_\text{scale}_\text{time}_3 \], which simply scales the time interval. i.e.,

\[ \text{bg}_\text{par}_\text{ocn}_\text{force}_\text{scale}_\text{time}_3 = 10.0 \]

causes the CO₂ injection to take place over 10 years. But since the flux is in units of PgC per year, you will get 1000.0 PgC carbon total (10 years x 100 PgC yr⁻¹). So a combination of both namelist scaling parameters (both flux scaling, and interval scaling) will be needed for the required total CO₂ injection.

Note that the integer at the end of the namelist parameter name corresponds to the index of the ocean tracer. 3 is DIC. 12 would allow you to inject alkalinity into the ocean (but the you would need to create additional forcing specification files).

The slightly harder way involves entering in the i,j,k location explicitly in the forcing configuration file configure_forcings_ocn.dat. Altering the magnitude and/or duration of the flux release requires editing biogem_force_flux_ocn_DIC_sig.dat.

The hardest way requires that two 3D fields explicitly specifying the spatial nature of the forcing flux are created and modified.

For these alternative options – see earlier section on tracer forcings (Section 4).

**Can I do carbon dioxide removal (CDR) experiments?**

Yes! See geoengineering HOW-TO.
19.7 Geologic carbon cycle questions

In GEMlite, does the adaptive step size control work with fixed/prescribed \( pCO_2 \)?

If \( pCO_2 \) if fixed/restored, the answer is 'no' (ish). Or rather: you’ll often get little difference compared to simply fixing the ratio of accelerated to non-accelerated time-steps. However, you will still get the advantage of adapting time-stepping depending on other changes to weathering (sedimentation) that may have been prescribed. i.e. even with \( pCO_2 \) restored during 'normal' time-stepping, \( pCO_2 \) will change during the accelerated mode if weathering is significantly out of balance with sedimentation. The greater this imbalance, the greater the change in \( pCO_2 \), and the sooner that time-stepping will be handed back to the normal (full updating) mode.

If you have prescribed changing \( pCO_2 \), e.g. a continual ramp upwards, GEMlite is not appropriate in the first place, as the atmosphere is intrinsically assumed to be in equilibrium with the ocean surface and steady-state geochemical gradients in the ocean have been established. (This assumption is broken if \( CO_2 \) is rapidly invading the ocean.) Acceleration (and GEMlite) is also not appropriate if ocean circulation and carbon cycling have not yet been spun-up, unless at least 5 to 10 kyr of normal time-stepping forms part of the total spin-up including acceleration.

How can I diagnose changes in the carbon budget due to weathering/sedimentation?

The following example assumes that you are only running with CaCO3 weathering (i.e silicate weathering and outgassing are both set to zero). In this case the weathering flux of DIC into the ocean is equal to the Ca weathering flux. This is output as a time series in:

\[
\text{biogem_series_diag_weather_Ca.res}
\]

in units of moles per year.

The system is closed with respect to organic matter, so that all POC is remineralised and returned to the ocean. For this reason, the exchange of DIC between the ocean and the sediments is equal to the exchange of Ca. i.e. the exchange of one mole of C is always associated with one mole of Ca, as the system is only open with respect to CaCO3. Therefore the net flux of DIC from ocean to sediments is equal to the difference between \text{biogem_series_focnsed_CaCO3.res} and \text{biogem_series_fsedocn_Ca.res}.

The net flux of DIC into the ocean from weathering and sediments is therefore equal to \( \text{weather}_\text{Ca} + \text{fsedocn}_\text{Ca} - \text{focnsed}_\text{CaCO3} \).

How can I change where the solute fluxes end up in the ocean?

ROKGEM routes all solute fluxes (from weathering) to coastal ocean grid cell, using the .k1 file to determine the size of the drainage basins and the point at which they exit into the ocean. By default, and when using the (standard) global average climate-weathering function options, ROKGEM divides up the global weathering rate in proportion to the size of each drainage basin. A number of major drainage basins drain into the North Atlantic and Arctic and this can bias North Atlantic surface tracer (and particularly isotope) values in some cases (e.g. in the case of dissolved silica).

As an alternative run-off routing, an option is provided that scales the (Si) weathering flux distributed to the ocean in proportion to the runoff from each drainage basin rather than simple area (which has the effect of e.g. moving much of the excess solute input away from the Arctic). To enable this, set:

\[
\text{rg_opt_weather_runoff=}.\text{TRUE}.
\]
19.8 Data saving / model output questions

Why is the netCDF data saved at 'odd' times?
There is a default sequence of points in time that BIOGEM will save data at. These points are specified in the file: save_timeslice.dat (which lives in cgenie.muffin/genie-biogem/data/input). This default sequence provides a useful generic starting point.

To specify different save points for an experiment:
1. Edit this file (not recommended).
2. Copy, or create a new file (with the same format). The file that BIOGEM uses for saving data is specified by the parameter:
   `bg_par_infile_slice_name='save_timeslice_historicalfuture.dat'`
   (in the example of a historical/future relevant series of save points being requested).

Note that always, at the very end of an experiment, data is automatically saved regardless of whether or not you remembered to specify a save point for the final simulated year.
Refer to the Chapter on muffin output.

How can I increase/reduce the amount (frequency/type) of data saved?
Refer to the Chapter on muffin output.

What the buck is 'convective cost'???
Convective cost – variable phys_cost in the 2D netCDF output, is the "column integrated adjustments per year". At each grid point, the number of convective adjustments – homogenization steps of a portion of the water column – is summed. The results are presented as a frequency – homogenizations per year by default.

Note that the homogenizations can occur between any number of layers, and anywhere in the water column. Only the water column integral frequency is saved.

---

18 As a default, the netCDF time-slices are saved an annual averages, centered on these points in time.
### Forcings questions

**Can I combine forcings together?**

Yes ... but it is not quite as simple as in the *user-config* writing:

```bash
# specify forcings
bg_par_forcing_name="worjh2.FeMahowald2006"
bg_par_forcing_name="pyyyyy.FRpCO2_Fp13C02"
```

In the example that you wanted to combine an atmospheric $CO_2$ emissions *forcing* with a surface ocean dust *forcing*, because only the last parameter value in a list of multiple definitions is used. i.e. the above is equivalent to just writing:

```bash
# specify forcings
bg_par_forcing_name="pyyyyy.FpCO2_Fp13C02"
```

Instead, you need to create a new forcing (assuming the combined forcing you want does not already exist):

1. Copy/rename one of the two individual *forcing* directories. This will be your new *forcing* name.
2. In the example above – if you have copied the directory for *worjh2.FeMahowald2006*, you simply need to add in the specific atmospheric $CO_2$ emissions *forcing* forcing files contained in *pyyyyy.FRpCO2_Fp13C02*, which are:
   
   ```
   biogem_force_flux_atm_pCO2_13C_sig.dat
   biogem_force_flux_atm_pCO2_sig.dat
   configure_forcings_atm.dat
   ```

Note that more care has to be taken when combining *forcings* that include the same phase of tracer, i.e. atmosphere and atmosphere, or ocean and ocean. In this case, you need to open up the *configure_forcings_*_.dat* file of one *forcing*, and copy the tracer selection line (or lines) to the equivalent file in the new *forcing* directory.

**Why does my ocean flux forcing does not do anything?**

As always if you apply a flux forcing and nothing appears to happen, check:

1. The flux has not been scaled to zero ...
2. The spatial locations, where you expect the flux to be applied, and not on land $(i,j)$ location is a land, not ocean point), or in the ocean crust $(i,j)$ is ocean, but the layer chosen is deeper than the ocean floor at that location).
3. That the model run, in time, overlaps with the time-dependent forcing information. e.g. you might start a forcing at year 2010, but only run the model to year 2000 ...

Careful comparison, e.g. difference maps or simply looking at some global diagnostic output provided as in the time-series data format, will confirm whether the impact truly is zero, or just very small. If very small, your issue is mostly simply one of scaling and too small of a flux to make much impact.

---

19 These occur between a pair of tags:

`-START-OF-DATA-`

`-END-OF-DATA-`
19.9 Forcings questions

Why does my ocean iron flux forcing does not do anything?

Start by referring to above (general flux forcing question). However, there is a special point of failure of a forcing, unique to the iron system, because there are two different ways of representing \( Fe \) and \( Fe \)-related species in eGENIE.muffin:

1. In the basic, and original Fe scheme, there are three separate tracers represented in the ocean:
   - \( Fe \) – tracer number 09 – dissolved iron III (Fe).
   - \( FeL \) – tracer number 23 – ligand-bound Fe.
   - \( L \) – tracer number 24 – free ligand (iron binding).

   In the forcing definition, a flux of Fe is selected in:
   
   `configure_forcings_ocn.dat`

   by:

   ```
   -START-OF-DATA-
   9 f f 0.0 t t -1 01 01 01 '[Fe]'
   -END-OF-DATA-
   ```

   Associated with this selection, is a field of time-dependent information for the forcing:

   `biogem_force_flux_ocn_Fe_sig.dat`

   and dependent on the nature of the forcing, potentially also a file containing a spatial pattern for the forcing, e.g.

   `biogem_force_flux_ocn_Fe_SUR.dat`

   Here: it is important to note that both file names contain the tracer short-name: \( Fe \).

2. In a newer scheme, there are just 2 tracers:
   - \( TDFe \) – tracer number 90 – total dissolved Fe.
   - \( TL \) – tracer number 42 – total dissolved ligand.

   (and e.g. free iron is derived by assuming an equilibrium partitioning based on the total iron and total ligand concentrations).

   Why am I telling you all this? For example, configurations using ECOGEM, use the newer (two tracer only) representation of iron cycling, whereas in e.g. geoengineering examples, BIOGEM is using the older three tracer representation. If you then wish to configure ECOGEM to use forcings based on the geoengineering examples, you have to:

   1. Firstly, in `configure_forcings_ocn.dat` change the selected tracer number from 9 to 90.
   2. Secondly, rename the time-dependent information file, and if present, the spatial file, changing the \( Fe \) bit of the filename to \( TDFe \).
   3. Lastly, the parameter in the user-config that scales the forcing (if used), has a name that ends in the tracer number and needs to be changed, so rather than e.g.

      `bg_par_ocn_force_scale_val_9`

      you would have:

      `bg_par_ocn_force_scale_val_90`

   If you select a tracer number in the `forcings` that does not exist in the ocean configuration you are using, such as the ‘wrong’ iron tracer – this is why the forcing appears not to do anything.
What follows are potted HOW-TO instructions for doing things, split into more technical installing and running the model (this Chapter) vs. configuring experiments (next Chapter).

There is some overlap with the FAQ Chapter, so please read all three!
20.1 HOW-TO ... get started with cGENIE.muffin

Find and Install muffin
See: Chapter 1.

Set up and analyse model experiments
See: Chapter 1. Also: subsequent tutorial chapters.

Do some thing dumb
Easy! Just close your eyes and change some parameter values at random. Better still, start using the model without reading the manual first ...
20.2 HOW-TO ... linux

Viewing directories; moving around the file system
When logging in, you always start from your 'home' directory. This is represented by a `~` before the command prompt ($). At the command line prompt in linux, you can view the current directory contents:

```
$ ls
```

or for a more complete output:

```
$ ls -la
```

To go down a directory (e.g. cgenie_output) relative to where you already are:

```
$ cd cgenie_output
```

and to go back up one is:

```
$ cd ..
```

It is often safer/easier at first, if you need to change more than one directory level to do this in stages. e.g. to change to cgenie_output/exp0_modern_SPINUP, change to cgenie_output (cd cgenie_output) but then check that you are in the place you think you are and/or remind yourself of the spelling of the next directory you need to change to by typing ls.

You can always return to your home directory (~) by typing:

```
$ cd $HOME
```

(or cd ~)

Copying and moving files
To copy a file myconfig to myconfig_new, assuming you are in the same directory where both the old file is and the new file will be:

```
$ cp myconfig myconfig_new
```

To move myconfig to the cGENIE user-config directory, assuming you are in the directory where the old file is but with the new file in a different directory, give the full path of the new directory:

```
$ mv myconfig ~/cgenie.muffin/genie-userconfigs/LABS/myconfig
```

To rename myconfig to useless_config:

```
$ mv myconfig useless_config
```

Creating directories
To create a directory mydirectory:

```
$ mkdir mydirectory
```
Repeating command lines

You do not have to re-enter lines of commands and options in their entirety each time – by pressing the UP cursor key you get the last command you issued. If you keep pressing the UP cursor key you can recover progressively older commands you have previously entered. When you have recovered a helpful line you can simply just edit it, navigating along with the LEFT and RIGHT cursor keys (press RETURN when you are done).

Changing passwords!

The command `passwd` Linux is used to change the user account passwords.

The usage can be as simple as:

```
$ passwd
```

You will then be asked your current password. And then the new one. (And then asked to confirm the new one.)

Use impossible to guess passwords ... never shorter than 8 characters. Ideally, use a mix of both UPPERCASE and lower case letters, numbers, and one or more 'special' characters, such as: #, %, &,* . But equally, make sure that you can remember the new password ...

```
vi
```

The `vi` editor is a text-based editor that you use at the command line (i.e. it does not open in its own window, nor have fancy menu items or icons to click). You use `vi` = to create/edit files, by typing:

```
$ vi FILENAME
```

where `FILENAME` is the ... name of the file you want to edit or create.

You start in the 'command' mode, in which you do not edit the contents directly, but instead can access file and copy-paste operations such as:

- `:q` == quit
- `:q!` == no, really quit
- `:x` == save and quit
- `dd` == cut line
- `p` == paste

To start editing and e.g. inserting text, press the i key.

To exit the editing mode and back to the command mode, press the Esc (escape) key.

---

1 The password can always be reset by the system administrator.

2 Then make sure the file is present in the directory you are currently in, or provide a full path to the file (+filename).
20.4 HOW-TO ... install Ubuntu muffins

This is a brief guide to installing muffin on Ubuntu. The example taken is for a generic Ubuntu distribution version 18.04 LTS ('Bionic Beaver') install that is fresh out of the box. For a more established installation, fewer components may need to be installed. Different/fewer components may also be necessary for an older Ubuntu distribution. The netCDF installation procedure has also been tested on Ubuntu 16.04 LTS and checks out A-OK.

Instructions are given step-by-step, although not all the components need be installed in this order. In brief: required installation components include: a fortran compiler, a git client, netCDF libraries. Note that the various netCDF component version numbers are not the most up-to-date, and newer can almost certainly be substituted (but not tested here).

Preparation
Get hold of a computer with Ubuntu installed on it. Make sure you have plugged in the internet connection. Log in. Get a strong cup of coffee.

Installation
1. gfortran [FORTRAN compiler]
   
   `sudo apt install gfortran`

2. g++ [C++ compiler]
   
   And ... also the GNU C++ compiler³:
   
   `sudo apt install g++`

3. netCDF [graphics libraries]
   
   These come rather inconveniently in 2 parts ... first the C libraries need to be installed, and then the FORTRAN libraries ....
   
   The example is given for the most recent version of both libraries. For details/most recent version, see: [https://www.unidata.ucar.edu/software/netcdf/](https://www.unidata.ucar.edu/software/netcdf/)
   
   But first ... missing in the default Ubuntu 18.04 is the m4 utility, which can be installed:
   
   `sudo apt install m4`

   and also make:
   
   `sudo apt install make`

   Now you are good to go ... First, we need install the main netCDF C libraries and then the FORTRAN and C++ libraries that depend on the C libraries. For the basic C libraries:
   
   
   `tar xzf netcdf-4.6.1.tar.gz`
   
   `cd netcdf-4.6.1`
   
   `./configure --disable-netcdf-4 --disable-dap`
   
   `make check`
   
   `sudo make install`

---

³This could probably be worked-around by editing some of the make files ...
Note that this installs the libraries in the default install location. Also disabling netCDF-4 to avoid added complexities of needing an HDF5 library built with zlib enabled (and hence the zlib library installed) ...

Then ... muffin requires netcdf-cxx (legacy) C++ libraries for netCDF that have since been retired\(^4\) and hence requiring a separate/additional installation step\(^5\) (first return back one directory level to where-ever you are downloading to and installing from):

```
wget ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf-cxx-4.2.tar.gz
tar netcdf-cxx-4.2.tar.gz
cd netcdf-cxx-4.2
./configure
make check
sudo make install
```

And ... then for the FORTRAN netCDF libraries. (At this point – return back one directory level to where-ever you are downloading to and installing from.)

```
wget ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf-fortran-4.4.4.tar.gz
tar xzf netcdf-fortran-4.4.4.tar.gz
cd netcdf-fortran-4.4.4
./configure
make check
sudo make install
```

NOTE: if for either or both libraries, you want them anywhere other than off of /usr/local, you’ll need to specify the installation directory, set NCDIR, and pass this to ./configure

```
-prefix=${NCDIR}
```

And probably engage in additional unpleasantness.

Finally – if when running make testbiogem you run into issues, specifically: libraries that cannot be ’found’, try forcing an update of the library link cache (the not found libraries may be links to the ’real’ library and somehow this link is not working/found):

```
sudo ldconfig
```

[’creates the necessary links and cache to the most recent shared libraries found in ... the file /etc/ld.so.conf, and in the trusted directories, /lib and /usr/lib’]

4. xsltproc [’a command line tool for applying XSLT stylesheets to XML documents’]

```
sudo apt install xsltproc
```

Why do we eve need this? :(

5. git [git client]

```
You’ll need a git client:
sudo apt install git
```

\(^4\)See: link

\(^5\)This version of the netCDF C++ library includes no changes since the 4.1.3 release, but is provided for backwards compatibility as a separate package. It was developed before key C++ concepts like templates, namespaces, and exceptions were widely supported. It’s not recommended for new projects, but it still works.’
6. **muffin** [the cGENIE.muffin code]
   And finally, you can obtain the code.
   NOTE: Recommended is changing directory to your HOME directory (cd) as the default **muffin** directory settings assume this.
   
   ```
   git clone https://github.com/derpycode/cgenie.muffin
   ```

7. **python** [python symbolic link]
   If an actual **muffin** experiment, rather than just make **testbiogem**, does not run, and e.g. in response to:
   
   ```
   which python
   ```
   
   you get nothing (although you still get a response to `python -V` or `type -a python`), it maybe that you need a python symbolic link. It might also then not hurt to install python2.7 ...
   
   ```
   sudo apt install python2.7
   ```
   to install python2.7, and:
   
   ```
   sudo ln -s /usr/bin/python2.7 /usr/bin/python
   ```
   
   to create a symbolic link from python -> python2.7
   
   Obviously, you are going to have to configured **muffin** and got it running first (see next section) to even find out whether you need to muck about with python or not ...

### Configuring muffin

The version of **muffin** available from GitHub is configured with default environmental settings that match the above instructions and require no configuration changes. If not – there are several environment variables that may need changing – the compiler name, netCDF library name, and netCDF path. These are specified in the file `user.mak` (**genie-main** directory). If the **muffin** code tree (`cgenie.muffin`) and output directory (`cgenie_output`) are installed anywhere other than in your account HOME directory, paths specifying this will have to be edited in: `user.mak` and `user.sh` (**genie-main** directory). If using the `runmuffin.sh` experiment configuration/launching scripts, you’ll also have to set the home directory and change every occurrence of `cgenie.muffin` to the model directory name you are using (if different). (Installing the model code under the default directory name (`cgenie.muffin`) in your $HOME directory is hence by far the simplest and avoids incurring additional/unnecessary pain (configuration complexity) ...)

### Testing muffin

To test the code installation – change directory to `cgenie.muffin/genie-main` and type:

```
make testbiogem
```

This compiles a carbon cycle enabled configuration of **muffin** and runs a short test, comparing the results against those of a pre-run experiment (also downloaded alongside the model source code). It serves to check that you have the software environment correctly configured. If you are unsuccessful here ... double-check the software and directory environment settings in `user.mak` (or `user.sh`) and for a **netCDF** error, check the value of the `NETCDF_DIR` environment variable. (Refer to the FAQ section for addition fault-finding tips.) If environment variables are changed: before re-trying the test, you will need to type:
make cleanall

That is is for the basic installation.
20.5 HOW-TO ... install RedHat muffins

No specific example is given at this time for RedHat ... refer to the installation instructions for Ubuntu in Section 20.4. Probably, not more installation will be needed. The netCDF requirements are the same.
This is a brief guide to installing **muffin** on a Mac.
(Also see the Section 19.1 FAQ (page 268).)

To install the **muffin** release of cGENIE on a Mac you will need a number of software packages, including Fortran, C++ and NetCDF. The best way to get hold of these is via a package management system, such as Homebrew (https://brew.sh) or MacPorts (https://www.macports.org). This guide is based on Homebrew, because it is slightly more user friendly than MacPorts, and is kept more up-to-date with changes in the Apple operating system. (If you are already a MacPorts user, please see Section 20.6).

1. First of all, you will need XCode, which can be downloaded from the app store or here...
   After installing XCode, it is necessary to enable command line tools, by entering at the command line...
   ```
   xcode-select –install
   ```
2. Get Homebrew by pasting the following at the terminal command line...
   ```
   /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
   ```
   Next, type ‘brew doctor’. This should tell you “your system is ready to brew”. If it doesn’t, see Section 20.6.
3. Install Fortran, C++, NetCDF and some other useful libraries at the command line (using Homebrew) as follows:
   ```
   brew install cmake
   brew install gcc
   brew install hdf5
   brew install netcdf
   brew install wget
   ```
4. Get hold of a current copy of the **muffin** code:
   ```
   git clone https://github.com/derpycode/cgenie.muffin
   ```
5. Check your netcdf version number by entering brew info netcdf
   This will return several lines, but the key one gives the netcdf path, and should look something like:
   ```
   /usr/local/Cellar/netcdf/4.6.1_2 (84 files, 6.2MB) *
   ```
   (This was from my most recent install, with version 4.6.1_2)
6. Finally, adjust the cGENIE environment variables for your machine and netcdf installation by editing
   `cgenie.muffin/genie-main/user.mak`, setting:
   ```
   MACHINE=OSX
   ```
   and
   ```
   NETCDF_DIR=/usr/local/Cellar/netcdf/4.6.1_2
   ```
to reflect your netcdf path (Step 5).
7. To test the code installation – change directory to cgenie.muffin/genie-main and type:

```
make testbiogem
```

This compiles a carbon cycle enabled configuration of cGENIE and runs a short test, comparing the results against those of a pre-run experiment (also downloaded alongside the model source code). It serves to check that you have the software environment correctly configured. If you are unsuccessful here ... double-check the software and directory environment settings in user.mak (or user.sh) and for a netCDF error, check the value of the NETCDF_DIR environment variable. (Refer to the User Manual for addition fault-finding tips.) If environment variables are changed: before re-trying the test, you will need to type:

```
make cleanall
```

That is it for the basic installation.

**MacPorts**

You are here because you have already installed MacPorts, presumably by following the instructions here: https://www.macports.org/install.php. You now have two options, either remove MacPorts entirely, replacing it with Homebrew, or install the required packages through MacPorts and a number of precompiled binaries.

**Option 1: Remove MacPorts from your system... (to be replaced by Homebrew)**

1. Back up your system (i.e. using Time Machine).
2. To uninstall MacPorts, enter at the terminal:

   ```
sudo port -f uninstall installed
   
   Then remove everything that is left from MacPorts:
   
sudo rm -rf /opt/local
   sudo rm -rf /Applications/DarwinPorts
   sudo rm -rf /Applications/MacPorts
   sudo rm -rf /Library/LaunchDaemons/org.macports.*
   sudo rm -rf /Library/Receipts/DarwinPorts*.pkg
   sudo rm -rf /Library/Receipts/MacPorts*.pkg
   sudo rm -rf /Library/StartupItems/DarwinPortsStartup
   sudo rm -rf /Library/Tcl/darwinports1.0
   sudo rm -rf /Library/Tcl/macports1.0
   sudo rm ~/.macports
   ```

   Note that the `sudo` command is inserted before the `rm` (i.e. remove) command in order to enable the correct permissions.
3. You may now continue with your installation as described in the main text. You may have to delete some files (using `sudo rm`), as recommended by `brew doctor`.

**Option 2: Install required packages through MacPorts and precompiled binaries...**

1. First of all, synchronize your installation of MacPorts:

   ```
sudo port -v selfupdate
   ```
2. Then install Netcdf and related C++ and Fortran libraries at the command line using MacPorts, as follows:
20.6 HOW-TO ... install macOS muffins

```
sudo port install netcdf
sudo port install netcdf-cxx
sudo port install netcdf-fortran
```

3. Download precompiled fortran and C++ binaries appropriate to your operating system (El Capitan & Sierra, etc.) from http://hpc.sourceforge.net. Install as follows, amending the file number to match your version of OSX (e.g. El Capitan & Sierra are associated with the 7.1 binaries).

```
cd ~/Downloads/
gunzip gcc-7.1-bin.tar.gz
sudo tar -xvf gcc-7.1-bin.tar -C /.
gunzip gfortran-7.1-bin.tar.gz
sudo tar -xvf gfortran-7.1-bin.tar -C /.
```

If your operating system is not listed here, you will either have to wait until it is, or install Homebew.

**Errors**

Errors identified by `brew doctor` are most likely associated with some incompatible files in your software libraries, perhaps from a previous installation of MacPorts. Try to follow the suggestions given to you by `brew doctor`, deleting any problematic files (using `sudo rm` to overcome any permission issues). Note that you may wish to do a system backup first.
**20.7 HOW-TO ... install Windows(!) muffins**

Natively ... *muffin* does not compile under Windows. However ... 6

**Cygwin**

So you want to run *muffin*, but previous sage advice has failed to deter you from wanting to do it on Windows... Well don’t say we didn’t warn you, but it *is* possible to do – but only by cheating.

*muffin* is not able to run directly on Windows, but there are ways to *pretend* that Windows is actually Unix and get it going anyway. But why oh why run *muffin* in Windows? Big runs are best sent to clusters (which are invariably a Linux machine of some sort) where they can happily run in native Linux and keep at it for ages. But maybe you want to tinker around with the model on your own laptop a bit and do some short runs without bothering with clusters and queueing and the like. Or maybe you don’t have access to a fancy cluster and want to play with a real earth system model on your laptop anyway. Or maybe the normal way of doing it was just too easy for you...

The main method described here uses **Cygwin**, which claims to “provide functionality similar to a Linux distribution on Windows”. Alternatively, if you have Windows 10 you could try using its new-ish Windows Subsystem for Linux (WSL) feature, which in theory means you can have supposedly actual Linux on your Windows machine without having to do anything awkward like dual-installation. However, it’s still a bit Beta, and when I last tried it put the requisite libraries all over the place which was a faff. It may work better at some point, but for the moment proceed there at your own risk (or do a better job than me).

**Cygwin** goes to pains to note that it only provides functionality similar to Linux, and isn’t actually Linux or “... a way to run native Linux apps on Windows”. What this means in practice is that you can’t just take stuff compiled on a Linux machine, dump it on your Windows machine and expect it to work via **Cygwin** (or vice versa). But if you’re coding and compiling stuff to run within **Cygwin** then it basically acts the same. Just remember the golden rule: What Compiles In **Cygwin**, Stays in **Cygwin**.

We’ll assume you don’t already have **Cygwin** installed (and if you do, you may need to check you’ve got all the packages you need anyway – skip to running the installation executable from wherever you saved it). It’s best to use this guide in parallel to the main muffin manual too (which provides extra troubleshooting). Here goes:

1. **Download Cygwin**: Head to [www.cygwin.com](http://www.cygwin.com) and click the link named `setup-x86_64.exe` under *Install* to download the executable (pick 64/32 bit depending on your system). Save it – maybe adding `Cygwin` to the name so you know what it is later – and then run it from your downloads tab when it’s ready (it will also no doubt have a security popup; say yes, or this will be a very short tutorial.)

2. **Start Cygwin Setup**: In **Cygwin** setup, say *Next* to all the things (assuming you want to install from the internet (rather than somewhere else?)), use the default root directory (best not change this), install for all users on your machine, and default local package directory, & use system proxy settings for downloading. It will ask you what mirror you want to download from too – it doesn’t really matter where, I tend to go for somewhere local but they should all be the same anyway.

---

6By Dr. David A. McKay &lt;david.armstrongmckay@su.se&gt;, tested May 2019, Jan 2020
3. **Select Packages:** Next in Setup, it will present you with Select Packages to install. **Cygwin** doesn’t automatically come with every useful Unix package and library ever (as it’d be enormous), so you have to go through the tedium of picking what you actually think you need (and inevitably miss something). Here is what you’ll need for **muffin** install to work beyond the default Basics (find them using the search bar, then change the dropdown bit where it says ’Skip’ to the latest non-test version number. Libraries for each package should be automatically selected by this as well):

(a) gcc-core; gcc-fortran; gcc-g++ – compilers for c, c++, & fortran for compiling the model
(b) nano – a standard text editor (or your favourite Unix one, e.g. emacs or vim) so you can edit
(c) git – for downloading/cloning **muffin** (and updating it in future)
(d) make – for testing and installing the model
(e) Python2
(f) gambas3-gb-xml-xslt – needed for xsltproc
(g) Be
(h) Anything else you fancy?
(i) Then click next for installation to happen

If you forget something, you can simply rerun the install executable and you can select new packages and update old ones (NB this is the only way it updates, so you have to do this occasionally anyway if you want any updates!)

NB-1: This how-to was most recently tested with Cygwin 3.1.2, downloaded/installed 22 Jan 2020.
NB-2: eagle-eyed readers may have noticed we didn’t mention **netCDF** here. There’s a good (and annoying) reason for this, to be revealed shortly
NB-3: you may later find **Cygwin** claiming some really core commands (like rm) is missing, so cgenie won’t compile. If this happens, uninstall and then reinstall **Cygwin** from scratch...

4. **Click Finish when Setup is done,** adding system icons if you want them

5. **Launch Cygwin:** Go find Cygwin Terminal (either from an icon or the Start bar) and launch it – you should get a bash-like terminal popping up ready to go.

6. **Download Netcdf:** This is the other annoying step, in that netcdf should already be available as a library via Cygwin but for irritating reasons (to do with naming, library locations, and compilers) isn’t properly recognised by **muffin** without some tiresome edits. So it’s a lot easier just to download it yourself, install it exactly where you want it, and tell **muffin** where it is. Inelegant, but effective. If you’d prefer to make Cygwin netcdf work instead, then feel free to have a go!

(a) Download the netcdf4 library from the **muffin** website and save it somewhere handy like in your brand new **Cygwin** directory user area (something like C:\cygwin64\home\User) where you can find it using the **Cygwin** terminal

(b) In **Cygwin** Terminal, navigate to where you downloaded the library to (using cd and ls commands) and then unzip and untar (you may need to use a utility like pzip in Windows 10), configure, and install the libraries with the following code:

    tar -xzf netcd4-4.0.tar.gz
cd netcdf-4.0
./configure --disable-shared --prefix=${HOME} FC=gfortran CC=gcc CXX=g+
# N.B. no new line here. This will also take a while...
make check
#this checks netCDF. All the tests should pass. If not, then try again...
make install
#this installs netCDF. It should finish with a congratulations. If not, then try again...
The key thing here is making sure we use *exactly* the same compilers that installing cGENIE will use to make sure it all matches up. Also note that this is an older version of netcdf with both c and fortran libraries combined – this is to make life simpler here, and still works fine

7. Download muffin: In Cygwin Terminal navigate back to your Home directory (where you started) and run the following code:

```bash
cd ..
git clone https://github.com/derpycode/cgenie.muffin.git
```

This clones the entire current cGENIE.muffin repository from github to your computer via Cygwin. Magical!

8. Configure cGENIE settings: Before installing, we need to edit a bit of code to tell cGENIE where we put netcdf. There are 2 files to edit:

(a) Type: cd cgenie.muffin/genie-main to go to the main muffin directory

(b) Type: nano user.mak in your Cygwin terminal to edit user.mak – at the end of the file, the netCDF path will need to be changed because we’ve put netcdf somewhere different (your home user area) to what it’s expecting. Change:
```
NETCDF_DIR=/usr/local
```
to:
```
NETCDF_DIR=$(HOME)
```
and make sure this line is the same:
```
NETCDF_NAME=netcdf
```
To do so, move to the line with your keyboard arrow keys, type it in, and save on exit (in nano, press ‘Ctrl-X’ to exit and then ‘y’ to save changes or ‘n’ to not, then hit enter)

(c) Now type: nano makefile.arc in your Cygwin terminal to edit makefile.arc – towards the end of the file, under the heading:
```
# === NetCDF paths ===
```
uncomment the line under (i.e. remove the # in front):
```
### FOR COMBINED C+FORTRAN NETCDF LIBRARIES #######
```
and comment the 2 lines under (i.e. add a # in front):
```
### FOR SEPERATE C AND FORTRAN NETCDF ###
```
so as to select, combined, rather than separate, netCDF libraries. Save and exit.
9. Test muffin: now we need to check muffin is working.
   (a) First, make sure you’re still in the directory cgenie.muffin/genie-main (type ls to find out). Then enter:

   make testbiogem

   After a whole bunch of text happens and time passes, you should get:

   **TEST OK**

   If not, try:

   make cleanall

   and then try again. If that doesn’t work, see the full muffin manual FAQs for pointers, or check the earlier steps with Cygwin packages, netCDF, and muffin configuration worked properly.

   (b) To see if it works for a proper model run, in the directory cgenie.muffin/genie-main, nano makefile.txt, find the section entitled “CLEANING RULES” and delete the leading backslashes in front of the four rm commands in this section, so those lines look like the following, and save the results:

   `rm -f ...`

   (c) Try out the following code to run a real experiment:

   `./runmuffin.sh cgenie.eb_go_gs_ac_bg.worbe2.BASE LABS LAB_0.EXAMPLE 10`

   with the command structure following:

   `./runmuffin.sh #1 #2 #3 #4 (#5)`

   #1 : The base config. Here it is: cgenie.eb_go_gs_ac_bg.worbe2.BASE
   #2 : The user config directory. Here it is: LABS
   #3 : The user config file (i.e. the experiment name). Here it is: LAB_0.EXAMPLE.
   #4 : Run the experiment for X years. Here it is: 10
   #5 : The (optional) restart file. Here there is no restart, so no 5th parameter passed

   You should see code compiling, and then after a while some results start appearing. After 10 model years (a few minutes), the model should save and finish, and you can find the outputs in cgenie_outputs in your Home directory.

10. Success! If you’ve made it this far, congratulations, you now having muffin running on your Windows machine! Remember that big runs will take a long time though, so beyond tinkering and short runs be prepared to leave your computer running for a lonnnng time. But on the upside, no queuing required!

---

7 no new line
20.8 HOW-TO ... run and submit muffins

runmuffin.sh options

Submitting jobs with Grid Engine

For submitting jobs using Sun Grid Engine (SGE) on a cluster, a basic command\(^8\) would look like this:

\[
\text{
$ qsub -S /bin/bash runmuffin.sh <options>
$
\]

Here: the \(-S /bin/bash\) part is to ensure that SGE uses the BASH shell to submit the job because this is the language that runmuffin.sh has been written in.

Take care that the installed FORTRAN compiler can be seen by the cluster nodes. If not, the muffin executable must have already been built prior to submitting a job. The easiest way to do this is to run muffin interactively briefly (e.g., with a run length of just a couple of years or kill it) and then submit the full run to the cluster.

