HOW-TO for cGENIE: ’muffin’ pre-release version

Andy Ridgwell

July 15, 2017

Contents

1 Introduction

This document provides HOW-TO's for cGENIE 'muffin' users.

2 HOW-TOs: Getting started

2.1 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 ...

2.2 Install cGENIE

See: cGENIE *Quick-start Guide.* (Also refer to the READ-ME file for e.g., details of changes in configuring and running cGENIE compared to GENIE.)

2.3 Find configurations for cGENIE

A series of (example) cGENIE configurations are provided, many of which are detailed in full in the cGENIE *Tutorial* document. Example configurations comprise *base-config* and *user-config* files, plus any *forcings* needed.

- cGENIE *base-configs* are stored in:
  `/cgenie/genie-main/configs`
  and all start with 'cgenie_', for example:
  `cgenie_eb_go_gs_ac_bg_hadcm3l_eocene_36x36x16_2i_080928_BASE.config`
- cGENIE user configs are stored in:
  `/cgenie/genie-userconfigs`
- cGENIE forcings are stored in:
  `/cgenie/genie-forcings`

3 HOW-TOs: General

4 HOW-TOs: Model output

4.1 Set the frequency of time-series and time-slice output

See: *c*GENIE *user-manual* (section 5).

4.2 Diagnose orbital (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 *c*GENIE 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) snap-shot.
3. `bg_par_sig_j_S` – sets the 'j' value for a southern hemisphere snap-shot.

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.

5 HOW-TOs: Climate

5.1 Set/un-set seasonal insolation forcing

Seasonal insolation forcing of the EMBM, GOLDSTEIN ocean, and sea-ice model, are set by the following parameters[1]:

```
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.`.

5.2 Adjust solar forcing in a time-dependent manner

The value of the solar constant in cGENIE is set by the *namelist parameter*:
`ma_genie_solar_constant` and by default is set to 1368 W m^{-2}, i.e.:

```
ma_genie_solar_constant=1368.0
```

Specifying a different value for `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:

```
ma_genie_solar_constant=1330.56
```

Modification of `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*:
`bg_ctrl_force_solconst`.
By default this is set to `.false.`. By adding to the user config file:

```
bg_ctrl_force_solconst=.true.
```

a time-varying change in the value of the solar constant will be imposed. For this, BIOGEM will expect the presence of a file called `biogem_force_solconst_sig.dat` in the forcing directory[2]. This 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[3]. GENIE 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 `radfor(genie_solar_constant)`[4] at the start of the time-stepping loop[5]. At each time-marker, BIOGEM sets the value of `genie_solar_constant` equal to the corresponding value specified in `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:

```
-START-OF-DATA-
 0.5   1367.0
 1.5   1366.0
```

[1] These are typically set in the *base-config* if needed (i.e. different from default).

[2] REMEMBER: The location of which is specified by the namelist parameter bg_par_fordir_name.

[3] REMEMBER: The filename of which is specified by the namelist parameter bg_par_infile_sig_name.

[4] `cgenie.muffin/genie-embm/src/fortran/radfor.F`

[5] `cgenie.muffin/genie-main/genie.F`

```
 2.5   1365.0
 3.5   1364.0
 4.5   1363.0
 5.5   1362.0
 6.5   1361.0
 7.5   1360.0
 8.5   1359.0
 9.5   1358.0
10.5   1357.0
-END-OF-DATA-
```

which will decrease the value of the solar constant by 1 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 W m^{-2} , the default. Although BIOGEM sets a new value of `genie_solar_constant` (1367 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.

6 HOW-TOs: Ocean biology and biogeochemical cycling

6.1 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_PO4MM"` 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 (or edit the existing line in the section '`--- BIOLOGICAL NEW PRODUCTION ---`'):

```
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.

6.2 Add a water mass age tracer

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.worjh2.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 progressive and time runs towards zero, so does the dye flux. i.e. for 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.

In the original, and still valid way of doing this, 2 surface ocean forcings need be specified in order to create the combined age tracer. An example forcing is provided: *forcing*: `pyyyyz.Rcolr_Rcolb`, which is yhen 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.)
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.

