

The Bumper Book of *.cookies*

(The cGENIE.cookie user-manual and
introduction to Earth system modelling)

Andy Ridgwell

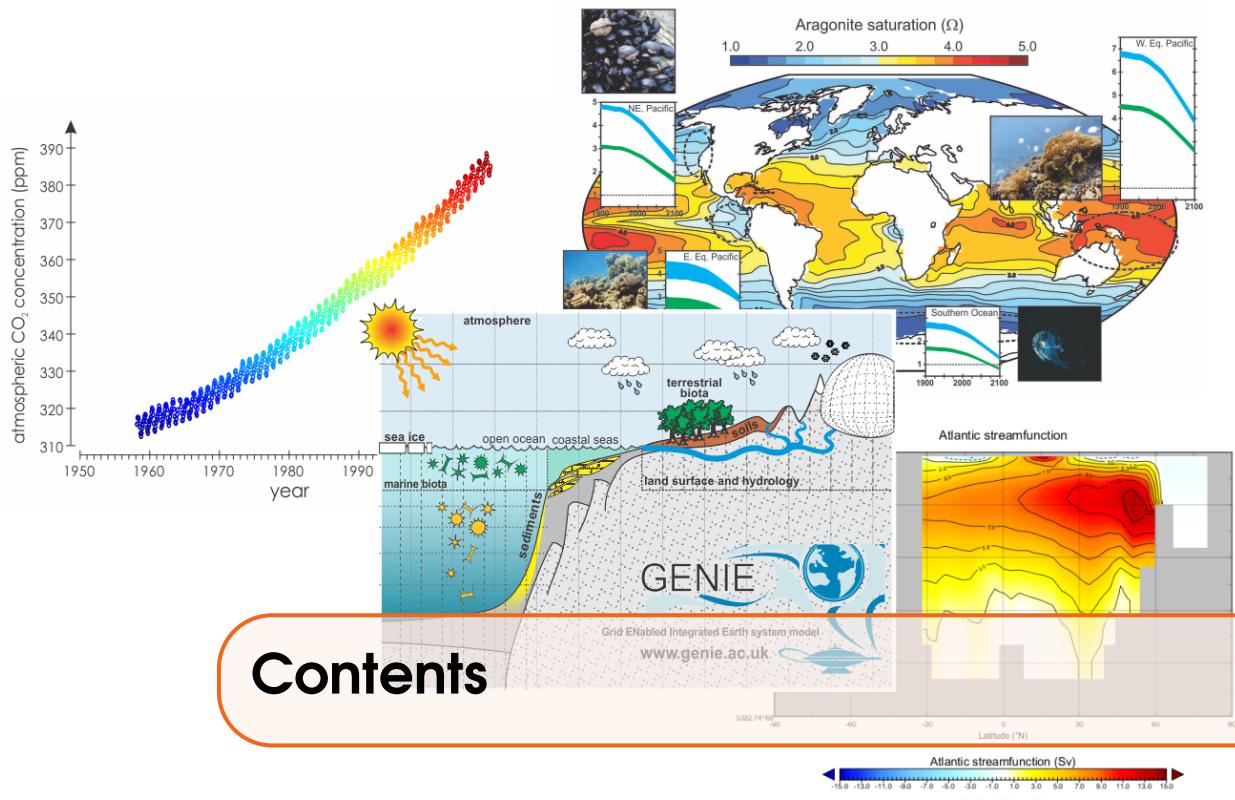
Copyright © 2017 Andy Ridgwell

PUBLISHED BY DERPY-COOKIES INC.

MYCOOKIE.SEAO2.ORG

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, October 2017



Contents

0	Installation, Configuration, Basic Usage	1
0.1	Before anything else ...	2
0.2	Starting (dozing?) off ...	6
0.2.1	Logging in!	6
0.2.2	Downloading/installing the model code	7
0.2.3	Configuring the code	7
0.2.4	Testing the model code	9
0.3	Running the model	12
0.4	Creating new experiments!	16
0.5	Model output	17
0.5.1	Time-slice output	17
0.5.2	Time-series output	18
0.5.3	File naming convention	18
0.6	Viewing model output	19
0.6.1	Time-series output	19
0.6.2	2- and 3-D time-slice output	20
0.7	Submitting experiment ‘jobs’	21
0.8	‘Restarts’	23
1	Climate dynamics & experimental design	25
1.1	Brrrrrrrrrrr – it’s chilly on ... snowball Earth!	27

1.2	Further ideas	31
1.2.1	Feedback loop analysis	31
1.2.2	Continental configuration (1)	31
1.2.3	Continental configuration (2)	32
1.2.4	Geothermal heat input	32
1.2.5	Seasonality	33
2	Ocean circulation I – modern	35
2.1	Simulating ocean circulation	37
2.2	Visualizing ocean circulation	39
2.3	Tracing ocean circulation	41
2.4	Tracing ocean ... ventilation	44
2.5	Poking the climate beast (1)	45
2.6	Poking the climate beast (2)	48
2.7	Further ideas	49
2.7.1	Assessing the spatial sensitivity of deep-water formation to shutdown	49
2.7.2	Forcing stronger deep-water formation ('anti-hosing' investigations)	49
2.7.3	Regional hosing – Poking the Mediterranean	49
3	Ocean circulation II (fake worlds)	51
3.1	(none)	52
4	Atmospheric dynamics	53
4.1	(none)	54
5	Fossil fuel CO₂ and 'ocean acidification'	55
5.1	Exploring the consequences of fossil fuel CO₂ emissions	57
5.1.1	Idealized emissions forcing	58
5.1.2	Where has my carbon gone???	61
5.1.3	Historical (real-world!) emissions forcing	62
5.1.4	Assessing future carbon emissions impacts	64
5.2	Further ideas	65
5.2.1	Assessing the importance of emissions rate	65
5.2.2	Determining thresholds of environmental impact	65
5.2.3	Future atmospheric CO ₂ concentration pathways ('RCPs')	68
5.2.4	Isotopic tracing of fossil fuel CO ₂ uptake	69
5.3	Appendix	70

6	Ocean biogeochemical cycles	73
6.1	Basic controls on biological productivity	78
6.2	Comparing model vs. observations	83
6.3	Temperature and the biological pump	85
6.4	Iron co-limitation of biological productivity	87
6.5	Further ideas	90
6.5.1	Further analysis ideas	90
6.5.2	Further experiment ideas	90
6.5.3	Other thoughts and suggestions	91
6.5.4	Biological pump background information	92
6.5.5	Advanced experiment ideas	93
7	Marine ecosystems and dynamics	95
7.1	(none)	96
8	The geological cycle of carbon	97
8.1	(none)	98
9	The land surface and carbon-climate feedbacks	99
9.1	(none)	100
10	Climates of Past Worlds	101
10.1	(none)	102
11	Proxies and Reconstructing the Past	103
11.1	(none)	104
12	Synthesis: carbon in the Earth system	105
12.1	(none)	106
13	cookie model output	107
13.1	(none)	108
14	Introduction to model results analysis	109
14.1	(none)	110

15	muffinplot	111
15.1	(none)	112
16	muffindata	113
16.1	(none)	114
17	muffingen	115
17.1	(none)	116
18	FAQ	117
18.1	(none)	118
19	HOW-TO (technical)	119
19.1	(none)	120
20	HOW-TO (experiment)	121
20.1	(none)	122
21	cookie Development	123
21.1	(none)	124

Before anything else ...