Other useful submission options for SGE:

• To redirect the standard output stream:

\[
\text{
$ qsub -o genie_log ... 
$
\]

• To redirect the standard error stream:

\[
\text{
$ qsub -e genie_log ... 
$
\]

• To merge standard output and error streams into standard output:

\[
\text{
$ qsub -j y ... 
$
\]

• To specify particular resources, such as the nodes with 8 GB of RAM:

\[
\text{
$ qsub -l mem_total = 8.0G ... 
$
\]

• To decrease the priority of a job\(^9\):

\[
\text{
$ qsub -p 1 ... 
$
\]

• To submit a job from the current working directory:

\[
\text{
$ qsub -cwd 
$
\]

\[
\text{
• Request an email is sent when the job starts and/or when it finishes – see the main pages for qsub for the required syntax).}
\]

\[
\text{
• A complete example for the domino UCR clusters would be:}
\]
$ qsub -q dog.q -j y -o cgenie_log -V -S /bin/bash runmuffin.sh
cgenie.eb_go.gs.ac.bg.worjh2.ANTH / EXAMPLE.worjh2.Caoetal2009.SPIN 10000

which merges standard output and error streams and redirects the resulting file to the directory
˜/cgenie_log. Note that to redirect output as per in this particular example, the directory
˜cgenie_log MUST be present. I have no idea what happens if it is not ... but it can’t be

good ;)

You can check the status of the SGE job queue\textsuperscript{10} with the command:

$ qstat -f

and you can kill a job with the qdel command, the job numbers being given by the qstat command.

\textsuperscript{10}Depending on the cluster setup, it may be possible to graphically check what is going on via the www.
20.9 HOW-TO ... re-start muffins

Modify the ocean inventory of a tracer of a re-started experiment

There are three different ways in which for a closed system, the inventory of a tracer can be modified:

1. Add a flux forcing, to the ocean surface or the ocean as a whole. The tracer change is then the total global flux times the duration of the forcing. Note that the forcing can be positive or negative (to effect a decrease in the tracer inventory)

2. There is a parameter that add to or subtract from, the tracer inventory at the very beginning of an experiment (and assuming it is running on from a re-start). The parameter name is bg_ocn_dinit_nn, where nn is the tracer number (which can be found in the parameter list table PDF, or by inspection of the file tracer_define.ocn (for ocean, dissolved tracers) in the directory cgenie.muffin/genie-main/data/input. For example:
   bg_ocn_dinit_8=1.0E-6
will add 1 μM kg⁻¹ of PO₄ (#8 is the tracer number of dissolved phosphate), uniformly to the ocean. Note that a negative value will result in the subtraction of a uniform concentration form every grid cell in the ocean (meaning that care has to be taken to ensure that negative numbers to not appear following subtraction).

3. There is a variant to the concentration adjusting parameter that is enabled by setting the parameter bg_ctrl_ocn_dinit to false (it is true by default). bg_ocn_dinit_nn now acts as a scaling factor that is applied to the tracer concentration field. The new concentration field is equal to the old concentration field (from the re-start), times \((1.0 + bg_ocn_dinit_nn)\), e.g.:
   bg_ctrl_ocn_dinit=.false.
   bg_ocn_dinit_8=0.5
will result in a 50% increase in the concentration of dissolved phosphate everywhere in the ocean (and a value of 1.0 doubles concentrations). Conversely, a value less than one will result in a proportional reduction everywhere, e.g.:
   bg_ctrl_ocn_dinit=.false.
   bg_ocn_dinit_8=-0.2
generates a 20% decrease everywhere.

Obviously, if the experiment is not being run from a re-start, or is being run from a re-start which does not include the particular tracer, then the initial value of the trace can be set. The parameter name is bg_ocn_init_nn, where nn is the tracer number.
What follows are potted HOW-TO instructions for doing things, split into more technical installing and running the model (previous Chapter) vs. configuring experiments (this Chapter).

There is some overlap with the FAQ Chapter, so please read all three!
21.1 HOW-TO ... diagnose how the system works

This HOW-TO will:

1. Implement a (single-tracer) water mass age tracer.
2. Implement a (dual-tracer) water mass age tracer. (Old scheme.)
3. Implement ‘preformed’ and other diagnostic biogeochemical cycling tracers.

______________________________
Add a water mass age tracer

To diagnose water mass age, a single diagnostic tracer is required – ocn_colr (tracer number 48). Once selected (in the base-config), you only then need set\(^1\) (user-config):

\[
\text{bg_ctrl_force_ocn_age1} = \text{.true.}
\]

The time-series output appears as file biogem_series_misc_col_age.res. 3D netCDF output is provided via the variable \text{misc_col_Dage} ("color tracer; ventilation age") with units of \text{yr}.

The way it works is that initially the ocean everywhere has zero age\(^2\). Then each time-step, the water in every interior (excepting those at the surface) ocean grid cell is aged by the time-step duration. The ocean surface is continually restored to zero (age). A \textit{re-start} age distribution can be directly re-started from.

The advantage of this simplified age tracer (other than its simplicity!!!) is that it frees up the numerical 'blue' tracer for other purposes.

\(^1\) (bg_ctrl_force_ocn_age1 to distinguish it from the earlier scheme (bg_ctrl_force_ocn_age) that required 2 tracers.

\(^2\) Hence, the total spin-up time needs to exceed the maximum age, for an ocean circulation at steady-state, or a run duration a factor several times this for a \textit{spin-up} from cold.
Add a dual-tracer water mass age (original, more complicated schemes)

[Note that this is pretty much entirely superseded by the previously-described, single-tracer alternative.]

Water masses (and hence something of ocean circulation) can be tagged with a color (dye) tracer. However, on its own, this can tell you nothing about water mass age. A second color tracer can be added, however, and configured in such a way that by analysing the ratio of the two tracers, water mass age (time since a parcel of water last saw the surface ocean on average).

First off, you are going to need a base-config that defines both color tracers. An example of this (but with no biology or carbon cycle and a modern configuration) is:

cgenie.eb_go_gs_ac_bg.worjhb2.rb

Obviously, this can be adapted or the 2 lines selecting the 'red' and 'blue' color tracers, copied over into a different base-config (but remembering that the total number of ocean tracers selected then increases by 2).

The way this is going to work is:

1. A restoring of red dye is applied evenly to the entire ocean surface. By itself, this will simply result in the ocean progressively filling up with dye until equal to the surface concentration and with no time information.

2. So a blue dye is also injected. The concentration of this is also restored at the surface. However, the surface concentration of the blue dye is scaled such that it reflects age in the model experiment. This counts 'down', such that at the start of the experiment the dye is at its highest concentration and hence representing the greatest amount of time (age). As the run progresses and time runs towards zero, so does the dye flux. i.e. for an experiment running for 10,000 years, the concentration of blue at the surface linearly declines from 10,000 to 0.

Or alternatively:

3. So far, even with the blue dye reflecting 'time', remote parts of the ocean will not have received much dye, so even though the water should be 'old' and the surface concentration high, the concentration and hence 'age' in the deep ocean will still be low. So the red dye is used to normalize for the dispersion and dilution of the blue dye.

Water mass age tracing can be configured quite simply, and with the tracer concentration manipulations carried out automatically, as follows. As base-config file with both red and blue color tracers defined is still required, but rather than have to define and use a set of forcings (and associated forcing configuration), a single parameter is added in the user-config file:

bg_ctrl_force_ocn_age=.true.

This will automatically create the age tracer and additional explicitly output (in netCDF) both the total age of a water parcel, as well as the age relative to the surface (ventilation age).

In this methodology, in the netCDF output, the concentration ratio of blue/red, should be 'age' – the mean time that a parcel of water was last at the surface.

In comparison, in the original (and still valid way of implementing water mass age tracing), 2 surface ocean forcings need be specified in order to create the combined age tracer. An example forcing is provided: forcing: pyyyy.Rcolr_Rcolb, which is then configured, for a 10,000 year run in this example, by adding the following 2 parameter settings in the user-config:

bg_par_ocn_force_scale_val_49=10000.0
bg_par_ocn_force_scale_time_49=10000.0
(If the experiment duration is longer than 10,000 years, the parameter values need be adjusted accordingly.)

Example `user-config` files for both approaches are provided:

```
EXAMPLE.worjh2.NONE_age.SPIN
EXAMPLE.worjh2.NONE_colage.SPIN
```

Note that the automatic approach (EXAMPLE.worjh2.NONE_colage.SPIN) will handle experiments started from a `re-start` (but not for the manual approach). The only advantage to the manual approach (EXAMPLE.worjh2.NONE_age.SPIN), which is provided for backwards code/experiment compatibility, is that it is possible to specify a surface age for a specific region, e.g. North Atlantic, meaning that the ventilation age is the time since a parcel of water last saw the North Atlantic rather than anywhere at the surface (as in the automatic approach).
Include 'preformed' tracers

Preformed tracers are selected via 2 user-config modifications:

1. By setting the parameter:
   \[
   \text{bg\_ctrl\_bio\_preformed} = \text{true}.
   \]

2. By including sufficient/appropriate 'color' (numerical) tracers in the base-config.

A total of 10 generic 'color' tracers are defined and can be selected (in the base-config), from \text{io\_co10} (tracer number #57) through \text{io\_co19} (tracer number #66), and with the preformed tracer parameter set true, represent:

<table>
<thead>
<tr>
<th>Ocean tracer number</th>
<th>Mnemonic</th>
<th>Represents the preformed version of the tracer</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>io_co10</td>
<td>DIC</td>
</tr>
<tr>
<td>58</td>
<td>io_co11</td>
<td>ALK</td>
</tr>
<tr>
<td>59</td>
<td>io_co12</td>
<td>(O_2)</td>
</tr>
<tr>
<td>60</td>
<td>io_co13</td>
<td>(PO_4)</td>
</tr>
<tr>
<td>61</td>
<td>io_co14</td>
<td>(NO_3)</td>
</tr>
<tr>
<td>62</td>
<td>io_co15</td>
<td>(Fe \text{ (i.e. } Fe^{3+}) \text{ or } TDFe)</td>
</tr>
<tr>
<td>63</td>
<td>io_co16</td>
<td>(SiO_2)</td>
</tr>
<tr>
<td>64</td>
<td>io_co17</td>
<td>(\delta^{13}C ) of DIC</td>
</tr>
<tr>
<td>65</td>
<td>io_co18</td>
<td>(\Delta^{14}C \text{ (of } DIC) \text{ or } \delta^{13}C \text{ of } C_{soft})</td>
</tr>
<tr>
<td>66</td>
<td>io_co19</td>
<td>(C_{soft})</td>
</tr>
</tbody>
</table>

(Obviously, you need to have the respective ocean tracer selected.)

One can have a generic base-config, with all 10 possible color tracers selected, and then only for the selected corresponding tracers will a preformed tracer be simulated and output generated. Or one can just select the specific color tracer corresponding to a selected biogeochem. tracer. For the \(C_{soft}\) tracer, in addition to the option:

\[
\text{bg\_ctrl\_bio\_preformed} = \text{true}.
\]

you also need to ensure that the redox diagnostics are saved\(^3\):

\[
\text{ctrl\_bio\_remin\_redox\_save} = \text{true}.
\]

You can also specify whether the \(C_{soft}\) tracer contains just the carbon remineralized from \(POC\), or from both \(POC\) and \(DOC\). By default, \(DOC\) is included in the \(C_{soft}\) tracer field. To exclude \(DOC\) from the regenerated \(DIC\) field, set:

\[
\text{ctrl\_bio\_preformed\_CsoftPOConly} = \text{true}.
\]

(by default the parameter is false and \(DOC\) is also included in accounting for remineralization).

Additionally, if \text{io\_co18}) is selected and radiocarbon (e.g. \text{io\_DIC\_14C}) is not, then the \(\delta^{13}C\) of \(C_{soft}\) is accounted for.

Preformed tracer output is saved in both time-series and time-slice form.

---

\(^3\)Unless you select a data save option that already includes this. But safest is to explicitly set this true.
• In addition to the standard tracer output – biogem_series_ocn_colx.res (where x is one of 0 through 9) – time-series output is also saved under the files: biogem_series_diag_pre_yy.res, where yy is the preformed tracer short name. For isotopes, the latter output provides values in units of per mill (‰) as well as mol. (The former, only as mol and concentrations.)

• netCDF output also includes the basic tracer values (as both 2D and 3D) – ocn_colx.res (where x is one of 0 through 9), and additionally, a series of (3D only) outputs: diag_pre_yy, where yy is the preformed tracer short name. For isotopes, diag_pre_yy variables are in units of per mill (‰). (The former, only as mol and concentrations.)

i.e. for both time-series and time-slice output, the diag_pre forms of output are the recommended ones.

Note that technically, C_{soft} is not a 'preformed' tracer but rather a 'remineralized' tracer. However, for easy of grouping the primary color tracer output together, it takes the same prefix as the other tracers in the saved variable notation.

A variety of 'derived' output are also included (depending on tracer selection) in both time-series and time-slice formats. Most (if not all) of the time-series variables are provided as a sort of check of tracer integrity (and to save on further processing.)

### time-series:

1. diag_reg_13Csoft – δ^{13}C of C_{soft}
2. diag_reg_13CsoftPLUSpre13C – δ^{13}C of C_{soft} plus δ^{13}C of preformed DIC.
   In the absence of CaCO_3, this should be equal to (mean ocean) DIC δ^{13}C.
3. diag_reg_CsoftPLUSpreDIC – C_{soft} plus preformed DIC.
   In the absence of CaCO_3, this should be equal to (mean ocean) DIC.
4. diag_reg_O2PCsoftPLUSpreO2 – C_{soft} converted to respired O_2 via the C : O_2 Redfield ratio, plus preformed O_2.
   This should be equal to (mean ocean) O_2.
5. diag_reg_PCsoftPLUSprePO4 – C_{soft} converted to regenerated PO_4 via the C : PO_4 Redfield ratio, plus preformed PO_4.
   This should be equal to (mean ocean) PO_4.

### time-slices:

1. diag_reg_O2sat – O_2(sat) – O_2 solubility calculated at every grid point using ambient (T,S).
2. diag_reg_AOU – AOU – estimated from O_2(sat) minus 'real' O_2 (i.e. O_2).
3. diag_reg_AOU_P – regenerated P – estimated from AOU multiplied by the P : O_2 Redfield ratio.
4. diag_reg_AOU_P_C – regenerated C – estimated from AOU multiplied by the P : O_2 Redfield ratio and then the C : P ratio.
5. diag_reg_AOU_P_C_d13C – regenerated δ^{13}C – estimated from AOU multiplied by the δ^{13}C of C_{soft} is the isotopic value of the remineralized carbon, not the contribution to δ^{13}C of DIC. See following section.

\(^4\)Note that the δ^{13}C of C_{soft} is the isotopic value of the remineralized carbon, not the contribution to δ^{13}C of DIC.
\( P : O_2 \) Redfield ratio and then the \( C : P \) ratio and then multiplied by \( \delta^{13}C \) of (overlying ocean) export POC ... to the moon!
The isotopic value is 'diluted' in the proportion of regenerated \( C \) over DIC, i.e. it is an estimate (from AOU) of the DIC \( \delta^{13}C \) contribution from respired carbon.

6. \( \text{diag}_\text{reg}_\text{OU} \) – 'real' oxygen utilization (\( OU \)) – calculated as preformed \( O_2 \) minus 'real' \( O_2 \) (\( \text{io}_\text{O2} \)).

7. \( \text{diag}_\text{reg}_\text{P} \) – 'real' regenerated \( P \) – calculated as actual \( PO_4 \) (\( \text{io}_\text{PO4} \)) minus preformed \( P \).

8. \( \text{diag}_\text{reg}_\text{P}_\text{C} \) – regenerated \( C \) – estimated as regenerated \( P \) multiplied by the \( C : P \) Redfield ratio.

9. \( \text{diag}_\text{reg}_\text{P}_\text{C}_\delta^{13}C \) – regenerated \( \delta^{13}C \) – estimated from regenerated \( P \) multiplied by the \( C : P \) Redfield ratio and then multiplied by \( \delta^{13}C \) of (overlying ocean) export POC. The isotopic value is 'diluted' in the proportion of regenerated \( C \) over DIC.

10. \( \text{diag}_\text{reg}_\text{C} \) – regenerated \( C \) – this is a simple copy of the \( C_{soft} \) tracer for completeness and is 'real' regenerated \( C \).

11. \( \text{diag}_\text{reg}_\text{C}_\delta^{13}C \) – regenerated \( \delta^{13}C \) – perhaps a little pointlessly ... estimated from ('real') regenerated \( C \) (aka \( C_{soft} \)) multiplied by \( \delta^{13}C \) of (overlying ocean) export POC. The isotopic value is 'diluted' in the proportion of regenerated \( C \) over DIC.

12. \( \text{diag}_\text{reg}_\text{C}_\delta^{13}C \) – regenerated \( \delta^{13}C \) – this is simply the \( C_{soft} \) \( \delta^{13}C \) tracer, but, 'diluted' in the proportion of regenerated \( C \) over DIC. i.e. it is an explicit (and 'real') measure of the \( \delta^{13}C \) contribution of respired carbon to DIC.

13. \( \text{diag}_\text{reg}_\text{DIC} \) – DIC (not really 'regenerated' \( per \ se \)) – calculated as the sum of preformed \( C \) plus regenerated \( C \). This is provided as a simple 'check' on the tracers, and in the absence of \( CaCO_3 \), should be equal to 'real' DIC (\( \text{io}_\text{DIC} \)).

14. \( \text{diag}_\text{reg}_\text{DIC}_\delta^{13}C \) – \( \delta^{13}C \) of DIC (not really 'regenerated' \( per \ se \)) – calculated as the weighted sum of preformed \( C \) plus regenerated \( C \) \( \delta^{13}C \). This is provided as a simple 'check' on the tracers, and in the absence of \( CaCO_3 \), should be equal to 'real' DIC \( \delta^{13}C \) (\( \text{io}_\text{DIC}_\text{13C} \)).
Diagnose a Transport Matrix

The physical circulation in muffin can be diagnosed as a Transport Matrix: a representation of the steady-state average circulation as a sparse matrix (see Khatiwala et al. [2005] Accelerated simulation of passive tracers in ocean circulation models. Ocean Modelling. 9 (1), pp. 51 - 69). There are a number of advantages of using a transport matrix such as finding preformed tracer distributions and using it as a surrogate circulation for a steady-state biogeochemical model (see Khatiwala [2007] – A computational framework for simulation of biogeochemical tracers in the ocean. Global Biogeochemical Cycles. 21 (3), GB3001).

At its most simplest, the method sets a passive tracer (i.e., affected by ocean circulation only) to 1 \(\mu\text{mol kg}^{-1}\) in a single grid-box, leaving every other grid-box as 0 \(\mu\text{mol kg}^{-1}\). The model is then integrated for a single timestep. The whole grid is then vectorised, i.e., grid-boxes are stacked on top of each other, and this forms one column of the matrix. This process is repeated for the next grid-box along forming the second column. The columns of the matrix represent the distribution of the tracer from each grid-box individually after one timestep. To diagnose a matrix within GENIE, a few extra steps are taken: 6 tracers are used in a gridded pattern to diagnose one depth level simultaneously; each depth level is diagnosed and averaged in a year in turn as the number of grid-boxes affected by convection is not predictable.

Diagnosing within a GENIE run

A full matrix can be diagnosed within a single run. To do so, first add the following six "colour" tracers selected to the base-config file:

```plaintext
gm_ocn_select_57=.true.
gm_ocn_select_58=.true.
gm_ocn_select_59=.true.
gm_ocn_select_60=.true.
gm_ocn_select_61=.true.
gm_ocn_select_62=.true.
```

(and remembering to increase the total number of selected ocean tracers).

Then in the user-config file, set the following parameters values in order to diagnose a transport matrix:

1. By ensuring preformed tracers are not implemented:
   ```plaintext
   bg_ctrl_bio_preformed=.false.
   ```

2. Select the transport matrix flag:
   ```plaintext
   bg_ctrl_data_diagnose_TM=.true.
   ```

3. Select the year to start diagnosing the matrix. The matrix requires the same number of years to diagnose as there are depth levels in the configuration, e.g., 16 years with a 16-level configuration.
   ```plaintext
   bg_par_data_TM_start=9980
   ```

4. Select the averaging interval. Setting to 1 will give one annual average matrix, 4 will give four seasonally averaged matrices and 12 will give monthly averaged matrices.
   ```plaintext
   bg_par_data_TM_avg_n=1
   ```

There are also a few other things to consider. The model will try and warn you...

- The "colour" tracers are used for other purposes such as preformed tracers. Diagnosing preformed tracers at the same time probably isn’t a good idea.
bg_ctrl_bio_preformed=.false.

- If you want to diagnose seasonal or monthly matrices, ensure you have seasonal variability otherwise you get will get a number of identical matrices:
  ea_dosc=.true.
  go_dosc=.true.
  gs_dosc=.true.

- If you want to diagnose seasonal or monthly matrices, you may want to save other output at the same intervals by setting the following (this is not the same as the matrix averaging parameter! Set to 24 for seasonal or 8 for monthly):
  bg_par_data_save_slice_n=24

Lastly, **biogem** needs to run with a 1:1 time-step ratio with **goldstein**. You can either edit the runmuffin.sh script yourself or better, use the runmuffin.TM.sh script which sets this ratio to 1 automatically.

The transport matrix data will be written to a netCDF file named transport_matrix_COO.nc in the biogem output (COO stands for the specific format of the sparse matrix). See the next sections on what to do with it...

### Loading the Transport Matrix in MATLAB

To load the Transport Matrix into MATLAB, copy transport_matrix_COO.nc to a directory MATLAB can see. Use the load_genie_matrix.m script from [www.github.com/derpycode/muffindata/Transport_Matrix](https://www.github.com/derpycode/muffindata/Transport_Matrix) to read the file and create the matrix in MATLAB and save the matrices to a file whatever you specified. A matrix is conventionally called \( A \). If there are multiple matrices, they are saved under one file as \( A1, A2 \) etc... e.g.

```
load_genie_matrix ( 'transport_matrix_COO.nc', 'output_filename')
```

The script also saves a file called matrix_vars.mat containing metadata:

1. \( v \_index \) - a structure array with 4 vectors containing the indices for longitude \( (i) \), latitude \( (j) \) and depth \( (k \) with 16 as surface, \( rk \) with 1 as surface). These are required to convert between a 3-D grid to vector formats (see below).
2. \( nb \) - the number of wet grid-boxes
3. \( Ib \) and \( Ii \) - indices containing the location of the surface and interior ocean grid-boxes in the matrix

You should perform a basic check on the transport matrices before using them to check there were no issues in diagnosing them. Each row of the matrix must sum to 1.0 to ensure mass conservation. In practice, due to the way the matrix is averaged over a single run, this is never exact but should be close within several decimal places. Running \( \text{sum}(A,2) \) will find the sum of each row. \( \text{sum}(\text{sum}(A,2)) \) should also equal the total number of boxes \( nb \). Hopefully, it should be obvious that the circulation should be as close to steady state as possible in your run!

### Useful MATLAB Functions

Additional MATLAB functions to help process and use transport matrices can be found at [www.github.com/derpycode/muffindata/Transport_Matrix](https://www.github.com/derpycode/muffindata/Transport_Matrix).

1. \( v2f.m \) and \( f2v.m \) - convert between vector and 3-D grid. Takes the vector indices in \( v \_index \) as arguments.
2. \( \text{read_genie_netcdf.m} \) - reads one or multiple variables from a genie netCDF output file into your MATLAB workspace. It will read in a 3-D field or convert to a vector format.
3. `split_transport_matrix.m` - splits a transport matrix into a boundary (surface grid-boxes only) and interior matrices, needed when solving for steady state solutions. The function takes the `Ib` variable as an argument.
21.2 HOW-TO ... screw with the climate system
Set/un-set seasonal insolation forcing

Seasonal insolation forcing of the EMBM, GOLDSTEIN ocean, and sea-ice model, are set by the following parameters:

ea_dosc=.true.
go_dosc=.true.
gs_dosc=.true.

and are .true. by default. To set an annual average insolation forcing with no seasonality, simply set these to .false.

---

5These are typically set in the base-config if needed (i.e. different from default).
Implement 'brine-rejection'

muffin has the capability to include this effect (at least crudely) and similarly to Bouttes et al. [2010]. For this, three namelist parameter values need to be set:

bg_ctrl_force_GOLDSTEInTS=.TRUE.
bg_par_misc_brinerejection_frac=0.1
bg_par_misc_brinerejection_jmax=9

The first, simply allows the BIOGEM biogeochem. module to directly influence ocean circulation. The second is the fraction of salt, rejected during sea-ice formation (e.g., see Bouttes et al. [2010]) that is transferred directly to the bottom-most (underlying) ocean cell in the model. The third sets a latitude limit (counted in cells) to the effect – on a 18 $\times$ 18 grid, a value of 9 will restrict brine rejection to the southern hemisphere (and hence Southern Ocean); a value of 18 will allow it to take place in the North Atlantic as well. (Note that in e.g., Bouttes et al. [2010], the effect of brine-rejection is considered only in the Southern Ocean.)

There is also option for treating all tracers the same way (including DIC ALK, PO4, etc.) and translocate biogeochem. tracers proportionally to salinity.
21.2 HOW-TO ... screw with the climate system

Add/modify geothermal heating'
(see: HOW-TO ... force the system)
### 21.3 HOW-TO ... iron (and dust)

This HOW-TO will:

1. Provide an overview of the different marine iron cycle schemes in **muffin**.
2. Describe the different options for tracers describing the main players in marine iron cycle.
3. Detail the various calibrated parameter sets.
4. Describe alternative ways of prescribing the solubility of iron in dust.
5. Introduce ... fake dust(!) (for paleo configurations).
6. Describe other sources of dissolved iron to the ocean.

---

**Overview of marine iron cycle schemes in muffin**

There are 3 different ways of describing the marine iron cycle in terms of which tracers in the ocean are defined (described immediately below). These are associated with one or more alternative tunings of the primary parameters controlling iron cycling – the solubility of iron in dust and the rate of scavenging of dissolved iron onto particulate organic matter, plus, some parameter scaling the characteristic biological nutrient (PO$_4$) uptake time-scale (described subsequently). In theory, the different tracer configurations are interchangeable and either could be used in conjunction with any given tuned parameter set. However, in practice and in publications, specific tuned parameter sets have been associated with a specific tracer combination choice (and e.g. not tested equally with both). The various available marine iron cycle configurations are summarized in Table 21.1.

### Table 21.1: muffin iron cycle configurations

<table>
<thead>
<tr>
<th>#</th>
<th>Tracers</th>
<th>Scheme</th>
<th>DOM$_\tau$ (yr)</th>
<th>bio$\tau$ (d)</th>
<th>Fe$_{\text{sol}}$ (%)</th>
<th>$k_{\text{scav}}$</th>
<th>Dust field</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fe, L, FeL</td>
<td>OLD</td>
<td>1.0</td>
<td>63.3</td>
<td>0.291</td>
<td>1.34</td>
<td>Mahowald</td>
<td>Tagliabue et al. [2016]</td>
</tr>
<tr>
<td>2</td>
<td>Fe, L, FeL</td>
<td>OLD</td>
<td>1.0</td>
<td>63.3</td>
<td>0.291</td>
<td>1.34 (various)</td>
<td>Lambert, et al. [2021]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Fe, L, FeL</td>
<td>(OLD)</td>
<td>1.0</td>
<td>63.3</td>
<td>n/a</td>
<td>n/a (none)</td>
<td>Meyer et al. [2016]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hülse et al. [2019]</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TDFe, TL</td>
<td>hybrid</td>
<td>1.0</td>
<td>95.6</td>
<td>0.201</td>
<td>0.344</td>
<td>Mahowald</td>
<td>Ward et al. [2018]</td>
</tr>
<tr>
<td>5</td>
<td>TDFe, TL</td>
<td>hybrid</td>
<td>1.0</td>
<td>n/a</td>
<td>0.201</td>
<td>0.344</td>
<td>Mahowald</td>
<td>Ward et al. [2018]</td>
</tr>
<tr>
<td>6</td>
<td>TDFe, TL</td>
<td>hybrid</td>
<td>0.5</td>
<td>60.0</td>
<td>0.125</td>
<td>0.225</td>
<td>Mahowald Albani</td>
<td>De’Ath et al. [in prep]</td>
</tr>
<tr>
<td>7</td>
<td>TDFe, TL</td>
<td>hybrid</td>
<td>0.5</td>
<td>60.0</td>
<td>0.100</td>
<td>0.100</td>
<td>Mahowald Albani</td>
<td>De’Ath et al. [in prep]</td>
</tr>
<tr>
<td>8</td>
<td>TDFe, TL</td>
<td>hybrid</td>
<td>0.5</td>
<td>60.0</td>
<td>0.244</td>
<td>0.225</td>
<td>Albani</td>
<td>Ödalen et al. [in prep]</td>
</tr>
<tr>
<td>9</td>
<td>TDFe, TL</td>
<td>hybrid</td>
<td>0.5</td>
<td>n/a</td>
<td>0.244</td>
<td>0.225</td>
<td>Albani</td>
<td>Jones et al. [in prep]</td>
</tr>
<tr>
<td>10</td>
<td>Fe, Fe2, TL</td>
<td>FeFe2TL</td>
<td>0.5</td>
<td>63.3</td>
<td>n/a</td>
<td>0.344 (none)</td>
<td>van de Velde et al. [2021]</td>
<td></td>
</tr>
</tbody>
</table>

---

**NOTES:**

- #3 – The tracers are present, but the iron cycle is ‘disabled’ so that the biological scheme operates with no Fe limitation.
- #5 – There is no bio$\tau$ value because the ECOGEM module is used.
- #7 – Tests an assumption of a spatially uniform iron solubility ($Fe_{\text{sol}}$ values in italics).
- #9 – There is no single bio$\tau$ value (the biological scheme is split into diatoms and non-diatoms).
Iron schemes and tracer configurations

In the original iron cycle in muffin, 3 separate tracers were defined:

- \( \text{Fe} \) – free dissolved iron
- \( L \) – free ligands
- \( \text{FeL} \) – iron bound to ligands

The associated tracer selection in the base-config looks like:

- `gm_ocn_select_9=.true. # Fe -- 'dissolved iron (Fe)'
- `gm_ocn_select_23=.true. # FeL -- 'ligand-bound Fe'
- `gm_ocn_select_24=.true. # L -- 'free ligand (iron binding)'

However, because the equilibrium between these tracers is re-solved every time-step (and used to determine the free \( \text{Fe} \) concentration for scavenging), there is redundancy, and in practice, only 2 tracers are needed. Subsequent iron cycle scheme hence use only 2 tracers:

- \( \text{TDFe} \) – total dissolved iron, including iron bound to ligands
- \( \text{TL} \) – total dissolved ligand concentration (both free and bound to iron)

The associated tracer selection in the base-config then looks like:

- `gm_ocn_select_90=.true. # TDFe -- 'total dissolved Fe'
- `gm_ocn_select_42=.true. # TL -- 'total dissolved ligand'

At each time-step, the concentration of 'free' iron (and iron bound to ligands) is solved for and used to calculate iron removal though scavenging.

But ... to distinguish between reduced (\( \text{Fe}^{II} \)) and oxidized (\( \text{Fe}^{III} \)) iron as well as account for the role of ligands ... 3 tracers are needed again (a sort of combination of the above two permutations):

- \( \text{Fe} \) – total dissolved iron III, including iron bound to ligands
- \( \text{Fe2} \) – total dissolved iron II (does not bind to ligands)
- \( \text{TL} \) – total dissolved ligand concentration (both free and bound to iron III)

or as tracers:

- `gm_ocn_select_9=.true. # Fe -- 'dissolved iron III (Fe)'
- `gm_ocn_select_83=.true. # Fe -- 'dissolved iron II (Fe)'
- `gm_ocn_select_42=.true. # TL -- 'total dissolved ligand'

Associated with these different sets of tracer and iron cycle configurations, are several different schemes for how the iron cycle is parameterized:

OLD – The very original scheme – based on 3 tracers (free iron, free ligand, ligand-bound iron), solves the equilibrium between them and based on this, scavenges free \( \text{Fe} \) from the water column, by any or all of: particulate organic matter, \( \text{CaCO}_3 \), opal, and/or detrital matter.

Note that surface layer scavenging is calculated within sub_calc_bio_uptake rather than as part of particulate remineralization subroutine (sub_calc_bio_uptake).
Iron speciation ($Fe$, $FeL$, and $L$) is updated within the main BIOGEM loop. This scheme was used e.g. in the ‘FeMIP’ paper of Tagliabue et al. [2016].

ALT – As per OLD, but with surface layer scavenging calculated within the particulate remineralization subroutine (sub_calc_bio_uptake).
Iron speciation ($Fe$, $FeL$, and $L$) is updated within particulate remineralization subroutine (sub_calc_bio_uptake).
This scheme has not yet been published with(!) although it should be very little different (if at all) from OLD. (Numerically, it is preferable to OLD.)

hybrid – As per ALT, but employing only 2 tracers – total dissolved iron, and total dissolved ligand (and solving for the equilibrium between (free iron, free ligand, and ligand-bound iron).
This is the recommenced modern modern (oxic ocean) configuration option (e.g. as used by Ward et al. [2018]).
In theory ... it should be identical in behavior to ALT (but with one fewer tracer simulated in the ocean).

lookup_4D – Yet to be fully implemented/published ... this assumes 2 tracers in the ocean (total dissolved iron, total dissolved ligand) and calculates the equilibrium partitioning between dissolved iron and solid FeOOH (assumed to be goethite), based on ambient temperature and pH (in addition to total dissolved iron, total dissolved ligand) and employing a lookup table. Solid FeOOH is the phase that is scavenged.

FeFe2TL – A 3-phase tracer scheme, but distinguishing the oxidized form $Fe^{III}$ (tracer: $Fe$) from reduced $Fe^{II}$ (tracer: $Fe2$). This scheme is primarily intended for predominantly ferruginous/euxinic deeper time esp. Precambrian) oceans, but is perfectly capable of being used in more geologically recent and predominantly oxic ocean times (as well as more 'mixed state' oceans, such as during OAE's or during the Paleozoic).
Only $Fe^{III}$ (tracer: $Fe$) is taken into account in the calculation for partitioning iron into free and ligand-bound iron, and only free (non ligand-bound) $Fe^{III}$ is then subject to scavenging.
Reduced $Fe^{II}$ (tracer: $Fe2$) is assumed truly dissolved and does not bind with ligands, nor is subject to scavenging.
In principal, in the absence of $is_POM_FeOOH$, the marine iron cycle should function identically to the original 'oxic-only schemes, with the exception that any dissolved iron that is present in the water column as $Fe^{II}$, will both reduce the rate of scavenging and of biological uptake (both processes only accounting for the availability of $Fe^{III}$).

A list of the different Fe-related processes, and which processes operate under which schemes, follows. See van de Velde et al. [2021] for further details.

(ALL) Biological uptake (cellular incorporation)
Iron is assumed to be co-limiting to biological uptake in a number of biological schemes in muffin\textsuperscript{7}, e.g. bio_PFe, as well as being a co-limiting nutrient in ECOGEM.
Bioavailable iron is assumed to be the sum of both 'free' and ligand-bound iron, which in the case of OLD and ALT is $Fe + FeL$, in hybrid and lookup_4D this is $TDFe$ (which implicitly includes both free and ligand-bound phases), and in FeFe2TL, is $Fe$ (which is equivalent to $Fe^{III}$).

\textsuperscript{6}In the water column, $Fe^{II}$ oxidizes to $Fe^{III}$ under oxic conditions, while $Fe^{III}$ is reduced under anoxic conditions.
\textsuperscript{7}Set by parameter: bg_par_bio_prodopt.
In the case of FeFe2TL, iron is assumed to be incorporated intracellularly and remineralized in its oxidized form. In reality, intercellular iron is reduced, and ideally, remineralized iron released back to the water column would be in the form of Fe(III).

**Ligand-binding (protection (stabilization) from scavenging)**

All schemes assume the presence of a single iron-binding ligand (although the ligand concentration could be set to zero). The equilibrium distribution between free iron (which in the case of FeFe2TL is Fe^{3+}, and all other schemes total dissolved iron (tracer TDFe or just Fe), free ligand, and iron-bound ligand, is solved each time-step for the purpose of calculating iron scavenging (see below).

The ligand stability constant is set via parameter: bg_par_{K_{FeL}}.

Future developments could consider multiple ligand classes, or a ligand cycle with sources and sinks.

**Scavenging**

Typically, scavenging only occurs (or is only configured to occur) by particulate organic matter.

