

A Hitchhikers Guide to the Black Arts (of Earth system modelling)

Lesson zero: modeling basics & experimental methodology

Stuff to keep in mind:

- cGENIE is a model. Models ARE NOT the 'real World'. (Don't get confused!)
- The very low resolution (for a 3-D ocean circulation model) of cGENIE limits its applicability for very short time-scale problems because the simplified atmospheric component does not simulate winds and there is no atmospheric dynamics or inter-annual variability.
- cGENIE is best thought of as a 'discovery and exploring' tool for learning how the Earth system (might) work rather than as a detailed 'simulation' tool.
- Have fun (or at least try) ☺

Some quasi-random and somewhat relevant reading:

Underlying climate model description/evaluation papers:

Edwards and Marsh [2005] (Climate Dynamics 24, 415-433)

→ description and calibration of the climate model component of (c)GENIE

Hargreaves *et al.* [2004] (Climate Dynamics 23, 745-760)

→ description of data assimilation methodology and calibrated climatology of the climate model

Ocean (and sediment) biogeochemical cycles model description/evaluation papers:

Ridgwell *et al.* [2007a] (Biogeosciences 4, 87-104)

→ description of basic ocean carbon cycle and its calibration against observations

Ridgwell and Hargreaves [2007] (Global Biogeochemical Cycles 21, doi:10.1029/2006GB002764)

→ description and calibration of the sediment model component and response to fossil CO₂ release

Example model applications – future/geoengineering:

Winkelmann *et al.* [2015] (Science Advances, DOI: 10.1126/sciadv.1500589)

→ long-term future projections of atmospheric pCO₂ and fate of the Antarctic ice sheet

Taylor *et al.* [2015] (Nature Climate Change, doi:10.1038/nclimate2882)

→ impacts on atmospheric pCO₂ and ocean acidification of enhanced weathering on land

Example model applications – paleo:

Chikamoto *et al.* [2008] (JGR 113, doi:10.1029/2007JG000666)

→ deep-sea CaCO₃ sediment and atmospheric CO₂ response to AMOC shutdown

Panchuk *et al.* [2008] (Geology 36, 315-318)

→ configuration of (c)GENIE for Palaeogene marine carbon cycling; assessing PETM CO₂ release

Hönisch *et al.* [2012] (Science 335, 1058-1063)

→ characterization of future vs. geologic time-scales, and nature, of ocean acidification

Gibbs *et al.* [2015] (Geology 44, 59-62)

→ assessment of marine environmental change across the PETM and relationship to biotic response

This document, plus the details of subsequent workshop exercises/activities can be downloaded from:

<http://www.seao2.info/mycgenie.html>

under 'cGENIE resources: TEACHING LABS [CURRENT]'

Andy (andy@seao2.org)

0. Before anything else ...

- 0.0 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 on to **Section 0.1** and save yourself an entire 15 seconds of reading words.*)

cGENIE has traditionally been configured and accessed (aka 'run') at the 'command line' of the linux (or Mac equivalent, which is sort of linux also) 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 username, computer name, and current directory, e.g.:

```
[username@sprout ~]$
```

which in this example is user 'username' (yours will be different!) on computer 'sprout' (Sprout will eventually appear under 'cat-of-the-day' on my homepage if you press F5 enough times – all the cGENIE model computing clusters are named after my cats ...) and the current directory is the 'home' directory (~).

If you are not or not very familiar with the linux/unix command line – Appendix 1 contains a quick summary of some of the more important/useful Commands you can use at the command line. Appendix 2 is a version of Appendix 1 for anyone especially nervous about linux.

NOTE: In the class instructions, example lines to be typed at the command line **are highlighted in yellow**. Be VERY CAREFUL that spaces are not missed out. 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: Text highlighted in grey represents instructions specific to using Windoz PCs, and hence may or will differ on different operating systems, e.g. if you have a Macbook. Other software choices for editing files or viewing results are possible as are other ways of configuring software and file editing/transferring methodologies.