Starting (dozing?) off ...

Logging in!

Downloading/installing the model code

Configuring the code

Testing the model code

Running the model

Creating new experiments!

Model output

Time-slice output

Time-series output

File naming convention

Viewing model output

Time-series output

2- and 3-D time-slice output

Submitting experiment 'jobs'

'Restarts'



0. Installation, Configuration, Basic Usage

Stuff to keep in mind:

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

0.1 Before anything else ...

ReadMe

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

[!]¹

Software version naming conventions

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

linux ...

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

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

```
[username@host ~]$
```

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

```
[username@host myfolder]$
```

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

¹Also read footnotes please.

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

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

NOTE: BE VERY CAREFUL that spaces are not missed out when typing out example lines. Also be careful not to confuse the number one (1) for the letter 'el' (l). Mis-spelling/typing is generally the most likely reason for things not working.

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

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

Required computer hardware/software

For running **cupcake**, your options are:

1. Remotely

To do this, you will need an account on a linux-based server or cluster. You can use any platform (**linux**, **macOS**, **WindoZ**, as well as **iOS** and **Android** (which is based on **linux** in any case)) to connect to the cluster. You will need the following software on your local machine:

- (a) A terminal ('shell') window. This is no problem for **linux** and **Mac** users (you already have one built in). For **Windows**, either download a simple (and old) **SSH** client (ssh-client) from my website⁴ or you can get hold of e.g. **PuTTY** (<http://www.putty.org/>).
- (b) A sftp (secure file transfer) client for convenience (i.e. dragging and dropping files between local and remote computers, and opening files directly on the remote computer cluster). If you have installed ssh-client (**Windows**, above) then a sftp client is already included as part of this software. If using **PuTTY** (**Windows**) you might try downloading **WinSCP** (<http://winscp.net/eng/index.php>). For **macOS**, you can connect to the server through the **Terminal**, but some sftp software for viewing/navigating server file structure include: **FileZilla** (recommended), **Cyberduck**, **TextWrangler**. For **linux**, maybe **FileZilla**.

2. Locally

It is possible to install and run the '**(cGENIE.)cookie**' Earth system model either on a linux box (e.g. **Ubuntu**) or on a **Mac**⁵⁶. At a minimum, you will need:

⁴<http://www.seao2.info/cgenie/software/ssh-client.exe>

⁵Sets of detailed installation instructions are available in the HOW-TO section of this manual.

⁶It is also possible to run **cookie** under **Windows**.

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

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

- (a) A viewer for netCDF format spatial data. A **Java** viewer called **Panoply** is provided by NCAR for all platforms – <http://www.giss.nasa.gov/tools/panoply/> (Note that you will need **Java** installed!) (Or alternatively: **MATLAB**, **python**, etc..)
- (b) A simple text editor, except not the rubbish default **Windows** one – you need one that can display **unix** ASCII text without screwing it up. Options for **Windows** users are: **notepad++** (<https://notepad-plus-plus.org/>) **SciTE** (<https://www.scintilla.org/SciTE.html>) (**linux** and **Mac** users need no special/different editor compared with your standard editor – everything will display just fine). You can also use **linux** command line based editors such as **vi**.

File editors ...

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

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

If you fiddle about with the settings under Options/Preferences in the **WinSCP** program and apply a little common sense, it should be possible to configure things so that you can simply double-click on a file in the remote (right-hand) window panel and it will open like magic (almost)! Saving the file after editing) should then result in the file being saved back to the cluster. Or you can select Edit With (and then **SciTE**) from right-mouse-button-clicking on the filename. Or ... a crude but workable approach is to use an sftp client to drag the file to your local machine (assuming **cookie** is installed remotely), edit it there, and then drag it back again.⁷

⁷Note that care still has to be taken to avoid certain **Microsoft** text editing programs under **Windoze**.)

Model documentation in general

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

1. On GitHub.

The **latex** source for the documentation lives here, allowing you to compile the most up-to-date PDF document. And ... make changes yourself and have them incorporated into the official documentation⁸.

However, note that you will have to compile the latex source yourself to create a PDF ...

2. On my website.

Here you can find compiled PDF versions of the documentation ... but it could be a little out of date (the up-to-date latex sources lives on **GitHub**).

This document in particular

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

Go!

OK – now we are ready to start ...

⁸First clone the **git** repository. Make changes. Commit them locally. Make a 'pull request' ...

0.2 Starting (dozing?) off ...

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

0.2.1 Logging in!

In running **cookie** locally – log into your **linux/macOS** box (and skip on to the next section)!⁹

Or ... and much more likely to be the case – if you are running **cookie** remotely (e.g. via a user account on a computing cluster or server), then log into the remote server or cluster account using a 'suitable terminal program',¹⁰:

- If logging in via a **linux/macOS** box, open the terminal/shell window and simply SSH in¹¹, e.g.

```
$ ssh username@clusternname
```

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

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

- On a **WindoZ** machine – first start the **WinSCP** program (an sftp file transfer client). Under Host Name, enter the remote server or cluster name (e.g. `catname.ucr.edu`):

The Port number should be set to 22 (except for `sterling.ucr.edu`). Enter your computing cluster user-name on the line below this ('User Name') and then the Password. Click on Login. You will also need a terminal window. This can be opened by clicking on the 'Open session in PuTTY' icon on the top icon row, or pressing `Ctrl+P`.

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

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

Note that the cluster you access may not use the standard 'port' number of 22. If your computer uses a different port number – in the login window of your sftp client, simply change the number in the Port box, or if you are using the command line, you will need to type:

```
$ ssh -p xxxx username@clusternname
```

where `xxxx` is the port number.

⁹If you fail at this step, you'll have to take up box-modelling instead.

¹⁰It very much depends on what software you are using. Provided are instructions for some examples, but only examples, and your reality may be rather different.