This is all quite a faff[6]. So a simpler methodology has been implemented. As *base-config* file with both red and blue color tracers defined is still required, but now, rather than 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).

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 *restart* (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).

6.3 Specify the CaCO3:POC export ratio

In the default[7] '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 $CaCO_3$: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 $CaCO_3$: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 $CaCO_3$: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 $CaCO_3$: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 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'[8], 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.

[6] Google it.

[7] The default biological scheme is given by: `bg_par_bio_prodopt='1N1T_PO4MM'`.

[8] `$RUNTIME_ROOT` being equal to `~/genie`.

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`[9]. Examples are given in the code repository[10].

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[11].

If you want to hold a previously-calculated (spatially variable) $CaCO_3$: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`[12]. Because this is a 3D data field ($36 \times 36 \times 8$), carefully highlight just the surface ocean (2D) distribution (e.g., from the Panoply viewer) or extract from the netCDF file by some other means, and then copy and paste into:
`CaCO3toPOCrainratio.dat` (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.

6.4 Implementing 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 remineralziation 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 * 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 6.5 and will act on the 'free' POC phase (i.e. the one not controlled by the ballasting parameterization).

6.5 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
```

[9] -260E by default

[10] e.g., `~/genie/genie-biogem/data/input/CaCO3toPOCrainratio_worbe2_preindustrial.dat`

[11] It doesn't matter if you specify a value over land because only values associated with wet cells will be acted on.

[12] You 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.

(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 (6.3). 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 (6.3)).

Note that ballasting is combined with an e-folding (or other) fixed profile remineralization schemes[13]. 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 6.4 for more and alternatives). The fraction of initial export assigned to ballasting vs. 'free' POC is calculated according to the available exported ballast flux.

6.6 Prescribe biological export production

Two 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 (i.e. would not recommend).
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"
   ```

[13] Although in a sense, the remineralization of POC is not 'fixed' in that it does nto 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).

) 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. is 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 yr-1. The main particulate flux output is in units of mol m-2 yr-1, but since *c*GENIE 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 *r*8825, *c*GENIE 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.

6.7 Include a R-DOM cycle in the ocean

6.7.1 R-DOM degradation

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:

```
bg_ctrl_bio_remin_RDOM_photolysis=.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).

6.8 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_PO4Fe_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_PO4Fe_Cd_SPIN 10001
```

6.9 Include an iodine cycle in the ocean

In order to run cGENIE with a marine iodine cycle, one of the following *base-config*s is needed:

```
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[14] as the oxygen minimum zones in the 8-level ocean `worbe2` configuration are much more poorly developed (see *Ridgwell et al.* [2007a]).

Example configurations of several different levels of complexity of iodine cycling are given in the follow sesions.

6.9.1 Basic ('abiotic') iodine cycle

This section outlines the most trivial possible configurtaion 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.PO4Ibasic.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;
  `bg_opt_bio_remin_reduce_IO3toI='threshold'`,
  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, IO_3^- is completely reduced to I^-.
  Alternative options also exist for how the re-oxidation of I^- occurs. In the simple parameterization specified here;
  `bg_opt_bio_remin_oxidize_ItoIO3='lifetime'`,
  a fixed lifetime of I^- 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 $2I^- + 3O_2 \rightarrow 2IO_3^-$ exists.
- `# set [O2] threshold (mol kg`$^{-1}$`) for (complete) reduction of I.`
  The parameter `bg_par_bio_remin_cO2_IO3toI` sets the dissolved oxygen concentration threshold (mol kg^{-1}, below which IO_3^- will be reduced.
- `#set I lifetime (yrs)`
  Finally, the parameter `bg_par_bio_remin_Ilifetime` then sets the lifetime of I^- in years.

Model output is saved in the 'normal way' (refer to the User Manual) and amongst the ocean tracers ('`ocn_*`'), are the tracers of dissolved iodide and dissolved iodate (in units of mol kg^{-1}). Model output can also be contrasted with observed data re-gridded to the cGENIE (`worjh2`) grid.

A typical command-line launching of a model experiment (10000 years integration in this case) would be:

```
./runmuffin.sh cgenie.eb_go_gs_ac_bg.worjh2.BASEI /
EXAMPLE.worjh2.PO4Ibasic.SPIN 10000
```

6.9.2 'Biotic' component to the iodine cycle

This section outlines the next component of the marine iodine cycle, involving phytoplankton, with the two processes:

1. uptate of IO_3^-
2. remineralization/release of I^-

[14] The difference between the two `worjh2` configurations is that one (with the '`S`' in '`BASESI`' also includes a sulphur cycle.

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), 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: *Ridgwell et al.* [2007a] for details). When either POM or DOM is remineralized and elemental constituents released, rather than returning IO_3^- back to solution, it is assumed that the dissolved iodine is in the form of I^-. (This also requires an accounting of release of O2, as IO_3^- is taken up by the cell and assume to be instantaneously internally reduced.) The result is: (i) a progressive transformation $IO_3^- \rightarrow I^-$ in the surface ocean as DOM is continually created and destroyed, and (ii) a 'nutrient-like' enrichment of I^- in the sub-surface as a consequence of the remineralization of POM.

An example experiment configuration *user-config* – `EXAMPLE.worjh2.PO4Ibio.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`.)

6.10 Determine the CH4 flux required to achieve a particular atmospheric pCH4 value

Unlike the concentration of CO_2 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), CH_4 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 CH_4 *restored* to a particular value in a spin-up, requires that restoring to be maintained in any *continuation* experiment or CH_4 will quickly decay to zero. However, doing this (*restoring* CH_4 concentrations), prevents the effect of CH_4 emissions on being assessed (as the atmospheric composition is being help constant).

An alternative would be carry out the *spin-up* experiment with no *restoring* of atmospheric CH_4 (or *restoring* to zero), and then run the *continuation* experiment with no CH_4 *restoring*. This would enable e.g. CH_4 emissions experiments to be carried out and the change in atmospheric CH_4 in response to be simulated. The problem here is that the lifetime of CH_4 in the atmosphere scales with the CH_4 concentration. So in starting with no CH_4 in the atmosphere, the CH_4 lifetime is relatively short, and the response to CH_4 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 CH_4 flux that balances the oxidation rate of CH_4 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 CH_4 that balances the rate of oxidative loss for the desired atmospheric CH_4 concentration. To do this:

1. Carry out a *spin-up* with atmospheric CH_4 *restored* to the desired concentration.[15]
2. Determine the total loss rate of CH_4 (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`[16].

[15] For an example: see experiment `EXAMPLE_p0055c_PO4_CH4_SPIN` described in *cGENIE.Examples*.

[16] Second column (the value in units of mol yr-1)

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 CH_4 to the atmosphere is set, is described in: *cGENIE.Examples – spin-up* example EXAMPLE_p0055c_PO4_CH4_SPIN2

6.11 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 PO_4 (#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 grip 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*.

6.12 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

7 HOW-TOs: Forcings of the system

Note: All the *forcings* described here assume the 'new' (simplified) methodology for prescribing `forcings`. This methodology is enabled by changing the default parameter setting of `bg_ctrl_force_oldformat` to `.false.`. This is set automatically as part of the `runmuffin.sh` shell script.

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
2 == 2D
0 == point
-1 == SURFACE
-2 == BENTHIC
```

the default (33) being that the forcing is applied uniformly to the entire (3D) ocean volume.

Options `3`, `2`, and `0`, as: uniform 3D[17] (volume), uniform 2D (surface), and point forcing, respectively, require no additional (spatial) information. Hence, with options `3`, `2`, or `0` set, only an additional file specifying the time-dependent information for each forcing need be provided, in files of the form:

`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`) and `-2` (`BENTHIC`) 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`

for restoring *forcings*.

Examples of point and 2D (benthic) ocean tracer *forcing* are given below.

For details of the 'old', fully 3D spatially-explicit forcing methodology, refer to the *c*GENIE *user-manual*.

7.1 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 pCO2 and emissions, and (2) applying a surface flux calculated internally to meet some objective – here, the value of atmospheric pCO2 at any point in time.