Do what suits you best – you can view the instructions as mostly represent an example methodology rather than a strict interpretation of European Law.

- 0.1 Some warnings and reminders 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! ;)
- 0.2 Additional documentation (of varying degrees of up-to-date-ness) lives on the interweb (<http://www.seao2.info/mycgenie.html>):
- (i) The cGENIE *User Manual* (the least up-to-date of all!) [cGENIE.muffin.User_manual.pdf]
 - (ii) A set of *Examples* [cGENIE.muffin.Examples.pdf]
 - (iii) A *HOW-TO* (explanations of how to get useful stuff done) [cGENIE.muffin.HOWTO.pdf]
 - (iv) A table of model parameters [cGENIE.muffin.Parameter_reference_guide.pdf]
 - (v) A Guide to installing cGENIE [cGENIE.muffin.QuickStartGuide.pdf]
 - (vi) A README that will inevitably remain un-read ... [cGENIE.muffin.README.pdf]
- 0.3 **Note that you will be using the development branch of the model and a version code-named 'muffin' (if Apple can have 'Leopard', 'Lion' etc., I can haz baked goods, right?). 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'. Also note that the instructions in general may not be bug-free – use your judgment.**
- 0.4 **OK – now we are ready to start ...**

1. Starting (dozing?) off ...

1.0 You are going to be installing the model from scratch – why? Why not? Hell, it saves me installing it a dozen times. Actually, it will be a happy character-building experience for you – trust me.

1.1 Log in to the account that has been created for you on the `sprout.ggy.bris.ac.uk` computing cluster. To do this – first start the WinSCP program. Under 'Host Name', enter:

```
sprout.ggy.bris.ac.uk
```

The 'Port number' should be set to 22. 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 ...).

1.2 The next step is to download a copy of the source code for the cGENIE model as follows (all this is done in the shell window of your `sprout` computer account and it installs the computer code in your `sprout` account rather than downloading it to your local computer/terminal):

(i) To get a copy of cGENIE: from your home directory type:

```
$ svn co  
https://svn.ggy.bris.ac.uk/subversion/genie/branches/cgenie.muffin  
--username=genie-user cgenie.muffin
```

All this must be on a continuous line, with a S P A C E before '--username', and before `cgenie.muffin`.

(If you get asked about a 'certificate' – enter 'p' to accept 'permanently'.)

(Note ... and you'll ignore this reminder ... the user-name (and password) for the code server is **not** the same as the user-name and password for the cluster account ... different computers ...)

Depending on the weather and phase of the moon, you may (or may not!) be asked for a password. If so – it is: **g3n1e-user**. Again – don't mix up the ONE ('1') with an 'el' ('l'). (The mystery character in question is a 'one' if you must know ...) Watch out that under linux, the characters of the password are not displayed, nor replaced by a '*'. (It is OK then to 'store the password unencrypted' if asked.)

Some of the files that are installed are quite large, and at times, it may appear like it is 'stuck'. Be patient! Installation should be complete within about 2-3 minutes (longer if everyone else is trying to do this at exactly the same time).

(ii) Change directory (see: Appendix I and also Figure 1.1) to `cgenie.muffin/genie-main` – if you are not 'linux-friendly' – maybe at first do this in steps – list the contents of the directory (`ls`) to check where you are (i.e. what directories are available to chance to), then change to `cgenie.muffin` (`cd cgenie.muffin`), then list again (`ls`) (and see what further directories are there), then change to `genie-main` (`cd genie-main`), and only then ... 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.