¹¹If your current directory looks something like this: `[username@clustername ~]$` then you are probably already logged in! Otherwise, it will look like: `username@localcomputername:~$`

0.2.2 Downloading/installing the model code

The next step is to download/install a copy of the source code for **cookie**. The current release of the cGENIE model (**cookie**) lives here:

```
https://github.com/genie-model/cgenie.cookie
```

There are 2 options:

1. 'cloning'

The preferred/advised way is to *clone* the repository to where you intend to run **cookie**¹². While you can also use a GUI based git client, easiest is at the command line (e.g. from your HOME directory), using the command `git clone`¹³:

```
$ git clone https://github.com/genie-model/cgenie.cookie.git
```

By doing this, you have created your own code repository (and an identical copy of the one hosted on GitHub). As part of the `git clone` command, you also automatically *check out* (from your very own personal repository) a copy of the code.¹⁴

2. Simple downloading.

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

0.2.3 Configuring the code

You may ... or may not, need to configure some local environment settings so that all the libraries etc. that are needed to compile **cookie** can be found. To a very limited extent, **cookie** will try and identify your computer and make the required setting automatically. The changes, if any, that you need to make will depend on the platform where you will be running **cookie** (i.e. the computer where you have just cloned (or downloaded) the code repository to). The currently recognized platforms and required actions are as follows:

The required configuration settings for **cookie** depend on the specific computing platform that you are running on. What follows is a list of the changes associated with some possible platforms that you might be using – find the list name (**bold**) that corresponds to the platform that you are using and ignore all the other options.¹⁵

¹²But see later for other/better ways of working.

¹³This: "... clones a repository into a newly created directory, creates remote-tracking branches for each branch in the cloned repository, and creates and checks out an initial branch that is forked from the cloned repository's currently active branch."

¹⁴Note that the major difference then with the **svn** system, is that previously, the GENIE code repository existed only on the University of Bristol server, and you *checked out* the code remotely from there.

¹⁵If you are using a platform not listed here, you may be able to simply adapt the instructions for one of the ones listed.

- **Ubuntu** – no changes are necessary (with caveats).

This is the default assumed platform. No changes are necessary IF the **netCDF** libraries are installed in their default locations and a relatively recent version of **netCDF** is used.¹⁶

- **sterling** (UCR cluster) – no changes are necessary (the cluster is automatically identified).

- **eevee** (UCR cluster) – no changes are necessary (the cluster is automatically identified).

- **macOS**

Please refer to the separate **macOS** instructions.

- **Windows**

Please refer to the separate **Windoz** instructions.

- **Otherwise ...**

... one or more edits will be required to the file user.mak, which lives in the genie-main directory.

1. At the end of the user.mak, the **netCDF** path needs to be changed.

First comment out¹⁷. i.e.,

```
### DEFAULT ###
NETCDF_DIR=/usr/local
```

to

```
### DEFAULT ###
#NETCDF_DIR=/usr/local
```

Then un-comment¹⁸ the line: #NETCDF_DIR=my_path

AND change my_path to be the path to your **netCDF** libraries ... the trick step :o)

2. For MacOS users, there are alternative machine type options depending on your CPU. You need to comment out MACHINE=LINUX in user.mak and then chose and un-comment one of the following:

```
#MACHINE=OSX # Intel processor
#MACHINE=OSX_M # Apple silicon (M1, M2 etc.)
```

If your computer is configured to use python v.2 by default and/or does not have python v.3, then you will need to make a few minor changes (see HOW-TO).

¹⁶By 'relatively recent' – the default settings assume that the **FORTRAN** and **C netCDF** libraries are separate.

¹⁷To 'comment out' a line – simply add a # symbol to the very start of the line. When **cookie** runs, this line will be ignored.

¹⁸To 'un-comment' a line – simply remove (delete) the # symbol from the beginning of the line.

0.2.4 Testing the model code

Finally, you need to test the code to ensure that all the files have been cloned/installed correctly.

First, change directory (see: Figure 1, and refer to the linux HOW-TO) to¹⁹:

```
cgenie.cookie/genie-main
```

If you are not ‘linux-friendly’ (see: Section ?? for **linux** basics) – maybe at first do this in steps – list the contents of the directory (`ls`) to check where you are (i.e. what directories are available to chance to), then change to `cgenie.cookie` (`cd cgenie.cookie`), then list again (`ls`) (and see what further directories are there), then change to `genie-main` (`cd genie-main`), and only then ... type²⁰:

```
$ make testbiogem
```

This compiles a basic carbon cycle enabled configuration of **cookie** and runs a short test, comparing the results against those of a pre-run experiment (also downloaded alongside the model source code). It serves to check that you have the software environment correctly configured. There may be some ‘Warnings’ reported (== somepony’s sloppy programming) but these are not detrimental to the ultimate science results (we hope!). If you see the error message:

```
make: *** No rule to make target
```

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

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

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

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

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

This is quite ‘normal’ – the model is thinking! Also – ignore the compiler warnings ...

This completes the basic code installation.

If the test doesn’t ‘work’,²¹ – try issuing the command:

```
$ make cleanall
```

and then re-try the test.

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

¹⁹Note: the model is *always* run from `cgenie.cookie/genie-main`

²⁰Remembering that the \$ is to indicate the command line, and you do not actually type it in.

²¹Note that if you have disabled the compilation of the C program that compares netCDF files, the test will run but never complete and you’ll not get a `**TEST OK**` reported, even if the test experiment ran correctly.

Note that **GitHub** does not host files larger than 100 MB, and the 'lookup table' for calculating opal dissolution in sediments (see e.g., *Ridgwell et al.* [2003]) is larger than this. It has hence been committed to **git** as an archived file. Only if a silica cycle is to be employed in your **cookie** experiments, does this file need to be unpacked. A script is provided for this – from `genie-main`, type:

```
$ ./installcookie.sh
```

and this will unpack the opal lookup table to `genie-sedgem/data/input` as well as unpacking a copy of the calcium carbonate lookup table²².

Some brief notes on git

The simplest workable installation of **cookie**, as described above, is to use `git clone`. You end up with a carbon copy of the cookie code repo (I am too lazy to type out 'repository' again) which you have also automatically now 'checked out'. Changes and developments will occur to the code on the GitHub repo from time-to-time, and it may be that either you might benefit by using a more up-to-date code base, or specific changes may have been made that you absolutely need. To determine the status of your repo, type²³ (from `cgenie.cookie`):

```
$ git status -uno
```

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

```
$ git fetch
```

which will 'download [new and modified] remote content but not update your local repo's working state'. If you now type `git status -uno`, **git** can tell you if there is newer content (e.g. 'Your branch is behind 'origin/master'²⁴). To merge in the fetched content:

```
$ git merge
```

Both these commands are also combined in a single command²⁵:

```
$ git pull
```

If you are not developing code, and hence not editing files in the repo (but e.g. only adding new model configuration files), your life should mostly be trouble-free with regards to updating your code via `pull`.²⁶

²²For now, the *CaCO₃* lookup table also included in the git repo in its expended (unpacked) form.