Scavenging occurs on any ‘free’ iron – iron not bound to ligands. In the case of scheme FeFe2TL, only Fe^{3+}

Scavenged iron, by default, is not ‘remineralized’ along with the (organic matter) particulate host and hence not returned to solution, and represents a loss (sink) of bio-available iron from the ocean. This behavior is controlled by parameters:

- # no scavenged regeneration
  bg_par_{scav_free}min=0.0

  which prevents return of iron scavenged by (e.g. particulate organic) matter when it is remineralized (dissolved) in the water column, and

- # return POFe
  bg_ctrl_bio_{NO_fsedFe}.false.

  which prevents scavenged iron being ‘reflected’ at the ocean floor (along with all the constituents of its particulate matter carrier).

The modifier (scaling) of scavenging rate ($k_{scav}$) for each particle type is set by the following parameters:

- bg_par_{scav_Fe_sf_POC}
- bg_par_{scav_Fe_sf_CaCO3}
- bg_par_{scav_Fe_sf_opal}
- bg_par_{scav_Fe_sf_det}

Calibrated values for $k_{scav}$ for POM are given in Table 21.1.

Note that the simplest way to ‘turn off’ scavenging is to set these parameters to zero.

**Isotopic fractionation**

With the current exception of hybrid, are isotope-enabled ($^{56}$Fe).

**Dissimilatory iron reduction**

If the solid (sediment) tracer is_POM_{FeOOH} is selected and the paired ocean tracer io_{FeOOH} is selected (so technically the scheme should be: ’FeFe2TLFeOOH’), when ambient dissolved $Fe^{2+}$ is not currently included in being ‘bioavailable’

---

8 Note that for reasons of simplifying the accounting for oxygen mass balance and not mixing different oxidation states during biological uptake, $Fe^{2+}$ is not currently included in being ‘bioavailable’

9 They are all zero by default anyway.
oxygen is low, the scavenged iron (III) can undergo further transformations and supply oxygen (and be reduced) associated with the remineralization of organic matter. In this, Fe$^{II}$ is released, in contrast with the default parameter settings under which scavenged iron is always ‘lost’ to the system. A pair of half-saturation and inhibition constant primarily control the behaviour of this reaction (and are not listed here) – see: van de Velde et al. [2021] for details. If is_POM_FeOOH is not selected (but the standard scavenged iron tracer, is_POM_Fe is), then the ‘fate’ of scavenged iron (now as a solid) proceeds as ‘normal’ (e.g. under under schemes OLD, ALT or hybrid).

FeFe2TL **Ferrosoiron oxidation**

Under ‘oxidizing’ conditions – defined as the (non-zero) presence of free $O_2$ in the water column – $Fe^{2+}$ is oxidized to $Fe^{3+}$.

The relevant subroutine (sub_box_oxidize_Fe) is called automatically if the required tracers are selected. The kinetic rate constant controlling the reaction, and van de Velde et al. [2021] default value, is:

\[ bg_par_bio_remin_kFe2toFe = 1.0 \times 10^9 \]

(so setting this to zero will disable iron oxidation).

FeFe2TL **Sulphide-mediated iron reduction (1)**

Under ‘reducing’ conditions – defined as the (non-zero) presence of free $H_2S$ in the water column – sulpha-mediated iron reduction occurs, and $Fe^{3+}$ is reduced to $Fe^{2+}$.

The relevant subroutine (sub_box_reduce_Fe) is called automatically if the required tracers are selected. The kinetic rate constant controlling the reaction, and van de Velde et al. [2021] default value, is:

\[ bg_par_bio_remin_kFetoFe2 = 2.3 \times 10^6 \]

(so setting this to zero will disable $Fe^{3+}$ reduction).

FeFe2TL **Sulphide-mediated iron reduction (2)**

Under ‘reducing’ conditions – defined as the (non-zero) presence of free $H_2S$ in the water column – sulpha-mediated iron reduction occurs, and (solid) FeOOH is reduced to $Fe^{2+}$.

The relevant subroutine (sub_box_react_FeOOH_H2S or sub_box_react_POMFeOOH_H2S) is called automatically if the required tracers are selected. The kinetic rate constant controlling the reaction, and van de Velde et al. [2021] default value, is:

\[ bg_par_bio_remin_kFeOOHtoFe2 = 1.7 \times 10^4 \]

(so setting this to zero will disable FeOOH reduction).

This reaction is applied to both ‘free’ solid FeOOH and POM-bound FeOOH (depending on which solid tracers have been selected).

FeFe2TL **Iron oxide precipitation (1)**

This is the solid (sediment) tracer: is_FeOOH

In the water column, if the concentration of $Fe^{3+}$ (total, including ligand-bound) exceeds some threshold (e.g. 1nM), as specified by the parameter (and van de Velde et al. [2021] default value):

\[ par_FeOOH_Fethresh = 1.0 \times 10^{-9} \]

solid FeOOH forms. This then sinks down through the water column along (but distinct from) other particles. All dissolved $Fe^{3+}$ above the threshold concentration is assumed precipitated (i.e. there are no kinetics involved in the reaction).

The relevant subroutine (sub_calc_precip_FeOOH) is called automatically if the required tracers are selected. The occurrence of FeOOH precipitation is also governed by the parameter:
21.3 HOW-TO ... iron (and dust)

bg_ctrl_bio_FeOOHprecip_explicit

which is .true. by default and precipitation can be disabled by changing this (or setting the precipitation threshold to some impossibly large concentration of dissolved iron).

Note that both forms (‘free’ and POM-associated) of FeOOH can precipitate simultaneously if both solid tracers are selected.

FeFe2TL **Iron oxide precipitation (2)**

If the solid (sediment) tracer *is_POM_FeOOH* is selected, FeOOH precipitation occurs associated with (or scavenged by) sinking POM. The amount/rate of precipitation is now controlled by the scavenging calculation rather than the explicit FeOOH precipitation subroutine\(^\text{10}\). The POM-associated FeOOH then sinks along with (and assumed bound to) POM. Note that by default, as per ‘normal’ scavenged Fe, dissolved iron is not released to the water column as the carrier POM is degraded (and is assumed lost to the sediments).

The same (POM) scavenging rate scalar (\(k_{\text{scav}}\)) as detailed earlier, controls this process.

Note that both forms (‘free’ and POM-associated) of FeOOH can precipitate simultaneously if both solid tracers are selected.

FeFe2TL **Pyrite precipitation**

If the solid pyrite tracer (\(FeS_2\)) is selected, then pyrite can form in the water column (from where it will settle down, unaltered, to the sediment surface).

The relevant subroutine (*sub_calc_precip_FeS2*) is called automatically if the required tracers are selected. The controlling parameters are ... complicated\(^\text{11}\). (See: *van de Velde et al.* [2021] for details ...)

The overall precipitation rate is governed by:

\[
bg\_par\_bio\_FeS2precip\_k=3.25e+03
\]

Note that FeS can also be selected as an explicit precursor solid tracer (calculated via subroutine *sub_calc_form_FeS* if this tracer is selected). This scheme is not published and will not described any further here.

FeFe2TL **Siderite precipitation**

Whatever. Probably doesn’t occur to any degree enough to worry about ever (mostly).

Occurs if you select it (solid/sedimentary tracer for FeCO\(_3\) (is_FeCO3)).

(Gonna sink to the bottom and do nothing in between.)

See: *van de Velde et al.* [2021] for details ...

FeFe2TL **Greenalite precipitation**

I don’t even know what this is.

It happens if you select it (solid/sedimentary tracer for Fe\(_3\)Si\(_2\)O\(_4\) (is_Fe3Si2O4)).

You have got to have dissolved silica ... lots and lots of it. So much so in fact that it will crash updating of carbonate chemistry and you will have to remove silicic acid from the definition of ‘alkalinity’ (*gm_ctrl_carbchem_noH3SiO4=.true.*).

(Gonna sink to the bottom and do nothing in between.)

See: *van de Velde et al.* [2021] for details ...

---

\(^{10}\) i.e. in the scavenging process, rather than scavenge iron as Fe, the same amount of scavenged iron is instead assumed to be in the form of FeOOH.

\(^{11}\) Please, for the love of god, leave this alone.
Iron diagnostic output

In addition to the actual ocean tracers, a set output variables are provided to break down (in some cases) and coalesce (in others), iron (and ligand) speciation.

For netCDF (and time-slice output), provided is:

- Parameter sets

User-configs (and the associated base-configs) can be found for most, if not all, of the published papers involving the iron cycle, on GitHub. For the more recent marine iron cycle configurations and tunings, the following example configuration files are provided in \genie-userconfigs\EXAMPLES (with the numbers corresponding to those in Table 21.1.). (Note that the names of the user-configs incorporate the corresponding base-config name to avoid any confusion re. which user-config should go with which base-config.)

#1,2 – cgenie.eb_go_gs_ac_bg.worjh2.BASESFe.FeMIP.SPIN
The marine iron cycle configuration used in the FeMIP inter-comparison paper of Tagliabue et al. [2016] (and other subsequent papers – see: Table 21.1). Note that the same marine iron cycle parameters, but with a variety of different dust fields, was used by Lambert, et al. [in revision].

#3 – cgenie.eb_go_gs_ac_bg.worjh2.BASESFe.NOFe.SPIN
The same biological configuration as cgenie.eb_go_gs_ac_bg.worjh2.BASESFe.FeMIP.SPIN, but with iron limitation of biological export ‘disabled’ by:
1. Setting a Fe half-saturation value of 0.1 pM Fe, which is to all practical purposes, zero (in comparison: the ocean is initialized with 0.6 nM Fe).
2. Setting a fixed Fe:C ratio and specifying the C/Fe organic matter ratio as effectively infinite (so that virtually no Fe is taken up in organic matter as compared to carbon).
3. Setting the scavenging rate of Fe by sinking organic matter to zero.
No dust field is required in this configuration, because there are no longer any processes that can remove (from the surface) the initial dissolved iron inventory.

#4 – (see: genie-userconfigs/MS/wardetal.2018/wardetal.2018.BIOGEM.SPIN)
This is the iron cycle as used by Ward et al. [2018] with the ECOGEM ecosystem model, but configured for the basic (non ecological) BIOGEM export scheme.

#5 – (see: genie-userconfigs/MS/wardetal.2018/wardetal.2018.ECOGEM.SPIN)
This is the iron cycle as used by Ward et al. [2018] with the ECOGEM ecosystem model.

#6 – muffin.CB.worjh2.BASESFeTDTL.Mahowald.SPIN
This is a new marine iron cycle tuning for the original Cao et al. [2009] configuration of the present-day ocean climatology (worjh2).

#6 – muffin.CB.worjh2.BASESFeTDTL.Albani.SPIN
Similar to muffin.CB.worjh2.BASESFeTDTL.Mahowald.SPIN, except the tuning is for the Albani et al. [2016] pre-industrial (1850 AD) dust field.

#7 – muffin.CB.worjh2.BASESFeTDTL.MahowaldUNIFORMsol.SPIN
Similar to muffin.CB.worjh2.BASESFeTDTL.Mahowald.SPIN, except with the iron cycle tuned
under the assumption of a spatially uniform solubility of iron in dust.

#7 – muffin.CB.worjh2.BASESFeTDTL.AlbaniUNIFORMsol.SPIN

Similar to muffin.CB.worjh2.BASESFeTDTL.Albani.SPIN, except with the iron cycle tuned under the assumption of a spatially uniform solubility of iron in dust.

---

**Alternative implementations for the dissolved iron flux to the ocean surface**

By default (and in all the marine iron cycle configurations published to date) – the solubility of iron in dust is an inverse function of the dust flux – i.e. at higher dust loadings, the fractional solubility is assumed to be lower. Following Baker et al. [2006], we adopt a relationship in which iron solubility scales with the square root of the dust flux, enacted by the parameter setting: The associated tracer selection in the base-config looks like:

```plaintext
#exponent for aeolian Fe solubility [use 1.0 for uniform solubility]
bg_par_det_Fe_sol_exp=0.500
```

The solubility parameter `bg_par_det_Fe_sol` determines the value of the overall iron solubility (as a fraction of the total iron in dust), on a global mean flux-weighted basis. For example:

```plaintext
# aeolian Fe solubility [Albani tuning]
bgs_par_det_Fe_sol=0.002441
```

specifies that when weighted by dust flux, the global mean solubility of iron in dust is 0.244%.

However, this parameterization can potentially lead to confusing and unintended consequences. For example, if the dust flux anywhere in the model grid is changed, because the global mean flux-weighted solubility must remain constant, the local solubility everywhere will slightly change, even though the dust flux itself only changes at a single grid point location.

As an alternative, the solubility of iron in dust can be fixed spatially at a single value. This is done by setting:

```plaintext
# Exponent for aeolian Fe solubility [use 1.0 for uniform solubility]
bgs_par_det_Fe_sol_exp=1.0
```

(In the solubility equation, the power is actually one minus the exponent parameter, hence a value of 1.0 makes the solubility spatially invariant.)

Calibrated marine iron cycle parameter sets a provided for both Mahowald et al. [2006] and Albani et al. [2016] pre-industrial dust field (see above and Table 21.1):

- muffin.CB.worjh2.BASESFeTDTL.MahowaldUNIFORMsol.SPIN
- muffin.CB.worjh2.BASESFeTDTL.AlbaniUNIFORMsol.SPIN

The spatial pattern of iron solubility (whether uniform or not), can also be prescribed. This firstly requires the parameter setting:

```plaintext
# Replace internal dust Fe solubility field?
bgs_ctrl_force_det_Fe_sol=true.
```

and then the presence of a 2D ASCII file delineating the spatial pattern of percent (not fractional solubility) iron solubility, which is specified by:

```plaintext
# Filename for dust Fe solubility field
bg_par_det_Fe_sol_file='worjh2.det_Fe_sol.NUL.dat'
```
where `worjh2.det_Fe_sol.NULL.dat` is a default filename for a `worjh2` modern continental configuration $36 \times 36$ grid (of zeros), with the file itself residing in the directory: `genie-biogem/data/input`

The internally calculated solubility field (in %) can be recovered from an experiment via the 2D `netCDF` parameter: `misc_sur_Fe_sol` (‘Aeolian iron solubility’), and requires that NaNs are converted to zeros (just in case), and if using Panoply, the array may also need to be flipped up-side-down.

Two examples are provided for the application of a prescribed 2D field of % solubility (and which are not listed in Table 21.1):

- `muffin.CB.worjh2.BASESFeTDTL.MahowaldFILEsol.SPIN`
  Specifies a spatially heterogeneous distribution of solubility that matches the spatial pattern of calibrated values from:
  `muffin.CB.worjh2.BASESFeTDTL.Mahowald.SPIN`  
  (and where the 2D solubility input file is: `worjh2.det_Fe_sol.Mahowald.dat`)

- `muffin.CB.worjh2.BASESFeTDTL.MahowaldUNIFORMsolFILEsol.SPIN`
  Specifies a field of uniform solubility that matches the calibrated value from:
  `muffin.CB.worjh2.BASESFeTDTL.MahowaldUNIFORMsol.SPIN`  
  (and where the 2D solubility input file is: `worjh2.det_Fe_sol.MahowaldUNIFORMsol.dat`)

(The equivalent `user-config` and iron solubility field files are also provided for the Albani dust field.)

Finally, as a further alternative – a dust field can be forgone entirely and instead, a (2D) flux of dissolved iron to the ocean surface explicitly specified – see `HOW-TO ... force the system`.

### Fake dust

For many (most?) deep-time applications, the spatial distribution of dust flux to the ocean surface may not be known (unless simulated in a paleo GCM). One can therefore:

1. Chose a biological scheme that does not require iron as a co-limiting nutrient.
2. Chose a biological scheme that does require iron as a co-limiting nutrient ... but disable the iron cycle.
   Examples of how this is implemented were given earlier.
3. Make up a dust field!!!  
   Actually ... yes ... but ... assume a modern (or other value) total flux of dust to the ocean surface, but distributed evenly.

The advantage of having *some* iron limitation of biological export is that it can prevent the occurrence of extreme nutrient-trapping and the possibility of carbonate chemistry calculation errors in the model, through small basins close to the Equator, and/or the implementation of temperature-dependent remineralization or short e-folding depths for remineralization, and/or highly elevated atmospheric $pCO_2$ and hence surface temperatures, and/or low atmospheric $pO_2$. Typically, problems arise if one or more of these factors are relatively extreme, and arises as a result of the trapping of very high $PO_4^{2-}$ concentrations in the subsurface, from where it fuels extreme export fluxes. Adding an iron co-limitation provides a negative feedback because dissolved Fe in the water column is scavenged by settling organic matter and lost, meaning that the more that export is fueled by $PO_4^{2-}$ trapping, the more bioavailable Fe is lost and hence the more export is restricted.

Example forcings are provided, for both Mahowald and Albani dust fields, as deposited to the ocean surface of the `worjh2` grid. These are:
Note that there is no (user-config) scaling parameter for sediment flux forcings and hence the total flux forcing appear explicitly in the forcing time-series file: `biogem_force_flux_sed_det_sig.dat`. 

Also note that for a different total continental configuration and hence ocean area, the flux per \( m^{-2} \) will change as only the global total flux to the ocean surface is conserved. This can be corrected e.g. by scaling the total dust flux given in the `biogem_force_flux_sed_det_sig.dat` file in the forcing, by the relative difference of the paleo ocean surface area vs. modern (\( \text{worjh2} - 0.3674774 \times 10^{15} \text{m}^2 \)) and whose value can be found near the top of the global summary output file (`diag_GLOBAL_AVERAGE.res *`).

The 2 different dust fields (with the same `worjh2` ocean grid are characterized by the following global dust fluxes (to the ocean surface):

- Mahowald \(- 16.252 \times 10^{12} \text{mol yr}^{-1} \)
- Albani \(- 9.154 \times 10^{12} \text{mol yr}^{-1} \)

Example user-configs in which the total modern dust flux to the ocean surface is distributed evenly (and keeping all the marine iron cycle parameters the same), are also provided:

- `muffin.CB.worjh2.BASESFeTDTL.MahowaldUNIFORM.SPIN`
- `muffin.CB.worjh2.BASESFeTDTL.AlbaniUNIFORM.SPIN`

Also refer to the chapter on `muffingen` and the configuration of fake worlds.

---

**Other sources of dissolved iron**
21.4 HOW-TO ... other (bio)geochemical cycle

In this HOW-TO, you’ll learn how to include additional ocean tracers and configure various (bio)geochemical cycles in the ocean. Specifically:

1. Include a $CH_4$ cycle in the ocean (and atmosphere)
2. Include a $R – DOM$ cycle in the ocean
3. Include a $Cd$ cycle in the ocean
4. Include an iodine cycle in the ocean
5. Include a neodymium cycle in the ocean
Include a CH4 cycle in the ocean (and atmosphere)

First ... obviously ... you need the appropriate ocean and atmospheric tracers in the base-config. For the atmosphere:

```plaintext
gm_atm_select_10=.true. # pCH4
gm_atm_select_11=.true. # pCH4_13C
```

and then also:

```plaintext
gm_atm_select_12=.true. # pCH4_14C
```

if you have a system including radiocarbon. For the ocean:

```plaintext
gm_ocn_select_25=.true. # CH4
gm_ocn_select_26=.true. # CH4_13C
```

(2 additional ocean tracers total) and then also:

```plaintext
gm_ocn_select_27=.true. # CH4_14C
```

(3 additional ocean tracers total) if you have a system including radiocarbon. Don’t forget to update the total number of ocean tracers accordingly ...

All these settings are also provided as part of as part of muffingen – file: TRACERADDITION.methanecycle.txt

For your user-config file settings, a set of parameter values is also provided as part of muffingen – file: USERADDITION.methanecycle.txt. Simply copy-past this block of parameter values into your user-config.

Note that these parameters settings are taken from the ‘modern’ (high-O2) benchmark experiment: 191210_GMD_mod_C06_20Tmol_SPIN of Reinhard et al. [2020], which can be obtained from: genie-userconfigs/MS/reinhardetal.GMD.2020, but that the settings are teh same for simulating methane cycling in 'deep time'.

Also note that as per CO2, CH4-induced radiative forcing is generally 'on' by default. Setting:

```plaintext
ea_36=n
```

turns off radiative forcing as a function of atmospheric composition. You can then scale the radiative impact of CH4 and CO2 directly via:

```plaintext
ea_radfor_scl_ch4=1.0
ea_radfor_scl_co2=1.0
```

which in this instance scales both CH4 and CO2 induced radiative forcing to ×1 the modern value.
Include a R-DOM cycle in the ocean

The parameter: bg_ctrl_bio_remin_RDOM_photolysis determines whether RDOM degradation is restricted to the surface layer and occurs only by/associated with photolysis. It can be .true. or .false. and by default is set to:

\[ \text{bg\_ctrl\_bio\_remin\_RDOM\_photolysis} = \text{.false.} \]

When set .true., RDOM degradation is set to zero everywhere in the ocean except the surface layer. Here, the lifetime (parameter: bg_par_bio_remin_RDOMlifetime) is modified in *inverse* proportion to the solar insolation integrated over the surface layer. (There is a field in the 2D netCDF of solar insolation at the ocean surface, and the average over the surface layer is approx 1/4 of this.). i.e., in lower latitude and higher insolation regions, the lifetime is shorter than specified by bg_par_bio_remin_RDOMlifetime (and by approx a factor of 1/4 of the solar insolation in W m-2).
21.4 HOW-TO ... other (bio)geochemical cycle

Include a Cd cycle in the ocean

In order to run cGENIE with ocean cadmium cycle, the following base config:
cgenie_eb_go_gs_ac_bg_itfclsd_16l_JH_BASEFeCd is provided.

A typical experiment command line, using the user config file: EXAMPLE_worjh2_P04Fe_Cd_SPIN
(also provided under SVN), would look like:

./runCCgenie.sh cgenie_eb_go_gs_ac_bg_itfclsd_16l_JH_FeCdBASE /
EXAMPLE_worjh2_FeCd_SPIN 11

To submit this job to the cluster (from $HOME):

qsub -q kitten.q -j y -o cgenie_log -S
/bin/bash subcgenie.sh cgenie_eb_go_gs_ac_bg_itfclsd_16l_JH_BASEFeCd /
EXAMPLE_worjh2_P04Fe_Cd_SPIN 10001
Include an iodine cycle in the ocean

In order to run muffin with a marine iodine cycle, several additional tracers need to be selected and initialized in the base configuration file:

```plaintext
# list selected biogeochemical tracers
gm_ocn_select_92=.true. # I -- 'iodide'
gm_ocn_select_93=.true. # IO3 -- 'iodate'
gm_sed_select_79=.true. # POI -- 'particulate organic iodine (POI)'

# list biogeochemical tracer initial values
bg_ocn_init_93=500.0E-09
bg_ocn_init_94=0.0E-09
```

Remember, when adding 2 ocean tracers, the value of the parameter defining the total number of tracers:

```
GOLDSTEINNTRACSOPTS='$(DEFINE)GOLDSTEINNTRACS=xxx'
```

must be increased by 2.

This information is also provided as part of the muffingen software release as the file TRACER-ADDITION.iodinecycle.txt A couple of EXAMPLE base-config files are also provided:

`cgenie.eb_go_gs_ac_bg.worjh2.BASEI`
`cgenie.eb_go_gs_ac_bg.worjh2.BASEI`
`cgenie.eb_go_gs_ac_bg.worbe2.BASESI`

Of these, recommended is one of the 16-level ocean worjh2 configurations\(^\text{12}\) as the oxygen minimum zones in the 8-level ocean worbe2 configuration are much more poorly developed (see Ridgwell et al. [2007a]).

The ocean iodine cycle can then be configured in several different ways and levels of complexity, as described in the following sections.

---

Basic (‘abiotic’) iodine cycle

This section outlines the most trivial possible configuration of the marine iodine cycle, in which the only processes are:

1. Reduction if IO3- to I- under dysoxic conditions
2. Re-oxidation of I- to IO3-

An example experiment configuration `user-config – EXAMPLE.worjh2.P04Ibasic.SPIN` is given. In this, the available parameters (listed under the heading '# -- MISC --' and '### IODINE CYCLE CONTROLS ###') controlling the iodine cycle are:

- **# set no biological IO3 uptake**
  For simplicity, this experiment configuration sets the iodate update in organic matter associated with biological production at the ocean surface, to zero. The parameter `bg_par_bio_red_POC_POI` specifies the ratio of I to C in new production (the cellular quotient) and is set to zero.

- **# select basic reduction and oxidation options**
  There are various alternative options for how IO3 is reduced in dysoxic conditions. The simplest parameterization is specified here;
  ```plaintext
  bg_opt_bio_remin_reduce_IO3toI='threshold',
  ```

---

\(^{12}\)The difference between the two worjh2 configurations is that one (with the 'S' in 'BASESI') also includes a sulphur cycle.
in which a threshold of dissolved oxygen is prescribed. In any regions (i.e. model grid boxes) of the ocean in which dissolved oxygen concentrations fall are below this, \( \text{IO}_3^- \) is completely reduced to \( \Gamma^- \).

Alternative options also exist for how the re-oxidation of \( \Gamma^- \) occurs. In the simple parameterization specified here;

\[
\text{bg_opt_bio_remin_oxidize_ItoIO3='lifetime'},
\]

a fixed lifetime of \( \Gamma^- \) in the ocean is prescribed. Oxidation proceeds at this rate regardless of the oxygenation state of the ocean, but as long as sufficient oxygen to accomplish the reaction

\[
2\Gamma^- + 3\text{O}_2 \rightarrow 2\text{IO}_3^- \]

exists.

- # set \([\text{O}_2]\) threshold \((\text{mol} \text{ kg}^{-1})\) for (complete) reduction of \( \Gamma^- \).
  
  The parameter \text{bg_par_bio_remin_cO2_IO3toI} sets the dissolved oxygen concentration threshold \((\text{mol} \text{ kg}^{-1})\), below which \( \text{IO}_3^- \) will be reduced.

- # set \( \Gamma^- \) lifetime \((\text{yrs})\)

  Finally, the parameter \text{bg_par_bio_remin_Ilifetime} then sets the lifetime of \( \Gamma^- \) in years.

Model output is saved in the 'normal way' (refer to the User Manual) and amongst the ocean tracers ('\( \text{ocn}_* \)'), are the tracers of dissolved iodide and dissolved iodate (in units of \( \text{mol} \text{ kg}^{-1} \)). Model output can also be contrasted with observed data re-gridded to the cGENIE (\text{worjh2}) grid.

A typical command-line launching of a model experiment (10000 years integration in this case) would be:

\[
./\text{runmuffin.sh cgenie.eb_go_gs_ac_bg.\text{worjh2}.BASEI / EXAMPLE.\text{worjh2}.P04Ibasic.SPIN 10000}
\]

---

'Biotic' component to the iodine cycle

This section outlines the next component of the marine iodine cycle, involving phytoplankton, with the two processes:

1. uptake of \( \text{IO}_3^- \)
2. remineralization/release of \( \Gamma^- \)

The way this cycling in the model works is as follows: First, as plankton biomass is created, alongside C and P (and if selected N, Fe and other trace elements), \( \text{IO}_3^- \) is taken up by the cell, in a specified proportion to carbon (see below). Of this biomass, a proportion is assumed to be exported in particulate organic matter (POM) form beneath the euphotic zone, with the remainder converted to labile dissolved organic matter (DOM) (see: \text{Ridgwell et al.} [2007a] for details). When either POM or DOM is remineralized and elemental constituents released, rather than returning \( \text{IO}_3^- \) back to solution, it is assumed that the dissolved iodine is in the form of \( \Gamma^- \). (This also requires an accounting of release of \( \text{O}_2 \), as \( \text{IO}_3^- \) is taken up by the cell and assume to be instantaneously internally reduced.) The result is: (i) a progressive transformation \( \text{IO}_3^- \rightarrow \Gamma^- \) in the surface ocean as DOM is continually created and destroyed, and (ii) a 'nutrient-like' enrichment of \( \Gamma^- \) in the sub-surface as a consequence of the remineralization of POM.

An example experiment configuration \text{user-config -- EXAMPLE.\text{worjh2}.P04Ibio.SPIN} -- is given. In this, the available parameters (listed under the heading '# -- MISC --' and '# -- IODINE CYCLE CONTROLS ---') controlling the iodine cycle are:

- # select option for no watercolumn reduction

For simplicity, this experiment configuration sets the reduction of iodate under low oxygen
conditions in the water to zero (i.e. leaving the biological pump as the only source of \( I^- \));
bg_opt_bio_remin_reduce_IO3toI='NONE'

- # set biological IO3 uptake
  Set a cellular I:C quotient. Here, a value of 1.0E-4 is given as an example;
  bg_par_bio_red_POC_POI = 1.0E-4

- # set I lifetime (yrs)
  Finally, the parameter bg_par_bio_remin_Ilifetime then sets the lifetime of \( I^- \) in years.
  (Here, for illustrative purposes, the lifetime is increased to 10 years compared to 1 year in the basic Example.)

Published configurations
Include a neodymium cycle in the ocean

The tracers needed for a Nd cycle are listed in TRACERADDITION.neodymiumcycle.txt in muffingen. For a restoring boundary-condition based implementation of Nd cycling in the ocean (currently the only option), only a restoring forcing of Nd concentration and isotopic composition is required (and no water column scavenging). The additional lines needed in a user-config are given in USERADDITION.neodymiumcycle.txt.

A complete EXAMPLE is provided and can be run:

```
```

In this particular EXAMPLE, the restoring forcing is applied to all the ocean surface data points where observations exists. Other combinations and regional forcings are also available:

- `worjh2.RNd_R144Nd_BENALL` – all benthic grid points
- `worjh2.RNd_R144Nd_BENPAC` – benthic grid points in the Pacific
- `worjh2.RNd_R144Nd_SURALL` – all surface grid points
- `worjh2.RNd_R144Nd_SURALL_BENALL` – all surface and all benthic grid points
- `worjh2.RNd_R144Nd_SURALL_BENPAC` – all surface, but only Pacific benthic grid points
- `worjh2.RNd_R144Nd_SURATL` – surface Atlantic grid points
- `worjh2.RNd_R144Nd_SURATL_BENPAC` – surface Atlantic plus benthic Pacific grid points

All forcings are for an abiotic ocean and hence no \( pCO_2 \) forcing is included, but it can of course be added. Forcings are also for the worjh2 modern continental configuration, but other modern or paleo forcings could be created.
21.5 HOW-TO ... biological production

Configure an abiotic ocean

Biological productivity in the ocean can be completely turned off to create an abiotic ocean (why you would want to do this is another matter ... perhaps analyzing the solubility pump or a 'deep-time' and prior to significant marine life study ... (?)). The biological option is set by the parameter:

```
bg_par_bio_prodopt
```

which by default takes a value of "1N1T_P04MM" which selects the scheme described in Ridgwell et al. [2007a]. To have no biological production in the ocean, add the following line to the end of the `user-config` file\textsuperscript{13}:

```
bg_par_bio_prodopt="NONE"
```

With this set, you do not have to specify any biological production or remineralization namelist parameter values in the `user-config` file.

\textsuperscript{13}Or edit the existing line under the section ‘-- BIOLOGICAL NEW PRODUCTION --’
Prescribe the CaCO₃:POC export ratio

In the default ‘biological’ scheme in GENIE the CaCO₃:POC export ratio from the surface ocean in BIOGEM is parameterized as a power law function of the degree of ambient over-saturation w.r.t. calcite [Ridgwell et al., 2007a,b]. The calculated CaCO₃:POC ratio will vary therefore both spatially, particularly w.r.t. latitude (and temperature), as well as in time, if the surface ocean saturation state changes. The latter can arise from climatic (temperature) or circulation changes, or through a change in the DIC and/or ALK inventory of the ocean (such as resulting from emissions of fossil fuel CO₂) or the re-partitioning of these species vertically within the ocean (e.g., as a result of any change in the strength of the biological pump).

There may be situations in which it is advisable to hold the CaCO₃:POC export ratio invariant. For instance, considering the current very considerable uncertainties in the impacts of ocean acidification on marine calcifiers [Ridgwell et al., 2007a] the safest assumption is arguably to exclude any acidification impact on calcification and carbonate export. Specifying a spatially uniform value of the CaCO₃:POC ratio ratio (e.g. 0.25 or 0.3) also allows comparison with the results of early carbon cycle model studies. For deeper-time geological studies where little about marine carbonate production may be known a priori, a spatially uniform value represents the simplest possible assumption (e.g., Panchuk et al. [2008]).

BIOGEM can be told to use a prescribed (spatially and temporally invariant) 2D field of CaCO₃:POC export rain ratios (instead of calculating these internally as a function of ocean chemistry) by setting the ‘Replace internal CaCO₃:POC export rain ratio?’ namelist parameter flag to .true.:

```
bg_ctrl_force_CaCO3toPOCrainratio=.true.
```

You must also then provide a 2D data field that specifies the value of the rain ratio at each and every surface ocean grid point. The filename of this field is set by default to:

```
bg_par_CaCO3toPOCrainratio_file="CaCO3toPOCrainratio.dat"
```

and the file must be located in the ‘BIOGEM data input directory’¹⁵, which by default is:

```
bg_par_indir_name=" $RUNTIME_ROOT/genie-biogem/data/input"
```

This 2-D field must be in the form of an ASCII file with space (or tab) separated values arranged in rows and columns of latitude and longitude. The format of the file must follow the GOLDSTEIN ocean grid with the first line being the most Northerly row, and the last line the most Southerly row of grid points. Data along a row is from West to East. The latitude of the first column of values must be consistent with the defined starting latitude of the model grid, which is specified by the namelist parameter gm_par_grid_lon_offset.¹⁶ Examples are given in the code repository.¹⁷

If you are using a uniform value, it is an easy enough job to create a 36 × 36 array of the value you so desire.¹⁸

If you want to hold a previously-calculated (spatially variable) CaCO₃:POC field constant, then the easiest way to achieve this is to copy the information contained in the time-slice results field: misc_sur_rCaCO3toPOC in the results netCDF file fields_biogem_2d.nc.¹⁹ Because this is a 3D data field (36 × 36 × 8), carefully highlight just the surface ocean (2D) distribution (e.g., from

---

¹⁴The default biological scheme is given by: `bg_par_bio_prodopt='1N1T_PO4MM'`.
¹⁵$RUNTIME_ROOT being equal to `~/genie`.
¹⁶-260E by default
¹⁷e.g., `~/genie/genie-biogem/data/input/CaCO3toPOCrainratio_worbe2_preindustrial.dat`
¹⁸It doesn’t matter if you specify a value over land because only values associated with wet cells will be acted on.
¹⁹You must have the ‘miscellaneous properties’ time-slice save flag set to: `bg_ctrl_data_save_slice_misc=.true.` (the default) for this field to be saved.
the Panoply viewer) or extract from the netCDF file by some other means, and then copy and paste into:

\[
\text{CaCO}^3\text{toPOCrainratio.dat} \ (\text{or whatever you have specified the filename as})
\]

When copying Panoply data, ‘NaN’s should be replaced by values of zero. Take care that the final (steady-state) time-slice is being copied and not the first (un-spunup) one ...

**TIP:** In order to quantify the importance of calcification feedbacks with CO2 and climate, two model integrations are required: one with the CaCO$_3$:POC ratio held constant and the other with it allowed to vary, thereby allowing the effect of a changing CaCO$_3$:POC ratio on the system to to elucidated.
Prescribe biological export production

There are two main possibilities:

1. **Via a full prescription of all particulate fluxes in surface ocean**
   
   Create a full set of particulate (sediment tracer) flux forcings fields for the surface ocean, one for each biologically-related sediment tracer selected in the model, including isotopes (and trace metals). Everything except for the surface layer can be left as a zero (0.0) in the two 3D spatial fields required for each tracer.

   You must also create a set of dissolved (ocean) tracer flux forcings fields for the surface ocean, one for each dissolved tracer associated with the particulates and selected in the model (including isotopes etc). The dissolved tracer flux fields must be create so as to exactly cancel out the particulate fields to conserve mass. For most tracers this is trivial, i.e., the fields for P in particulate organic matter (sed_POP) need be associated with fields for dissolved PO4 (ocn_PO4) which will simply be equal in magnitude but opposite in sign to POP. Complications start to arise for CaCO3 and there is also the question of alkalinity changes associated with organic matter creation/destruction (via changes in NO3), so this method, whilst the most flexible, is not without its complications and degree of tediousness ...