There may be some 'Warnings' reported (== someone's sloppy programming) but these are not detrimental to the ultimate science results (we hope!). 'Success' of this test is indicated by:

```
**TEST OK**
```

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 cGENIE. Note that the model will pause for a l 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!
```

Ignore the compiler warnings ... (reflecting my lack of adequate software engineering skills).
(iii) That is it as far basic installation goes.

1.3 Later on you will be editing some configuration files. 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 ... so I will not *make* you ;)

OR ... Use a suitable linux-friendly text editor (NOT Micro\$oft 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.

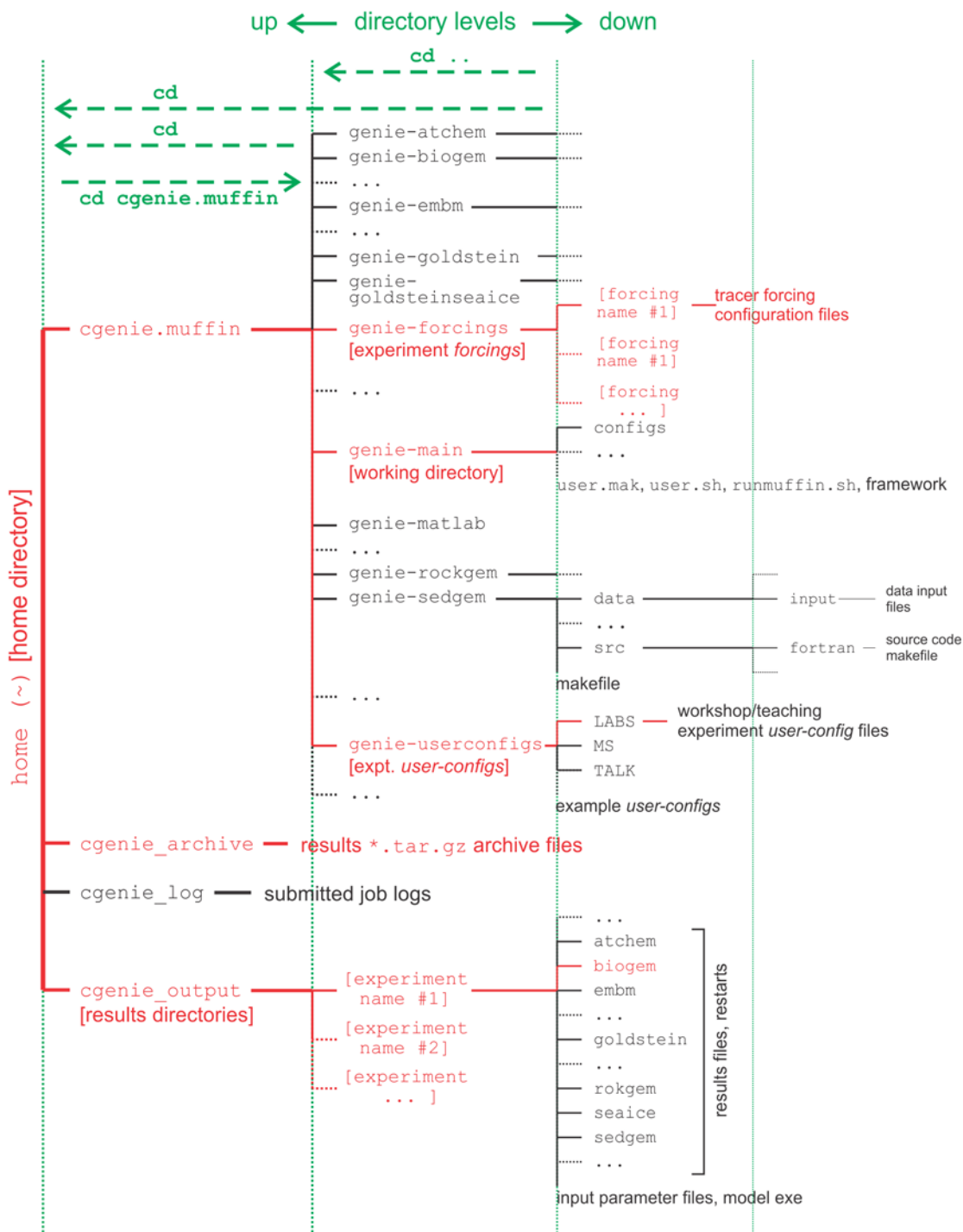


Figure 1.1 cGENIE model directory structure

Directory structure of the cGENIE 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.

2. Running the model ('interactively', in a shell window)

- 2.1 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)
```

In which 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.
- There is also one optional (5th) parameter (described later).