²³Here, the `-uno` ensures that files (e.g. experiment configuration files) you have created, but not added and committed, are not listed.

²⁴`origin` is the origin of the repo – GitHub, and `master` is the name of the branch, in this case, the default branch name.

²⁵If there are no changes (to `fetch` and `merge`), the you get the message: `Already up-to-date.`

²⁶One exception is, if any of the model installation/configuration files that you might have edited, i.e. one or all of: `genie-main/user.mak`, `genie-main/makefile.arc`, and/or `genie-main/makefile`, have also changed on `origin/master`, then `pull` (or `merge`, after `fetch`) will fail (with the message: 'Please, commit your changes or stash them before you can merge. Aborting'). The root issue is a conflict between remote file changes, and local ones that you had not committed to your repo.

You probably want to keep your local configuration changes, otherwise you'll probably end up re-doing them all over again. One solution is to sneakily 'hide' (`stash`) them out of sight, by: `$ git stash` You can now `pull`, updating your repo with respect to `origin/master`. Then – you want your changes back, so apply the stashed changes: `$ git stash apply` If unlucky, there will be conflict as your stashed changes are merged back onto your local repo branch (master). If so, you'll need to edit the file, deciding which line(s) is correct, and delete the version you do not want, along with the ASCII tags/labels.

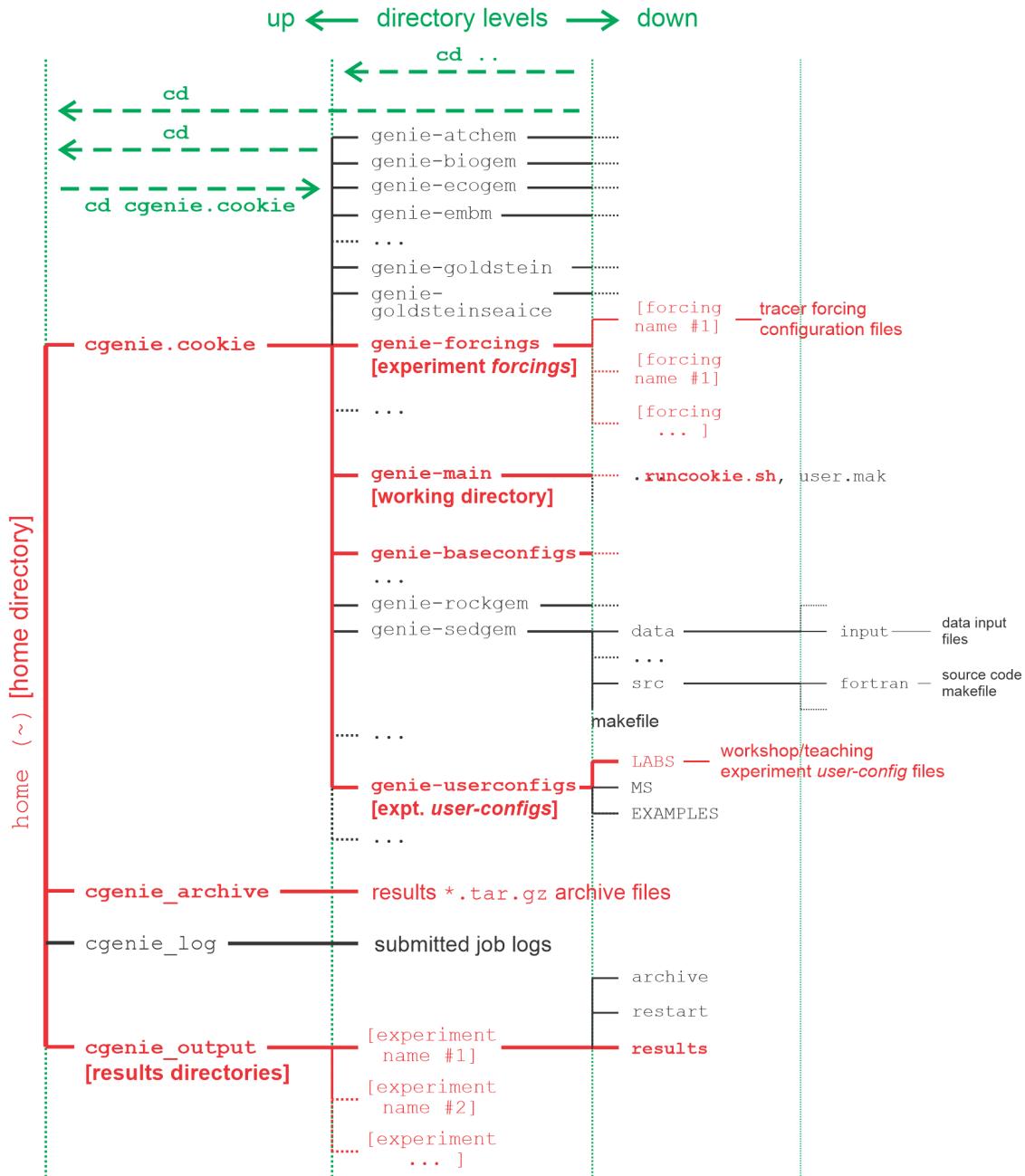


Figure 1: Directory structure of the **cookie** model. Highlighted in red are directories and sub-directories that you will need to access at some point. Vertical green lines designate directory levels, with example commands shown for moving between them.

0.3 Running the model

The overall sequence of configuring and running **cookie** (job submission to a cluster queue), is shown in Figure 2. Refer to this if in any doubt at any point.

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

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

(don't type it yet!) It requires that you must list at least 4 parameters after `./runcookie.sh`, separated by S P A C E S and on a single continuous line (even if it ‘wraps’ around across 2 lines of the screen). These parameters are:

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

As an example of running the **cookie** Earth system model:

- #1 : The *base-config* is: `cookie.CB.p_worbe2.BASES`
- #2 : The *user-config* directory is: LABS
- #3 : The *user-config* file (the experiment name) is: LAB.0.3.EXAMPLE.
- #4 : Run the experiment for ten years: 10
- #5 : (There is no restart file, and so no 5th parameter needs to be passed ...)

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

```
$ ./runcookie.sh cookie.CB.p_worbe2.BASES LABS LAB.0.3.EXAMPLE 10
```

(you can try it now!)

REMEMBER: This must be entered on a single CONTINUOUS LINE.

The (single) S P A C E S are vital.

ALSO – take care not to confuse an el (‘l’) with a one (‘1’) ... (it is a ‘one’ here).