1. The *forcing*:

   `pyyyyz_RpCO2_Rp13CO2_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 pCO2 (and associated d13C), while

   `pyyyyz_FpCO2_Fp13CO2_FRALK_FDIC_F13DIC_FCa`

[17] 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*).

is similar, except with a prescribed time-history of flux (emissions) of CO2 (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 pCO2 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*:

```
pyyyyz_FRpCO2_Fp13CO2_FRALK_FDIC_F13DIC_FCa
```

differs firstly in specifying both a CO2 emissions flux and restoring value for atmospheric pCO2. 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 pCO2, ALK additions to the ocean is made in order to try and maintain the prescribed history of atmospheric pCO2, regardless of the CO2 emissions also specified. If at any one time-step atmospheric pCO2 is higher than the target value, ALK is added to the ocean with the flux specified in the ALK flux forcing. If atmospheric pCO2 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[18].

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 CO2 flux forcing and restoring forcing time-series are specified 'as normal', and may constitute an SRES CO2 emissions scenario and RCP based pCO2 time-history, respectively, for example. For d13C, only the d13C of emissions is specified[19]. If an atmospheric CO2 emissions is not required, simply set the value in the time-series file to zero. Note that no scaling of atmospheric CO2 *forcings* (e.g. `bg_par_atm_force_scale_val_3`) must be used because it will scale both restoring and emissions ...

If the model cannot quite attain the pCO2 target specified, you probably do not have a sufficiently large ALK flux specified. Conversely, if the pCO2 target is significant over-shot (technically: under-shot) then you might have prescribed too large a flux.

7.2 Prescribe a time-varying history of radiocarbon production in the atmosphere

The original way for doing was this was to create a CO_2 flux forcing and give its $\delta^{14}C$ signature an extremely positive value. However, inputs and outputs of isotopic compositions in *c*GENIE are limited to being between -999 and +999 per mil. Prescribing a small flux of CO_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_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 *c*GENIE to interpret the value specified as its $\delta^{14}C$ forcing signature as an absolute flux (in units of mol yr^{-1}) rather than convert from a per mil notation. *c*GENIE knows to use this alternative if the flux of CO_2 specified in the CO_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:

```
ac_par_atm_F14C=0.387E+03
```

one would now set an identical value of `0.387E+03` in both *forcing* files:

```
biogem_force_flux_atm_pCO2_14C_sig.dat
biogem_force_flux_atm_pCO2_sig.dat
```

[18] This is likely pretty unphysical for most applications, hence the default is `.false.`.

[19] A restoring must *not* also be set.

(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_2)).

Selection of the required forcings in `configure_forcings_atm.dat`[20] then looks like this:

```
-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]'
-END-OF-DATA-
```

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.

7.3 Prescribe an injection of radiocarbon-dead DIC

First ... a *base-config* with 14C tracers is needed, e.g.,:

`cgenie_eb_go_gs_ac_bg_itfclsd_16l_JH_ANTH`

`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.:

`pyyyyx_FDIC_F13DIC_F14DIC`

and under the heading `--- FORCINGS ---`, might look something like:

- `bg_ctrl_force_oldformat=.false.`
  `bg_par_forcing_name="worjh2_FDIC_F13DIC_F14DIC"`
  which prescribes a forcing of DIC plus its (13C and 14C) isotopes to the ocean (somewhere or everywhere).
- `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[21].
- 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.

[20] 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_2. Actually ... DO NOT add a ^{13}C forcing, just in case ...