2.2 As an example of running the cGENIE Earth system model:

parameter #1: The base config is: `cgenie.eb_go_gs_ac_bg.worbe2.BASE`

parameter #2: The user config directory is: `LABS`

parameter #3: The user config file (the experiment name) is: `LAB_0.EXAMPLE`

parameter #4: Run the experiment for ten years: `10`

parameter #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
```

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 ('l') with a one ('1') when typing this in ... (it is a 'one' here).

2.3 What should happen is: First, you will end up twiddling your thumbs a while, as all the components of cGENIE 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)
d13CO2     – mean δ13C value of atmospheric CO2 (‰) (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)
```

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)?'

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

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

      model year   * pCO2(uatm)   dl3CO2   * AMO(Sv)   ice(%)   <SST>   <SSS>   * <DIC>(uM)   <ALK>(uM)

$N$      0.00      278.000   -6.500      0.000   0.000   -0.000   34.900      2244.000   2363.000
>>> SAVING BIOGEM TIME-SLICE AVERAGE CENTERED @ year :      0.500
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      0.500
$N$      1.00      279.960   -6.598      13.613   0.744    2.509   34.901      2241.498   2363.111
>>> SAVING BIOGEM TIME-SLICE AVERAGE CENTERED @ year :      1.500
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      1.500
$N$      2.00      279.525   -6.580      12.828   3.499    4.471   34.901      2240.173   2363.135
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      2.500
$N$      3.00      279.258   -6.568      11.695   5.028    5.996   34.901      2239.169   2363.161
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      3.500
$N$      4.00      279.044   -6.558      10.444   5.929    7.209   34.901      2238.354   2363.191
>>> SAVING BIOGEM TIME-SLICE AVERAGE CENTERED @ year :      4.500
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      4.500
$N$      5.00      278.899   -6.551       9.380   6.191    8.156   34.902      2237.664   2363.220
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      5.500
$N$      6.00      278.777   -6.545       8.500   6.623    8.975   34.902      2237.069   2363.246
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      6.500
$N$      7.00      278.680   -6.541       7.922   6.629    9.637   34.903      2236.548   2363.267
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      7.500
$N$      8.00      278.601   -6.537       7.917   6.738   10.225   34.903      2236.087   2363.285
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      8.500
$N$      9.00      278.528   -6.534       7.952   6.740   10.732   34.904      2235.682   2363.301
>>> SAVING BIOGEM TIME-SLICE AVERAGE CENTERED @ year :      9.500
>>> SAVING BIOGEM TIME-SERIES AVERAGE CENTERED @ year :      9.500
$N$     10.00      278.466   -6.531       8.025   6.694   11.176   34.904      2235.325   2363.314

*****
*** Simulation complete: shutdown starting ...
*****
```


3. Model output

- 3.1 The first thing to note about output (i.e., saved results files) from cGENIE 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, ATCHEM does not save its own results (BIOGEM can save information about atmospheric composition and air-sea gas exchange) while SEDGEM essentially saves results 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, in order 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

Time-slices

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 36×36×8 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 then assumes that the specified time point represents the mid-point of the annual (by default) average with the results that output years end up being reported as e.g.

```
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.).

Time-series

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).

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

- (i) `_restart.nc` (is the re-start file created from the run you have just complete, and can be ignored).
- (ii) `biogem_series_*.res` – these are the time-series files (in ASCII / plain text format).
- (iii) `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).
- (iv) `fields_biogem_2d.nc` – 2-D fields of ocean and atmosphere properties, as NetCDF.
- (v) `fields_biogem_3d.nc` – 3-D fields of ocean properties, as NetCDF.