What should happen is: First, you will end up twiddling your thumbs a while, as all the components of **cookie** are compiled from the raw source code (**FORTRAN**). When it has finished doing this, the model will initialize and carry out some brief self-checking. Only then will it start actually ‘running’ and doing something ... in this particular example, the output should look something like this:

```
*****
*** Initialisation complete: simulation starting ...
*****
>   model year | pCO2(uatm)  SAT(°C)  AMOC(Sv)  ice(%)  SST(°C)  SSS(PSU)  | pH  OHMEGA  | [O2] (uM) fPOC(PgC/yr)
>   0.5 | 285.042 -3.667 | 12.715  0.274  1.408  34.902 | 8.048  2.537  | 181.360 12.327
>   1.5 | 295.295 -2.531 | 13.189  2.126  3.450  34.902 | 8.079  2.896  | 191.671 12.636
>   2.5 | 302.383 -1.270 | 12.266  4.282  5.004  34.904 | 8.088  3.131  | 196.879 12.394
>   3.5 | 307.657 -0.274 | 11.020  5.525  6.214  34.906 | 8.093  3.318  | 199.930 12.216
>   4.5 | 311.667  0.628 | 9.831   6.037  7.185  34.908 | 8.096  3.469  | 201.861 12.096
>   5.5 | 314.811  1.437 | 8.876   6.387  7.987  34.910 | 8.099  3.597  | 203.157 11.994
>   6.5 | 317.299  2.150 | 8.091   6.628  8.667  34.911 | 8.101  3.707  | 204.047 11.899
>   7.5 | 319.290  2.782 | 7.872   6.673  9.256  34.913 | 8.103  3.800  | 204.669 11.822
>   8.5 | 320.852  3.342 | 7.886   6.734  9.773  34.914 | 8.105  3.882  | 205.073 11.742
>>> SAVING BIGGEM TIME-SLICE AVERAGE CENTERED @ year : 9.500
>   9.5 | 322.066  3.848 | 7.935   6.684  10.232 34.914 | 8.106  3.954  | 205.320 11.670
*****
*** Simulation complete: shutdown starting ...
*****
```

(Note that you might want to increase the width of your terminal window in order to get the values reported for each year appearing on the same line and not wrapped-over to the next one.)

What do all these numbers mean? From left-to-right:

model year – ... guess!

Then:

pCO_2 (uatm) -- mean atmospheric CO_2 concentration (in units of μatm)
 $\langle SST \rangle$ – global mean surface air temperature (‘SAT’) °C

Note that if you are not running an experiment without a carbon cycle, there will be no values for pCO_2 .

AMO (Sv) – Atlantic meridional overturning circulation (AMOC) (Sv)
 $ice(%)$ – global sea-ice fraction (%)
 $\langle SST \rangle$ – global sea surface temperature (‘SST’) °C
 $\langle SSS \rangle$ – global sea surface salinity ‘SSS’ (‰)

Note that if you are not running an experiment with a modern-like Atlantic basin, no value is given for the AMOC.

After that, a couple of carbonate chemistry indicators:

pH – Mean ocean surface pH.
OHMEGA – Mean ocean surface OHMEGA ...

which will become apparent later.

Note that if you are not running an experiment without a carbon cycle, no values will be given.

Finally:

- 02 – Dissolved oxygen in the ocean(global mean). ($\mu mol kg^{-1}$)
- fPOC – Global total annual export production (in terms of particulate organic carbon ($Pg Cyr^{-1}$)).

Note that you need some sort of biological pump in the ocean for these to be meaningful.

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

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

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

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

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

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

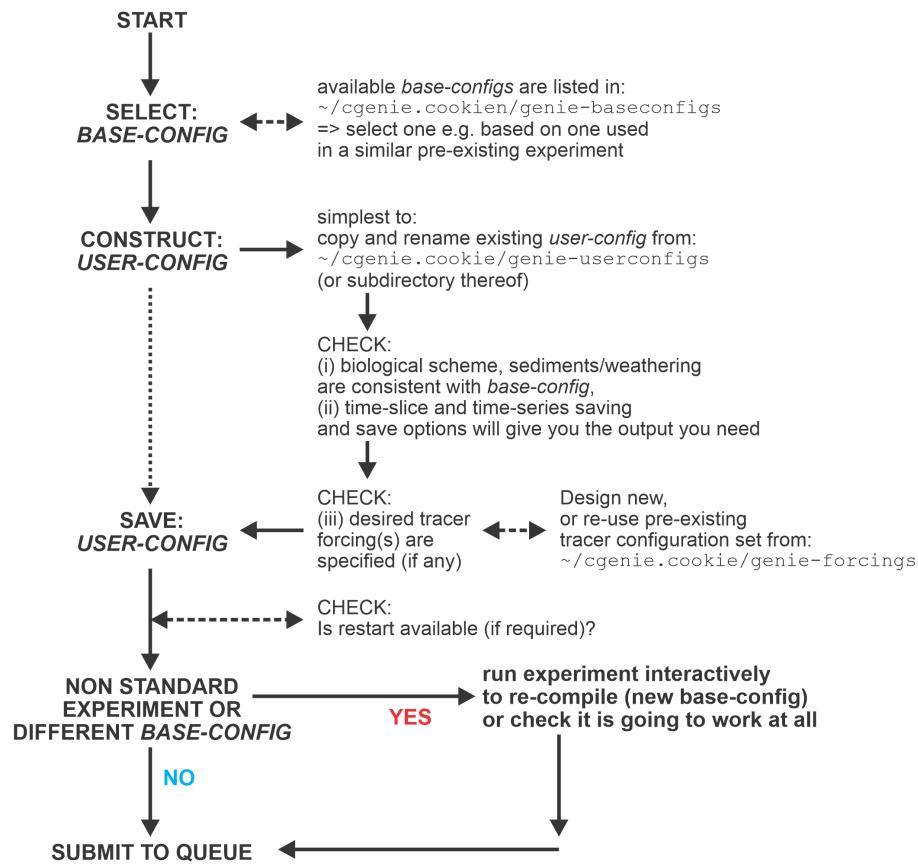


Figure 2: Schematic of the sequence-of-events in configuring and running an experiment.

0.4**Creating new experiments!**

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

There are two obvious ways to create a new *user-config* file and hence new experiment:

1. Create a blank text file ²⁷ and populate the contents with the parameter value assignments you need for your new experiment.

Inevitably, it is difficult to remember all the names and even values you want to specify, meaning that you'll end up looking at existing *user-config* files and copying and pasting lines (or even the entire contents) from the old file(s) into your new *user-config* file. So then you may was well ...

2. Copy and edit an existing file!

This is the most practical approach – pick an existing *user-config* file that is closest to the specific experiment that you want to run – copy and rename it, then edit the contents and save. For the purpose of trying this out, you can literally pick on any existing file in the `LABS` subdirectory (of `genie-userconfigs`).

You can do this by:

- Using your sftp client (program):

First – drag the existing *user-config* file to your local computer. On your local computer, rename the file as per how you would 'normally' rename any file.