[21] Note this is on the scale of d14C not D14C

7.4 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
2 == 2D
0 == point
-1 == SURFACE
-2 == BENTHIC
```

with the default requiring a (3D) spatially explicit field to be provided. Options `-1` (`SURFACE`) and `-2` (`BENTHIC`) 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 either 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 `NaN`s 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: `1`s can be replaced by `0`s to remove unwanted locations.[22] 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.[23]

[22] This can be quite time-consuming and tedious and there is no particular short-cut :(

[23] The values in the forcing map need not be all 1.0 of course.

8 HOW-TOs: Sediments and weathering

8.1 Spin-up the full marine carbon cycle including deep-sea sediments

By a 2-step process[24]:

1. **First-guess '*closed system*' *spin-up***
   As of **r4211** this it is possible to carry out the initial spin-up, **with** a solute input to the ocean via rivers, but also with the system configured 'closed', i.e.,:

   ```
   bg_ctrl_force_sed_closedsystem=.true.
   ```

   The weathering flux is subtracted from ocean cells overlying the sediments to balance the global budget and ensure a closed system. This subtraction involves partitioning the total global weathering flux between each ocean floor cell with a subtraction in proportion to the estimated CaCO3 preservation and burial rate. To utilize this methodology now requires that the ROKGEM module is used, i.e., a *base config* such as:

   ```
   cgenie_eb_go_gs_ac_bg_sg_rg_itfclsd_16l_JH_BASE
   ```

   A first guess for the weathering flux must now be prescribed. This could be derived from a previous closed system model experiment with no weathering flux specified (diagnosing weathering from total global CaCO3 burial as described earlier), or from the literature, e.g., *Ridgwell* [2007] cites 20 Tmol HCO3- yr-1, an equivalent CaCO3 weathering rate of 10 Tmol yr-1:

   ```
   rg_par_weather_CaCO3=10.00E+12
   ```

   The following *user config* file

   ```
   EXAMPLE_worjh2_PO4_S36x36_SPIN
   ```

   can be used for the closed system spin-up.

   To launch an experiment, type (all in one line; notes space separators between line items in this document format):

   ```
   ./runCCSgenie.sh cgenie_eb_go_gs_ac_bg_sg_rg_itfclsd_16l_JH_BASE /
     EXAMPLE_worjh2_PO4_S36x36_SPIN 20001
   ```

   To submit to the cluster type:

   ```
   qsub -q kitten.q -j y -o cgenie_log -S /bin/bash subcgenie.sh
   cgenie_eb_go_gs_ac_bg_sg_rg_itfclsd_16l_JH_BASE /
     EXAMPLE_worjh2_PO4_S36x36_SPIN 20001
   ```

   20000 years (20001 if using the default *time-series* saving points in order to record the last (annual averaged) year of the experiment) is probably about the minimum practical *spin-up* time. Primarily – you are looking for convergence in the mean wt% CaCO3 value (averaged sediment composition), which is recorded in the *BIOGEM time-series* file:

   ```
   EXAMPLE_worjh2_PO4_S36x36_SPIN
   ```

2. **Open system spin-up**
   The last stage is an open system spin-up as described previously. The prescribed weathering flux (`rg_par_weather_CaCO3`) is revised and set equal to the diagnosed global CaCO3 burial rate ('`Total CaCO3 pres (sediment grid)`') as reported in the SEDGEM module results file:
   `seddiag_misc_DATA_GLOBAL.res`. In addition, an *open system* must now be specified in the *user config*:

   ```
   bg_ctrl_force_sed_closedsystem=.false.
   ```

   50000 years (50001 if using the default *time-series* saving points in order to record the last (annual averaged) year of the experiment) is probably about the minimum practical *spin-up* time. Again – you are looking for convergence in the mean wt% CaCO3 value.

[24] This is a revised methodology compared to that described in the GENIE-1 HOW-TO.

There is still some departure of ocean Ca and ALK inventories during the revised multi-stage spin-up compared to observed (and the initialized values), but this is substantially reduced compared to the original 2-part spin-up methodology as well as to a single spin-up methodology.

TIP: Having completed the full marine carbon cycle spin-up, it is recommended that the CaCO3:rain ratio is set invariant – see earlier HOWTO. If the default CaCO3 parameterization setting is retained, CO2-calcification feedback as described in *Ridgwell et al.* [2007b] is enabled.

NOTE: There is no climate feedback by default. To run experiments with feedback between CO2 and climate, add:

```
ea_36=y
```

at the end of the *user config*.

8.2 Run the sediments at higher resolution (as compared to the ocean grid)

By default (as set in the *base config* file in `~/genie/genie-main/configs`) the SEDGEM sediment grid is configured at a resolution of 36x36 (and on an equal area grid), by:

```
SEDGEMNLONSOPTS='$(DEFINE)SEDGEMNLONS=36'
SEDGEMNLATSOPTS='$(DEFINE)SEDGEMNLATS=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:

  ```
  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 in.
- The required sediment grid topography (bathymetry):

  ```
  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].