4. Viewing time-series output

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

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

- 4.2 One way of viewing the contents of files is the change directory to the experiment results directory and opening the file in the `vi` editor. 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 2 columns – time and mean (whole) ocean temperature (°C). (However, in subsequent exercises a fuller output will be created with additional columns, with one for mean surface ocean temperature ‘SST’ (°C) as well as mean benthic (bottom water) temperature (°C)). Other results files may differ in the numbers of columns but all should be identifiable from the header information.

Note: The Secure File Transfer Client **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 Secure File Transfer Client needs to be refreshed – there is an icon at the top of the program window or hit the ‘F5’ key.

- 4.3 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: Chose 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.

5. Viewing time-slice output of 2-D and 3-D environmental property fields

- 5.0 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/>.
- 5.1 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 of 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).

For all three: there may be multiple time-slices (i.e., you can plot data saved from different years).

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. Grey cells represent 'dry' grid points, i.e., continental or oceanic crust.

NOTE: The default settings in Panoply can mislead:

(1) By displaying the very 1st time-slice (often year mid-point 0.5) time-slice rather than the experiment end. (This can confuse as it can look like an experiment has not done anything!)

(2) By interpolating the data (not always misleading). To remove interpolation, untick '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 is to 'Show' is the shorter and less confusing list of 'Georeferenced variables'.

(5) In Longitude-Latitude plots, by overlaying the modern continental output. (GENIE 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'.

- 5.2 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).
- 5.3 To save plots in Panoply:
- File
- Save Image As ...
- Then select the location, filename, and graphics format.

6. Submitting experiment 'jobs' to a cluster

- 6.0 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 pub vs. sat bored watching a computer screen ☺
- 6.1 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 ...
- 6.2 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 :)

Unless you want to read the cGENIE *User Manual*, for now, take the following suggested way of submitting jobs on trust: to run the same experiment to 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) by submitting the experiment as a job to the cluster queue, type:

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

(Again: SINGLE, CONTINUOUS LINE.)

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 6 compute nodes, numbered 0-0 to 0-5), although it might briefly reside as a 'PENDING JOB'. For each node, there are 24 processing cores, meaning that up to 24 instances of cGENIE can run simultaneously on each node (i.e., a total of 144 across the entire cluster). For an 8-level ocean based configuration of cGENIE, 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 ...). If you periodically re-issue a `qstat -f` command you can follow your job's progress.

A rough rule of thumb is that 8-level ocean cGENIE @ a horizontal grid resolution of 36×36 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.

In the special case of this low resolution model with no carbon cycle, it is more like 10000 years per CPU hour.

NOTE: If your experiment appears on the queue but vanishes after a few seconds, it has most likely died ☹

7. 'Restarts'

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

7.1 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 CO₂ 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 CO₂ emissions scenarios all starting from current year 2012 conditions). A facility is thus provided for requesting that a 'restart' 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 (*restart*) 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 *restart* – it is simply an experiment that you have already run. Equally, there is nothing special about the *restarts* you will download next – these you could have generated yourself – it simply saves time to have them provided.

7.1 To experiment with using a *restart*, 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 <RETURN>`), and then changing to `cgenie_output` – refer to linux commands crib sheet and Figure 1.1), and type:

```
$ wget http://www.seao2.info/cgenie/labs/AWI.2017/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:

```
$ tar xfvz 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 `cgenie.muffin/genie-main`).

7.3 A *restart* can be requested in running an experiment by setting the 5th and last (optional) parameter when entering in the `runmuffin.sh` command. A spin-up of the climate state is provided: `LAB_0.SPIN` which you have just unpacked to the `cgenie_output` results output directory.