Now you can edit the parameter values, re-save it, drag it back to the cluster (from your local computer) using the sftp client ... and finally run the experiment.

There are also ways of configuring **sftp** clients so that you can double-click on a file in the remove window, edit it, and save it back to the remote computer.²⁸

- Or, if you are comfortable working at the linux command line; working from the same directory (e.g. `LABS`) that the file you want to copy lives in, you can copy a file (to a new filename) by:

```
$ cp oldfile newfile
```

(after which you can edit the new *user-config* file `newfile`, re-save, and then run the experiment.)

Remember that the new *user-config* file needs to be saved in (or copied to) the `genie-userconfigs/LABS` sub-directory (in the case of experiments carried out as part of the tutorials described in this manual), or `LABS` itself or any sub-directory (or sub-sub- etc directory) ... as long as you specify the correct path to the directory where you save the new file to²⁹.

²⁷ \$ `touch file.txt` will achieve this at the linux command line.

²⁸ Actually what happens is that the file is transferred locally, opened, and you edit a local copy. When you save it (locally), it is automatically transferred back.

²⁹ See previous Section and the use of the 2nd parameter passed to `runcookie.sh`

0.5 Model output

Experiment results are saved in a single sub-directory of the experiment results directory (along with a sub-directory for *restart* files, and one for copies of your parameter choices). The directories containing the results of all your different experiments live in:

```
~/cgenie_output
```

and will be assigned a directory name something like:

```
LAB.0.3.EXAMPLE
```

(this being the results directory name for an experiment called LAB.0.3.EXAMPLE) and the actual results live in a sub-directory of this – results, hence:

```
~/cgenie_output/LAB.0.3.EXAMPLE/results
```

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

cGENIE.cookie has a flexible and powerful facility of saving results by means of spatially explicit ‘time-slices’, and as a semi-continuous ‘time-series’ of a single global (or otherwise representative mean) variable, described as follows.

0.5.1 Time-slice output

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

For example, to save regularly every 10 years you would set:

```
bg_par_data_save_slice_timeinterval=10.0
```

and which would give you save points at:

```
9.5  
19.5  
29.5  
39.5  
...
```

(the mid-points of averages made over the intervals: 9-10, 19-20, 29-30, 39-40 years, etc.). Otherwise, you can specify the name of a file containing times to save at, or simply only save at the end. (see later)

0.5.2 Time-series output

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

The much smaller and simpler (text) file format now allow you to save much more frequently. For example, for experiments up to a few 100 years, you could save every single year, which you would set by:

```
bg_par_data_save_sig_timeinterval=1.0
```

Otherwise, you can specify the name of a file containing times to save at. (see later)

0.5.3 File naming convention

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

- fields_biogem_2d.nc – 2-D fields of ocean and atmosphere properties, as **NetCDF**.
 - fields_biogem_3d.nc – 3-D fields of ocean properties, as **NetCDF**.
 - timeseries_*.txt – these are the time-series files (in ASCII / plain text format).
 - SUMMARY_AT_year_*_diag_GLOBAL.txt – these contain (global diagnostics) summary information and are saved at the same frequency as the time-slices (also as ASCII / plain text).
-

Alternative file naming convention ...

Having the e.g., 2D and 3D **netCDF** files always called the same name (fields_biogem_2d.nc, fields_biogem_3d.nc) in each and every experiment, has the potential to get confusing if you have multiple experiments open simultaneously in a viewer such as **Panoply**.

You can specify that **netCDF** files are named following the name of your experiment, by adding the following line to your *user-config*:

```
bg_ctrl_ncout_expid_name=.true.
```

0.6 Viewing model output

0.6.1 Time-series output

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

- timeseries_atm_humidity.res - mean atmospheric (surface) humidity
- timeseries_atm_temp.res - mean atmospheric (surface) air temperature
- timeseries_misc_opsi.res - overturning stream-function (e.g. AMOC) strength
- timeseries_misc_seaice.res - mean ocean sea-ice cover and thickness
- timeseries_ocn_sal.res - mean ocean surface and whole ocean salinity
- timeseries_ocn_temp.res - mean ocean surface and whole ocean temperature

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

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

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

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

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

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

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

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

0.6.2 2- and 3-D time-slice output

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

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

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

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

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

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

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

To save plots in **Panoply**:

File

Save Image As ...

Then select the location, filename, and graphics format.

0.7 Submitting experiment ‘jobs’

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

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

1. The connection between your terminal and the server computer running the model must remain unbroken. Anything more than a fleeting loss of internet connectively may result in the experiment terminating.
 2. You can only run one experiment at a time ... unless you want to have thousands of separate terminal open ... ? I thought not ...
 3. Any cluster or computer you are likely to be accessing using a shell will not have many computing cores itself, either because it is a single machine with only one or two processors, or if a cluster, by using a terminal you are running on the ‘head node’, which will have similar computing core limitations to running on a single machine. The more experiments you run simultaneously, the slower they will all run ...
-

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

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

```
$ qsub -q QUEUE.q -j y -o cgenie_log -V -S /bin/bash
runcookie.sh cookie.CB.p_worbe2.BASES LABS LAB.0.3.EXAMPLE 100
```

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

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

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

After submitting an experiment, you receive a job number. This number appears in the first column in the queue status information when you issue a `qstat -f` command. You should see your job appear on one of a number of compute nodes, perhaps numbered 0-0 through 0-5), although it might briefly reside as a ‘PENDING JOB’. For each node, there are multiple processing cores (depending on the specific cluster and queue), meaning that multiple instances of `cookie` can run simultaneously on each node. For an 8-level ocean based configuration of `cookie`, being run for 100 years, the job should remain there in the queue for a few minutes before ‘disappearing’ (your clue that it has finished, or died³⁰ ...). If you periodically re-issue a `qstat -f` command you can follow your job’s progress.

³⁰If your experiment appears on the queue but vanishes after a few seconds, it has most likely died :(

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

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

0.8 'Restarts'

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

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

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

```
$ wget --no-check-certificate http://www.seao2.info/cgenie_output/
cookie.CB.p_worbe2.BASES.ridgwelletal.SPIN.tar.gz
```

(all one line!)

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

```
$ tar xfzv cookie.CB.p_worbe2.BASES.ridgwelletal.SPIN.tar.gz
```

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

A *re-start* can be requested for running on a new experiment from the end of a previous one, by providing a 5th (optional) parameter when entering in the `runcookie.sh` command. A spin-up of the modern World climate state is provided for you as a *restart* – `cookie.CB.p_worbe2.BASES.ridgwelletal.SPIN` – which you have just unpacked to the cgenie_output results output directory.