2. **By just prescribing just the POC flux**
   
   An alternative has been provided enabling a full biological productivity in the surface ocean, but controlled by prescribing just the particulate organic carbon export flux. This ‘biological’ scheme is selected with:
   
   bg_par_bio_prodopt="bio_POCflux"

   What happens in practice is that the POC flux is used to calculate the equivalent PO4 change in the surface ocean, and then this is passed to the biological scheme and export production calculated ‘as usual’. (The POC flux forcing is set to zero once the associated PO4 (uptake) flux has been calculated.)

   A particulate (sediment tracer) flux forcing for POC in the surface ocean still has to be defined and selected, but no other forcings (including the associated removed dissolved tracers) are required. An example forcing configuration is given in worjh2.FPOC_Caoetal2009 (and selected by: bg_par_forcing_name="worjh2.FPOC_Caoetal2009") An example user-config: EXAMPLE.worjh2.Caoetal2009_FPOC illustrating this is provided.

   **NOTE**: Take care with dissolved organic matter (DOM) production, as the specified POC flux is automatically increased to take into account DOM production. i.e. 50% of export is specified to be partitioned into dissolved rather than particulate organic matter export, whatever is specified in the POC export forcing would be internally doubled, before being partitioned into POM and DOM. **NOTE**: Also take care with the units of the flux forcing to the surface layer in the ocean, which are in mol m-2 yr-1. The main particulate flux output is in units of mol m-2 yr-1, but since cGENIE is invariably run on a equal area grid it is not difficult to convert export production densities to mol yr-1 (you either need to divide the area of the Earth’s surface by the number of grid points, or save the ocean grid information – see netCDF save options in the User Manual). Alternatively, as of revision r8825, cGENIE saves the primary particulate flux fields also in units of mol yr-1 (assuming you have the biological or full netCDF save options selected – see netCDF save options in the User Manual).

   Be aware that if there is insufficient PO4 to support the require POC flux, the entire POC flux will still be created, meaning that you may end up with regions of negative nutrient concentration.
21.6 HOW-TO ... ecology

Prescribe SSTs for plankton only

Plankton growth in ECOGEM is temperature-dependent and so a change in climate will drive both a physical response from ocean circulation and an ecological response. You can separate these effects by prescribing an annual mean field of SSTs for plankton in ECOGEM as done by Wilson et al., [2018] in Paleoceanography and Paleoclimatology.

To enable this option set the following parameter in the user-config:

```
eg_ctrl_force_T=.true.
```

The field of prescribed SSTs (in °C) arranged by longitude-latitude) should be saved as a text file within `~/cgenie.muffin/genie-ecogem/data/input`. See the corresponding forcing files from Wilson et al. [2018] as an example, e.g.

```
~/cgenie.muffin/genie-ecogem/data/input/20170811.worj2.PO4.8P8Z.pal.1x_SST.dat
```

Specify the SST field in the user-config for plankton using:

```
eg_par_ecogem_force_T_file="20170811.worj2.8P8Z.pal.3x_SST.dat"
```
Implement 'instantaneous remineralization'

By default in the original version of cGENIE, biogenic particles exported out of the surface ocean layer were assumed to settle at velocity of $125 \text{md}^{-1}$. The parameter controlling this was: bg_par_bio_remin_sinkingrate. The consequence was that particles took more than a single time-step to reach the ocean floor and be fully remineralized. Hence, particles could be found at intermediate depth at the ocean at the end of any particular time-step; settling further during the next time-step (and so on to the deepest ocean layer at that grid point).

An associated parameter was defined to control the residence time of particles in any one layer for the purpose of calculating scavenging. A separate parameter was created so that the residence time and hence scavenging rate could be varied independently of the settling rate of the bulk particles. This parameter was: bg_par_bio_remin_sinkingrate_scav. The default value of this is also $125 \text{md}^{-1}$.

This all became pretty confusing ... what is 'real' settling and what the 'scav' bit is ... so a new pair of parameters was created. These are:

- bg_par_bio_remin_sinkingrate_physical
- bg_par_bio_remin_sinkingrate_reaction

The former defining the 'real' (physical) sinking speed, and the latter the effective sinking speed for the purpose of calculating the residence time in each ocean layer and hence reaction rates. Here, reaction rates include scavenging as well as opal dissolution, and organic matter remineralization driven by temperature.

By default these are set to zero, and the values of bg_par_bio_remin_sinkingrate and bg_par_bio_remin_sinkingrate_scav are used.

As an alternative to the default, of particles setting a finite distance down through the water column each time-step, the products of remineralization can be instantaneously distributed down through the water column – as if the particles settled with infinite velocity, yet still dissolved and reacted as the same rates as before, i.e. at a finite settling speed. To do this one would set, e.g.:

bg_par_bio_remin_sinkingrate_physical=1.0E9
bg_par_bio_remin_sinkingrate_reaction=125.0

At a velocity of $1.0^9 \text{md}^{-1}$, particles would reach the ocean floor (regardless of the depth at that grid point) in a single time-step. Remineralization is therefore calculated throughout the water column within that time-step – 'instantaneous'. The second parameter says that for the purposes of residence time and reactions, sinking should be considered to have occurred at $125 \text{md}^{-1}$.

Why do this? It simplifies calculation of flux mass balances, as material leaving the ocean surface is 'seen' (arriving) at the sediment surface at the same time-step. This behaviour is likely to become the muffin default in the future.
Alternative remin schemes

Implement an alternative fixed remineralization profile for POC (e.g. Martin curve)

There are several options for utilizing a fixed remineralization profile for POC, which by default is a double exponential (See: Ridgwell et al. [2007a]). The fixed remineralization profile scheme is set by the string parameter: bg_par_bio_remin_fun. By default, it has a value of 'efolding'. Currently available options are:

- **Martin1987**, which applies a globally-uniform power, set by:
  
  bg_par_bio_remin_martin_b

  (which by default has a value of -0.858)

- **Henson2012**, which calculates the value of b according to sea surface temperature (SST):
  
  \[ b = (0.024 \times \text{SST}) - 1.06 \]

  To user either (on their own), all organic matter should be assigned to a single phase, with the 2nd (recalcitrant) fraction set to zero:

  bg_par_bio_remin_POC_frac2=0.0

  Note that these parameterizations can be combined with ballasting and will act on the 'free' POC phase (i.e. the one not controlled by the ballasting parameterization).

Implement particulate organic carbon 'ballasting'

The default particulate organic carbon (POC) ocean interior remineralization scheme is based on fixed, prescribed profiles of relative POC flux to depth (e.g. see: Ridgwell [2001]; Ridgwell et al. [2007a]). A 'ballasting' control on POC transport to depth can instead be implemented by:

bg_ctrl_bio_remin_POC_ballast=.true.
bg_ctrl_bio_remin_POC_fixed=.false.

The POC 'carrying coefficients' for CaCO3, opal, and detrital (lithogenic) material are set by the parameters:

bg_par_bio_remin_ballast_kc
bg_par_bio_remin_ballast_ko
bg_par_bio_remin_ballast_kl

(for CaCO3, opal, and lithogenics, respectively). Note that the ballast coefficient units are: g POC m-2 yr-1 (g ballast m-2 yr-1)-1 (i.e. g g-1), which are internally converted to: mol POC m-2 yr-1 (mol ballast m-2 yr-1)-1 (i.e. mol mol-1).

A fixed (in time), but spatially heterogeneous field can also be prescribed instead of global uniform values (akin to setting a pattern of the CaCO3:POC export rain ratio (??)). The parameters setting whether to substitute a globally-uniform value with a specified pattern are:

bg_ctrl_force_CaCO3ballastcoeff=.true.
bg_ctrl_force_opalballastcoeff=.true.
bg_ctrl_force_detballastcoeff=.true.
and which by default are .false. The patterns of carrying coefficient are determined by files read in from cgenie/genie-biogem/data/input. The filenames are specified by:

bg_par_CaCO3ballastcoeff_file
bg_par_opalballastcoeff_file
bg_par_detballastcoeff_file

(again, akin to the methodology for setting the CaCO3:POC export rain ratio (??)).

Note that ballasting is combined with an e-folding (or other) fixed profile remineralization schemes\(^\text{20}\). Ballasting is calculated with respect to the 2nd (recalcitrant) fraction of POC only. The remaining POC export is be degraded by an alternative algorithm, which by default, is an e-folding decay (see previously for more and alternatives). The fraction of initial export assigned to ballasting vs. 'free' POC is calculated according to the available exported ballast flux.

---

**Impose a fake 'sea floor' to remineralization**

Why ... ?

There may be reasons ... but I am not going to explain.

By default, BIOGEM takes the physical ocean grid ('k1 file') and uses it to determine the maximum the depth that particles can sink (and be remineralized) and hence the depth of the sea-floor. At that bottom ocean layer, particles that have not been remineralized/dissolved in the water column can be passed to SEDGEM if selected, remineralized and returned to the overlying ocean, or even remineralized and returned directly to the surface.

As an alternative, there is an option to tell BIOGEM that the sea-floor is 'shallower' than the actual physical ocean at any particular grid point. This would result in less total remineral-ized/dissolved occurring in the water column and a larger flux passed to the sediments (SEDGEM) (if selected). The option is set by\(^\text{21}\):

bg_ctrl_force_Vgrid=.true.

This requires that a file is present in BIOGEM/data/input or genie-paleo/[CONFIG], depending no how you have configured the boundary conditions. This file must be named according the parameter setting:

bg_par_force_Vgrid_file='FILE'

(or vice versa) where FILE is the filename.

This file contains the replacement grid of 'k' levels (as integer values). Note that this file differs slightly from the format of the 'k1' file loaded into GOLDSTEIN etc. as it has no boundary/padding rows and columns, and has the same dimension (lon x lat) as the physical grid. It does not matter what land grid points are marked by (a zero will do). If a deeper 'k' level is given in the alternative file as compared to the true physical grid, the depth of the true physical grid is used (so the fake ocean floor can only be shallower (or the same), not deeper, than the true ocean grid).

As to 'why' you might want to use this ...

---

\(^{20}\)Although in a sense, the remineralization of POC is not 'fixed' in that it does not have a predetermined profile but instead is set by the changing flux of CaCO3, opal and lithogenic fluxes with depth, bg_ctrl_bio_remin_POC_fixed should still be set to .true. (the default).

\(^{21}\) 'Vgrid' for Virtual grid.
21.8 HOW-TO ... force the system

Note: All the forcings described here assume a newer (simplified) methodology for prescribing forcings.\(^{22}\) This methodology is enabled by setting:

```
bg_ctrl_force_oldformat=.false.
```

This is set automatically as part of the runmuffin.sh shell script\(^{23}\).

Taking the example of the ocean (dissolved tracers): flux and restoring forcings are defined in the forcings specification file: `configure_forcings_ocn.dat`. As detailed in the notes to this file, there is a flag (COLUMN #6) which sets the spatial attributes of the forcing as follows:

- 3 – ‘3D’ – force the entire ocean volume uniformly
- 2 – ‘2D’ – force the entire ocean surface uniformly
- 1 – [currently there is no option for forcing a single ocean column]
- 0 – ‘0D’ – force a specified point in the ocean
- -1 – ‘SURFACE’ – force with/to specified surface ocean pattern
- -2 – ‘BENTHIC’ – force with/to specified benthic pattern
- -3 – ‘LEVEL’ – force a specific layer in the ocean with/to specified pattern\(^{24}\)
- -4 – ‘SURFACE+BENTHIC’ – simultaneously force with/to specified (and separate) surface and benthic pattern

The default (3) is that the forcing is applied uniformly to the entire (3D) ocean volume. Options 3, 2, and 0, as: uniform 3D\(^{25}\) (volume), uniform 2D (surface), and point forcing, respectively, require no additional (spatial) information and only an additional file specifying the time-dependent information for each forcing need be provided:

- `biogem_force_flux_ocn_xxx_sig.dat` for flux forcings, and
- `biogem_force_restore_ocn_xxx_sig.dat`

where: `xxx` represents the mnemonic of the tracer (e.g., DIC is dissolved inorganic carbon. CH4 is methane, etc.).

Options -1 (SURFACE), -2 (BENTHIC) -3 and -4, require a 2D field to be provided, in addition to the time-dependent information for each forcing. The grids for both are the same – i.e., all ‘wet’ grid points (non dry land) in the model. The filename for these 2D files is of the form:

- `biogem_force_flux_ocn_xxx_SUR.dat` for flux forcings, and
- `biogem_force_restore_ocn_xxx_SUR.dat`

with BEN for the equivalent BENTHIC, and LEV for LEVEL forcing selections.

---

\(^{22}\)For details of the ‘old’, fully 3D spatially-explicit forcing methodology, refer to the cGENIE user-manual.

\(^{23}\)(but it not the actual default namelist parameter setting)

\(^{24}\)For k=k_max, this duplicates the SURFACE forcing.

\(^{25}\)Note that here: ‘3D’ does not mean a spatially explicit 3D pattern and hence the original (‘old’) way of specifying forcings, but instead: that the forcing is applied uniformly in 3D space (i.e., is in effect a volume forcing).
For option -3, the grid level is set via the parameter:

```
bg_par_force_point_k
```

(e.g. bg_par_force_point_k=7) which by default is 0.

For sediment (particulate) forcings, there is no restoring forcing. In addition, the flux forcings are limited, with the choices being:

-2 – '2D' – force the entire ocean surface uniformly
-0 – '0D' – force a specified point in the ocean
-1 – 'SURFACE' – force with/to specified surface ocean pattern
-2 – 'BENTHIC' – force with/to specified benthic pattern
-5 – 'SUB-BENTHIC' – force with/to specified (sub-) benthic pattern

For option -2, particles are injected into the layer overlying the sediment surface, at each grid point. However, the bottom-most layer is used in the numerical water column remineralization scheme to 'park' particles before they are added to the sediment surface, and particles present in the benthic layer are assumed to have already interacted (reacted/scavenged) with and dissolved and other particulate tracers in that layer. In other words, particles injected (flux forced) into the benthic layer will not scavenge or react and will simply be added to the sediment surface (or reflected and remineralized, for a closed system) at the next time-step.

Option -5 (chosen to avoid and hence not overlap and be confused with ocean flux options -3 and -4), requests that particles are injected into the sub-benthic layer, i.e. the layer immediately above the bottom most layer. Particles to not react in the sub-benthic layer, but at the next time step, settle through the benthic layer and allowing for a single time-step of reactions (e.g. scavenging) to occur between dissolved tracers in the benthic layer and the injected particles (and prior to settling out of the water column and being incorporated into underlying surface sediments on the time-step after that).

Note that although there is no separate user-config scaling factor as exists for ocean and atmosphere flux and restoring forcings, a 2D flux forcing pattern for particulate tracers (options: -1, -2, -5) can still be scaled via the time-series (*.SigCof) file, e.g. swapping the value of 1.0 for -1.0 to make the flux negative.

---

26 Or the surface layer, for the occasional situation of the benthic layer also being the surface layer (and the ocean only being 1 layer deep).
Add a salinity (freshwater adjustment) forcing

For a limited number of 'modern' configurations (principally worbe2 and worjh2), a (e.g. Atlantic -> Pacific) freshwater flux balance adjustment can be made by changing a single scaling parameter (and/or adjusting 3 parameter controlling individual freshwater transfers). Refer to the tutorial chapter on the modern AMOC for further details.

For non-standard modern continental configurations, or paleo configurations, to re-balance surface freshwater fluxes requires that a 2D surface forcing is created. Refer to Chapter 12 on Proxies and Reconstructing the Past for a description of a pair of late Holocene and LGM configurations which employ a 2D salinity forcing field in order to control the strength of the AMOC.

To re-balance surface freshwater fluxes you need to create a new, or modify an existing, forcing.

You need a 2D spatial field and (scaling) time-series file:

- biogem_force_flux_ocn_sal_SUR.dat
- biogem_force_flux_ocn_sal_sig.dat

and in configure_forcings_ocn.dat, you set:

2 f f 0.1 t f -1 01 01 01 ' [salinity (PSU)]'

which requests a flux forcing for salinity (tracer 2), with the -1 specifying that a 2D field for the surface layer (only) will be used (-1 == SURFACE).

Easiest is to set biogem_force_flux_ocn_sal_sig.dat constant and scaled so as to convert units from mol per grid point yr$^{-1}$ to Sv, e.g.

0.0 1.10e+18
999999999.0 1.10e+18

In the user-config, the additional scaling parameter:

bg_par_ocn_force_scale_val_2=0.4

can then be used to set the flux transfer in units of Sv.

It then remains to devise the spatial pattern in biogem_force_flux_ocn_sal_SUR.dat. The key here is that the sum of all the grid point values, should be zero (unless you intend to have a net input or loss of freshwater from the ocean and hence a globally decreasing or increasing mean salinity).

Where you want to remove freshwater from should have positive salinity flux values, and where you wish to add the freshwater to, should have negative salinity values.

Finally, for the scaling (to Sv) to work out, the values in the two regions must sum to one (losing freshwater) and minus one (gaining freshwater).

To create this 2D map – refer to the examples mentioned above. For the Holocene/LGM pair, salinity is added to the entire North Atlantic (north of the Equator), and balanced by a salinity loss in the North Pacific. (Note that the salinity forcing conforms to a mask that is common to both Holocene and LGM land-sea masks.) For the Miocene-through-Holocene series, salinity input is restricted to more extreme northern latitudes in the Atlantic (north of the isthmus of Panama), while the balancing salinity removal is spread evenly over the entire remaining surface ocean. (Again, a mask common to all continental configurations is used and so a salinity flux is only applied to grid points that are 'wet' in all continental configurations.)

---

27 In practice implemented by a (negative) salinity flux.

28 Also see: user-configs/MS/crichtonetal.CP2021 for further examples.
**Add a second salinity forcing**

Soooooo greedy! This really shouldn’t be allowed. (But it is, hence the manual entry ...)

Why would you even want do this? Well, you might want regions of salinity forcing to operate independent of each other. For instance, a pattern of salinity that provides an AMOC-controlling flux adjustment, together with a salinity forcing elsewhere that might vary in time, or simply to simplify the additional forcing rather than have to combine it within the same spatial pattern file. Or you might want 2 different patterns of flux adjustment that both vary in time and independently of each other. A practical situation would be to have 2 flux forcings – one for the North Atlantic and one for the Southern Ocean (independently controlling northern vs. southern soured deep water).

If you have already selected and configured a salinity flux forcing, then selecting (in the base-config) the blue numerical dye tracer – **ocn_colb** (tracer number 49) and configuring a flux forcing for this, will automatically be interpreted by...

You configure the spatial pattern and units scaling of the blue salinity forcing in exactly the same way as for salinity. e.g. in the forcing file `configure_forcings_ocn.dat`, you would have:

```
-START-OF-DATA-
  2 f f 0.1 t f -1 01 01 01 'salinity (PSU)'
  49 f f 0.1 t f -1 01 01 01 'blue salinity (PSU)'
-END-OF-DATA-
```

The files `biogem_force_flux_ocn_sal_diag.dat` and `biogem_force_flux_ocn_sal_SUR.dat` can be simply copied and renamed with **colb** replacing **sal** in the filename (and then edited as required).

Note that with the blue numerical tracer used for a second salinity forcing, if you want to also diagnose water mass age, you will have to use the (new) single-tracer approach that only requires the red numerical tracer. If so, set:

```
bg_ctrl_force_ocn_age1=.true.
```

Ensure that you have:

```
bg_ctrl_force_ocn_age=.false.
```

or simply do not list that parameter in the `user-config` file, otherwise the blue tracer will be used along-side the red one to create a combined age tracer (and no additional salinity input will occur) ...
Modify a dissolved Fe (dust) forcing

Dissolved iron, required in co-limitation of biological productivity in a number of BIOGEM biological options (as well as in ECOGEM), in the basic marine iron cycling schemes, is supplied via dust deposited to the ocean surface. This is implemented as a sediment flux forcing (but to the ocean surface rather than ocean floor) – the solid detrital component deposited to the ocean surface is assumed to sink through the water column to the sediment surface.

For example – worjh2.RpCO2_Rp13CO2.FeMahowald2006 – in addition to implementing a restoring of atmospheric pCO₂ and associated δ¹³C, applies a dust flux forcing on the worjh2 continental grid. The dust flux is activated in configure_forcings_sed.dat by the line:

```
22 t F -1 01 01 '[detrital (refractory) material (wt%)]'
```

and the associated 2D surface flux distribution biogem_force_flux_sed_det_SUR.dat and time-series biogem_force_flux_sed_det_sig.dat files

To change the supply of iron to the ocean surface, it is as simple as changing the pattern of dust flux, either by substituting a completely different field (e.g. using a LGM-derived dust flux field in place of a present-day one, or using an idealized field, perhaps completely uniform or with a simple regional pattern). Note that the units of flux to the ocean in muffin are mol (dust) per grid cell(!), so some units conversion may be required in setting a particular dust (and hence iron) flux.

However, the default configuration of the solubility scheme for iron in dust employs a non-linear relationship between dust loading and solubility, and adjusts local solubilities to achieve a specified global flux-weighted mean solubility, given by parameter:

```
bg_par_det_Fe_sol
```

The consequence of this is that if the dust flux is changed anywhere in the 2D field, all local (in each grid cell) solubilities will change in order to retain the same global flux-weighted mean solubility. To implement a globally uniform proportion change in solubility (and hence uniform % increase or decrease in dissolved iron supply to the surface) is easy – by changing the specified global flux-weighted mean solubility or the assumed (uniform) iron content of dust, but otherwise, not :(

Modifying the pattern of iron supply to the ocean surface is hence most easily accomplished either by (1) creating a flux of iron in addition to a dust field (with its inherent global mean iron solubility), or (2) completely substituting dust-supplied iron with an explicit dissolved iron flux to the ocean surface that can then be more readily modified.

1. For the first option – see: 7.4.1 Iron fertilization

2. For the second – first de-select the dust forcing – simply a matter of deleting the line in configure_forcings_sed.dat.

Then create a 2D dissolved Fe forcing – adding to the configure_forcings_ocn.dat file, the line (for iron cycle schemes: ALT, OLD, FeFe2TL):

```
9 f f 0.0 t f -1 01 01 01 '[Fe]'
```

(where the -1 specifies a surface ocean pattern of iron flux) file, OR, if you are using a scheme involving the total dissolved iron tracer (e.g. hybrid_lookup_4D), adding:

---

29 Dust may react with species in the water column and/or in the sediments, and/or act as a ballasting or scavenging particle, depending on the options selected.

30 Option 1 in ‘COLUMN #03: make forcing uniform over this dimension’ in the configure_forcings_sed.dat file.

31 Instead, you can select a uniform iron solubility by setting: bg_par_det_Fe_sol_exp=1.0

32 Between the -START-OF-DATA- and -END-OF-DATA- tags.
Check which iron scheme you are using and select the corresponding line to be added to configure_forcings_ocn.dat. In the user-config file, the parameter: bg_opt_geochem_Fe specifies the iron scheme.

Then you then just need to obtain the spatial field of dissolved iron flux to the ocean surface that was calculated and applied in the dust-driven version. This can be found in fields_biogem_2d netCDF variable: misc_sur_fFe ('Solulablized aeolian iron flux to surface grid points') in units of mol Fe yr\(^{-1}\) (per model grid point).

From MATLAB or Panoply, copy the 2D array (table) of values, paste into a text file, and save it as biogem_force_flux_ocn_Fe_SUR.dat (or biogem_force_flux_ocn_TDFe_SUR.dat, depending on which primary iron tracer is selected). You will also need a corresponding time-series (*.sig.dat) file. In creating the *.SUR.dat file, Ensure:

- That the iron flux field is the right way up!!!! (It should be map-orientation, with North upwards.) By default, newer versions of Panoply plot the grid up-side-down.
- That it is not in MS-DOS/Windows format ASCII – the line-ending format must be Unix (LF). DO NOT use Windows Notepad to create the file.
  
  If you are unsure/concerned, from the directory containing the file, at the linux command line, type:
  
  $ dos2unix biogem_force_flux_ocn_TDFe_SUR.dat
  
  which will strip out any accidental CR characters.

- Ensure there is a blank line at the bottom of the file.

You should run a control experiment using the dissolved iron forcing, following on from the dust-driven spin-up, to confirm you have correctly substituted the source of iron to the ocean (and no changes in atmospheric \(pCO_2\) or global biological export flux (or any of the ocean dissolved iron outputs) change.
Determine the CH4 flux required to achieve a particular atmospheric pCH4 value

Unlike the concentration of CO2 in the atmosphere, which if restored to a chosen value during a spin-up experiment, will remain at that value in a continuation experiment (if no other perturbation of the carbon cycle or CO2 emissions have been prescribed), CH4 in the atmosphere decays with a lifetime of ca. 8 years (with a small fraction dissolving in ocean surface waters and being oxidized in the ocean). Hence, atmospheric CH4 restored to a particular value in a spin-up, requires that restoring to be maintained in any continuation experiment or CH4 will quickly decay to zero. However, doing this (restoring CH4 concentrations), prevents the effect of CH4 emissions on being assessed (as the atmospheric composition is being held constant).

An alternative would be carry out the spin-up experiment with no restoring of atmospheric CH4 (or restoring to zero), and then run the continuation experiment with no CH4 restoring. This would enable e.g. CH4 emissions experiments to be carried out and the change in atmospheric CH4 in response to be simulated. The problem here is that the lifetime of CH4 in the atmosphere scales with the CH4 concentration. So in starting with no CH4 in the atmosphere, the CH4 lifetime is relatively short, and the response to CH4 emissions will be underestimated.

What is in effect ‘missing’ are the (natural) sources of CH4 to the atmosphere such as wetlands, which at steady state, provide a CH4 flux that balances the oxidation rate of CH4 in the atmosphere (and ocean). cGENIE has a parameter for this: ac_par_atm_wetlands_FCH4 (mol yr\(^{−1}\)) (with the isotopic composition of this source set by: ac_par_atm_wetlands_FCH4_d13C). All that then remains is to determine the flux of CH4 that balances the rate of oxidative loss for the desired atmospheric CH4 concentration. To do this:

1. Carry out a spin-up with atmospheric CH4 restored to the desired concentration.\(^{33}\)
2. Determine the total loss rate of CH4 (including both atmospheric oxidation and invasion (and subsequent oxidation) into the ocean) – this is recorded in the time-series results file: biogem_series_focnatm_pCH4.res\(^{34}\).
3. Set the parameter ac_par_atm_wetlands_FCH4 equal to this value.

An example of a spin-up in which a prescribed (‘wetland’) flux of CH4 to the atmosphere is set, is described in:
cGENIE.Examples – spin-up example EXAMPLE_p0055c_P04_CH4_SPIN2

\(^{33}\)For an example: see experiment EXAMPLE_p0055c_P04_CH4_SPIN described in cGENIE.Examples.

\(^{34}\)Second column (the value in units of mol yr\(^{−1}\))
Determine the atmospheric radiocarbon flux required to achieve a steady state 14C system

First, you need to spin the system with the atmosphere restored to the e.g. pre-industrial $\Delta^{14}C$ value desired. There are two ways then to diagnose the equivalent $^{14}C$ flux:

1. Sum the total mol inventory of $\Delta^{14}C$ in the ocean (as DIC) and atmosphere (CO$_2$), both of which can be found in the relevant time-series output. The radiocarbon decay rate multiplied by the (steady state) inventory then gives the total decay, which of course is equal to the cosmogenic input flux needed to maintain steady state.

2. Rather simpler, enabled by some recent output code changes, is simply to read off the reported atmospheric flux forcing that is being applied to restore the atmosphere $\Delta^{14}C$ value. The relevant time-series output file is:

   biogem_series_diag_misc_specified_forcing_pCO2_14C.res

Once the balancing $^{14}C$ flux value has been obtained, the namelist parameter controlling the rate of cosmogenic $^{14}C$ production (by default, zero): ac_par_atm_F14C
Prescribe a time-varying history of radiocarbon production in the atmosphere

The original way for doing was this was to create a CO\textsubscript{2} flux forcing and give its $\delta^{14}$C signature an extremely positive value. However, inputs and outputs of isotopic compositions in cGENIE are limited to being between -999 and +999 per mil. Prescribing a small flux of CO\textsubscript{2} to the atmosphere with a $\delta^{14}$C of +999 per mil will not give an equivalent flux as if it were pure $^{14}$C (hence needing the calculation of the equivalent total CO\textsubscript{2} flux with a +999 per mil signature).

There is now a simple hack (not yet extended to flux forcings of the ocean) to direct cGENIE to interpret the value specified as its $\delta^{14}$C forcing signature as an absolute flux (in units of mol yr\textsuperscript{-1}) rather than convert from a per mil notation. cGENIE knows to use this alternative if the flux of CO\textsubscript{2} specified in the CO\textsubscript{2} flux forcing is *identical* to the value given for $^{14}$C. i.e. in order to prescribe a production rate of $^{14}$C in the atmosphere sufficient to balance ocean-atmosphere carbon cycling (in the Cao et al. [2009] configuration of the model), instead of setting the atmospheric $^{14}$C production parameter:

\begin{verbatim}
ac_par_atm_F14C=0.387E+03
\end{verbatim}

One would now set an identical value of 0.387E+03 in both forcing files:

\begin{verbatim}
biogem_force_flux_atm_pCO2_14C_sig.dat
biogem_force_flux_atm_pCO2_sig.dat
\end{verbatim}

(or equivalently, re-scale a unit flux forcing given in the forcing files using the bg_par_atm_force_scale_val_5 parameter in the example of $^{14}$C (and bg_par_atm_force_scale_val_3 for bulk CO\textsubscript{2})).

Selection of the required forcings in configure_forcings_atm.dat\textsuperscript{35} then looks like this:

\begin{verbatim}
-START-OF-DATA-
03 f 0.1 t t F 2 01 01 'carbon dioxide (CO2)'
05 f 0.1 t t F 2 01 01 '[14C CO2]'
-START-OF-DATA-
\end{verbatim}

Now that $^{14}$C production is being specified explicitly by means of a forcing, one can easily then implement a time-dependent change in $^{14}$C production.

\textsuperscript{35}There is no need to specify a $^{13}$C flux (ignore any warnings at start-up). Indeed, formally, there is no $^{13}$C associated with $^{14}$C production form N\textsubscript{2}. Actually ... DO NOT add a $^{13}$C forcing, just in case ...
21.8 HOW-TO ... force the system

**Prescribe an injection of radiocarbon-dead DIC**

First ... a base-config with $^{14}$C tracers is needed, e.g.:

cgenie_eb_go_gs_ac_bg_itfclsd_16l_JH_ANTH

or
cgenie_eb_go_gs_ac_bg_itfclsd_16l_JH_ANTHFe

(with Fe co-limitation of marine biological productivity).

Then, in the user-config, an appropriate forcing needs to be specified, e.g.:

`pyyyyyx_FDIC_F13DIC_F14DIC`

and under the heading -- FORCINGS --, might look something like:

- For forcing selection:

  bg_ctrl_force_oldformat=.false.
  bg_par_forcing_name="worj2_FDIC_F13DIC_F14DIC"

  which prescribes a forcing of DIC plus its (13C and 14C) isotopes to the ocean (somewhere or everywhere).

- Then:

  bg_par_ocn_force_scale_val_03=0.0833e15

  sets the flux (mol yr$^{-1}$), which is equivalent to 1 PgC yr$^{-1}$.

- To scale the isotopic composition:

  bg_par_ocn_force_scale_val_04=-60.0
  bg_par_ocn_force_scale_val_05=-999.0

  for example gives -60 per mil for 13C like methane and 14C that is pretty isotopically dead$^{36}$.

- By default in the forcing, the duration of the emission 1 year, and can be re-scaled (e.g.,
to 1000 years duration) by:

  bg_par_ocn_force_scale_time_03=1000.0
  bg_par_ocn_force_scale_time_04=1000.0
  bg_par_ocn_force_scale_time_05=1000.0

- Finally – the emission location is specified by, e.g.:

  bg_par_force_point_i=18
  bg_par_force_point_j=26
  bg_par_force_point_k=7

Alternatively, the DIC release can be made over the entire ocean floor or simply sections (or depth intervals) of the ocean floor instead of a point source.

$^{36}$Note this is on the scale of d14C not D14C
Prescribe a spatial map of benthic tracer release

Flux *forcings* to the ocean with a variety of different spatial attributes can be specified in the *forcings* specification file: `configure_forcings_ocn.dat`. As detailed in the notes to this file, there is a flag (COLUMN #6) which sets the spatial attributes of the *forcing*:

- 3 – '3D' – force the entire ocean volume uniformly
- 2 – '2D' – force the entire ocean surface uniformly
- 0 – '0D' – force a specified point
- -1 – 'SURFACE' – force with/to specified surface ocean pattern
- -2 – 'BENTHIC' – force with/to specified benthic pattern
- -3 – 'LEVEL' – force a specific layer in the ocean with/to specified pattern
- -4 – 'SURFACE+BENTHIC' – simultaneously force with/to specified (and separate) surface and benthic pattern

Options -1 through -4 require a 2D field to be provided. The grids for both are the same – i.e., all 'wet' grid points (non dry land) in the model. Templates for these can be created as follows:

1. Open up the BIOGEM results file: `fields_biogem_2d.nc` (any experiment).
2. Display the variable: `grid_mask`.
3. Select the Array 1 tab (to display the actual gridded values rather than the color-coded map); highlight the grid of values and then copy-and-paste to a text editor.
4. You should have a grid of values, with a ’1.0’ representing ocean, and 'NaN' land. The NaNs can then be search-and-replaced to ’0.0’ and you have a grid valid for either the entire surface ocean or entire benthic surface.

From here: 1s can be replaced by 0s to remove unwanted locations.

In the forcing configuration file, if the COLUMN #5 flag ('scale flux forcing of tracer?') is set to 't', then the flux applied at each selected location is scaled such that the total applied flux is equal to that given in the forcing time-signal file.

---

37 For k=k_max, this duplicates the SURFACE forcing.
38 This can be quite time-consuming and tedious and there is no particular short-cut :(
39 The values in the forcing map need not be all 1.0 of course.
Applying a geoengineering 'liming' flux to the ocean surface

[see geoengineering HOWTO]
Prescribe a spatial map of geothermal heat input to the ocean floor

**muffin** provides a single parameter to set a uniform geothermal heat flux to the ocean floor:

```plaintext
# add a geothermal heat flux (W m-2)
bg_par_Fgeothermal=100.0E-3
```

(for which a plausible mean modern flux value would be $100 \text{mW m}^{-2}$ ($100.0 \text{E}^{-3}$), although it should be noted that the default parameter setting is zero). Obviously, this parameter can be changed (e.g. perhaps increased in value to represent higher heat flow on the earlier Earth ... or other planets) and its importance to large-scale ocean circulation patterns and mixing (and further, to cold extensive sea-ice climate states), explored.

However, other questions require spatial patterns of heat flow to be considered, such as from the emplacement of magmatic provinces at the sea-floor, and/or the importance of the location of spreading ridges and associated elevated localized heat fluxes. To address these, more flexibility than a single uniform flux assumption is needed.

A pair of parameters are hence provided to implement an explicit 2D field of geothermal heat input. Firstly:

```plaintext
# Use 2D geothermal heat input field?
bg_ctrl_force_Fgeothermal2D=.true.
```

specifies that a 2D field rather than uniform value is to be applied, and:

```plaintext
# Filename for 2D geothermal heat input field
bg_par_force_Fgeothermal2D_file='worjh2.Fgeothermal.NULL'
```

specifies the filename itself – here a files of zeros (`worjh2.Fgeothermal.NULL`) based on the $36 \times 36$ `worjh2` modern ocean grid.

Whatever file is used, needs to reside in the directory: \texttt{genie-biogem/data/input}. The units need to be $\text{W m}^{-2}$.