The directory location of the required files is set by input directory namelist setting, which by default is:
`sg_par_indir_name=" $RUNTIME_ROOT/genie-sedgem/data/input"`

As described in Ridgwell and Hargreaves [2007], SEDGEM can be sub-gridded to a resolution of 72x72 (equal area). The following namelist additions are necessary to the *user config* file:

```
SEDGEMNLONSOPTS='$(DEFINE)SEDGEMNLONS=72'
SEDGEMNLATSOPTS='$(DEFINE)SEDGEMNLATS=72'
sg_par_sed_topo_D="sedgem_topo_D.72x72"
sg_par_sedcore_save_mask_name="sedgem_save_mask.72x72"
```

NOTE: Carbonate chemistry stability problems (= model crash) may occur in the 16-level configuration in conjunction with 72x72 resolution sub-gridded sediments. Who knows why?! :(

8.3 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 $CaCO_3$ (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 $CaCO_3$ 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[25] (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 sg_par_sed_Dmax_neritic (in units of m below the ocean surface). A typical value is 176.0[26], 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.

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[27] 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:

```
sg_ctrl_sed_neritic_reef_force=.TRUE.
```

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 (sg_par_sed_Dmax_neritic).

Additional information and tips on setting up shallow water sediment grids can be found in some of the EXAMPLES.

Relevant EXAMPLES:
EXAMPLE.p0251b.PO4.SPIN0

8.4 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: 8.3). 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:

```
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 *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, [Ca2+], etc. and (preindustrial) atmospheric pCO_2, and in conjunction with an adequate simulation of the large-scale productivity of the ocean and recycling in the interior ocean.

[25] Set by parameter sg_par_sed_topo_D_name that must point to a file in cgenie.muffin/genie-sedgem/data/input.

[26] Note that the simple (esp. $CaCO_3$ diagenesis schemes) arguably only applicable below depths of ca. 1000 m, although even this is complicated by varying patterns of productivity across the ocean.

[27] The parameter name to set the reef mask is sg_par_sed_reef_mask_name.

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 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 pCO_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 pCO_2 and $[Ca^{2+}]$. The parameters specifying the initial mean ocean ALK and DIC concentrations (μmol kg^{-1}) are:

   ```
   bg_ocn_init_12=2.3630E-03
   bg_ocn_init_3=2.244E-03
   ```

   for their respective modern mean ocean values.
   In order to estimate appropriate past concentrations for a given pCO_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 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).
   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 pCO_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: 8.7) 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 shreshold, 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 8.7 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 atmospheric pCO_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 ompeting 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[28].
   An aragonite saturation value can be forced towards rather than calcite by setting:

   ```
   bg_ctrl_force_ohmega_calcite=.false.
   ```

   The setup for this is a little involved and involves specifying a particular forcing that cGENIE 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.PO4.SPIN0` and described in the `cgenie.muffin.EXAMPLES` document. A typical configuration would like like the following:

   ```
   bg_par_forcing_name="pyyyyz.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 $p\text{CO}_2$ 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[29] 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[30], 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 $p\text{CO}_2$*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 8.7) 3rd spin-up phase is conducted with no forcing.

   Example *user-configs* of this sequence are provided (`EXAMPLE.p0251b.PO4.SPIN*`). Note that the 3rd phase of spin-up could be carried out with a prescribed $p\text{CO}_2$ *plus* associated isotopic composition if not already done so (e.g. as part of the phase #1 of the spin-up).
   Conclusion: work-able.
   Relevant EXAMPLES:
   `EXAMPLE.p0251b.PO4.SPIN0`, `EXAMPLE.p0251b.PO4.SPIN1`, `EXAMPLE.p0251b.PO4.SPIN3`

[28] A negative flux of ALK can be enabled, which would act to reduce saturation, by setting the following: `bg_ctrl_force_invert_noneg=.false.`
[29] These values represent the *total* flux that distribution throughout the ocean relative to grid cell volume.
[30] SEDGEM file seddiag_misc_DATA_GLOBAL.res