Create a new (*user config*) experiment configuration file by using the given file `LAB_0.EXAMPLE` as a template (no parameter changes need to be made yet). You can make copies (`cp` command) of the experiment configuration files (e.g. `LAB_0.EXAMPLE`) and assign them different names, e.g., `twilight`, `rarity`, `applejack` ... (ideally, use a name that helps you remember what the experiment did). Or ... if you and the linux command lines are simply not BFFs, you can drag the *user-config* file (`LAB_0.EXAMPLE`) you want to use as a template to your local PC (/Mac) directory in the file transfer window, rename it either in the file transfer program or e.g. via the Windoz file-manager (e.g. to `LAB_0.NEW`), and drag it back to the cluster directory again.

You specify the use of the *restart* state by adding the *restart* experiment name as the 5th parameter, e.g.:

```
$ ./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 ready an equilibrium, steady-state.

Appendix 1 – linux 101

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/LAB_0.EXAMPLE, 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`. Then move on, e.g. `cd LAB_0.EXAMPLE`. Note that to go straight from your home directory in one jump, type: `cd cgenie_output/LAB_0.EXAMPLE`. If you start from somewhere other than your home directory, give the full path: `cd $HOME/cgenie_output/LAB_0.EXAMPLE`.

You can always return to your home directory (~) by typing:

```
$ cd
```

or:

```
$ 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).



Appendix 2 – linux 101 – non-intimidating version



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
```

or:

```
$ 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

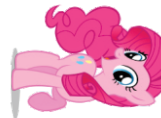
To create a directory `mydirectory`:

```
$ 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).



Appendix 3 – Using the vim (/vi) text editor

To start Vim, enter this command:

```
$ vim file.txt
```

If the file (`file.txt` in this example) is a new file, you get a blank window and the screen will look like:

```
█
~
~
~
~
"file.txt" [New file]
```

where █ is the cursor position.

The tilde (~) lines indicate lines not in the file. In other words, when Vim runs out of file to display, it displays tilde lines. At the bottom of the screen, a message line indicates the file is named `file.txt` and shows that you are creating a new file. The message information is temporary and other information overwrites it.

The Vim editor is a modal editor. That means that the editor behaves differently depending on which mode you are in. The two basic modes are called **Normal** mode and **Insert** mode. In **Normal** mode the characters you type are commands.

In **Insert** mode the characters are inserted as text. When you first start Vim it will be in Normal mode. To start Insert mode you type the "i" command (i for Insert). Then you can enter the text which will be inserted into the file. At the bottom of the window, -- INSERT -- will be displayed. This indicates you are in Insert mode.

Press the <Esc> key to stop Insert mode and go back to Normal mode (-- INSERT -- will disappear from the bottom of the screen). It can get confusing which mode you are in, or accidentally typing a command that switches modes. To get back to Normal mode, no matter what mode you are in, press the <Esc> key. Sometimes you have to press it twice. If Vim beeps back at you, you already are in Normal mode.

Moving around in the file

In both **Normal** and **Insert** modes, you can move around the file by using the cursor (arrow) keys.

In **Normal** mode, the <Ctrl-U> command scrolls up half a screen of text. (Up in this case is backward in the file; the text moves down on the screen.) The <Ctrl-D> command scrolls you down half a screen.

Editing

In **Insert** mode, press the <Enter> key to start a new line. In **Normal** mode, the "o" command creates a new, empty line below the cursor and puts Vim in Insert mode. Then you can type the text for the new line.

In **Insert** mode, the <Backspace> and <Delete> keys can be used to delete characters. In **Normal** mode: to delete a character, move the cursor over it and type "x", to delete a whole line use the "dd" command.

Saving

In **Normal** mode, use the "ZZ" (in CAPITALS!) command to exit. This command writes the file and exits.

If you want to discard your change and not save the file – Vim has a quit-and-throw-things-away command – in **Normal** mode, type:

```
:q!
```

SUMMARY

'i' enters **Insert** mode. Hit <Escape> to leave **Insert** mode and enter **Normal** mode.

From **Normal** mode, you can save-and-exit with 'ZZ', and exit-without-saving-changes by typing ':q!'.