What reasonable specific values might be, for a magmatic province or spreading ridge ... (compared to a mean global value of $100 \text{mW m}^{-2}$ ($100.0 \text{E}^{-3}$) ...)
21.8 HOW-TO ... force the system

**Force tracer conservation**

Without sediments and an explicit representation of the fractional preservation and burial of settling particulate matter reaching the seafloor, there is the question of what to 'do' with this material. The typical solution is to apply a 'reflective' boundary condition whereby all settling particulate matter reaching the seafloor is remineralized back to its respective dissolved components. This is enacted by the default parameter setting:

```
bg_ctrl_force_sed_closedsystem=.true.
```

(Note that the local redox conditions at the seafloor are employed to calculate the consumption and release of dissolved species associated with remineralization, as per is the case in the water column.)

An exception to this is scavenged iron, which by default is removed from the system, controlled by the parameter setting:

```
bg_ctrl_bio_NO_fsedFe=.true.
```

which prevents 'reflection' of scavenged Fe. This then requires that a new supply of dissolved iron, e.g. at the surface in the form of a dust flux, is always applied to balance the loss at the seafloor of scavenged Fe.

The major consequence of a reflective boundary condition, in addition to not being able to simulate changing mass balance (between weathering inputs and sedimentation losses) and hence inventories in the ocean, is that the forced return of dissolved species at the seafloor can potentially exacerbate nutrient trapping and/or anoxia.

An alternative, is to re-route the reflective fluxes to the ocean surface (immediately overlying). This can be achieved via:

```
bg_ctrl_force_sed_closedsystem=.true.
bg_ctrl_force_sed_closedsystem_SUR=.true.
```

and in order to retain a dissolved iron inventory in the ocean without having to prescribe a dust flux:

```
bg_ctrl_bio_NO_fsedFe=.false.
```

Note that in calculating the remineralization of particulate matter reaching the seafloor, the surface redox conditions are used (rather than those at the seafloor). (The alternative (application of seafloor redox conditions) would probably give similar results, given the likely rapid re-oxidation of reduced species at the surface.)

Also note that these new parameter settings only apply to a muffin configuration without the **SEDGEM** model enabled.
Set a specific ocean chemistry or saturation state

In the absence of a significant pelagic plankton derived deep-sea carbonate sink, the global burial sink of CaCO$_3$ is likely to have been dominated by shallow water depositional environments. cGENIE can represent something of these systems and the fundamental dynamical difference between deep-sea pelagic and shallow water CaCO$_3$ sinks (the former is predominantly controlled by preservation and input of C$_{org}$, whilst the latter is primarily a function of primary production by benthic calcifiers) by specifying particular shallow water grid cells as 'reefal' (see: ??). Also in contrast to the deep-sea pelagic CaCO$_3$ sink, which is mechanistically simulated (both export production and particularly early diagenesis and CaCO$_3$ dissolution within the sediments), the shallow water CaCO$_3$ sink in cGENIE is heavily parameterized and is treated as little more than a function relating deposition of CaCO$_3$ (in units of mol cm$^{-2}$ yr$^{-1}$) at each and every designated reefal grid point. A parameter is provided:

\texttt{sg\_par\_sed\_reef\_CaCO3precip\_sf}

that scales the CaCO$_3$ burial flux so as to achieve some desired global value, such as to balance the global weathering rate. Unfortunately, the value of this parameter is not \textit{a priori} known. Its value (and hence sink strength) also depends on the ambient saturation state. Hence, from the outset, the carbonate saturation properties of the ocean surface must be assumed and set. For the modern ocean, this arises naturally from a system initialized with observed concentrations of ALK, DIC, [Ca$^{2+}$], etc. and (preindustrial) atmospheric $p$CO$_2$, and in conjunction with an adequate simulation of the large-scale productivity of the ocean and recycling in the interior ocean.

For deeper in the geological past and particularly in the absence of an effective deep-sea pelagic CaCO$_3$ buffer, ALK and DIC in particular \textit{a priori} not known, although either sensitivity experiment assumptions of the CO$_2$ concentration required to generate a specific climate and/or proxy data, can give some ideas of $p$CO$_2$. It is beyond the scope of this HOW-TO to discuss what geological constraints can be applied in constraining surface ocean saturation (and hence, in conjunction with a given weathering flux, to identify the value of the scaling parameter) but having identified a number (as calcite or aragonite), there are three possible ways of setting an appropriate ocean chemistry:

1. To achieve a specific mean ocean surface saturation state: mean ocean ALK and DIC values can be set consistent with assumptions regrading $p$CO$_2$ and [Ca$^{2+}$]. The parameters specifying the initial mean ocean ALK and DIC concentrations ($\mu$mol kg$^{-1}$) are:

\begin{align*}
\text{bg\_ocn\_init\_12} & = 2.3630E-03 \\
\text{bg\_ocn\_init\_3} & = 2.244E-03
\end{align*}

for their respective modern mean ocean values.

In order to estimate appropriate past concentrations for a given $p$CO$_2$ and [Ca$^{2+}$], trial and error can be employed, aided by use of a carbonate calculator program such as CO2SYS, although bearing in mind that the saturation conditions calculated in CO2SYS will be \textit{w.r.t.} the surface, while the 2 model initialization parameters are setting the mean ocean composition – ocean circulation and particularly biological productivity in the open ocean will create an offset between mean surface and bulk ocean properties. Once a reasonable ocean saturation state has been employed, the reefal CaCO$_3$ depositional parameter can be played with to (re)balance weathering and global burial, although if the initial guess is too far off, this will need to be briefly iterative (as localized CaCO$_3$ removal and burial will affect the local saturation state and hence in turn burial).

\textbf{Conclusion}: do-able, but tedious.
2. An alternative methodology, somewhat akin to how the system with a responsive deep-sea pelagic CaCO$_3$ buffer is configured, is to set the system ‘open’, and allow the balance between weathering and shallow water CaCO$_3$ burial to set ocean chemistry (whilst restoring $p$CO$_2$ to a required value). This is potentially even more tedious, because the value of the scaling parameter is not known. In fact, it is the scaling parameter that will set the mean ocean surface saturation in order to balance weathering and shallow water CaCO$_3$ burial. An acceleration technique can be used (see: ??) but the process is still iterative (and tedious). A further provision is hence made in the SEDGEM module that will flux force the ocean with Ca (and also DIC if requested) at each and every reefal cell if the local saturation state falls below a specified target value. In this:

```
sg_ctrl_sed_neritic_reef_force=.TRUE.
```

will turn ‘on’ the provision of a forcing of ocean chemistry towards a local saturation target,

```
sg_par_sed_ohmegamin
```

sets the saturation threshold, and

```
par_sed_ohmegamin_flux
```

specifies the flux in units of mol Ca$^{2+}$ cm$^{-2}$ per time-step at each and every reefal grid point. Setting:

```
ctrl_sed_forcedohmega_ca=.false.
```

will enable the corresponding fluxes of ALK and DIC to be applied. This methodology can be accelerated as per Section ?? but with the downside that what is being set in practice is the minimum saturation at any reefal grid point – i.e. one should obtain a reefal grid point, typically at relatively high latitudes, with ambient chemistry close to the specified saturation, with all other grid points characterized by higher saturation. There is no simple way of deriving the global mean surface saturation in advance.

Having forced ocean chemistry (e.g. just by altering the Ca$^{2+}$ and ALK input to the ocean) and restored to a specified atmosphere $p$CO$_2$ value, the value of the reefal CaCO$_3$ burial scaling parameter can be adjusted in order to balance weathering and global burial. A little iteration may probably be required to get a good balance as the competing flux input and CaCO$_3$ removal are very localized (at reefal cells).

**Conclusion:** do-able, but does not give a simple-to-predict mean global saturation.

3. With a desired global mean surface saturation value in mind, cGENIE can be configured to determine the appropriate ocean ALK and DIC value directly. In brief – a mean global saturation target value is set by the parameter:

```
bg_par_force_invert_ohmega
```

ALK, and if requested: DIC (and isotopes) and Ca, is fluxed evenly throughout the ocean if the current ocean surface saturation value falls beneath the target, and not fluxed at all otherwise$^{40}$. An aragonite saturation value can be forced towards rather than calcite by setting:

```
bg_ctrl_force_ohmega_calcite=.false.
```

---

$^{40}$A negative flux of ALK can be enabled, which would act to reduce saturation, by setting the following:

```
bg_ctrl_force_invert_noneg=.false.
```
The setup for this is a little involved and involves specifying a particular forcing that cGENE-NIE identifies as requiring a saturation target to be matched and requires some additional configuration. An example `user-config` to spin-up the ocean to a specified saturation value is provided:

```
EXAMPLE.p0251b.P04.SPIN0
```

and described in the Examples chapter. A typical configuration would like like the following:

```
bg_par_forcing_name="pyyyyy.RpCO2_Rp13CO2_FRALK_FDIC_F13DIC_FCa"
bg_par_atm_force_scale_val_3=2800.0E-06
bg_par_atm_force_scale_val_4=-6.5
bg_par_ocn_force_scale_val_3=1000.0E12
bg_par_ocn_force_scale_val_4=0.0
bg_par_ocn_force_scale_val_12=2000.0E12
bg_par_ocn_force_scale_val_35=0.0
bg_par_force_invert_ohmega=10.0
```

of which the first line sets the forcing (provided), the second line specifies the atmospheric pCO$_2$ value to be restored to (and the 3rd its isotopic composition). Because the ocean is either fluxed or not, depending on surface saturation compared to the target, how aggressively the ocean is fluxed is set by lines #4 (for DIC) and #6 for ALK. The values shown here are approximately x10 global weathering for reference. The last line is the required mean ocean surface saturation state. (Lines #5 and #7 set the isotopic composition of injected DIC, and any associated Ca flux, respectively.)

In terms of methodology:

(a) This saturation restoring would be carried out in a closed system for typically 10 or 20 kyr, depending on whether an initial `spin-up` was being used and how far off any guess as to initial ALK and DIC is. Depending on the reported global CaCO$_3$ burial rate, the reefal CaCO$_3$ burial scaling parameter is adjusted. This is also a good time if climate-dependent feedback is to be used, to adjust e.g. the baseline mean global temperature.

(b) The saturation restoring `forcing` is replaced by a simple atmospheric pCO$_2`forcing` (or none at all), and a short, perhaps just 1 kyr experiment is carried out with an open system. Fine-tuning of the CaCO$_3$ burial scaling parameter is carried out (plus any fine-tuning required to set an exact initial weathering flux).

(c) A long, open-system `spin-up`, likely accelerated (Section ??) 3rd spin-up phase is conducted with no forcing.

Example `user-configs` of this sequence are provided (EXAMPLE.p0251b.P04.SPIN*). Note that the 3rd phase of spin-up could be carried out with a prescribed pCO$_2$ *plus* associated isotopic composition if not already done so (e.g. as part of the phase #1 of the spin-up).

Conclusion: work-able.

---

41 These values represent the *total* flux that distribution throughout the ocean relative to grid cell volume.

42 SEDGEM file seddiag_misc_DATA_GLOBAL.res
21.9 HOW-TO ... sediments and weathering

Overview

There are some complications involved in setting up sediment deposition and terrestrial weathering, particularly when you add organic matter burial (and/or want to consider shallow sedimentary systems rather than just open ocean/deep sea environments). There are also a lot of varying subsections in this section and it would be easy to get lost/overwhelmed ... so what follows is a guide to setting things up as well what the relevant sub-sections are.

0. Creating/choosing a base-config

Before anything – you are going to need a base-config with SEDGEM and ROCKGEM enabled (and if you want to accelerate spin-ups, also GEMLITE).

The mnemonics for these in the old-style base-config naming convention, are: _sg, _rg, and _gl, respectively.

In the new style naming convention, they are: S, R, and G, respectively.

Several different examples of base-configs are given and used in this section:

- cgenie_eb_go_gs_ac_bg_sg_rg.worbe2.BASE (OLD format)
  - a 36 × 36, 8-level ocean, modern configuration as used in Ridgwell and Hargreaves [2007].
- muffin.CBSRG.p_worbe2.BASES
  - a 36 × 36, 8-level ocean, modern configuration as used in Ridgwell and Hargreaves [2007] but in a 'paleo' file directory format.
  - Includes the GEMLITE module.
- cgenie_eb_go_gs_ac_bg_sg_rg_itfclsd_16l_JH_BASE (OLD format)
  - a 36 × 36, 16-level ocean, modern configuration as used in Archer et al. [2009].
- muffin.CBSRG.fkl_pp20.BASES – a 18 × 18, 16-level ocean, conceptual configuration (single continent from pole-to-pole).
  - Includes the GEMLITE module.

Also see the muffingen chapter – Section 18.4 on page 251, for how to create new base-configs (including sediments etc.).

1. Creating/choosing a basic user-config

Now ... you need a user-config to define your initial experiment (which will be a spin-up).

To help you – again, some example files are provided and referenced in this Section:

- EXAMPLE.worbe2.RidgwellHargreaves1997_S36x36.SPIN1
  - is the initial (1st stage) spin-up from Ridgwell and Hargreaves [2007].
- muffin.CBSR.p_worbe2.BASES.RidgwellHargreaves1997_S36x36.SPIN1 – is a newer version of the initial spin-up from Ridgwell and Hargreaves [2007] that uses a paleo directory format for boundary condition files.
- EXAMPLE.worjh2.Archeretal2009.SPIN1
  - is the initial (1st stage) spin-up used in Archer et al. [2009].
- muffin.CBSRG.fkl_pp20.BASES.CaCO3_burial.SPIN1

---

43Designed for rapid testing.
44Using the base-config: cgenie_eb_go_gs_ac_bg_sg_rg.worbe2.BASE
45Using the base-config: muffin.CBSRG.p_worbe2.BASES
46Using the base-config: cgenie_eb_go_gs_ac_bg_sg_rg_itfclsd_16l_JH_BASE
Chapter 21. HOW-TO (experiment)

muffin.CBSRG.fkl_pp20.BASES.CaCO3POC_burial.SPIN1, muffin.CBSRG.fkl_pp20.BASES.CaCO3POCPoP_burial.SPIN1
– conceptual worlds with increasing complexity of burial: CaCO₃-only, CaCO₃ + POC, and CaCO₃ + POC + POP, respectively.⁴⁷
These are arguably the cleanest and best annotated examples to work from.

• muffin.CBSR.p_worbe2.BASES.Corg_burial.simple.EXPT, muffin.CBSR.p_worbe2.BASES.Corg_burial.dunne2007.EXPT
– illustrating two different simply organic carbon preservation/burial schemes.⁴⁸

In the old (non ‘paleo’) format file structure, you had to specify the resolution of the sediment grid in the user-config, plus sediment depth grid and files for defining areas of ‘reef’ and sediment core save locations, via a block of lines similar to:

```bash
# Sediment grid
SEDGEMNLONSOPTS='$(DEFINE)SEDGEMNLONS=36'
SEDGEMNLATSOPTS='$(DEFINE)SEDGEMNLATS=36'

# sediment grid options
sg_par_sed_topo_D_name="worbe2_topo.36x36"
sg_par_sed_reef_mask_name="worbe2_reef_mask.36x36"
sg_par_sedcore_save_mask_name="worbe2_save_mask.36x36"
```
(taking as an example: EXAMPLE.worbe2.RidgwellHargreaves1997_S36x36.SPIN1).

In the newer file structure, the equivalent parameter settings are included in the base-config and the files live in the corresponding genie-paleo directory.

In addition, you need to define a water depth, above which any seafloor cells not defined as ‘reef’ in the grid specified by `sg_par_sed_reef_mask_name`, will become ‘mud’.⁴⁹ This depth is set by:

```bash
# max cut-off depth for neritic (non deep sea) sediment grid points
# NOTE: sediment grid points shallower that this and not defined as reefal cells,
# are assumed to be ‘mud’
sg_par_sed_Dmax_neritic=0.0
```
(in this example, no ‘mud’ cells are allowed).

To change the grid resolution or files – refer to the sub-section on page 399.
To add or configure areas of ‘reef’ deposition, refer to the sub-section on page 377.
To change the assumed mineralogy of reefal carbonate, refer to the sub-section on page 378.

2. Choosing sediment accumulation/diagenesis options

The basic component of the sediment systems is detrital material and you always have this solid component by default. The flux of material can either be taken from a spatial pattern of dust flux at the ocean surface, in which case you need a dust forcing, or if you do no a priori (e.g. in deep time) know the paleo dust flux, you can prescribe a fixed (in time) spatially uniform flux directly to the sediment surface via:

```bash
# generic paleo detrital flux (g cm⁻² kyr⁻¹)
# NOTE: set to zero when applying a dust field
# (unless an additional detrital flux is required)
sg_par_sed_fdet=0.180
```

⁴⁷ All using the base-config: muffin.CBSRG.fkl_pp20.BASES
⁴⁸ Using the base-config: muffin.CBSR.p_worbe2.BASES
⁴⁹ ‘mud’ is a sediment grid cell classification in the model that allows only organic matter preservation and burial (plus detrital), with all carbonate assumed dissolved.
(which is a typical modern open-ocean Pacific value as well as being a representative Eocene detrital accumulate rate value).

In addition to a detrital flux that accumulates in the sediments, you can have any or all of (as solid species): calcium carbonate, opal, organic matter. These are set by:

- **sg_par_sed_diagen_CaCO3opt**
  - 'NONE’ – simply results in all $CaCO_3$ reaching the sediment surface, dissolving.
  - 'archer1991explicit’ (the default)
  - 'ridgwell2001lookup’ – much faster than the default but cannot deal with non-standard (paleo) $Mg/Ca$ ratios.
  - 'ridgwell2001lookupvec’ (not currently used)
  - ‘archer2002nn' (not currently used)
  - ‘ALL’ – simply results in all $CaCO_3$ reaching the sediment surface, being preserved.

- **sg_par_sed_diagen_opalopt**
  - 'NONE’ (the default) – simply results in all opal reaching the sediment surface, dissolving.
  - 'ridgwelletal2003lookup’
  - 'ridgwelletal2003explicit’
  - ‘ALL’ – simply results in all opal reaching the sediment surface, being preserved.

- **sg_par_sed_diagen_Corgopt**
  - 'NONE’ (the default) – simply results in all organic matter reaching the sediment surface, being remineralized (oxic-only).
  - 'simple’ – see sub-section starting on page 379.
  - 'dunne2007’ – see sub-section starting on page 379.
  - 'huelse2016’ – see sub-section 21.9.1 on page 400.

3. Sediment mixing (bioturbation) options

Generally – to capture the more correct dynamics of ocean-sediment interaction and to simulate sediment records, you want to apply a mixing profile to the upper sediments layers. If instead, you simply want to spin-up the system to achieve a balance between weathering and sediment burial more quickly, or to equilibrate the surface sediment composition (more quickly), apply no mixing.

The later option is the simplest, and requires only setting:

```bash
# bioturbation
# NOTE: turn OFF bioturbation to equilibriate the system faster
sg_ctrl_sed_bioturb=.FALSE.
```

Alternatively, to apply mixing (biodiffusion) to the upper sediment layers:

```bash
# bioturbation
# NOTE: turn ON bioturbation to capture carbonate compensation dynamics
sg_ctrl_sed_bioturb=.TRUE.
```

which utilizes a non-dimensional profile read in from a file\(^{50}\), and which is subsequently

\(^{50}\)Itself, given by the parameter: `sg_par_sed_mix_k_name`
scaled by the parameter:

```plaintext
# scale (maximum) surface biodiffusion rate
sg_par_sed_mix_k_sur_max=0.15
```

(in units of cm² yr⁻¹).

The alternative bioturbation option:

```plaintext
sg_ctrl_sed_bioturb_Archer=.true.
```

uses a different depth profile and scales biodiffusion as a function of ambient bottom water [O₂] and F_POC, following Archer et al. [2002].

NOTE: As of a recent commit (d569cd6), the default is: sg_ctrl_sed_bioturb_Archer=.false. and hence `user-config` files no longer need specify this setting each time.

NOTE: There is also an option: `sg_par_n_sed_mix` which prescribes how many sediment layers are mixed often appears in `user-config` files. However, this invariably takes a value of 20, which is the default ... and hence also needs not be re-specified.

NOTE: Finally ... previous `user-configs` additionally invariably specified a value for the minimum biodiffusion coefficient (`sg_par_sed_mix_k_sur_max`) when using the Archer et al. [2002] scheme. This parameter definition is also redundant.

In summary – the simplest description (post commit d569cd6) for implementing bioturbation, is just:

```plaintext
# bioturbation
# NOTE: turn ON bioturbation to capture carbonate compensation dynamics
sg_ctrl_sed_bioturb=.TRUE.
# scale (maximum) surface biodiffusion rate
sg_par_sed_mix_k_sur_max=0.15
```

with all other settings being correct in their default values.

4. Final (weathering) configuration and running the system

If you think you know, or simply want to explore, the rates of carbonate and silicate weathering and volcanic outgassing, then you can go ahead and let SEDGEM calculate the solute loss as CaCO₃, and the system will eventually come into balance with a specific ocean carbonate chemistry enabling burial to balance weathering. The configuration required for this is effectively the same as any of the *.SPIN2 EXAMPLE `user-config` files.

If, however, you know the chemistry of the ocean (and perhaps biological export) as in the modern ocean (or simply want to explore the consequences of changing ocean chemistry and/or biological assumptions, then the recommended approach is as per Ridgwell and Hargreaves [2007] and as per the paired *.SPIN1 + *.SPIN2 EXAMPLE `user-config` files. This is the more common methodology and you may want to utilize a prior climate-only (or climate plus ocean-atmosphere carbon cycle) spin-up (effectively, SPIN0) so that you start with stable circulation and ocean biogeochemical cycling.

This paired sequence of spin-ups works as follows:

(a) **First-guess closed system spin-up**

If you have *a priori* knowledge of ocean chemistry, then obviously you do not want this to drift (e.g. away from observations) during model spin-up. A facility is provided to always balance solutes lost through preservation and burial of e.g. solid CaCO₃ by

---

51The Archer et al. [2002] scheme has not in fact been published with to date.
instantaneously adjusting the global weathering flux such that it always tracks the burial loss term. This is selected by:

\[ \text{bg_ctrl_force_sed_closedsystem=.true.} \]

To utilize this methodology requires that the ROKGEM module is used.\(^{52}\) Additionally, for the spatial distribution of weathering flux calculated by ROKGEM to be re-scaled (to balance the global loss rate), a non-zero weathering flux must be prescribed. This could be derived from a previous closed system model experiment with no weathering flux specified (diagnosing weathering from total global \(\text{CaCO}_3\) burial as described earlier), or from the literature, e.g., Ridgwell [2007] cites 20 \(\text{Tmol HCO}_3 \text{yr}^{-1}\), an equivalent \(\text{CaCO}_3\) weathering rate of 10 \(\text{Tmol yr}^{-1}\):

\[ \text{rg_par_weather_CaCO3=10.00E+12} \]

Or ... it can be completely \textit{made up} \(^{53}\) (as long as it is not zero).

As an example – see:

muffin.CBSRG.fkl_pp20.BASES.CaCO3 burial.SPIN1

(Or muffin.CBSR.p_worbe2.BASES.SPIN1 or muffin.CBSR.p_worjh2.BASES.SPIN1, which are the paleo file structure equivalents to Ridgwell and Hargreaves [2007] and Archer et al. [2009], respectively.)

20,000 years is probably about the minimum practical \textit{spin-up} time. Primarily – you are looking for convergence in the mean wt% \(\text{CaCO}_3\) value (averaged sediment composition), which is recorded in the BIOGEM time-series file: biogem_series_sed_CaCO3.res. This relatively short time to equilibrium is also aided by turning off bioturbation and mixing of the sediments.

There are a few complexities if you want the burial of phosphorous in organic matter, because the ocean \(\text{PO}_4\) inventory must also be balanced. Refer to the sub-section on page 378.

(b) 2\textit{nd stage}, open system \textit{spin-up}

In the simplest possible configuration of weathering, the prescribed weathering flux (\(\text{rg_par_weather_CaCO3}\)) is now revised and set equal to the diagnosed global \(\text{CaCO}_3\) burial rate ("Total \(\text{CaCO}_3\) pres (sediment grid)") as reported in the SEDGEM module results file: seddiag_misc_DATA_GLOBAL.res.

In addition, an \textit{open system} must now be specified in the \textit{user config}:

\[ \text{bg_ctrl_force_sed_closedsystem=.false.} \]

50,000 years is probably about the minimum practical \textit{spin-up} time. Again – you are looking for convergence in the mean wt% \(\text{CaCO}_3\) value and it is up to you to judge how long the \textit{spin-up} needs to be.

For a full description of how to set up the weathering parameters, see:

- The subsection: 'Set up a (silicate) weathering feedback' on page 387 (this is sufficient if your configuration does not include organic carbon burial).
- The subsection: 'Include organic carbon burial' on page 379 for how to partition the various weathering fluxes (if your system configuration includes organic carbon burial).

\(^{52}\)Which in the original naming convention was indicated by \_rg\ in the base-config filename, but now appears simply as \_R.

\(^{53}\)Whatever the global value is, will be re-scaled anyway ...
As an example of a more complex partitioning of weathering fluxes into silicate vs. carbonate, also see:

muffin.CBSRG.fkl_pp20.BASES.CaCO3_burial.SPIN2

You can also accelerate the 2nd stage spin-up – see the sub-section 'Accelerate the weathering-sedimentation mass balance (‘GEMlite’)’ on page 395. An example user-config is given:

muffin.CBSRG.fkl_pp20.BASES.CaCO3_burial.SPIN2gl

This can also be used as a single-state spin-up, although you need to know the weathering fluxes in advance, and ideally you’d either run from a climate (and ocean biogeochemistry spin-up), or run much much longer than 50,000yr.
Include shallow water depositional systems

By default, the entire seafloor grid is considered as 'deepsea' and sedimentary diagenesis options are provided accordingly. In practice, because the simple diagenesis options available, such as for CaCO\textsubscript{3} (e.g. Archer [1991]; Ridgwell [2001]; Ridgwell et al. [2003]) tend not to be applicable to shallower water and particularly high organic carbon delivery (especially in the case of CaCO\textsubscript{3} diagenesis) environments, the deep-sea grid is restricted. This can be prescribed 'hard', by defining the sediment grid only at deeper depths and classifying shallowed ocean grid points as invalid (a value of 0.0) in the SEDGEM depth definition file\textsuperscript{54} (e.g. worbe2.depth.36x36x08). Or, and more flexibly, the entire ocean grid can be defined as potentially valid (e.g. worbe2.depth.36x36x08.ALL) and SEDGEM directed to treat any grid points lying shallower than a specific depth as shallow water sediments. This depth cut-off is set via the parameter \texttt{sg_par_sed_Dmax_neritic} (in units of m below the ocean surface). A typical value is 176.0\textsuperscript{55}, which for an 8-level configuration will exclude the shallowest ocean depth level from the deep-sea grid, and for a 16-level ocean will exclude the two shallowest ocean depth levels.

Sediments that are not classed as 'deepsea', are automatically classed as 'mud', i.e. potentially detrital and organic carbon rich. Such sediment locations currently have no option for carbonate burial associated with them. Instead, a limited range of extremely simple and crude assumptions regarding organic carbon preservation (and P regeneration) are provided. For a new, more mechanistic representation of organic carbon degradation and burial in marine sediments and associated biogeochemical benthic-pelagic fluxes see Section ??.

Locations at which carbonate can be produced and preserved are classified as 'reef' locations. Note that the assumption here is that carbonate is precipiced benthically and buried, rather than produced pelagically and settle to the seafloor. A mask needs to be provided\textsuperscript{56} in order to define the cells which are reef rather than mud. The mask consists of the sediment grid, with values being either 1.0 (reef) or 0.0 (other). Grid points that are shallower than the depth cut-off and not associated with a value of 1.0 in the corresponding reef mask file, are classified as mud.

If can be a little tedious (not really that tedious) to create a reef mask than exactly matches all the shallow grid points, so a parameter is provided to force all non deepsea grid points to be reef. This is achieve by setting:

\begin{verbatim}
sg_ctrl_sed_neritic_reef_force=.TRUE.
\end{verbatim}

This parameter can also be used to force a pattern of reef cells as defined by the reef mask, even if they lie deeper than the depth cut-off value (\texttt{sg_par_sed_Dmax_neritic}).

Additional information and tips on setting up shallow water sediment grids can be found in some of the EXAMPLES (e.g. EXAMPLE.p0251b.PO4.SPIN0).

\textsuperscript{54}Set by parameter \texttt{sg_par_sed_topo_D_name} that must point to a file in: cgenie.muffin/genie-sedgem/data/input.

\textsuperscript{55}Note that the simple (esp. CaCO\textsubscript{3} diagenesis schemes) arguably only applicable below depths of ca. 1000 m, although even this is complicated by varying patterns of productivity across the ocean.

\textsuperscript{56}The parameter name to set the reef mask is \texttt{sg_par_sed_reef_mask_name}.
Specify a particular carbonate mineralogy

Firstly – pelagic carbonate production and deep-sea sedimentary diagenesis is inherently assumed to be (all) calcite. Currently there is no alternative option, nor mixed phase (including high-Mg calcite) assemblage option.

Shallow water (neritic) carbonates, however, can be specified as either calcite (the default) or aragonite, the difference being simply in which saturation state is used to calculate precipitation and burial rate. Reefal carbonate precipitation assumed in the form of aragonite is simply set by:

\[
\text{sg\_par\_sed\_reef\_calcite} = \text{.false.}
\]

Obviously, the same scaling value as per for calcite cannot be used (aragonite will require a higher scaling to achieve the same global CaCO$_3$ depositional flux).

In terms of a spin-up to a specified saturation state (see: 21.8) – an aragonite, rather than calcite (the default) saturation value can be forced towards rather than calcite by setting:

\[
\text{bg\_ctrl\_force\_ohmomega\_calcite} = \text{.false.}
\]
Include organic carbon burial

There are several different ways of implementing organic carbon burial in muffin (all of which requiring you to have the SEDGEM module included):

1. Specify a simple fraction of organic matter as assumed preserved and buried.
   (This also allows for differentiating C vs. P burial, as well as different fractional burial under oxic vs. anoxic vs. sulphidic conditions).
2. Utilize the empirical flux-dependent scheme of Dunne et al. [2007].
   (This also allows for differentiating C vs. P burial.)
3. Implement the OMEN-SED organic matter preservation/burial scheme – see Section ??

All three schemes are applied to both 'deep sea' and 'mud' benthic grid-points, but not to grid-points categorized as 'reef' – see Page ??.

Scheme options #1 and #2 are summarized below.

1. Fixed fraction ('simple')

   This is the simplest possible representation of organic matter preservation and burial and simply imposes the burial of a fixed fraction of the organic matter flux reaching the sea-floor.
   It is selected by specifying:
   \[ \text{sg\_par\_sed\_diagen\_Corgopt='simple'} \]
   (by default this parameter is set to 'none')

   However, there are a couple of complications/subtleties:

   (a) The fraction can be different, depending on the redox conditions, which is controlled by 3 parameters – one fraction parameter for each distinct redox environment:
       \[ \text{sg\_par\_sed\_diagen\_fracCpres_ox=0.0} \]
       for oxic conditions (defined as \([H_2S] < [O_2]\), when the \(H_2S\) tracer is selected)
       \[ \text{sg\_par\_sed\_diagen\_fracCpres_eux=0.0} \]
       for euxinic conditions (defined as \([H_2S] > [O_2]\), when the \(H_2S\) tracer is selected)
       \[ \text{sg\_par\_sed\_diagen\_fracCpres_anox=0.0} \]
       for anoxic conditions (defined as \([O_2] < 0\), when the \(H_2S\) tracer is NOT selected)
   
   (b) You can also specify that the entire recalcitrant (fraction #2) organic matter flux is preserved and buried:
       \[ \text{sg\_ctrl\_sed\_diagen\_preserve\_frac2=.true.} \]
       (by default this parameter is set to false)

   (c) You can decouple the relationship between \(C\) and \(P\) preservation and burial, e.g. simulate burial of organic carbon but impose remineralization of the phosphorous associated with the buried carbon. For example, you might want to have an 'open system' with respect to carbon (and isotopes), but 'closed' with respect to phosphorous.
   There are 3 parameters, corresponding to the same redox zonations as defined above:
   \[ \text{sg\_par\_sed\_diagen\_fracC2Ppres_ox=0.0} \]
   \[ \text{sg\_par\_sed\_diagen\_fracC2Ppres_eux=0.0} \]
   \[ \text{sg\_par\_sed\_diagen\_fracC2Ppres_anox=0.0} \]

   Each parameter sets the relative return (as a fraction of the \(P\) associated with buried \(C\).
   i.e. a value of 0.0 will return to the ocean all the \(P\) associated with buried \(C\), while a value of 1.0 (the default), will bury all the \(P\) associated with buried \(C\), resulting in an 'open system' for both elements.
2. Dunne et al. [2005] scheme

This is the next simplest representation of organic matter preservation and fractional burial of the organic matter flux reaching the sea-floor, and is based on the empirically-derived equation of Dunne et al. [2005]. It is selected by specifying:

\[ \text{sg_par_sed_diagen_Corgopt='dunne2007'} \]

The calculated fractional preservation and burial of POC can be scaled via:

\[ \text{sg_par_sed_diagen_fracCpres_scale} \]

which is assigned a default value of 1.0. The maximum calculated preservation of organic matter is capped at 100%. Otherwise, the fraction of organic matter that is preserved under this scheme is solely a function of the organic matter flux to the seafloor.

With this option, the ‘simple’ scheme P vs. C parameters are not used in conjunction with the ‘dunne2007’ scheme. Instead, an option to apply an empirical (observational-based) parameterization, taken from Wallmann [2010]. This option is specified by:

\[ \text{sg_ctrl_sed_diagen_fracC2Ppres_wallmann2010=.true.} \]

which varies the C : P ratio of the remineralized fluxes (and hence the C : P ratio of buried organic matter) as a function of ambient [O$_2$].

If not set, the C : P ratio of the remineralized fluxes follows the C : P ratio of organic matter deposited to the sediment surface and hence there is no preferential regeneration of dissolved phosphate under low [O$_2$] conditions.

As implemented here, there is a maximum relative return of dissolved P relative and the C/P ratio is never lower than 11 (i.e. an approximately 10-fold enhancement of regenerated P vs. C when compared to a standard Redfield value of 106). This limit is set by:

\[ \text{sg_par_sed_diagen_fracC2Ppres_off} \]

in ‘units’ of C/P.\[^{57}\]

The sensitivity of preferential phosphate regeneration to ambient oxygenation can also be adjusted. This is set by parameter:

\[ \text{sg_par_sed_diagen_fracC2Ppres_c0_O2} \]

and by default has a value corresponding of 32$\mu$MO$_2$. Increasing this value enhances the preferential regeneration of dissolved P (but does not affect carbon preservation and burial).

(It is similar to, but not quite the same in the formulation, as a half-saturation constant.)

Note that when using the ‘dunne2007’ scheme, in order to bury no organic phosphorous and hence obtain a ‘closed system’ just for P, one must set:

\[ \text{sg_ctrl_sed_dunne2007_remin_POP=.true.} \]

This is an alternative to using:

\[ \text{bg_ctrl_force_sed_closed_P=.true.} \]

which requires that a non-zero PO$_4$ weathering flux is set in order for weathering to always track burial and hence close the system.\[^{58}\]

\[^{57}\]Note that the dissolved phosphate return flux is ultimately capped to be no greater than the organic matter phosphorous rain flux to the sediment surface.

\[^{58}\]Note that using bg_ctrl_force_sed_closedsystem in conjunction with POP burial also requires that a non-zero PO$_4$ weathering flux must be set.
For an open system with silicate weathering feedback (see subsequent Section), you need to balance any POC and POP burial with weathering. Kerogen carbon is assumed to be weathered along with silicate rocks and is defined as a ratio with the $Ca^{2+} (+Mg^{2+})$ silicate rock weathering flux:

$$\text{rg\_par\_weather\_CaSiO3\_fracC}$$

(Refer to later for how to set this fraction in light of the need to achieve a carbon isotopic balance of the system.)