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

```
~/cgenie.cookie/genie-userconfigs/LABS
```

taking the file `LAB_0.3.EXAMPLE` (which is provided) as a template (no parameter changes need to be made yet). As described earlier – copy this file and give it a new name – here, name it: `LAB.0.3.NEW`

You can then specify the use of the *restart* in your new LAB_0.3.NEW experiment:

```
$ ./runcookie.sh cookie.CB.p_worbe2.BASES LABS LAB.0.3.NEW 100  
cookie.CB.p_worbe2.BASES.ridgwelltal.SPIN
```

The run-time output should now look noticeably different. There should be no (or perhaps just very little) drift in any of the various variable values outputted to the screen – this is because you have (re-)started from the end of a run that had already ready an equilibrium (or close).

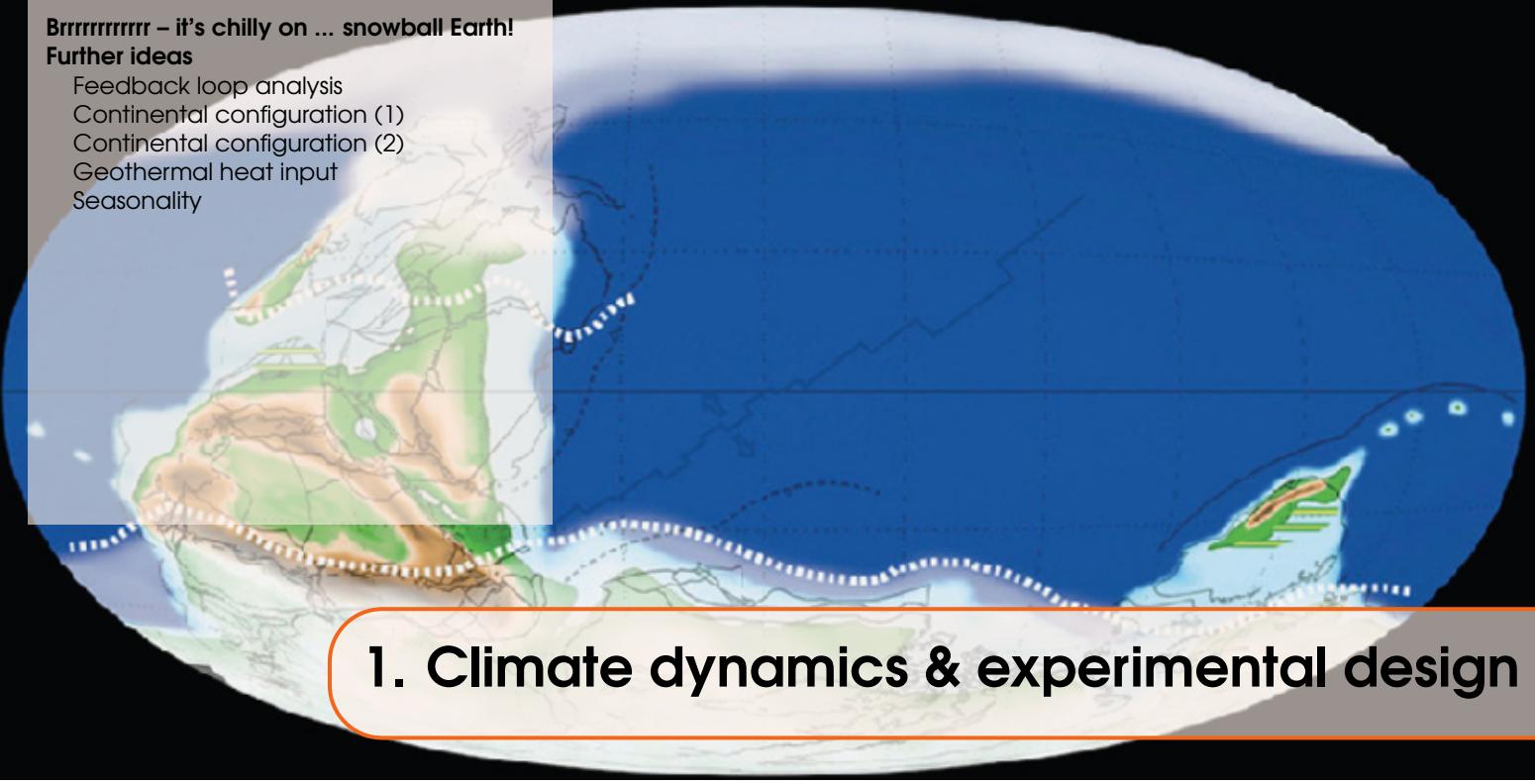
Because LAB.0.3.EXAMPLE (and hence your copy, LAB_0.3.NEW) does not prescribe a value of pCO_2 but instead allows it to 'float', your experiment is a good test of whether the carbon cycle in experiment

cookie.CB.p_worbe2.BASES.ridgwelltal.SPIN was at steady state.

Brrrrrrrrrrr – it's chilly on ... snowball Earth!

Further ideas

- Feedback loop analysis
- Continental configuration (1)
- Continental configuration (2)
- Geothermal heat input
- Seasonality



Stuff to keep in mind:

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

Readme

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

```
$ wget --no-check-certificate  
http://www.seao2.info/cgenie_output/cookie.C.p_woreq1.NONE.SPIN.tar.gz
```

(all one line)

This downloads an archived/compressed copy of the experiment cookie.C.p_woreq1.NONE.SPIN – effectively, just an experiment (spin-up) that has already been run for 5,000 years for you. Extract the contents of this archive by typing (also from the cgenie_output directory):

```
$ tar xfzv cookie.C.p_woreq1.NONE.SPIN.tar.gz
```

A new experiment results directly will then appear as if you had just run the entire 10,000 year experiment yourself (and you could in fact have done so). Remember to refresh the **WinSCP** directory view window if you are using this particular software, or it might appear that nothing has been extracted.

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

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

1. That you in fact downloaded it to the correct directory, which should be: cgenie_output, and not randomly to e.g. your home directory ...
2. That you unpacked it.

1.1 Brrrrrrrrrr – it's chilly on ... snowball Earth!

To illustrate how ‘easy’ it can be to configure an Earth system / climate model such as **cookie** and explore the behavior of the Earth system and its response to perturbation – you are going to induce an extreme cooling of the climate system and see what happens.

Solar output was weaker during the late Neoproterozoic, a time when the Earth experienced a series (2-ish) of extreme glaciations. Thus, having a mild climate state to start with must have been dependent on sufficient CO_2 and/or CH_4 in the atmosphere which presumably must have been highly elevated compared to the modern World ... sort of opposite to the problem we have today ...