8.5 Specify a particular carbonate mineralogy

Relevant EXAMPLES:

Also see: 8.4

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:

```
sg_par_sed_reef_calcite=.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: 8.4) – an aragonite, rather than calcite (the default) saturation value can be forced towards rather than calcite by setting:

```
bg_ctrl_force_ohmega_calcite=.false.
```

8.6 Set up a (silicate) weathering feedback

Relevant EXAMPLES: `EXAMPLE.worbe2.Colbournetal2013.EMISSIONS`

Also see: 8.7

To create a temperature (only) dependency for the weathering of carbonate and silicate rocks, the following two parameter values need to be set (for carbonate and silicate weathering, respectively):

```
rg_opt_weather_T_Ca=.true.
rg_opt_weather_T_Si=.true.
```

A reference temperature governs the dependency of weathering on climate change and modifies the solute fluxes from weathering scaled (non-lineary) to the deviation of mean global land surface temperature from the reference temperature. The mean global land surface temperature is given in the **BIOGEM** *time-series* file `biogem_series_misc_SLT.res` and is set equal to the reference temperature parameter:
`rg_par_ref_T0`.

The baseline (unmodified) solutes fluxes from terrestrial weathering are set by the parameters:

```
rg_par_weather_CaCO3
rg_par_weather_CaSiO3
```

for carbonate and silicate weathering, respectively. At steady-state (no climate perturbation), these must sum up to the total weathering flux which is given by the total global $CaCO_3$ burial flux found in the file `seddiag_misc_DATA_GLOBAL.res` (in the `genie-sedgem` results sub-directory). To balance the silicate weathering, volcanic CO_2 out-gassing is then assigned a value equal to silicate weathering flux by setting the parameter `rg_par_outgas_CO2`.

There are different 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.::

   ```
   rg_par_weather_CaCO3=10.0E+12
   rg_par_weather_CaSiO3=0.0
   ```

   for a hypothetical example with 10 Tmol yr^{-1} total global $CaCO_3$ burial. No volcanic CO_2 out-gassing should then be prescribed.

2. Secondly, silicate and carbonate weathering could be assumed to be split evenly, e.g.:

   ```
   rg_par_weather_CaCO3=5.0E+12
   rg_par_weather_CaSiO3=5.0E+12
   ```

   Volcanic CO_2 out-gassing needs to be set equal to the baseline silicate weathering flux:

   ```
   rg_par_outgas_CO2=5.0E+12
   ```

3. Thirdly, volcanic CO_2 out-gassing could be assumed, which then constrains the silicate flux, with the carbonate flux being whatever is required to make the total Ca^{2+} weathering flux equal to global $CaCO_3$ burial.

A difficulty arises because setting the reference temperature rg_par_ref_T0 equal to the mean global land surface temperature only gives rise to weathering fluxes exactly equal to the reference parameter values (rg_par_weather_CaCO3 and rg_par_weather_CaSiO3) for a non-seasonally forced climate. Because weathering is non-linear in climate (here just temperature), a seasonally forced configuration of the model with give rise to slightly modified weathering fluxes. The result is a slight drift in atmospheric $p\mathrm{CO}_2$) and climate, even after a fairly long (100s of kyr), or alternatively, a slightly different steady state $p\mathrm{CO}_2$) value (for a ca. 1 Myr *spin-up*).

The reason is that long-term climate and hence $p\mathrm{CO}_2$) is controlled by the silicate weathering component of total weathering, not total weathering. Under a seasonally forced rather than annual average climate, the non-linearity in the weathering response to temperature deviations means that silicate weathering will generally slightly exceed volcanic CO_2 out-gassing. Atmospheric $p\mathrm{CO}_2$) 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:
biogem_series_misc_exweather_Ca.res
which records the absolute excess of weathering compared to out-gassing, in 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.

If the carbon isotopic signature of volcanic CO_2 out-gassing is assigned a value of e.g. -6.0‰. The $\delta^{13}C$ of weathered $CaCO_3$ is then simply set in order that 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 For example, assuming equal fluxes of 5 Tmol yr^{-1} for both weathering components and for volcanic CO_2 out-gassing (at -6.0‰) and assuming a mean carbonate burial $\delta^{13}C$ of 3‰, requires:

```
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 ...