Other weathered organic matter (kerogen) components are then defined as a ratio to the kerogen carbon weathering flux. For POP:

$$\text{rg\_par\_weather\_kerogen\_fracP}$$

which is the ratio of weathered POP vs. POC.

Note that this ratio is not necessarily as simple as buried $\frac{\text{POP}}{\text{POC}}$, because some of the buried POC might be balanced by enhanced volcanic $CO_2$ outgassing (see later).

Finally, note that because by default, the alkalinity associated with implicit nitrogen transformations is associated with phosphorous, any loss of $PO_4$ from the system through POP burial must be accompanied by a negative alkalinity weathering flux equal to $-16.0 \times PO_4$. Again, the alkalinity associated with kerogen weathering is defined as a ratio with kerogen carbon:

$$\text{rg\_par\_weather\_kerogen\_fracALK}$$

Its value will typically be $-16$ times the value of $\text{rg\_par\_weather\_kerogen\_fracP}$.

Of course, it would be far simpler to run with no ALK associated with implicit N transformations (for any experiment). You could set this via:

$$\text{bg\_par\_bio\_red\_PON\_ALK}=0.0$$

(the default is $-1.0$ and in the absence of $N$ tracers, the alkalinity is associated with the uptake and remin of $PO_4$ instead).

Example user- (in: user-configs/EXAMPLES) and base-config files are given for the application of both scheme, and are based on the $36 \times 36$ sediment resolution ‘worbe2’ world configuration of Ridgwell and Hargreaves [1997]. These configurations also make use of a simplified grouping of the various module configuration and boundary condition files, as used per used in the ‘paleo’ configurations and described as part of the muffingen documentation (see: Chapter 18).

To run the 1st and 2nd stage spin-ups described in Ridgwell and Hargreaves [1997]:

```
./runmuffin.sh muffin.CBSR.p_worbe2.BASES EXAMPLES
muffin.CBSR.p_worbe2.BASES.RidgwellHargreaves1997_S36x36.SPIN1 20000
./runmuffin.sh muffin.CBSR.p_worbe2.BASES EXAMPLES
muffin.CBSR.p_worbe2.BASES.RidgwellHargreaves1997_S36x36.SPIN2 50000
muffin.CBSR.p_worbe2.BASES.RidgwellHargreaves1997_S36x36.SPIN1
```

Examples (here, for 20,000 years) for the two different simple organic carbon burial options, with the marine $P$ cycle ‘closed’, but with carbon (and isotopes) ‘open’ (there is carbon loss through sedimentary burial) … but effectively ‘closed’ by the application of restoring boundary conditions for atmospheric $pCO_2$ and $\delta^{13}C$, are:

---

As an alternative, you can also specify a $PO_4$ weathering flux as a ratio to silicate rock weathering via:

$$\text{rg\_par\_weather\_CaSiO3\_fracC}$$
The results of the two different organic carbon burial configurations when run in the 36 × 36 sediment resolution 'worbe2' world configuration of Ridgwell and Hargreaves [1997], are shown in Figures 21.1 and 21.2.

Figure 21.1: Results of 'simple' scheme with 10% assumed burial efficiency vs. the empirical curve of Dunne et al. [2007] (red line).

Figure 21.2: Results of 'dunne2007' scheme vs. the empirical curve of Dunne et al. [2007] (red line)

From the figures, it is noticeable that the 'simple' scheme, in applying a uniform 10% organic
manner preservation efficiency, leads to a substantive over-estimation in the fractional preservation in the range of organic matter rain fluxes that are simulated in this configuration (although it should be noted that these are rain fluxes to the deep sea only). The ’dunne2007’ scheme, while faithfully following the fractional preservation vs. flux relationship, also lacks the higher rain fluxes observed in the modern ocean in the Dunne et al. [2007] and that are not reproduced in the model. Because in these simple examples there are no shallow sedimentary locations, it is likely that both schemes will underestimate global burial. (The carbon burial fluxes as reported in sedgem/seddiag_misc_DATA_GLOBAL.res are: 0.088 PgC yr\(^{-1}\) and 0.027 PgC yr\(^{-1}\), respectively. The latter is equivalent to a mean burial efficiency of 3.1\%).

Finally, examples are provided as to how to configure a fully ’open system’ (excepting that of P). The command-line usage for these examples is:

```bash
./runmuffin.sh muffin.CBSRG.p_worbe2 BASES EXAMPLES muffin.CBSRG.p_worbe2.BASES.Corg_burial.simple.EXPT gl 200000
./runmuffin.sh muffin.CBSRG.p_worbe2 BASES EXAMPLES muffin.CBSRG.p_worbe2.BASES.Corg_burial.dunne2007.EXPT gl 200000
```

(Here – note that the base-config now selected the gemlite module and the experiments make use of the previous examples as restarts.)

The important changes in these user-configs concerns how volcanic CO\(_2\) out-gassing fluxes, and the carbon isotopic signature of weathered carbonates, are set.

In the basic (open-system) 36 \times 36 sediment resolution ’worbe2’ world configuration of Ridgwell and Hargreaves [1997], you would set up with something like:

```bash
# # --- WEATHERING ---------------------------------------------
# # set a ’OPEN’ system
bg_ctrl_force_sed_closedsystem=.false.
# set CaCO3_weathering-temperature feedback
rg_opt_weather_T_Ca=.TRUE.
# set CaSiO3_weathering-temperature feedback
rg_opt_weather_T_Si=.TRUE.
# weathering reference mean global land surface temperature (C)
rg_par_ref_T0=8.48
#CO2 outgassing rate (mol C yr\(^{-1}\))
rg_par_outgas_CO2=5.58E+12
# global silicate weathering rate (mol Ca2+ yr\(^{-1}\))
rg_par_weather_CaSiO3=5.58E+12
# global carbonate weathering rate (mol Ca2+ yr\(^{-1}\))
rg_par_weather_CaCO3=5.58E+12
# d13C
rg_par_outgas_CO2_d13C=-6.0
rg_par_weather_CaCO3_d13C=12.0
```

where there are equal contributions of carbon to the ocean-atmosphere from volcanic CO\(_2\) out-gassing and CaCO\(_3\) weathering, and to balance a 2nd stage spin-up \(\delta^{13}C\) of CaCO\(_3\) burial of 3.4\%/oo, you need a (unrealistic!) \(\delta^{13}C\) of weathered CaCO\(_3\) of approximately 12\%/oo (whereas ideally, it might be close to 3.4\%/oo ...).
There are two ways of isotopically balancing the addition of organic carbon burial in the system – either all via enhanced CO$_2$ outgassing, or a combination of kerogen weathering and enhanced CO$_2$ outgassing. The calculation is presented for enhanced CO$_2$ outgassing only first, and then combining kerogen weathering and enhanced CO$_2$ outgassing. The second scheme is arguably more realistic/better.

1. **Enhanced CO$_2$ outgassing only.**

   - ’simple’
     
     The organic carbon burial flux is 0.088 PgC yr$^{-1}$ and has a mean carbon isotopic signature (taken from the benthic rain flux) of $-23.1^{\circ}/oo$.
     
     If we start by balancing the bulk carbon mass budget and simply increase the rate of volcanic CO$_2$ out-gassing to compensate for carbon loss through organic matter burial, then outgassing becomes:
     
     # CO$_2$ outgassing rate (mol C yr$^{-1}$)
     
     # NOTE: add Corg burial flux to 5.58E+12
     
     rg_par_outgas_CO2=1.2927e+13
     
     The flux-weighted $\delta^{13}$C of the sinks (in units of T mol C $\times^{\circ}/oo$) is:
     
     $7.35 \times -23.1 + 11.59 \times 3.4$
     
     and must balance the input, which is now:
     
     $12.93 \times -6.0 + 5.58 \times x$
     
     where $x$ is the isotopic signature of weathered CaCO$_3$ and in this example has a value of $-9.5^{\circ}/oo$ to achieve isotopic mass balance.
     
     Note that this mass balance is done without any assumed weathering or organic (kerogen) carbon on land. Off-setting some of the enhanced CO$_2$ out-gassing with kerogen weathering creates an more negative net input of reduced carbon and hence requires a more positive carbonate carbon weathering signature.
     
     - ’dunne2007’
       
       A similar calculation can be made for the mass balance of the ‘dunne2007’ scheme option, which gives a 0.027 PgC yr$^{-1}$ burial flux with a mean isotopic composition(taken from the benthic rain flux) of $-23.1^{\circ}/oo$.
       
       The required $\delta^{13}$C of weathered CaCO$_3$ ends up at about 5.7$^{\circ}/oo$.
       
       Note that under a topography with a higher shallow sediment area and hence a higher organic carbon burial than ‘dunne2007’ and ‘worbe2’, you would end up requiring a less positive (or negative) weathered CaCO$_3$ isotopic signature.

2. **Kerogen weathering plus enhanced CO$_2$ outgassing.**

   The bulk carbon mass balance is:
   
   $F_{out(Corg)} + F_{out(CaCO3)} = F_{in(CO2)} + F_{in(Corg)} + F_{in(CaCO3)}$
   
   and isotopically:
   
   $\delta^{13}C_{out(Corg)} \times F_{out(Corg)} + \delta^{13}C_{out(CaCO3)} \times F_{out(CaCO3)} = $
   
   $\delta^{13}C_{in(CO2)} \times F_{in(CO2)} + \delta^{13}C_{in(Corg)} \times F_{in(Corg)} + \delta^{13}C_{in(CaCO3)} \times F_{in(CaCO3)}$
   
   We can simplify things if we assume that: (1) the isotopic composition of weathered kerogen is the same as newly buried organic matter ($\delta^{13}$C$_{out(Corg)}$) and (2), the isotopic composition of weathered carbonate is the same as newly buried CaCO$_3$ ($\delta^{13}$C$_{out(CaCO3)}$)$^{60}$.

$^{60}$ As a point of reference, the long-term Phanerozoic average value of ca. 2 – 4$^{\circ}/oo$.  
Further, the isotopic composition of buried inorganic (carbonate) vs. organic carbon in the model can be related by a fixed offset, $\alpha$. Eliminating $\delta^{13}C_{\text{out}(\text{CO}_3)}$ gives:
\[
(\delta^{13}C_{\text{out}(\text{CaCO}_3)} + \alpha) \times F_{\text{in}(\text{CO}_2)} + \delta^{13}C_{\text{out}(\text{CaCO}_3)} \times F_{\text{out}(\text{CO}_2)} = \delta^{13}C_{\text{in}(\text{CO}_2)} \times F_{\text{in}(\text{CO}_2)} + \delta^{13}C_{\text{in}(\text{CO}_3)} \times F_{\text{in}(\text{CO}_3)}
\]

Now, assuming that a fraction, $\gamma$, of CO$_2$ outgassing helps balance organic carbon burial, we can balance the reduced and carbonate carbon fluxes as follows:
\[
F_{\text{out}(\text{CO}_2)} = \gamma \times F_{\text{in}(\text{CO}_2)} + F_{\text{in}(\text{CO}_3)}
\]
\[
F_{\text{out}(\text{CaCO}_3)} = (1 - \gamma) \times F_{\text{in}(\text{CO}_2)} + F_{\text{in}(\text{CaCO}_3)}
\]

and substituting, gives:
\[
(\delta^{13}C_{\text{out}(\text{CaCO}_3)} + \alpha) \times (\gamma \times F_{\text{in}(\text{CO}_2)} + F_{\text{in}(\text{CO}_3)}) + \delta^{13}C_{\text{out}(\text{CaCO}_3)} \times ((1 - \gamma) \times F_{\text{in}(\text{CO}_2)} + F_{\text{in}(\text{CaCO}_3)}) = \delta^{13}C_{\text{in}(\text{CO}_2)} \times F_{\text{in}(\text{CO}_2)}
\]
from which we can eliminate form both sides the terms $(\delta^{13}C_{\text{out}(\text{CaCO}_3)} + \alpha) \times F_{\text{in}(\text{CO}_3)}$ and $\delta^{13}C_{\text{out}(\text{CaCO}_3)} \times F_{\text{in}(\text{CaCO}_3)}$ to give:
\[
(\delta^{13}C_{\text{out}(\text{CaCO}_3)} + \alpha) \times (\gamma \times F_{\text{in}(\text{CO}_2)}) + \delta^{13}C_{\text{out}(\text{CaCO}_3)} \times ((1 - \gamma) \times F_{\text{in}(\text{CO}_2)}) = \delta^{13}C_{\text{in}(\text{CO}_2)} \times F_{\text{in}(\text{CO}_2)}
\]

and then divide by $F_{\text{in}(\text{CO}_2)}$:
\[
(\delta^{13}C_{\text{out}(\text{CaCO}_3)} + \alpha) \times \gamma + \delta^{13}C_{\text{out}(\text{CaCO}_3)} \times (1 - \gamma) = \delta^{13}C_{\text{in}(\text{CO}_2)}
\]

and hence:
\[
\alpha \times \gamma + \delta^{13}C_{\text{out}(\text{CaCO}_3)} = \delta^{13}C_{\text{in}(\text{CO}_2)}
\]

and finally:
\[
\gamma = \frac{\delta^{13}C_{\text{in}(\text{CO}_2)} - \delta^{13}C_{\text{out}(\text{CaCO}_3)}}{\alpha}
\]

The value of $\delta^{13}C_{\text{in}(\text{CO}_2)}$ is prescribed in the model and is typically $-6^{\circ/oo}$. $\delta^{13}C_{\text{out}(\text{CaCO}_3)}$ is principally dependent on the assumed atmospheric $\delta^{13}C$ and is typically ca. $3^{\circ/oo}$. The value of $\alpha$ ($\delta^{13}C$ of organic matter minus the $\delta^{13}C$ of carbonate carbon) in the model also depends on atmospheric $\delta^{13}C$ as well as ocean surface temperature and atmospheric $p$CO$_2$ and is typically something like $-27^{\circ/oo}$.

Taking the 'simple' configuration as an example and $\gamma = 0.333$, together with equal carbonate and silicate weathering fluxes of $5.58T_{\text{myr}}^{-1}$ (as before), volcanic CO$_2$ outgassing is now adjusted higher by a value equal to $\times 0.333$ the organic carbon burial flux of $7.33T_{\text{myr}}^{-1}$ (0.088Pg$\text{Cy}r^{-1}$), to $5.58 + 2.44 = 8.02T_{\text{myr}}^{-1}$. Kerogen weathering, with a value equal to $\times 0.667$ the organic carbon burial flux $4.89T_{\text{myr}}^{-1}$, is now prescribed in the user-config as a ratio to the silicate weathering flux:
\[
\text{rg\_par\_weather\_CaSi03\_fracC\_d13C=0.87}
\]
\[
(4.89/5.58) \text{and with an isotopic composition equal to organic carbon burial} (-24^{\circ/oo}):
\]
\[
\text{rg\_par\_weather\_CaSi03\_fracC\_d13C=-24.0}
\]

In other words – in this example, we balance the 0.088Pg$\text{Cy}r^{-1}$ of projected organic carbon burial with an additional 0.029Pg$\text{Cy}r^{-1}$ of volcanic CO$_2$ outgassing at $-6^{\circ/oo}$, with the remainder $-0.059Pg\text{Cy}r^{-1}$ – balanced by kerogen weathering with the same (as organic carbon burial) isotopic composition $(-24^{\circ/oo})$.

To put it yet another way – with the weathering and burial of CaCO$_3$ isotopically balancing, we need to isotopically balance $5.58T_{\text{myr}}^{-1}$ of additional CaCO$_3$ burial sourced (in
terms of alkalinity supplied to the ocean) from silicate weathering, with $8.02T_mol yr^{-1}$ of volcanic $CO_2$ outgassing at $-6\%_{oo}$. We do this by now removing $2.44T_mol yr^{-1}$ of organic carbon at $-24\%_{oo}$, i.e. $-6 \times 8.02 = 3 \times 5.58 + -24 \times 2.44$. This implies that for any model-projected organic carbon burial flux less than $2.44T_mol yr^{-1}$, the above assumptions are no longer possible and e.g. the isotopic composition of weathered $CaCO_3$ must become increasingly positive and larger than $3\%_{oo}$. For any projected organic carbon burial flux greater than $2.44T_mol yr^{-1}$, you simply balance the 'excess' burial with kerogen weathering of any identical (to burial) isotopic composition.

The end.
Set up a (silicate) weathering feedback

First ... ensure the system is 'open' (i.e. ocean chemistry can evolve depending on the balance between weathering and burial):

# set an 'OPEN' system
bg_ctrl_force_sed_closedsystem=.FALSE.

To create a temperature (only) dependency for the weathering of carbonate and silicate rocks on land, the following two parameter values need to be set (for carbonate and silicate weathering, respectively):

# set CaCO3_weathering-temperature feedback
rg_opt_weather_T_Ca=.true.

# set CaSiO3_weathering-temperature feedback
rg_opt_weather_T_Si=.true.

(by default they are both .false.).

A prescribed reference temperature governs the dependency of weathering on climate change and modifies the solute fluxes from weathering scaled (non-linearity) to the deviation of mean global land surface temperature from the reference temperature. The mean global surface air temperature over land is given in the BIOGEM time-series file biogem_series_misc_SLT.res and is set equal to the reference temperature parameter:

# weathering reference mean global land air surface temperature (C)
rg_par_ref_T0

The baseline (unmodified) solutes fluxes from terrestrial weathering, in units of mol yr\(^{-1}\), are set by the parameters:

# global carbonate weathering rate (mol Ca2+ yr\(^{-1}\))
rg_par_weather_CaCO3

# global silicate weathering rate (mol Ca2+ yr\(^{-1}\))
rg_par_weather_CaSiO3

for carbonate and silicate weathering, respectively. At steady-state (no climate perturbation) and assuming no other sources of dissolved calcium to the ocean, these must sum up to the total global CaCO\(_3\) burial flux, found in the file: seddiag_misc_DATA_GLOBAL.res (in the genie-sedgem results sub-directory).

Now, to balance the silicate weathering component (and assuming no organic carbon weathering or burial), volcanic CO\(_2\) out-gassing must be assigned a value equal to silicate weathering flux. This is specified by the parameter:

# CO2 outgassing rate (mol C yr\(^{-1}\))
rg_par_outgas_CO2

Even in the absence of organic carbon weathering and/or burial, there are a variety of ways in which the total weathering flux (equal to total CaCO\(_3\) burial at steady state) can be split between carbonate and silicate weathering and hence a value for volcanic CO\(_2\) out-gassing assigned:

1. All solutes could simply be assumed to be derived from carbonate weathering (which is the default assumption in open system experiments without a weathering feedback), e.g.:

   \[\text{rg\_par\_weather\_CaCO3}=10.0\times10^{12}\]
   \[\text{rg\_par\_weather\_CaSiO3}=0.0\]
for a hypothetical example with 10 Tmol yr\(^{-1}\) total global CaCO\(_3\) burial. In the example, no volcanic CO\(_2\) out-gassing is prescribed (because there is no silicate weathering to consume it)\(^{61}\).

2. Silicate and carbonate weathering could also be assumed to be split evenly, e.g.:

\[
\text{rg\_par\_weather\_CaCO3}=5.0E+12 \\
\text{rg\_par\_weather\_CaSiO3}=5.0E+12
\]

(again for a hypothetical example with 10 Tmol yr\(^{-1}\) total global CaCO\(_3\) burial). Volcanic CO\(_2\) out-gassing then needs to be set equal to the baseline silicate weathering flux:

\[
\text{rg\_par\_outgas\_CO2}=5.0E+12
\]

This is the simplest implementation of the silicate weathering feedback, e.g. as used in Lord et al. [2015].

3. A more (modern) 'realistic' partitioning can be made as follows.

Modern continental weathering studies generally partition the atmospheric CO\(_2\) consumption during weathering between silicate and carbonate weathering in the ratio \(\sim 12\) Tmol CO\(_2\) yr\(^{-1}\) to \(7\) – \(12\) Tmol CO\(_2\) yr\(^{-1}\) (e.g. Gaillardet et al. (1999), Ludwig et al. (1998), Amiotte-Suchet et al. (2003), Munhoven (2002)). Assuming that (1) all silicate CO\(_2\) consumption is associated with Ca\(^{2+}\) and that for silicate weathering, 2 mol CO\(_2\) are consumed for every 1 mol Ca\(^{2+}\) released, but (2) CO\(_2\) is consumed only in a 1:1 with CaCO\(_3\) in carbonate weathering, and (3) that a simple mean estimate of CO\(_2\) consumption from the literature range is 9 Tmol CO\(_2\) yr\(^{-1}\) ... leads to an approximate 2:3 (\(\frac{12}{3}\) : 9) split in riverine Ca\(^{2+}\) supply between silicate vs. carbonates.

For our hypothetical 10 Tmol yr\(^{-1}\) of total global CaCO\(_3\) burial, we then have:

\[
\text{rg\_par\_weather\_CaCO3}=6.0E+12 \\
\text{rg\_par\_weather\_CaSiO3}=4.0E+12 \\
\text{and:} \\
\text{rg\_par\_outgas\_CO2}=4.0E+12
\]

4. A final possibility is to assume volcanic CO\(_2\) out-gassing, which then constrains the silicate weathering flux component. The carbonate weathering flux then becomes whatever is required to make the total Ca\(^{2+}\) weathering flux equal to global CaCO\(_3\) burial.

---

\(^{61}\)Unless you have an additional e.g. hydrothermal source of calcium (and alkalinity) to the ocean, when you need an additional source of CO\(_2\) to balance this.
and with it global temperatures will hence be gradually drawn-down until net (carbonate) carbon removal exactly matches the rate of new (volcanic) carbon input. The imbalance between silicate weathering and volcanic CO$_2$ out-gassing is given in the BIOGEM output file: biogem_series_misc_exweather_Ca.res, which records the absolute excess of weathering compared to out-gassing, in units of mol Ca$^{2+}$ yr$^{-1}$ as well as a percentage. In an open system (not a closed one!), this excess needs to be adjusted 'close' (how close? sub 1 percent or 0.1 Tmol Ca$^{2+}$ yr$^{-1}$, certainly) to zero. This adjustment is done by running a short (a few or 10s of years) experiment, reading off the excess weathering value, and doing one of the following:

1. Adjust the reference temperature parameter $rg\_par\_ref\_T0$ – to a lower value if there is an excess of silicate weathering over out-gassing, or
2. Adjust the reference silicate weathering parameter $rg\_par\_weather\_CaSiO3$, or
3. Adjust the volcanic CO$_2$ out-gassing flux parameter $rg\_par\_outgas\_CO2$.

An exact adjustment can be made for the second and third possibilities, with the easiest being to reduce the out-gassing flux value by the amount of excess weathering. For the first option, a couple of iterations will typically be required in order to determine the new reference temperature value that gives rise to a silicate weathering balance. The second option is the recommended one, with the third being the least recommended. Regardless of which of the 3 options, the total weathering flux, given in the BIOGEM time-series file biogem_series_diag_weather_Ca.res, will now be slightly different from the required burial flux. This either requires that the value of the reference carbonate weathering parameter ($rg\_par\_weather\_CaCO3$) is now slightly adjusted, or that a slightly different global carbonate burial flux is allowed.

4. You can of course simply live with a slightly different (long-term), or drifting (short-term), atmospheric CO$_2$ value.

This might be your best bet if the difference is small, and/or it doesn’t really matter, and/or you did not understand any of the above.

Finally, you need to consider the carbon isotopic mass balance of the system. If the carbon isotopic signature of volcanic CO$_2$ out-gassing is assigned a value of e.g. -6.0‰, then the $\delta^{13}C$ of weathered CaCO$_3$ is simply set in order that flux weighted isotopic inputs equal the mean $\delta^{13}C$ of carbonate burial, which as given in the BIOGEM time-series file: biogem_series_sed_CaCO3_13C.res as well as in the SEDGEM summary file: seddiag_misc_DATA_GLOBAL.res (better). The values reported in the 2 files very slightly differ because the BIOGEM time-series average excludes very low wt% sediments and weights the average with wt% carbonate content, while the SEDGEM estimate is burial flux weighted (and hence more appropriate for use in mass balance calculations).

For example, assuming equal fluxes of 5 Tmol yr$^{-1}$ for both weathering components and for volcanic CO$_2$ out-gassing set at -6.0‰, and... assuming a mean carbonate burial $\delta^{13}C$ of 3‰, would require:

$rg\_par\_outgas\_CO2\_d13C$=-6.0
$rg\_par\_weather\_CaCO3\_d13C$=12.0

Of course, in reality organic carbon burial is important and including it would enable a much more realistic value of weathered carbonate $\delta^{13}C$ to be set ... which was discussed previously.
Have some weathering components fixed and others climate-dependent

By default, when carbonate and/or silicate weathering feedbacks are selected (see page 387):

```plaintext
gr_opt_weather_T_Ca=.true.
gr_opt_weather_T_Si=.true.
```

then the fluxes of all (selected) tracers (e.g. nutrients, trace elements) that are associated with carbonate/silicate rocks. also vary with climate. For example, is pyrite is selected (as a solid tracer), then the weathering fluxes of sulphur (as $H_2S$ or oxidized as $SO_4^{2-}$) and iron to the ocean will co-vary with silicate weathering.

Options are provided to decouple bulk carbonate and silicate weathering from their 'daughter' tracers such that e.g. one could have varying silicate weathering (and hence $CO_2$ consumption) but a fixed rate of pyrite weathering, or, fixed silicate weathering by varying pyrite weathering.

1. **Invariant bulk carbonate/silicate weathering**

   Although the carbonate/silicate weathering feedbacks themselves can be turned off, this means that the weathering fluxes derived from all the 'daughter' tracers will also be climate invariant. Instead, 2 parameters are provided that will keep carbonate/silicate weathering invariant, but allow the weathering of 'daughter' tracers to vary in response to climate change.

   These are:

   - `rg_opt_weather_fixed_CaCO3=.true.`
     which will keep the fluxes of $DIC$, $ALK$, and $Ca^{2+}$ associated with the weathering of carbonate rocks, invariant.
   - `rg_opt_weather_fixed_CaSiO3=.true.`
     which will keep the fluxes of $Ca^{2+}$ and $Si$ and the consumption of $CO_2$ associated with the weathering of silicate rocks, invariant.

   Note that climate-responsive rates of carbonate/silicate weathering are still calculated, just not applied to the weathering of the parent rock.

2. **Invariant daughter tracer weathering**

   The options for retaining climate-responsive changes in carbonate/silicate parent rock weathering but keeping daughter tracer weathering invariant, are:

   - `rg_opt_weather_fixed_Li=.true.`
     Fixes (maintains invariant) $Li$ assumed associated with both carbonate
     (rg_par_weather_CaCO3_fracLi) and silicate (rg_par_weather_CaSiO3_fracLi) parent rock weathering.
   - `rg_opt_weather_fixed_Sr=.true.`
     Fixes (maintains invariant) $Sr$ assumed associated with both carbonate
     (rg_par_weather_CaCO3_fracSr) and silicate62 (rg_par_weather_CaSiO3_fracSr) parent rock weathering.
   - `rg_opt_weather_fixed_Os=.true.`
     Fixes (maintains invariant) $Os$ assumed associated with both carbonate
     (rg_par_weather_CaCO3_fracOs) and silicate (rg_par_weather_CaSiO3_fracOs) parent rock weathering.

---

62 Including subdivision into granite and basalt.
• \texttt{rg\_opt\_weather\_fixed\_kerogenC=.true.}.
  Fixes (maintains invariant) kerogen carbon (\textit{DIC}) weathering fluxes assumed associated with silicate (\texttt{rg\_par\_weather\_CaSiO$_3$\_fracC}) parent rock weathering.

• \texttt{rg\_opt\_weather\_fixed\_kerogenP=.true.}.
  Fixes (maintains invariant) kerogen phosphorous (\textit{PO$_4^{3-}$}) weathering fluxes assumed associated with silicate (\texttt{rg\_par\_weather\_kerogen\_fracP*rg\_par\_weather\_CaSiO$_3$\_fracC}) parent rock weathering.\textsuperscript{63}

• \texttt{rg\_opt\_weather\_fixed\_kerogenS=.true.}.
  Fixes (maintains invariant) kerogen sulphur (\textit{H$_2$S}) weathering fluxes assumed associated with silicate (\texttt{rg\_par\_weather\_kerogen\_fracS*rg\_par\_weather\_CaSiO$_3$\_fracC}) parent rock weathering.

• \texttt{rg\_opt\_weather\_fixed\_FeS2=.true.}.
  Fixes (maintains invariant) pyrite weathering (and derived fluxes of \textit{H$_2$S} (or \textit{SO$_4^{2-}$}) and \textit{ALK}) assumed associated with silicate (\texttt{rg\_par\_weather\_CaSiO$_3$\_fracFeS2}) parent rock weathering.

• \texttt{rg\_opt\_weather\_fixed\_CaSO$_4$=.true.}.
  Fixes gypsum) weathering (and derived fluxes of \textit{SO$_4^{2-}$} and \textit{Ca}$^{2+}$)\textsuperscript{65} assumed associated with carbonate (\texttt{rg\_par\_weather\_CaCO$_3$\_fracCaSO$_4$}) parent rock weathering.

• \texttt{rg\_opt\_weather\_fixed\_FeCO$_3$=.true.}.
  Fixes (maintains invariant) siderite weathering (and derived fluxes of \textit{Fe}$^{2+}$ and \textit{DIC}) assumed associated with carbonate (\texttt{rg\_par\_weather\_CaCO$_3$\_fracFeCO$_3$}) parent rock weathering.

• \texttt{rg\_opt\_weather\_fixed\_Ca$_5$PO$_4$$_3$=.true.}.
  Fixes (maintains invariant) apatite weathering (and derived fluxes of \textit{P}, \textit{Ca}$^{2+}$, and \textit{ALK}) assumed associated with silicate (\texttt{rg\_par\_weather\_CaSiO$_3$\_fracCa$_5$PO$_4$$_3$}) parent rock weathering.

• \texttt{rg\_opt\_weather\_fixed\_SiO$_2$=.true.}.
  Fixes (maintains invariant) silica (\textit{H$_4$SiO$_4$}) weathering fluxes assumed associated with silicate (\texttt{rg\_par\_weather\_CaSiO$_3$\_fracSi}) parent rock weathering.\textsuperscript{66}

\textsuperscript{63}And for backward compatibility, also non-kerogen phosphorous associated with silicate (\texttt{rg\_par\_weather\_CaSiO$_3$\_fracP}) parent rock weathering.

\textsuperscript{64}Note that these weathering flux components are independent of any apatite weathering.

\textsuperscript{65}Note that gypsum weathering is neutral w.r.t. alkalinity.

\textsuperscript{66}Note that by default, bulk silicate rock (\textit{CaSiO$_3$}) weathering does not release dissolved \textit{Si} unless a ratio (\texttt{rg\_par\_weather\_CaSiO$_3$\_fracSi}) is explicitly set.
2D(!) weathering

ROKGEM contains one 0D and two 2D weathering schemes. In the 0D scheme, prescribed, global baseline weathering fluxes from terrestrial carbonates and silicates (scaled according to climate conditions if chosen) are distributed across coastal grid cells according to calculated run-off. In 2D weathering schemes, weathering fluxes are a function of local run-off and are calculated for specific lithologies. For each lithology its abundance in each land grid cell as well as the scaling factor with run-off need to be prescribed. Furthermore, the relative contributions of the weathering of each lithology to the terrestrial fluxes of Ca and Si have to be specified. The resulting weathering fluxes can then be scaled to achieve desired total terrestrial inputs of $Ca^{2+}$ and alkalinity, and altered according to temperature and plant productivity.

Two 2D weathering schemes exist, GEM_CO2 based on the Global Erosion Model for CO2 fluxes (Amiotte Suchet and Probst [1995]) and GKWM, based on the Gibbs and Kump Weathering Model (Bluth and Kump [1994]; Gibbs and Kump [1994]; Gibbs et al. [1999]). These two weathering models primarily differ in the assumed functionality between runoff and weathering flux, the underlying distribution of lithologies and whether all granites are treated equally or are divided into felsic and acidic granites.

Each weathering scheme requires consistent sets of lithological abundance maps (one map for each lithology) and weathering parameters specific to the assumed functionality between weathering fluxes and run-off and lithology abundances. These input files (for the pre-industrial Earth system) are provided in: genie_rokgem/data/input/

In order to activate either of the the 2D weathering schemes, ‘GEM_CO2’ or ‘GKWM’, the respective input folders have to be set in the user-config:

```
rg_par_weathopt='GEM_CO2' #or 'GKWM'
rg_par_lith_data='GEM_CO2' #or 'GKWM'
```

(as an example for ‘GEM_CO2’).

- `rg_par_weathopt` – prescribes which ROKGEM subroutine is used to calculate the weathering flux, as well as the name of the file which contains the names of all relevant lithology abundance maps and the file with the corresponding weathering parameters.
- `rg_par_lith_data` – provides the name of the folder which contains the lithology abundance map. (And optionally, also: `rg_par_lith_data_2` and `rg_par_lith_data_3`, all strings will be concatenated by ROKGEM.)

The provided lithology maps might not exactly match the muffin land-sea-mask. With the following to parameters, the lithology maps can be trimmed to the muffin land area and the remaining lithology abundances can be re-scaled to preserve the total amount of each weathered lithology:

```
rg_truncate_to_land=.true. #or .false.
rg_scale_to_landarea=.true. #or .false.
```

---


It can then be chosen whether the weathering fluxes calculated based on run-off and weathering parameters should be scaled:

\[ \text{rg\_calibrate\_weath}\] = true. or false.

and how this scaling should be done:

\[ \text{rg\_calibrate\_weather\_GEM\_CO2\_CaCO3}=1.122913136 \] example value
\[ \text{rg\_calibrate\_weather\_GEM\_CO2\_CaSiO3}=0.484189947 \] example value

or:

\[ \text{rg\_calibrate\_weather\_GKWM\_CaCO3}=1.122913136 \] example value
\[ \text{rg\_calibrate\_weather\_GKWM\_CaSiO3}=0.484189947 \] example value

Furthermore, additional dynamic scaling according to environmental conditions can be applied, either as a global average using the same parameters as in the 0D weathering scheme, or spatially explicit using the following parameters:

\[ \text{rg\_opt\_calibrate\_T\_2D}=\] true. or false., temperature
\[ \text{rg\_opt\_calibrate\_R\_2D}=\] true. or false., run-off
\[ \text{rg\_opt\_calibrate\_P\_2D}=\] true. or false., productivity

If any of these environmental scalings are used, the folders containing the reference maps and base patterns for the calibration need to be provided, e.g.:

\[ \text{rg\_par\_ref\_T0\_2D}=\]
\[ \text{rg\_par\_data\_T0\_2D}=\]

Finally, the calculated weathering fluxes can be altered by specifying local weathering regimes (i.e. whether local weathering is energy or transport limited). This requires the respective option to be set 'true' and the name of the weathering regime mask:

\[ \text{rg\_opt\_weath\_regimes}=\] true. or false.
\[ \text{rg\_weath\_regimes}=\]

2D maps of the weathering fluxes are generated as ascii or netCDF files if the following options are set to true.:

\[ \text{rg\_opt\_2d\_ascii\_output}=\] true.
\[ \text{rg\_opt\_2d\_netcdf\_output}=\] true.

There are several ways in which the 2D weathering schemes can be adapted to Earth system configurations other than the pre-industrial.

1. **Scaling the total Ca and Si input from weathering.**
   The 2D weathering schemes extrapolate weathering fluxes from local run-off and lithology abundance and characteristics. The resulting global fluxes of Ca\(^{2+}\), DIC and alkalinity can be scaled by constants which are then applied to the local inputs transferred to BIOGEM.

2. **Changing weathering parameters.**
   The constants which individualize the parameterized relationship between run-off and weathering flux for each lithology, as well as the relative abundance of carbonates and silicates in each lithology, are specified in an ASCII file named: \text{rg\_par\_weathopt\_consts.dat} in cgenie.muffin/genie-rockgem/data/input, and with \text{rg\_weathop} being the identifier of the chosen 2D weathering scheme.
   The content of the default files provided for pre-industrial Earth system configurations is taken from Table 2 in Colbourn et al. (2013)\(^{71}\).

---

\(^{71}\)Colbourn, G., Ridgwell, A., Lenton, T. M. (2013). The rock geochemical model (RokGeM) v0. 9. Geoscientific Model Development, 6(5), 1543-1573.
3. **Providing different lithology abundance maps.**

New lithology abundance maps require the creation of a new folder in: `cgenie.muffin/genie-rockgem/data/input/` named according to the format: `lithologies_ + rg_par_lith_data + resolution`, with the `lith_data` parts of the name matching those prescribed in the `user-config` file and `resolution` being the horizontal resolution of the maps, written e.g. as `36x36`.

This folder should then contain one map for each lithology covered by the chosen 2D weathering scheme as an ASCII (.dat) file.

By default, the 2D weathering schemes only calculate weathering fluxes of $Ca^{2+}$, alkalinity and DIC (and including $^{13}C$ and $^{14}C$). Any other tracer needs to be added to the 2D weathering schemes before their marine inputs can be calculated based on the specified lithology maps.

For example, $Os$, $^{187}Os$ and $^{188}Os$ were added to the 2D weathering schemes with the simplifying assumption that $Os$ supply from weathering scales to run-off via the same functionality that is parameterized for the weathering DIC fluxes. The local weathering fluxes of $Os$ and its isotopes can then be calculated by scaling the local rates of carbonate and silicate weathering with the prescribed abundances of $Os$, $^{187}Os$ and $^{188}Os$ in the respective lithology.

These abundances are prescribed in the `cgenie.muffin/genie-rockgem/data/input/rg_par_weathopt + _consts.dat` input files, which have been extended to include the abundance of $Os$ (column 5), the $^{187}Os/^{188}Os$ of $Os$ in this lithology (column 6), and its $^{188}Os/^{192}Os$ (column 7). For the simulation of Os weathering fluxes with a 2D weathering scheme shown in Adloff et al. (2021), it was assumed that $Os$ is predominantly derived from basalt and organic shale weathering. The $^{187}Os/^{188}Os$ signatures of these lithologies were derived from Lu et al. (2017) and Dubin and Ehrenbrink (2015), respectively, and the $Os$ abundances were set in such a way that the total $Os$ flux and its $^{187}Os/^{188}Os$ are equal to those prescribed in the 0D weathering simulations of Adloff et al. (2021).

---

72 And similarly for `rg_par_lith_data_2, rg_par_lith_data_3, ...`

73 The horizontal resolutions of the maps and the `muffin` configuration have to match!


Accelerate the weathering-sedimentation mass balance ('GEMlite')

Also see: The FAQ Chapter.

**NOTE:** The principal assumption underlying accelerated time-stepping is that the gradients in the ocean, including nutrients and hence also the biological pump, do not change. Weathering products are added (and sedimentary burial removed) uniformly to the ocean, which is treated as a single box. Acceleration is hence for when you have a 100% spun-up circulation and biological pump, and just want to accelerate the mass balance of long-lived trace element and isotopic tracers (such as Ca, Sr, Li etc.). And ... with care ... in accelerating the long tail of CO₂ weathering draw-down.

A (pseudo) module is provided: 'GEMlite' which provides a means of much more rapidly solving the weathering-sedimentation mass balance – i.e. the long-term (>10 kyr) carbon cycle processes and feedbacks. The motivation behind GEMlite is the stark disparity between the time-scales of ocean circulation and biological pump (ca. 0.1-1000 years) and those of sedimentation and weathering (2-20 kyr) and particularly the silicate weathering feedback (>100 kyr). This makes running cGENIE to an open system steady state (with or without the silicate weathering feedback) challenging. Is there any way of 'accelerating' the calculation of the 'long tail' [Archer et al., 2009] of the CO₂ curve (e.g. in response to fossil fuel CO₂ emissions)?

The philosophy is as follows: the long-term weathering-sedimentation processes are effectively just an imbalance between the supply of solutes via weathering and preservation and burial of esp. carbonates in deep-sea (and shallow) marine sediments. For a small imbalance between weathering and sedimentation, atmospheric pCO₂ and climate (and hence the solute flux when including weathering feedbacks) will only change very slightly. For long intervals characterized by only a small imbalance in weathering-sedimentation the key assumption is made: Ocean circulation and the biological pump, and hence the gradients of dissolved species in the ocean can be considered invariant*. Hence, for the purpose of solving weathering-sedimentation over an intervals of time: The ocean can be treated as a *single box*. It further assumes that: The ocean is initially in equilibrium with the atmosphere (w.r.t. CO₂). (This latter assumption does place important limitations on under what circumstances GEMlite can be employed to accelerate experiments.)

This is what GEMlite does – it solves for weathering-sedimentation and applies the mass difference uniformly throughout the ocean (as if it were a single box), hence preserving the tracer gradients in the ocean. It also (optionally) calculates and re-partitioning of carbon between ocean and atmosphere. Because ocean circulation and the biological pump etc. do not have to be re-calculated, the accelerated quasi box-model phase can be calculated very considerably faster than the 'full' model. Obviously, if atmospheric pCO₂ and hence climate are changing at an appreciable rate then the assumption of invariance in ocean tracer gradients breaks down and it is not 'safe' to apply the accelerated calculation. Similarly, appreciable changes in nutrient inventories will affect the biological pump and hence also change tracer gradients.

The key to employing GEMlite, in addition to knowing when it is appropriate/not appropriate to employ it, is to decide what balance of accelerated (GEMlite) time-stepping vs. normal (full system update of ocean circulation, biological pump, etc.) time-stepping to employ. This division is implemented by creating a sequence of accelerated vs. non-accelerated time-stepping. This can be done in one of two ways:

1. Fixed sequence.
    By default, GEMlite will employed a fixed, pre-determined sequence of accelerated vs. non-accelerated time-stepping. The parameters to specify this sequencing are:
Chapter 21. HOW-TO (experiment)

ma_gem_notyr – which sets the number of years (the assumed time-step of GEMlite) for 'normal' time-stepping.

ma_gem_yr – which sets the number of years for accelerated time-stepping.

For instance: if \( \text{ma_gem_notyr}=50 \) and \( \text{ma_gem_yr}=50 \), you would have a sequence with 50 years of full updating, followed by 50 years of accelerated.

For instance: if \( \text{ma_gem_notyr}=10 \) and \( \text{ma_gem_yr}=90 \), you would have a sequence with 10 years of full updating, followed by 90 years of accelerated.

etc.

Note that the GEMlite cycle phase of 'normal' time-stepping is *always* done first.

Also note that choosing e.g. \( \text{ma_gem_notyr}=10 \) and \( \text{ma_gem_yr}=100 \), while appearing a desirably simple ratio, would result in the change-over point in cycle phase (to accelerated) occurring at the end of year 10, 120, 230, 240, etc. – something that might affect/influence your choice of data saving pattern (i.e., the sequence of time-points for time-series and time-slice data saving).

By default, the parameter values are: \( \text{ma_gem_notyr}=999999 \) and \( \text{ma_gem_yr}=1 \) meaning that in practice you will never get to the end of the 'normal' time-stepping phase. Note that these parameters are integers (setting real numbers, e.g. 1.0E6 will not work ...).


Here, GEMlite attempts to be clever and optimizes the ratio between the duration of each phase of the cycle.

The motivation for this is that often in model experiments, environmental parameters will tend to change faster at the beginning of an experiment compared to towards the end. Fossil fuel CO2 release and its long tail of declining pCO2 is a good example of this. Obviously this complicates the choice of a (fixed) ratio of cycle phases – 100:100 (or more likely: 1000:1000) might not lead to too much degradation of the simulation, but you would only gain a speed advantage of x2 for the experiment as a whole, which if 100-1000 kyr in total duration, is still going to be long. On the other hand: 10:90 would give you a factor almost x10 increase in overall speed, but would seriously degrade the simulation during the initial, rapidly changing environment following CO2 release.

In adaptive sequencing, time-stepping is adjusted via 2 criteria:

- In the normal time-stepping phase, if the rate of change of pCO2 is *more than* a specified threshold over any one year, then the total duration of this phase is extended by one year.

- In the accelerated time-stepping phase, if the total change in pCO2 since the last normal phase is *less than* a specified threshold, then the total duration of this phase is extended by one year.

The result is that the phase durations are always a minimum of the values set by \( \text{ma_gem_notyr} \) and \( \text{ma_gem_yr} \). If it is 'unsafe' to switch to accelerated mode, because pCO2 is changing rapidly, then the model stays in normal mode. If it is safe to stay in the accelerated mode, because pCO2 has not changed much in total during the phase, then the model stays in the accelerated phase.

The parameter names are default values for the two thresholds are:

- \( \text{ma_gem_adapt_dpC02dt}=0.1 \) (ppm yr-1)

- \( \text{ma_gem_adapt_dpC02}=1.0 \) (ppm)

but these will not necessarily be the ideal of any particular experiment (and some trial-and-
error ma be called for). Adaptive time-stepping is enabled by setting:

```
ma_gem_adapt_auto=.true.
```

(by default it is .false.).

The switching between normal (non accelerated) and accelerated phases is saved in a time-series file: biogem_series_misc_gemlite.res

As a further refinement, the accelerated phase can be set to be relatively short to begin with, but gradually increasing in length. The parameters controlling this are:

- `ma_gem_yr` – the initial accelerated phase duration
- `ma_gem_yr_max` – the maximum accelerated phase duration
- `ma_gem_adapt_dgemyr` – the (minimum) fractional increase in duration each cycle (or 1.0 yr, whichever is greater)

A reasonable set of parameters would be:

- `ma_gem_notyr=10`
- `ma_gem_yr=10`
- `ma_gem_yr_max=990`
- `ma_gem_adapt_dgemyr=0.05`
- `ma_gem_adapt_dpCO2dt=0.10`
- `ma_gem_adapt_DpCO2=0.01`
- `ma_gem_adapt_auto=.true.
- ma_gem_adapt_auto_unlimitedGEM=.false.

Finally ... you will need a `base-config` that has GEMlite enabled. This actually requires nothing more than the addition of a couple of lines (to a `base-config` file):

```
ma_flag_gemlite=.TRUE.
```

which can go e.g. near the start of the file under # GENIE COMPONENT SELECTION. Plus:

```
ma_kgemlite=xx
```

which can go e.g. under # TIME CONTROL AND TIME-STEPPING.

Here, xx will depend on the time-step assumed in the base-config. This is likely to be either 96: the standard for most `base-configs`, or 48: for low resolution and faster model configurations, which typically have .t48 in their filename. By convention, I name `base-configs` including GEMlite with _gl, e.g. cgenie_eb_go_gs_ac_bg_sg_rg_gl.p0000c.BASESLi.t48.config but you can name it BobTheLeglessPony for all I care.

The most important thing is to ensure you are not seriously degrading model fidelity (of carbon cycle simulation) by your adoption and configuration of GEMlite. Test different assumptions of how the time-stepping phases are scheduled and compare (of possible) against a full experiment in which GEMlite is not used.

It is important to recognize that when the model switches into the GEM phase, it assumes all ocean tracer gradients are fixed, and updates only ocean composition as a whole according to weathering vs sedimentation imbalance (and also tries to re-equilibrium ocean and atm). As part of this, the flux to the sediments is taken from the average of the last year of the preceding normal phase, and fixed. This also means that the d13C of the CaCO3 deposited to the sediments is fixed
... even if the ocean d13C is being updated and changing ... So, basically you lose the feedback that leads to d13C converging as sinks balance (weathering and volcanic) inputs\textsuperscript{76}.

The solution is to not run in the GEM phase for such long intervals – instead giving the normal phase a chance to make a brief update of ocean gradients and also d13C of export flux. BUT, if pCO2 hardly changes, cGENIE runs the risk of staying in the GEM phase for ever (ish)!

A further option: `gem_adapt_auto_unlimitedGEM` sets whether GEM is allowed an unlimited phase duration or not. By default it is `.false.`. This means that the maximum GEM duration is limited to the normal `gem_yr` parameter. Also, if excessive \( pCO_2 \) drift occurs, the model will immediately switch to the normal phase.

By default then:
- `gem_notyr` specifies a MINIMUM duration for a normal phase..
- `gem_yr` specifies a MAXIMUM duration for a GEM phase.

Values of `gem_yr` much less than 100 are not advisable as you will not reestablish a new equilibrium gradient of tracers in the ocean in that time.

\textsuperscript{76} Adjusting the fluxes themselves during the GEM intervals would break the underlying assumption inherent in the acceleration approximation.
Run the sediments at higher resolution (as compared to the ocean grid)

By default (as set in the base-config file in ~/cgenie.muffin/genie-main/configs) the SEDGEM sediment grid is configured at a resolution of $36 \times 36$ (and on an equal area grid), by:

```bash
SEDGENLONSOPTS='$(DEFINE)SEDGENLONS=36'
SEDGENLATSOPTS='$(DEFINE)SEDGENLATS=36'
```

Several data input files are required by SEDGEM consistent with the specified grid:

- A mask, which specifies the sediment grid locations (if any!) at which sediment cores (see: Ridgwell [2007]) are to be generated at:
  ```bash
  sg_par_sedcore_save_mask_name="sedgem_save_mask.36x36"
  ```
  The example provided on SVN contains some illustrative locations set (by a ‘1’) for cores to be generated.

- The required sediment grid topography (bathymetry):
  ```bash
  sg_par_sed_topo_D="sedgem_topo_D.36x36"
  ```
  This particular grid is derived from observed bathymetry and excludes sediment locations shallower than the surface ocean layer (of the 8-level model) as described in Ridgwell and Hargreaves [2007].

As described in Ridgwell and Hargreaves [2007], SEDGEM can be sub-gridded to a resolution of $72 \times 72$ (equal area). The following namelist parameter additions are necessary to the user-config file:

```bash
SEDGENLONSOPTS='$(DEFINE)SEDGENLONS=72'
SEDGENLATSOPTS='$(DEFINE)SEDGENLATS=72'
sg_par_sed_topo_D="sedgem_topo_D.72x72"
sg_par_sedcore_save_mask_name="sedgem_save_mask.72x72"
```
21.9.1 ... OMEN-SED

Quick background

In the original version of cGENIE, (particulate) organic matter (OM) reaching the bottom of the ocean is completely remineralized, instantaneously releasing dissolved carbon and nutrients (i.e. a reflective boundary approach). Thus the cycles of OM and associated nutrients (PO$_4$) are closed. The remineralization of OM occurs either in the deepest ocean cell (if SEDGEM is not coupled) using the terminal electron acceptors (TEA) available or within the SEDGEM module assuming all OM is respired using oxygen thus completely neglecting the characteristic redox zonation in marine sediments and its impacts on marine nutrient and carbon cycles. As this reflective boundary approach does not simulate any OM-related burial fluxes it overestimates benthic recycling fluxes and does not allow for direct comparison of model output with the sedimentary record.

In order to overcome these limitations the new, one-dimensional, numerically efficient early diagenetic model OMEN-SED has been developed and was coupled to cGENIE (Hülse et al. [2018]). OMEN-SED is the first analytical diagenetic model to explicitly describe OM cycling as well as associated dynamics of the most important TEAs (i.e. O$_2$, NO$_3$, SO$_4$), related reduced substances (NH$_4$, H$_2$S), the full suite of secondary-redox reactions, macronutrients (PO$_4$) and associated pore water quantities (ALK, DIC). Thus, OMEN-SED captures most of the features of a complex, numerical diagenetic model, however, its computational efficiency allows the coupling to global Earth system models and therefore the investigation of coupled global biogeochemical dynamics over different timescales.

Couple OMEN-SED to cGENIE

base-config

As the new Organic Matter ENabled SEDiment model is a sub-module of SEDGEM and you might want to balance simulated burial fluxes with appropriate (or arbitrary) weathering fluxes using ROKGEM both modules need to be enabled. Thus ensure that at the top of the base-config file the following flags are set:

ma_flag_sedgem=.TRUE.
ma_flag_rokgem=.TRUE.

user config

The default version of OMEN-SED is selected by setting:

sg_par_sed_diagen_Corgopt = 'huelse2016'

Because the nitrogen cycle is not modeled explicitly in the current version of BIOGEM alkalinity (ALK) is directly linked to uptake and remineralization of PO$_4$ using the standard 16:1 N:P ratio (compare Ridgwell et al. [2007]). Because ALK in OMEN-SED is tied to the remineralization of OM we now also tie ALK in BIOGEM to POC by setting the following namelist flag:

bg_ctrl_bio_red_ALKwithPOC=.true.

In the following experiments we assume the simplest scenario and focus on OM burial alone, i.e. neglecting any CaCO$_3$ burial (hence also no CaCO$_3$ weathering rate). Therefore, respective CaCO$_3$-diagenesis options are set to NONE and .false. and weathering fluxes are specified as 0.0 (an alternative closed-system setup without ROKGEM also exists and will be described at a later stage). As we now employ OMEN-SED to calculate benthic recycling and burial fluxes an open system must be specified in the user config:

---

This section was written by Dominik Hülse. Any complaints/suggestions should be directed to him: dominik.huelse@ucr.edu or dominik.huelse@gmx.de.
# sediment diagenesis options
sg_par_sed_diagen_CaCO3opt="NONE"
sg_ctrl_sed_bioturb=.false.
sg_ctrl_sed_bioturb_Archer=.false.
...

# --- WEATHERING -----------------------------------------------
#
# set an 'OPEN' system
bg_ctrl_force_sed_closedsystem=.false.
# set total weathering rate
rg_par_weather_CaCO3=0.0
# set bulk weathering isotopic composition
rg_par_weather_CaCO3_d13C=0.0

Atmospheric pCO$_2$ and associated $\delta^{13}$C is restored by specifying a forcing file and values (i.e. setting bg_par_forcing_name, bg_par_atm_force_scale_val_3 and bg_par_atm_force_scale_val_4. As we do not model CaCO$_3$ preservation we have to apply a globally uniform nonbiogenic, noncarbonate detrital flux in order to simulate (somehow) meaningful OM weight percentages (OM wt%) in the sediments. This is set by:

sg_par_sed_fdet=0.180

The prescribed flux of 0.18 g cm$^{-2}$ky$^{-1}$ represents a value characteristic of early Cenozoic open-ocean sediments [Zeebe&Zachos, 2007]. (For the modern ocean (as in Hülse et al. [2018]) we ran experiments with prescribed fixed solid fields of CaCO$_3$, opal and detritus to the sediments. These were regridded from observations.)

The most important parameters in OMEN-SED are the first-order degradation rate constants of the two simulated OM pools (comparable to BIOGEM a labile and a refractory pool is simulated). A considerable amount of effort has been expended to link these rate constants for instance to sedimentation rate and various empirical parameterizations have been proposed of which some are implemented in OMEN-SED (e.g. Tromp et al. [1995], Boudreau [1997], Stolpovsky et al. [2015]. Also a temperature dependent degradation scheme exists). However, here only the use of globally uniform degradation rate constants is described. By default the degradation rate constant for the refractory OM pool depends on the oxygenation of the overlying ocean waters (i.e. a lower rate constant is specified for [O$_2$] < 5.0 μM). The degradation rate constant for the labile OM pool does not depend on oxygen concentration. The three degradation rate constant can be specified by setting:

# global OM degradation rate constants (1/yr)
sg_par_sed_huelse2017_k1=0.0065
sg_par_sed_huelse2017_k2=0.005
# k2 under anoxic bottom water:
sg_par_sed_huelse2017_k2_anoxic=1E-6

where the values above represent the default degradation rate constants as used in in Hülse et al. [2019].

**Do something with the P-cycle**
The default OMEN-SED configuration omits P dynamics and all organic phosphorus reaching the bottom of the ocean is instantaneously recycled and returned to the deepest ocean grid-cell as PO$_4$. 
However, various options for P burial/regeneration are available. All options described below cause a decline in oceanic [PO$_4$] due to P-burial which needs to be restored via weathering. One option is to again setup a closed P-system by setting the following namelist flag:

bg_ctrl_force_sed_closed_P = .true.

Here, the P-weathering flux equals P-burial and thus closes the P-cycle. A second option (relevant for transient simulations) is to define a weathering input of riverine dissolved P (in mol yr$^{-1}$, i.e. by specifying rg_par_weather_CaSiO3=1.0 and rg_par_weather_CaSiO3_fracP). The second option will be described in more detail in the section on transient simulations at a later stage.

**1) P-cycle as in Hülse et al. [2018]:**
OMEN-SED represents a redox-dependent P-cycle which accounts for P recycling through OM degradation and the formation and burial of Fe-bound P and authigenic Ca–P minerals. Note, that this representation is not able to represent enhanced P-regeneration under anoxia (i.e. lower than ’Redfield’ C/P regeneration ratios – for this see option 3) below). The P-cycle as described in Hülse et al. [2018] can be selected by changing the P-cycle ’flag’ to true

sg_par_sed_huelse2017_P_cycle=.true.

**2) Organic P burial equals organic C burial**
A simpler scheme is to assume that the same fraction of organic P is buried as organic carbon. This means in every grid cell the burial of organic P (P$_{orgb}$) is calculated from the amount of buried organic C (C$_{orgb}$) as P$_{orgb}$ = 1/106 $\times$ C$_{orgb}$. This option can be selected by setting:

sg_par_sed_huelse2017_P_cycle=.false.
sg_par_sed_huelse2017_sim_P_loss = .true.
sg_par_sed_huelse2017_sim_P_loss_pres_fracC = .true.

**3) Redox dependent P-regeneration**
OMEN-SED furthermore provides a P-scheme that is able to simulate the positive feedback between productivity, bottom water anoxia and enhanced P-regeneration. In order to simulate the oxygen dependence of the C/P regeneration ratio we employ the empirical function derived by Wallmann [2010]. This will be described in more detail at a later stage. This option can be selected by setting:

sg_par_sed_huelse2017_P_cycle=.false.
sg_par_sed_huelse2017_sim_P_loss = .true.
sg_par_sed_huelse2017_sim_P_regeneration = .true.

**Running transient simulations**
Two step process:
1) Run to steady-state with forced atmospheric concentrations (e.g. CO$_2$, O$_2$ and $\delta^{13}$C)
2) Diagnose organic carbon burial in seddiag_misc_DATA_GLOBAL.res as total POC pres – actually better: diagnose as the mean burial (in mol yr$^{-1}$ over the e.g. last 2000 years using the time-series files: biogem_series_focnsed_POC.res - (biogem_series_fsedocn_DIC.res - biogem_series_fsedocn_Ca.res)
3) Set the weathering rates accordingly: rg_par_weather_CaSiO3=1.0 and rg_par_weather_CaSiO3_fracC (and rg_par_weather_CaSiO3_fracP)
This will be described in more detail at a later stage.
Example config-files

This will be described in more detail at a later stage. An example base-config file has been created (muffin.CBS.fkl_pp01.BASES_DH) and is available in genie-main/configs/. It is the same as the respective fake world configuration only here a flat bottom ocean is used. The following user-config files are available in genie-userconfigs/OMEN:

1) No P-cycle in OMEN-SED:
Here the P-cycle in the sediments is completely ignored, i.e. all organic P reaching the ocean floor is instantaneously remineralized and given back to the lowest ocean grid-cell. This can be executed for instance as:

```
$ ./runmuffin.sh muffin.CBS.fkl_pp01.BASES_DH OMEN
muffin.CBS.fkl_pp01.BASES.OMEN_Porgoff.SPIN 20000
```

2) Organic P burial equals organic C burial:
This can be executed for instance as:

```
$ ./runmuffin.sh muffin.CBS.fkl_pp01.BASES_DH OMEN
muffin.CBS.fkl_pp01.BASES.OMEN_Porgfix.SPIN 20000
```

3) Redox dependent P-regeneration
This can be executed for instance as:

```
$ ./runmuffin.sh muffin.CBS.fkl_pp01.BASES_DH OMEN
muffin.CBS.fkl_pp01.BASES.OMEN_Porgredox 20000
```

**OMEN-SED model output**

As for CaCO$_3$, 2D sediment properties can be found in the sedgem sub-directory of your experiment directory in a netCDF file called fields_sedgem_2d.nc. The 2D distribution of wt% organic matter is saved for two depth levels in the sediment column. First OMEN_wtpct_top saves the mean weight fraction of OM in the upper 5cm of the sediment layer. Second OMEN_wtpct_bottom saves the mean weight fraction of OM at a sediment depth of 1m — which is the total depth of the simulated sediment column in OMEN-SED and is assumed as lost to long-term burial.

The file seddiag_misc_DATA_GLOBAL.res summarizes the mean global preservation flux of organic matter (here POC). Note, that in this file (when using OMEN-SED) ‘POC rain’ always equals ‘Total POC preservation’. The actual value for POC rain is saved in the corresponding BIOGEM summary file biogem_year_*_diag_GLOBAL.res.
21.10 HOW-TO ... geoengineer

Set up carbon dioxide removal (CDR) geoengineering experiments

There are various ways to implement different carbon dioxide removal strategies in cGENIE. For instance – additions of dissolved iron and phosphate can be implemented as simple flux forcings to the ocean surface, as described in the Tutorial exercises. Similarly, ocean ’liming’ can be implemented as a flux forcing of alkalinity to the surface (with or without associated Ca2+ and with or without additional CO2 emissions to the atmosphere due to the creation of lime). There is also the facility for automatically calculating the liming required for a specific policy target (atmospheric CO2 history or desired mean ocean surface pH or saturation state). This is described in a subsequent HOWTO. This section describes a framework created for applying additional geoengineering modifications and particularly ones that cannot be implemented as a simple flux forcing.
Applying a geoengineering 'liming' flux to the ocean surface

There are two different methodologies provided for: (1) applying a surface flux of alkalinity (and Ca, and DIC) with a uniform or spatial pattern, regardless of atmospheric $pCO_2$ and emissions, and (2) applying a surface flux calculated internally to meet some objective – here, the value of atmospheric $pCO_2$ at any point in time.

1. The forcing:

```
pyyyyyz_RpC02_Rp13C02_FRALK_FDIC_F13DIC_FCa
```

provides a facility for applying alkalinity (ALK) (as well as Ca, and DIC, if selected) to the ocean surface concurrently with a prescribed time-history of restoring of atmospheric $pCO_2$ (and associated $\delta^{13}C$), while

```
pyyyyyz_FpC02_Fp13C02_FRALK_FDIC_F13DIC_FCa
```

is similar, except with a prescribed time-history of flux (emissions) of $CO_2$ (and associated isotopic composition) to the atmosphere. By default in each, only an ALK flux to the ocean surface is selected. The other three ocean tracers listed in `configure_forcings_ocn.dat` are set to 'f' for flux forcing. As provided in these examples, atmospheric $pCO_2$ is held at 278 ppm for the restoring, and emissions of 1 PgC yr\(^{-1}\) (8.3333e+013 mol yr\(^{-1}\)) for the flux forcing version.

By default, `forcings` are set such that the ALK (and Ca, and DIC if selected) is applied uniformly over the ocean surface (the '2' in COLUMN #06). Point sources can be specified by changing this to a '0', or an explicit spatial pattern with a '-2' (which must then be provided in a separate file).

2. The forcing:

```
pyyyyyz_FRpC02_Fp13C02_FRALK_FDIC_F13DIC_FCa
```

differs firstly in specifying both a $CO_2$ emissions flux and restoring value for atmospheric $pCO_2$. For alkalinity (ALK), both a flux and restoring of the ocean are selected.

Basically, what happens here is when a flux + restoring of ocean ALK is selected together with flux + restoring of $pCO_2$, ALK additions to the ocean is made in order to try and maintain the prescribed history of atmospheric $pCO_2$, regardless of the $CO_2$ emissions also specified. If at any one time-step atmospheric $pCO_2$ is higher than the target value, ALK is added to the ocean with the flux specified in the ALK flux forcing. If atmospheric $pCO_2$ is lower than the target value, no ALK is added or taken away, unless the parameter:

```
bg_ctrl_force_invert_noneg=.true.
```

is set, which enables a negative ALK flux to be applied\(^{78}\).

Note that the restoring value of ALK has no meaning and the value set in:

```
biogem_force_restore_ocn_ALK_sig.dat
```

is not important. Both atmospheric $CO_2$ flux forcing and restoring forcing time-series are specified 'as normal', and may constitute an SRES $CO_2$ emissions scenario and RCP based $pCO_2$ time-history, respectively, for example. For d13C, only the d13C of emissions is specified\(^{79}\).

If an atmospheric $CO_2$ emissions is not required, simply set the value in the time-series file to zero. Note that no scaling of the atmospheric $CO_2$ forcing\(^{80}\) must be applied because it will scale both restoring and emissions ...

---

\(^{78}\) This is likely pretty un-physical for most applications, hence the default is `.false..`

\(^{79}\) A restoring must *not* also be set.

\(^{80}\) e.g. `bg_par_atm_force_scale_val_3`
Chapter 21. HOW-TO (experiment)

If the model cannot quite attain the $pCO_2$ target specified, you probably do not have a sufficiently large ALK flux specified. Conversely, if the $pCO_2$ target is significant over-shot (technically: under-shot) then you might have prescribed too large a flux.
Geoengineering with ... 'pipes'

Pipes are parameterized following Yool et al. [2009] (Yool, A., J. G. Shepherd, H. L. Bryden, and A. Oschlies (2009), Low efficiency of nutrient translocation for enhancing oceanic uptake of carbon dioxide, J. Geophys. Res., 114, C08009, doi:10.1029/2008JC004792.) This is selected by setting:

\[ \text{bg\_opt\_misc\_geoeng='pipes'} \]

(Currently, there is no other option and anything passed other than a value of 'pipes' results in the default: no geoengineering.) A series of option then control the working of the pipes:

- A mask file is provided to designate the grid points of the ocean with pipes in them, via:
  \[ \text{bg\_par\_misc\_2D\_file} \]

  with a default of 'misc.dat'. The default file location is: `cgenie.muffin/genie-biogem/data/input`. This file is treated in a similar way to the normal 2D forcing files. The values at each grid point can be scaled via the parameter: `bg\_par\_misc\_2D\_scale`. The units are m³ per year. E.g. setting `bg\_par\_misc\_2D\_scale=1E13`, assuming values of 1.0 or 0.0 in misc.dat will create an annual vertical advective flux at each grid point equivalent to 30% of the volume of the surface cell (3.2E13).

- The ocean depth level associated with the base of the pipes is set via:
  \[ \text{bg\_par\_misc\_kmin\_pipe=12} \]

- Three parameters are then provided to control what is advected, with a number of combinations of tracers possible (useful for diagnosing the relative importance of e.g. nutrients vs. resired CO2 vs. temperature and salinity (and hence ocean circulation changes)):

  - # pump T and S?
    \[ \text{bg\_ctrl\_force\_GOLDSTEInTS=.false.} \]
    (the default) prevents T and S being advected.
  - # ONLY pump T and S?
    \[ \text{bg\_ctrl\_force\_GOLDSTEInTSonly=.true.} \]
    results in *only* T and S being advected. This requires that `bg\_ctrl\_force\_GOLDSTEInTS=.false.`. Its default is .false..
  - # pump no DIC?
    \[ \text{bg\_ctrl\_misc\_geoeng\_noDIC=.false.} \]
    prevent DIC from being advected.

The following combinations are then valid (shown commented out (#) are the settings that are the same as the default settings and hence do not need to be re-defined, although it would not hurt to):

1. `bg\_ctrl\_force\_GOLDSTEInTS=.true.`
   \[ #bg\_ctrl\_force\_GOLDSTEInTSonly=.false. \]
   \[ #bg\_ctrl\_misc\_geoeng\_noDIC=.false. \]
   results in everything being advected.
2. \#bg_ctrl_force_GOLDSTEInTS=.false.
   \#bg_ctrl_force_GOLDSTEInTSonly=.false.
   \#bg_ctrl_misc_geoeng_noDIC=.false.
   results in everything (nutrients, DIC, ALK, isotopes, etc.) except T and S being advected.

3. \#bg_ctrl_force_GOLDSTEInTS=.false.
   \#bg_ctrl_force_GOLDSTEInTSonly=.false.
   bg_ctrl_misc_geoeng_noDIC=.true.
   results in everything except T and S *and* DIC being advected (i.e. just nutrients, alkalinity, isotopes etc.).

4. bg_ctrl_force_GOLDSTEInTS=.true.
   bg_ctrl_force_GOLDSTEInTSonly=.true.
   #bg_ctrl_misc_geoeng_noDIC=.false.
   results in only T and S being advected.

Combination #1 is arguably the only realistic setting – the others being for diagnosing how the model works and the primary controls on the effectiveness or otherwise of pipes, only. The difference between #2 and #1 indicate the importance of changes in ocean circulation driven by T and S, which can also be assessed in isolation via option #4. The difference between option #3 and #2 indicates the importance of the respired CO2 'leak' in the effectiveness of ocean pipes. (Note that there is no option for removing DIC only and e.g. advecting T and S and nutrients etc etc.)
Adjust solar forcing in a time-dependent manner

The value of the solar constant in cGENIE is set by the namelist parameter:

\texttt{ma\_genie\_solar\_constant}

which by default is set to 1368 W m\(^{-2}\), i.e.:

\texttt{ma\_genie\_solar\_constant=1368.0}

Specifying a different value for \texttt{ma\_genie\_solar\_constant} in the user-config file allows the solar forcing of the EMBM to be altered. For example, to induce a 'snowball Earth' like state under a solar constant applicable to the late Neoproterozoic (some 6% less than modern) you would set:

\texttt{ma\_genie\_solar\_constant=1330.56}

Modification of \texttt{ma\_genie\_solar\_constant} can be turned into a time-dependent forcing of solar forcing, but only by frequent re-starting using a sequence of short model integrations.

Alternatively, a crude (temporary) hack is provided to allow a semi-continual adjustment of solar forcing. Whether you wish to vary the solar constant in a time-dependent manner is determined by the parameter:

\texttt{bg\_ctrl\_force\_solconst}

By default this is set to .false.. By adding to the user-config file:

\texttt{bg\_ctrl\_force\_solconst=.true.}

a time-varying change in the value of the solar constant will be imposed. For this, \texttt{BIOGEM} will expect the presence of a file called \texttt{biogem\_force\_solconst\_sig.dat} in the forcing directory\(^{81}\). This file must contain two columns of information: the first is a time marker (year) and the second is a paired value for the solar constant. In the current crude incarnation of this feature, the time markers (1st column) must correspond exactly to the time markers in the time-series specification file\(^{82}\). \texttt{cGENIE.muffin} will exit with an appropriate error message if this is not the case.

When using the time-varying solar constant hack, seasonal solar insolation is re-calculated each year with a call to \texttt{radfor(genie\_solar\_constant)}\(^{83}\) at the start of the time-stepping loop\(^{84}\). At each time-marker, \texttt{BIOGEM} sets the value of \texttt{genie\_solar\_constant} equal to the corresponding value specified in \texttt{biogem\_force\_solconst\_sig.dat}. Thus, regardless of how closely-spaced the time-marker years are, (seasonal) solar insolation is only adjusted every year. For a longer time-marker interval than yearly, no interpolation is performed on the series of solar constant values, and in this way time-dependant solar forcing currently differs from the calculation of other forcings.

A simple example file might look something like:

\begin{verbatim}
-START-OF-DATA-
0.5  1367.0
1.5  1366.0
2.5  1365.0
3.5  1364.0
4.5  1363.0
\end{verbatim}

\(^{81}\)REMEMBER: The location of which is specified by the namelist parameter \texttt{bg\_par\_fordir\_name}.
\(^{82}\)REMEMBER: The filename of which is specified by the namelist parameter \texttt{bg\_par\_infile\_sig\_name}.
\(^{83}\)\texttt{cgenie.muffin/genie-embm/src/fortran/radfor.F}
\(^{84}\)\texttt{cgenie.muffin/genie-main/genie.F}
which will decrease the value of the solar constant by $1 \text{ W m}^{-2}$ each year. Note that because the solar forcing is only updated each year (with the call to `radfor.F`), the first year will be characterized by climate with a solar constant of $1368 \text{ W m}^{-2}$, the default. Although BIOGEM sets a new value of `genie_solar_constant (1367 \text{ W m}^{-2})` mid way through the first year, it is only at the start of the second year that solar insolation is recalculated according to the reduction in solar constant.