8.7 Accelerate the weathering-sedimentation mass balance ('GEMlite')

Relevant EXAMPLES: EXAMPLE.p0251b.PO4.SPIN2

Also see: cGENIE user-manual 'FAQ' (further comments on GEMlite applicability).

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:
   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 **ma_gem_notyr=50** and **ma_gem_yr=50**, you would have a sequence with 50 years of full updating, followed by 50 years of accelerated.
   For instance: if **ma_gem_notyr=10** and **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. **ma_gem_notyr=10** and **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: **ma_gem_notyr=999999** and **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 ...).
2. Adaptive sequencing.
   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.
Adaptive sequencing adjust the time-stepping 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 `ma_gem_notyr` and `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:

- `ma_gem_adapt_dpCO2dt=0.1` (ppm yr-1)
- `ma_gem_adapt_DpCO2=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:

```
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[31].
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, *c*GENIE 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 (pCO2) 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.

[31] Adjusting the fluxes themselves during the GEM intervals would break the underlying assumption inherent in the acceleration approximation.

9 HOW-TOs: Visualization

10 HOW-TOs: Model development

Best not to. But ...

10.1 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`[32].

   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 loking 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 catagory 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 ...

10.2 Define a new tracer

You probably should not be doing this ... but ... just in case ...

10.2.1 Basic definition procedure

1. The starting point to adding a new tracer in *c*GENIE 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`

[32] `cgenie.muffin/genie-biogem/src/fortran`

- `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[33] 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[34], 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 δ 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
'`5`' → 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 netCDF output.

#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[35]:

   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.

[33] The meaning of the columns is also summarized at the end of the file.
[34] The elemental components of organic matter (P, N, etc) count as bulk tracers for the purpose of dependency.
[35] Watch out that the line numbers may have changed somewhat ...

   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[36] :(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.

   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.

[36] Note that there are no sediment (solid) tracer arrays for either initial composition or forcings.

10.2.2 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*[37] 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.)

10.2.3 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`[38]

   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`[39].

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[40]. 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

[37] `/cgenie.muffin/genie-main/configs`

[38] `cgenie.muffin/genie-main/src/fortran/cmngem/gem_util.f90`

[39] `cgenie.muffin/genie-biogem/src/fortran/biogem_data.f90`

[40] 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

carbon in both organic matter and $CaCO_3$ (and for that matter, opal)) and of the isotope ratios of specifies to their bulk equivalent. The array is (re)populated each time-step in `sub_calc_bio_uptake`[41], 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.
```

Scavenged tracers are automatically remineralized along with the corresponding parent particulate tracer by default[42]. However, there is no scavenging or creation of scavenged tracers by default. A call would need to be added to `sub_box_remin_part`[43] (ca. L2961), e.g. following the examples of Fe scavenging and H2S reaction 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.

10.3 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.

10.3.1 Data saving 101

For new *time-slice* saving, code needs to be added to either `sub_save_netCDF_2d_USER` (`biogem_data.netCF.f90`), or `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).

[41] `cgenie.muffin/genie-biogem/src/fortran/biogem_box.f90`

[42] 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.

[43] cgenie.muffin/genie-biogem/src/fortran/biogem_box.f90

- 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) ...

10.3.2 (full) Time-slice saving

10.3.3 (full) Time-series saving

BLAH

11 HOW-TOs: Miscellaneous

11.1 Set up carbon dioxide removal (CDR) geoengineering experiments

There are various ways 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*.

11.1.1 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:
`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:

  ```
  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:

  ```
  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. respired CO2 vs. temperature and salinity (and hence ocean circulation changes)):

  -
    ```
    # pump T and S?
    bg_ctrl_force_GOLDSTEInTS=.false.
    ```

    (the default) prevents T and S being advected.

  -
    ```
    # ONLY pump T and S?
    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?
    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.)

12 Contact Information

- Andy Ridgwell: `bandy@seao2.org`