Hacking the solar constant in a time-varying manner is, of course, a (albeit crude) way of addressing SRM geoengineering impacts.
21.11 HOW-TO ... orbit

Specifying a fixed orbital configuration

The parameters to prescribe a fixed (but different from modern) orbital configuration are:

- $ea_{par\_orbit\_osce}=0.0167$
- $ea_{par\_orbit\_oscsob}=0.397789$
- $ea_{par\_orbit\_oscgam}=102.92$

which specify, respectively:

- eccentricity
- sine of obliquity
- longitude of perihelion

and here are shown with their modern, default values.

To specify an alternative (different from modern) orbital configuration, such as for the last glacial maximum (21 ka)\(^{85}\), you would add (to the user-config):

- $ea_{par\_orbit\_osce}=0.018994$
- $ea_{par\_orbit\_oscsob}=0.389911$
- $ea_{par\_orbit\_oscgam}=114.42$

For comparison, the mid-Holocene (6 ka) parameter set\(^{86}\) is:

- $ea_{par\_orbit\_osce}=0.018682$
- $ea_{par\_orbit\_oscsob}=0.408410$
- $ea_{par\_orbit\_oscgam}=0.87$

\(^{85}\)Taken from: PMIP2 experiment boundary conditions.
\(^{86}\)Taken from: PMIP2 experiment boundary conditions.
Specifying time-varying orbits

For model simulations over long timescales you might want to consider the effects of changes with time in the astronomical forcing. In this case, the amount of daily mean insolation at each latitude can vary as a function of eccentricity, obliquity and (climatic) precession, as opposed to under a fixed insolation pattern. Such transient forcing can be applied in muffin by adding the following set of parameter settings to the user-config file:

```
# Call orbit_radfor: Applies astronomical forcing
ea_38 = "y"
# Specify the type of orbital forcing: 0 (default), 1 (time-varying)
ea_39 = 1
# Number of data points in orbits file
ea_40 = 1001
# Interval between data points in units of goldstein time steps
ea_41 = 96000
# filename for orbital parameters
ea_42 = "orbits_La2004_1Myr.dat"
```

In this specific example, a file: `orbits_La2004_1Myr.dat` is specified, which provides the astronomical solution for the past 1 million year based on the astronomical solution of Laskar (2004) in time-steps of 1,000 years.\(^87\) For 1 million year in time-steps of 1,000 years, there are 1,000,000 / 1,000 = 1,000 data points (+ 1 for year zero) = 1,001 data points total. The value of parameter `ea_40` needs to be set equal to this value, i.e. prescribing how many data points are present in the file specified by `ea_42`.

Note that the file containing the astronomical solution is read in reverse order, i.e. from the bottom to the top. Thus, for file `orbits_La2004_1Myr.dat`, the values for eccentricity and obliquity used for the first 1000 year of the simulation, are 0.03576 and 0.40080, respectively, corresponding to the astronomical solution 1 million year ago. To change the time slice that you would like to simulate\(^88\) (for example from 500 ka to recent), simply copy the astronomical solution into a new text file and remove the last 500 lines so that the simulation begins with the solution at 500 ka.

The value for parameter `ea_41` depends on the resolution of the model that is used. By default, for a 16-level ocean circulation model configuration, muffin employs 96 time-steps per year in the GOLDSTEIN ocean circulation model component. When the astronomical solution is provided as one orbital value per 1000 years, this equates to 1000 \(\times\) 96 = 96000 GOLDSTEIN time steps. For lower resolution configurations of muffin, GOLDSTEIN may be operating on 48 time steps per year, which means that `ea_41` need to be changed to 48000 (when the astronomical solution is provided 1000 year time steps).

Important. When using one of the paleo configurations of muffin defined in the genie-paleo directory, the file containing the astronomical solution must be present in that directory. For all other configurations, the file lives in genie-embm/data/input. A warning message will be generated and the muffin will exit if the file is not found in the correct directory.

---

\(^{87}\) The data file `.dat` can be found in sub-directory: genie-embm/data/input

\(^{88}\) Don’t forget to update parameter `ea_40`!
Save orbitally-relevant data

The challenge here is that changes in seasonal characteristics are critical to understanding orbital modulated insolation forcing and response, so seasonal or even monthly resolved data is needed ... yet the experiments may be several 100 kyr to Myr in duration ... Saving seasonal or monthly resolved netCDF sufficiently frequently to resolve precession – every few kyr – is possible, but the resulting data file size maybe near unmanageable, and in any case, the net CDF files would have to be laboriously processed. The situation can be helped a little by only saving 2D netCDF (see earlier), but it still remains a task to process a large seasonal/monthly netCDF file. In contrast, time-series data is cheap to save, but is currently saved only as global and annual average, masking the details of any orbital response.

The solution provided here is to save data at specific points in the model grid (that are specified prior to an experiment being run), and to be very specific about what data is saved (i.e. rather than large generic sets of variables being saved as in the time-series and time-slice data saving approaches, only certain variables are saved). Furthermore, because orbital variability is slow compared to the seasonal cycle, the data can be aliased. i.e., rather than saving every time-step during the seasonal cycle, and saving every year (of every seasonal cycle), one can devise an interval of data save, that samples a complete seasonal cycle only when $x$ (where $x > 1$) complete years have been simulated. For example – saving data every 13 months would mean that Jan is saved in year 1, Feb in year 2, Mar in year 3, etc, with all months saved over the course of 12 years and then Jan again in year 13. This would not work so well if significant inter-annual variability was present (i.e. as per in a fully coupled model), but in the basic EMBM-based configuration of muffin, there is no inter-annual variability and hence this strategy should work just fine.

To configure this way of saving data requires that two files are be present in the: genie-biogem/data/input sub-directories, are identified by name via 2 new user-config parameters:

1. bg_par_infile_orb_pts_loc_name
   which directs BIOGEM to a file containing a list of model grid locations, in (i,j) format, and
2. bg_par_infile_orb_pts_var_name
   which directs BIOGEM to a file containing a list of variables to be saved.

The default setting of bg_par_infile_orb_pts_loc_name is file: save.orb_pts_loc.EXAMPLE.dat, whose content looks like:

```
01 01
01 02
01 35
01 36
```

For bg_par_infile_orb_pts_var_name is file: save.orb_pts_var.EXAMPLE.dat, and whose content looks like:

```
ipoa_solfor.
ipoa_fxsol.
ipoa_seaice.
ipoa_cost.
ocn_sur_temp.
ocn_ben_temp.
```

89Regardless of whether or not a paleo configuration is used.
BIOGEM will save *each* variable at *each* location, saving the data in files with names:

- biogem_orb_i01j01.res
- biogem_orb_i01j02.res
- biogem_orb_i01j35.res
- biogem_orb_i01j36.res

The data is written out on each time-slice save step. Internally, in between time-slice saves, BIOGEM can store up to a maximum number of data points, determined by the variable:

bg_n_orb_pts_nmax=0

By default, this is set to zero, and BIOGEM will therefore not save orbital data, nor look to see if the files bg_par_infile_orb_pts_loc and bg_par_infile_orb_pts_var_name even exist. BIOGEM will warn you if you run out of internal storage, should you have e.g. too low a value of bg_n_orb_pts_nmax and/or too infrequent a time-slice save.

The frequency of data sampling is given in units of GOLDSTEIN time-steps, and is set by the parameter:

ma_conv_kocn_korb=9999999999

The default value, effectively disabling saving).

For example:

- ma_conv_kocn_korb = 2 would sample each and every BIOGEM time-step (48 per year) for the standard 16 level ocean (96 time-steps per year in the ocean model).
- ma_conv_kocn_korb = 96 would sample once per year for the standard 16 level ocean (96 time-steps per year in the ocean model).
- ma_conv_kocn_korb = 98 would alias the sampling, so that each year, you sample one BIOGEM time-step later in the year. So after 48 years, you will have sampled each of the BIOGEM time-steps once (and hence have sampled the variability in a year), but only have 48 lines of data, rather than 48*48 lines for 48 years sampling each and every BIOGEM time-step.

For the variable list – the format is similar to the netCDF data names. If the sub-string ‘sur’ appears in the variable string, surface data is saved. If the sub-string ‘ben’ appears in the variable string, then benthic data is saved.

You can pick any of the ocean tracers listed in tracer_define.ocn as part of the string (assuming they are even selected and simulated ...)

- ocn_sur_PO4.
- ocn_surDIC.
- ocn_ben_sal.

---

90(For this example)

91This means that for non orbital data saving, you do not even need to have these default files present.

92I would not recommend exceeding a product of bg_n_orb_pts_nmax X number of locations X number of variables of ... 10,000,000 (whether disk space exists to save a long run, is another matter)

93Likely you’ll want a larger value of ma_conv_kocn_korb than the above examples, because it is still effectively saving some data each and every year.

94Note that if you select an isotope tracer, e.g. the 13C tracer, the output is saved as per mol.
Carbonate variables can also be chosen – the full list of possibilities is:

- \( H \),
- \( \text{fug}_\text{CO}_2 \),
- \( \text{conc}_\text{CO}_2 \),
- \( \text{conc}_\text{CO}_3 \),
- \( \text{conc}_\text{HC}O_3 \),
- \( \text{ohm}_\text{cal} \),
- \( \text{ohm}_\text{arg} \),
- \( \text{dCO}_3\text{cal} \),
- \( \text{dCO}_3\text{arg} \),
- \( \text{RF}0 \).

2D atm-ocn physics variables (no sur or ben) include:

- \( \text{seaice} \),
- \( \text{seaice}_\text{th} \),
- \( \text{solfor} \),
- \( \text{fxws} \),
- \( \text{cost} \),
- \( \text{KCO}_2 \),
- \( \text{MLD} \),
- \( \text{evap} \),
- \( \text{precip} \).

In addition, atmospheric – \( \text{atm}_* \) and biological export – \( \text{bio}_\text{fpart}_* \) tracers are also supported.

**Important.** A DOT is required at the end of each output variable line, which helps the algorithm distinguish different tracers.

Note that because there is an ambiguity between atmosphere and ocean temperature, as they have the same name, when specifying any of atm, ocn, or sed tracers for saving, add ’atm’, ’ocn’, or ’sed’ to the string. Use underscores (‘_’) to separate parts of the string. For now, all ’sed’ tracers requested are assumed to be particulate fluxes. (Again, as before.)

Also note that for saving carbonate chemistry anywhere in the ocean and at any time (not just associated with time slice saving), you need to specify that the carbonate system is re-solved frequently. The setting for this is:

\[ \text{bg_ctrl_carbchemupdate_full=}.true. \]

Note that there may be considerable computational overheads to this. However, the surface equilibrium is solved frequently by default, and if you only want surface, you do not need to add this setting. (So I guess it is only if you want benthic carbonate chem saved is there any issue.)

There is also a previous/old way of saving just orbital insolation:

For evaluating and checking orbital experiments in \texttt{cGENIE.muffin}, it is useful to be able to extract commonly used diagnostics of orbital variations in insolation, e.g. June 21 65N. \texttt{cGENIE.muffin} can be configured to save the \textit{time-series} (at the standard save frequency for \textit{time-series} output) at two different ’j’ grid positions (latitude) – typically one Northern and one Southern hemisphere, and at specific points in the seasonal cycle. The relevant parameters are:
• **bg_par_sig_j_N** Sets the ‘j’ latitude grid point location for extracting the insolation. On a 18 × 18 grid, a value of 17 is as close as you are going to get to 65N.

• **bg_par_sig_j_S** As above, except for the Southern hemisphere (although there is nothing stopping you from choosing 2 Northern or 2 Southern hemisphere points, just what out for the automatic output column labelling that might cause confusion).

• **bg_par_t_sig_count_N** Sets the time-step in the **BIOGEM** tie-stepping cycle at which the insolation value is extracted. Care is needed here in distinguishing between the primary ocean (**GOLDSTEIN**) time-step, and **BIOGEM**, which is typically sub-stepped. For example, on an 18x18x8 grid, the **GOLDSTEIN** time-step is 48 (per year), and **BIOGEM** time-steps every other **GOLDSTEIN** step, i.e. 24 per year. A value for this parameter of 12 is then approximately June 21.

• **bg_par_t_sig_count_N** As above, except for the Southern hemisphere.

The **time-series** output files is called: **biogem_series_misc_ocn_insol.res** (and has pretty self-explanatory columns).
Saving global/annual (insolation) changes

Two new misc category time-series files have been provided:

biogem_series_misc_ocn_insol.res

and

biogem_series_misc_ocn_swflux.res

with the the SW (shortwave) flux (swflux) being equivalent to the incident strength of solar radiation at the surface (insol) but accounting for the prescribed planetary albedo. Both variables are calculated and saved on a global mean ocean grid basis (2nd data column) and have units of W m-2.

In addition, to help diagnose orbital variability, biogem_series_misc_ocn_insol.res includes two further insolation variables (3rd and 4th columns). These reflect the strength of insolation at a single point in the annual cycle and at discrete latitudes (i.e. j grid indices). (The insolation at 2 different latitudes are saved so that both .e.g. N and S hemisphere insolation signals can be simultaneously recorded.)

Three new namelist parameters are provided to configure this:

1. bg_par_t_sig_count – which sets the BIOGEM ‘time-step’ in the annual cycle at which the insolation value will be saved. e.g. for 96 time-steps in the ocean physics, and a 2:1 GOLDSTEIN:BIOGEM gearing (the default for the 16 level configuration), there are 48 BIOGEM time-steps. (It is left to the user to work out which part of the annual cycle cGENIE starts at (i.e. time-step #1) – I haven’t a clue …)

2. bg_par_sig_j_N – sets the ‘j’ value for a northern hemisphere (but could be southern) snapshot.

3. bg_par_sig_j_S – sets the ‘j’ value for a southern hemisphere snapshot.

By default bg_par_sig_j_N is assigned a value of 2 and bg_par_sig_j_S a value of 1 (on account of the maximum grid size not being a priori known). The default setting of bg_par_t_sig_count is 1 – i.e. the first (BIOGEM) time-step in the annual cycle.
Testing
Common Modifications
Uncommon (Ill-advised) Modifications
Other Modifications!

22. Model Code

Best not to touch it, but...
22.1 Testing
## 22.2 Common Modifications

### Add 'name-list' (run time) parameters

In order to create a new 'namelist' parameter, i.e. a parameter whose value can be set in a user-config file, you need to edit a total of 4 files:

1. \*._lib.f90
   - Add an entry in the relevant module library file, which for BIOGEM would be:
     - biogem_lib.f90
   - The parameter must be defined (with an appropriate type) and added to the NAMELIST section at the top of the tile. Simply follow the format of the existing entries and add to the most appropriate section of parameter categories. Note that the parameter name appears *twice* — one in defining its type, and once in adding the the parameter NAMELIST.

2. \*._data.f90
   - For completeness, there is an entry in the subroutine that reports the selected parameter options upon model start-up (if this reporting is selected in the first place ...), which for BIOGEM, is subroutine:
     - sub_load_goin_biogem
   - that lives in:
     - biogem_data.f90.
   - Again – simply follow the format of existing entries for creating a new one. Again: add in an appropriate section of parameter categories to prevent future coders going mad looking for something.

3. Add a new definition (including brief description and default value) in the xml definition file:
   - definition.xml
   - that can be found in the directory:
     - cgenie.muffin/genie-main/src/xml-config/xml You will have to scroll down to find the section for the appropriate module, and then within that, the section for that category of parameter. Simply follow the format of the existing entries.

4. \<FILE\>.f90 (or \<FILE\>.f77)
   - Finally, edit into the appropriate FORTRAN source file the code that incorporates the parameter that you wish to use. D’uh!

Having ensured the model compiles and seems to do what you wish it do to – run a standard test to confirm that nothing obvious has been unintentionally affected by the change. To do this, simply type

```
make testbiogem
```

and confirm that the test passes ...

### Add additional results output

If a new data field can be derived from an existing field, then creating additional results saving is relatively straightforward because no new time-averaging has to be carried out (i.e. you create the new field based on annual (or sub-annual) averages that are already calculated and available. (If not – refer to the ’full’ data saving sub-sections.) Note that new tracers are automatically saved.

### Data saving 101

For new time-slice saving, code needs to be added to either:

```
sub_save_netCDF_2d_USER
```

or:

```
cgenie.muffin/genie-biogem/src/fortran biogem_data.netCF.f90
```
sub_save_netCDF_3d_USER

depending on whether the field is 2D or 3D, respectively. Add the code to the end of the subroutine (as marked). Follow the format of previous data saving as far as possible. For general format is:
- Add a conditional to define under what circumstances, particularly selected save options, the data is saved. DO NOT save it by default ... unless it is of vital importance to the future of the planet. Refer to the user-manual for the categories of save options, investigate what options are used for similar data fields, and use your common sense ...
- Specific a units name for loc_unitsname, or set to ’n/a’ if not applicable (or non-dimensional).
- Initialize (zero) the local 2D (loc_ij) or 3D (loc_ijk) (depending on the data field) data array.
- Calculate the data, employing a nested loop if necessary (i.e. simply matrix math cannot be employed) and assign to the local data array.
- Add a call to sub_adddef_netcdf (don’t ask questions – follow the general format).
- Add a call to sub_putvar2d (ditto).

For new time-series saving ... it is sort of both more and less complicated :o) The new code goes in biogem_data_ascii.f90 but in two different places: sub_init_data_save_runtime and sub_data_save_runtime. For former creates the (ASCII) file and adds a header (to define the columns), and then closes the file. The latter opens the now existing file, writes the output, including the (time-series save point) time, and the closes the file. The code needed (again – follow the generla format and best to add the new code to the end of the subroutines) is hence ...

In sub_init_data_save_runtime:
- Add a conditional to define under what circumstances, particularly selected save options, the data is saved. DO NOT save it by default ... unless it is of vital importance to the future of the planet. Refer to the user-manual for the categories of save options, investigate what options are used for similar data fields, and use your common sense ...
- (If multiple different variables stored in or based on the same array, set up a loop.)
- Create the filename to be sued (loc_filename) via a call to fun_data_timeseries_filename. All files are *_series with the specific variable type and variable after, or if not a specific variable type, then _misc.
- Create the header text. If you start with a ’%’ then it is all the more MUTLAB friendly.
- Follow the sequence format of: CHECK / OPEN / CHECK / WRITE / CHECK / CLOSE / CHECK (yawn) ...

In sub_data_save_runtime:
- Add the same conditional as used in sub_init_data_save_runtime.
- (If multiple different variables stored in or based on the same array, set up a loop.)
- Construct the local filename *exactly* as before (sub_data_save_runtime) or it will not find the fiel you have created ...
- Calculate the data value(s) (loc_sig).
- CHECK / OPEN / CHECK / WRITE / CHECK / CLOSE / CHECK (yawn) ...
22.3 Uncommon (Ill-advised) Modifications

Define a new tracer

You probably should not be doing this ... but ... just in case ...

Basic definition procedure

1. The starting point to adding a new tracer in cGENIE is to add the its definition to the relevant tracer definition file.

   There are three tracer definition files that live in cgenie.muffin/genie-main/data/input:
   - tracer_define.atm
   - tracer_define.ocn
   - tracer_define.sed

   that house a list of atmospheric (gaseous), oceanic (dissolved), and sediment (solid) tracer definitions. Each file has a similar format, with a series of columns holding information on:

   #01 The short (mnemonic) name for the tracer. This is used in creating the output filenames and netCDF variable names. In theory, this could be anything you like, but to a limit of 16 characters.
   #02 The identifier (index) of the tracer. This must be a unique number – one number for each tracer.
   #03 The ‘dependency’ of the tracer. For instance, an isotope depends on the bulk (and lower mass) tracer. A scavenged element depends on bulk organic matter. For a bulk tracer, its dependency is itself. The dependency is used in the code to automatically determine any tracer that it depends on and in the case of isotopes, to calculate the $\delta$ value.
   #04 Is the tracer variable ‘type’, used internally to determine what to ‘do’ with a specific tracer:
     - '1' → assigned to primary biogenic phases; POM (represented by POC), CaCO3, opal (all contributing to bulk composition)
     - '2' → assigned to abiotic material (contributing to bulk composition); det, ash
     - '3' → assigned to elemental components associated with POC; P, N, Cd, Fe
     - '4' → assigned to elemental components associated with CaCO3; Cd
     - '5' → assigned to elemental components associated with opal; Ge
     - '6' → assigned to elemental components associated with det; Li
     - '7' → assigned to particle-reactive scavenged elements; 231Pa, 230Th, Fe
     - '8' → assigned to carbonate ‘age’
     - '9' → assigned to the fractional partitioning of biogenic material (for remineralization purposes)
     - '10' → assigned to misc / 'inert'
     - '>10' → assigned to isotopic properties: 11==13C, 12==14C, 13==18O, 14==15N, 15==34S, 16==30Si, 17==114Cd, 18==7Li, 19==144Nd, 20==44Ca, 21==98Mo, 22==56Fe.
   #05 Tracer long name. Cannot be more than 128 characters in length.
   #06 Units.
   #07 Minimum valid value for the tracer. Values lower than this are classed as NaN in the

---

3 The meaning of the columns is also summarized at the end of the file.
4 The elemental components of organic matter (P, N, etc) count as bulk tracers for the purpose of dependency.
#08  Maximum valid value. Values higher than this are classed as NaN in the netCDF output. Hence, the first thing to do is to add an entry for the required tracer(s) to the relevant file(s), using the above information and keeping a consistent format and convention with the existing tracers.

If you have an elemental or isotopic tracer that is taken up by a growing phytoplankton cell and incorporated structurally into POM, you will also need to define equivalent dissolved (and recalcitrant) dissolved organic matter tracers (DOM and RDOM). If the tracer is associated with organic matter (or other particles), then you require a scavenged particulate tracer but no corresponding dissolved tracer. For something like iron, which is both incorporated into the cell, and scavenged, you need both types of particulate tracer plus a set of dissolved organic matter tracers ...

2. A set of tracer-related definitions also exists in:
cgenie.muffin/genie-main/src/fortran/cmngem/gem_cmn.f90
Requiring some editing-attention here is:
ca. L24-26  Three parameters define the main tracer array sizes. (Unfortunately this information has to be duplicated elsewhere due to the peculiarities of the GENIE code structure – see below.) Their values must be equal to the total number of tracers defined in the tracer definition files.
ca. L72-261  So as to simplify referencing the tracers in the code, each tracer is assigned a simple mnemonic. This mnemonic may or may not be the same as the short name defined in the tracer definition files. With a little bit of code, the tracer definition short name could have been turned into an index value, but to date this has not been implemented. However, while slightly tedious to set up, created fixed and compile-time rather than run-time mnemonic assignments is going to be somewhat faster as well as making the code slightly more compact.
ca. L630-642  Definition of the isotope standards. Only if creating a new isotope system does this need to be edited.
So effectively, just the total number of tracers, and the addition of the tracer mnemonic name, has to be edited in this file.

3. An extremely unfortunate fact of GENIE code structure life is that the total tracer numbers are defined a second time in:
cgenie.muffin/genie-main/genie_control.f90
at ca. L144-146, and need to be edited consistent with the values in gem_cmn.f90 (see above). Why ... ? Please don’t ask.

4. Equally, or arguably even more annoying and opaquely justified, is the need to edit a number of entries in:
cgenie.muffin/genie-main/src/xml-config/xml/definition.xml
A number of sets of runtime parameters exist with one entry per tracer. Given that all runtime parameters must be defined in the xml definition file, means that every time a new tracer is created, a number of xml entries must be created (The approximate line occurrence of this outrage against all good programming practice, are as follows:
ca. L397-1159  The tracer arrays holding information about which tracers are selected.
ca. L2538-3271  Tracer arrays for the initial value of ocean (dissolved) tracers, as well as a tracer value modification array.

5 Watch out that the line numbers may have changed somewhat ...
6 Note that there are no sediment (solid) tracer arrays for either initial composition or forcings.
22.3 Uncommon (ill-advised) Modifications

ca. L4343-5234 Tracer arrays for atmospheric (gaseous) and ocean (dissolved) tracers governing tracer forcings (value and tie-scale).

ca. L5292-5370 Tracer array for the initial value of atmospheric (gaseous) tracers.
To edit in new entries – simply follow the format of the last entry.
In addition – at the start of each array definition (xml tag starting <ParamArray), the array size is defined (extent=). This also needs to be edited to reflect the inclusion of additional tracers.

5. Even more annoying, if possible, is the need to then edit the array entries in the Python xml parameter translation script:
cgenie.muffin/genie-main/config2xml.py
It should be obvious where there additional entries corresponding to the new tracers need to be added. Just be careful of the formatting (the last entry in the list of tracers for each array not having a ’,’ terminating the line. Otherwise, simply follow the existing format.

That is it for the basic tracer definition. The model should now compile without error (and still pass its make testbiogem) although you’ll need to do a make cleanall first. However, whilst ‘knowing’ about the possibility of the new tracers, at this point you have not actually selected any for an experiment and no initial conditions will be read in, nor relevant output created.

Testing
Having checked the model compiles and passes the basic BIOGEM test, the next step is to check that the new tracer(s) is initialized correctly (atm and ocn tracers) and that appropriate output is generated.

1. Create a new base-config by taking an appropriate/comparable existing base-config\(^7\) and modifying it.
The first modification is to increase the number of ocean tracers, defined on the line headed by:
   # Set number of tracers
incrementing the total by the number of additional tracers to be included in the new base-config as compared to the original one.
The second modification is to select the additional tracers – simply add appropriate entries to the list of selected tracers (the tracers are selected by setting the parameter value to .true. as by default the are .false.).
The final modification, in the case of atmospheric (gaseous) and ocean (dissolved) tracers is to set an initial value. If you do not do this, by default, concentrations are initialized to zero.

2. Now go create a user-config file. Copy an EXAMPLE config file – one corresponding to the unmodified base-config, if possible. You should rename it, and although no modifications are required to the parameter settings in order for the model to run, to ensure all possible output is produced, set:
   bg_par_data_save_level=99

3. Run a brief experiment and check that the tracer appears in the output – both time-series and netCDF. For ocean tracers, the concentration field should progressively look like salinity as concentration changes at the surface will be influenced by P-E. (Remember that at this point, no other transformations or changes of tracer have been defined – just that there is a tracer and it is initialized to a certain value.)

\(^7\)/cgenie.muffin/genie-main/configs
Adding (and testing) a basic tracer biogeochemical cycle

Now for the trickier part, assuming you do not just want a simple passive tracer (there are plenty of ‘color’ tracers defined already!) and assuming you have got the tracer already successfully configured and running as a simple passive tracer (i.e. previous steps).

The example will be for an ocean tracer that is incorporated into particulate organic matter (POM) (and hence creating an associated sedimentary tracer) during biological productivity. Other and much more fun and entertaining complexities will apply if e.g. the ocean tracer exchanges with the atmosphere and hence is associated with an atmospheric tracer.

1. The relationships between different sorts of tracers, e.g. dissolved and gaseous, and dissolved and solid, are defined in subroutine sub_def_tracerrelationships.

   If there is an equivalent dissolved organic matter tracer corresponding to the particulate organic matter one, the relationship between POM and DOM (and also RDOM) also needs to be defined.

   Typically, the relationship between a particulate and dissolved inorganic, or particulate and dissolved organic will be 1.0, but depending on the species concerned, it may be 2.0 (or its reciprocal) and/or negative. These values can be modified later if necessary, and this occurs depending on the redox state of the ocean in:

   sub_data_update_tracerrelationships

2. For a dissolved inorganic species being taken up biologically, a ‘Redfield’ like ratio is defined and used to relate the cellular quotient of the tracer versus carbon. An array (bio_part_red) stores the relationship of every particulate tracer to every other particulate tracer. It is hence mostly zeros, except for the ratios of the particulate tracers to carbon in both organic matter and CaCO$_3$ (and for that matter, opal) and of the isotope ratios of specifics to their bulk equivalent. The array is (re)populated each time-step in sub_calc_bio_uptake, typically by directly applying a globally applicable ratio that is read in at run-time (and hence requiring a new namelist parameter to be defined), by some function of ambient environmental conditions that is often a modification of the run-time parameter.

   If steps #1 and #2 have been completed correctly, there should now be a biological cycle of the new tracer, with it being taken up at the ocean surface and incorporated into POM with a specific ratio compared to carbon (set by the new namelist parameter) and resulting in depletion of the inorganic dissolved tracer at the ocean surface. Conversely, there should be elevated concentrations of the inorganic tracer at depth, mirroring the pattern of e.g. [PO$_4$]. Creation and subsequent remineralization of a corresponding tracer incorporated into DOM (and RDOM if selected) should occur automatically. However, the recommended first step in testing the newly defined biogeochemical cycle is to disable all DOM formation, by setting:

   bg_par_bio_red_DOMfrac=0.0

   Also recommended is to enable ‘auditing’ of all the tracer inventories to ensure that tracers are not being spuriously created or destroyed by:

   bg_ctrl_audit=.true.

---

8 cgenie.muffin/genie-main/src/fortran/cmngem/gem_util.f90
9 cgenie.muffin/genie-biogem/src/fortran/biogem_data.f90
10 A carbon currency is used in the model rather than phosphate, despite the classic Redfield ratio being defined relative to a phosphate quotient of 1.0
11 cgenie.muffin/genie-biogem/src/fortran/biogem_box.f90
Scavenged tracers are automatically remineralized along with the corresponding parent particulate tracer by default. However, there is no scavenging or creation of scavenged tracers by default. A call would need to be added to:

```
sub_box_remin_part
```

(see L2961), e.g., following the examples of Fe scavenging and H2S reactions with organic matter (treated as a form of ‘scavenging’ for simplicity of code structure), plus a corresponding subroutine added in which is the scavenged particulate tracer concentration is calculated and the removal of the corresponding dissolved tracers set.

---

12 They can instead be set to remain in the scavenging medium by setting a non zero value of `par_scav_fremin`, which sets the fraction of the scavenged tracer that is remineralized along with the degraded parent particulate tracer.

13 `egenie.muffin/genie-biogem/src/f90/biogem_box.f90`
22.4 Other Modifications!

Add a new functional type to ECOGEM

1. First, define the new PFT in:
   `~/cgenie.muffin/genie-ecogem/src/fortran/ecogem_data.f90`
   In the subroutine `sub_init_plankton`, add a set of definitions for a new PFT consistent with the formatting – the PFT has a name and various true/false statements as to what it does, e.g.

   ```
   if (pft(jp).eq.'new_PFT') then
       NO3up(jp) = 0.0
       Nfix(jp) = 0.0
       calcify(jp) = 0.0
       silicify(jp) = 0.0
       autotrophy(jp) = 1.0
       heterotrophy(jp) = 0.0
       palatability(jp) = 1.0
   end if
   ```

2. Define parameters relevant to the PFT such as size-scaling parameters (a,b,c) for vmax. (See existing manual section for adding parameters.)

3. Use the new parameters to update the PFT parameter definitions in `~/cgenie.muffin/genie-ecogem/src/fortran/ecogem_data.f90`, using the PFT name defined in step 1:
   ```
   ! generic maximum photosynthetic rate
   vmax(iDIC,:) = (vmaxDIC\_a + log10(volume(:))) / (vmaxDIC\_b + vmaxDIC\_c * log10(volume(:)) + log10(volume(:)**2) * autotrophy(:))
   
   ! modify rates for new functional types
   vmax(iDIC,:) = merge(vmaxDIC\_a\_pft\_newPFT * volume(:)**2, vmax(iDIC,:), pft.eq.'new\_PFT')
   ```

4. Set up a plankton definition file in `~/cgenie.muffin/genie-ecogem/data/input/new_PFT_setup.eco` that includes the name of the new PFT defined in step 1 and the cell size.

Note that to date, only the growth rate parameters (Vmax for DIC) have been changed, and this approach has worked. No changes in other parameters, specifically for PFTs such as quotas or other nutrient uptake, have been tested but I guess it would be similar. Overall, this is a little bit clumsy in that it requires new at least 2-3 new parameter definitions for each PFT.
23. Examples
Chapter 23. Examples

23.1 xxx
## 24.1 Data save levels

Table 24.1: Categories of saved data by save level (#) – *time-slices*

<table>
<thead>
<tr>
<th>Data category</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>ocn</td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>ocnatm</td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>ocnsed</td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>misc</td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>sur</td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>fairsea</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>focnatm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>focnsed</td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>fsedocn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>carb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>carbconst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phys_atm</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>phys_ocn</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>diag_bio</td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>diag_geochem</td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>diag_proxy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>diag_tracer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>derived</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>redox</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GLOBAL</td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

---

- == not saved
- × == always saved for that saved level
- ⊗ == generically saved if SEDGEM is selected
Table 24.2: Description of data categories – *time-slices*

<table>
<thead>
<tr>
<th>Data category</th>
<th>Description of associated data saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>ocn</td>
<td>–</td>
</tr>
<tr>
<td>ocnatm</td>
<td>–</td>
</tr>
<tr>
<td>ocnosed</td>
<td>–</td>
</tr>
<tr>
<td>fairsea</td>
<td>–</td>
</tr>
<tr>
<td>focnatm</td>
<td>–</td>
</tr>
<tr>
<td>focnsed</td>
<td>–</td>
</tr>
<tr>
<td>fsedocn</td>
<td>–</td>
</tr>
<tr>
<td>bio</td>
<td>–</td>
</tr>
<tr>
<td>carb</td>
<td>–</td>
</tr>
<tr>
<td>carbconst</td>
<td>–</td>
</tr>
<tr>
<td>phys_atm</td>
<td>–</td>
</tr>
<tr>
<td>phys_ocn</td>
<td>–</td>
</tr>
<tr>
<td>misc</td>
<td>–</td>
</tr>
<tr>
<td>diag_bio</td>
<td>–</td>
</tr>
<tr>
<td>diag_geochem</td>
<td>–</td>
</tr>
<tr>
<td>diag_proxy</td>
<td>–</td>
</tr>
<tr>
<td>diag_tracer</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 24.3: Categories of saved data by save level (#) – *time-series*

| Data category | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 14 | 15 | 16 | 99 |
|---------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| ocn           | – | – | × | × | × | × | × | × | × | × | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| ocnatm        | – | – | × | × | × | × | × | × | × | × | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| ocnosed       | – | – | ⊗ | ⊗ | ⊗ | ⊗ | ⊗ | ⊗ | ⊗ | ⊗ | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  |
| fairsea       | – | – | – | × | × | – | – | – | – | – | –  | –  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| focnatm       | – | – | – | × | × | – | – | – | – | – | –  | –  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| fexport       | – | – | × | – | – | – | × | × | – | × | –  | –  | –  | –  | –  | –  | –  | –  | –  | –  |
| focnsed       | – | – | ⊗ | ⊗ | ⊗ | ⊗ | × | × | × | × | ×  | ×  | ⊗  | ⊗  | ⊗  | ⊗  | ×  | ×  | ×  | ×  |
| fsedocn       | – | – | ⊗ | ⊗ | ⊗ | ⊗ | ⊗ | ⊗ | ⊗ | ⊗ | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  | ⊗  |
| ocn_sur       | – | – | × | × | × | × | × | × | × | × | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| carb_sur      | – | – | – | × | × | × | × | × | × | × | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| misc          | – | – | × | × | × | × | × | × | × | × | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| diag          | – | – | – | × | × | × | × | × | × | × | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| diag_bio      | – | – | × | – | – | × | × | × | × | × | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| diag_geochem  | – | – | – | × | × | – | × | × | – | × | –  | –  | ×  | ×  | ×  | ×  | ×  | ×  | ×  | ×  |
| diag_redox_old| – | – | – | – | – | – | – | – | – | – | –  | –  | –  | –  | –  | –  | –  | –  | –  | –  |