You are going to be running experiments in a similar manner to before, and using the *re-start* experiment that you downloaded. You could then, for example, take the experiment configuration file LAB.1.1.EXAMPLE (which is provided for you), and run the experiment, which has a new *base-config*: cookie.C.p_woreq1.NONE, by typing:

```
$ ./runcookie.sh cookie.C.p_woreq1.NONE LABS LAB.1.1.EXAMPLE 100
cookie.C.p_woreq1.NONE.SPIN
```

(All one line ... but remember there will need to be a space separating the experiment duration and the name of the *restart*).

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

So – copy and rename the file LAB.1.1.EXAMPLE to ... LAB.1.1.EXPT or whatever (in the subdirectory LABS), and then run your newly created experiment:

```
$ ./runcookie.sh cookie.C.p_woreq1.NONE LABS LAB.1.1.EXPT 100
cookie.C.p_woreq1.NONE.SPIN
```

Remember that because you have changed *base-config* since the exercises in the previous chapter, running the model first interactively (at the command line) is essential in order for the model code to be re-compiled consistent with the new configuration. i.e. you cannot directly start submitting experiments using the *base-config* cookie.C.p_woreq1.NONE, straight-away after having previously run the model using the cookie.CB.p_worbe2.BASES *base-config*. Once you have re-compiled the model with the new *base-config* and starting running (any experiment), you need not re-compile again and can submit as many jobs to the cluster queue as you like, up until you change *base-config* once again.

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

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

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

Note that the model configuration of an idealized super-continent you are using and as defined by the `cookie.C.p_woreq1.NONE base-config` file, positioned symmetrically about the Equator, is pretty unrealistic. But the further you go back in time, the more uncertain it becomes as to exactly where and in what orientation the continents were. Sometimes modelers have to resort to somewhat idealized experiments if the uncertainties are too great. In addition, one can conduct sensitivity experiments to test whether the continental configuration is important to the results. For instance, Hoffman and Schrag [2002] discuss the potential importance of continental configuration, while the hypothesis of Donnadieu *et al.* [2004] rests on specific details of the continental configuration.

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

```
ma_genie_solar_constant= 1285.92
```

which is set at the top of the provided *user-config* file `LAB.1.1.EXAMPLE`. (For reference, the modern value is 1368Wm^{-2} .)

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

Near the top of the file you should see something like:

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

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

PARAMETER=VALUE

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

`ea_radfor_scl_co2=20.0`

specifies a radiative forcing of climate by CO_2 equivalent to x20 modern ($20 \times 278 = 2560$ ppm). If you instead wrote:

`ea_radfor_scl_co2=1.0`

this would give you a modern¹ (x1, or $1 \times 278 = 278$ ppm) radiative forcing. Note: CO_2 is not being explicitly modeled in this experiment², but the long-wave radiative forcing associated with a specified concentration of CO_2 (in ratio to modern) is being set instead.

¹Technically: the pre-industrial CO_2 value rather than ‘modern’ *per se*.

²And hence the parameter:

```
# set no CO2 climate feedback  
ea_36=n
```

which tell cookie to ignore the concentration of CO_2 in the atmosphere, in case, as here, there isn't one defined ...

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

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

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

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

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

In answering the question ('How low does CO_2 have to be to trigger a 'snowball'?'), think about what an appropriate degree of precision (rather than accuracy) might be for your experiments. Just because computer models generally calculate to around 16 significant places of precision, does not mean you have 16 significant figures of realism. For instance – how many significant figures is the solar constant quoted to and what do you think is the uncertainty in this? Harder to judge is how the assumed (incorrect) continental configuration creates additional uncertainty, or the simple physics assumed in the ocean or sea-ice, or lack of snow on land ...

Other questions to think about with regards to numerical modeling are:

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

Associated with the question of 'precision' is: '*How long do I have to run the experiment for?*'

This also has a vague-to-no answer. It depends on the science question and what you might judge to be appropriate precision in the context of the various uncertainties ... In other words – if you can define a sufficient or minimum precision of some property of the system, you only need to run the model as long as needed to achieve this. However, to determine the precision of the model at any point in time, you need to know the final or equilibrium value. So ideally you would run the model once, for much longer than you think you need to, and then determine how as a function of time, precision increases and error between current state and final steady state decreases.

Here I am crudely equating the concept precision with how far the model is from steady-state. You could think about it in terms of accuracy if you assumed that the steady-state of the model was 'perfect' and deviations from that steady-state (i.e. incomplete spin-up) equates to model 'error'.

Once you are happy with determining snowball threshold, try and answer the associated question:

How high does the (CO₂) radiative forcing have to be in order to escape from a snowball?

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

Note that a good *re-start* is one for which the experiment did not sit too long in the snowball state before finishing (the more sea-ice thickness you create in the first experiment, the more you are going to have to melt in the next and hence the longer this new series of experiments will take to ruin...). To achieve this, you can fine-tune the number of years the experiment is run, i.e. having determined the appropriate radiative forcing value required to create a snowball state, find out when (in terms of number of years) in the experiment the snowball state first occurred, and then run a new experiment that finishes only a decade or so after the snowball state was initiated.³

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

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

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

³You cannot select when the *re-start* is saved – it is always saved at the end of an experiment.

1.2 Further ideas

1.2.1 Feedback loop analysis

To quantify the snowball Earth hysteresis loop in **cookie** as per Figure 2 in *Hyde et al.* [2000] you will need to extract from the model ‘meaningful’ measures of climate (e.g., global surface air temperature, fractional sea-ice coverage) as a function of CO_2 multiples, CO_2 concentration, or (better) radiative forcing. For the latter, in **cookie**, the radiative forcing for a doubling of CO_2 is set at: $5.77Wm^{-2}$. See: *Myhre et al.* [1998] (*Geophys. Res. Lett.* 25, 2715–2718) and/or *IPCC* [2007] for more on what radiative forcing is and how it is related to a relative change in CO_2 concentration. Also, for making a comparison with *Hyde et al.* [2000] – for going into the snowball, note that they plot the change in radiative with a ‘cooling’ as positive (a bit daft). Their baseline radiative forcing state (an anomaly of $0Wm^{-2}$) you might assume is equivalent to 278 ppm and hence ~ 130 ppm is an approximately halving of CO_2 and hence creates $\sim 5Wm^{-2}$ of cooling. (You might prefer to plot the radiative forcing change as warming being positive, which makes rather more sense ...)

For coming out of a snowball, because the CO_2 and hence radiative forcing threshold is so high as compared to going in, you may want to be creative in the plotting (assuming attempting to combine both thresholds into a single plot) and, for instance, one might break the scale between the low radiative forcing interval spanning going in and the high one spanning coming out.

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

1.2.2 Continental configuration (1)

It was mentioned earlier that the position of the continents is an area of modeling uncertainty and might be important. You can test for this. Four alternative *base-configs* are provided, each defining a different continental configuration (all with an ocean resolution of 36x36 with 8 vertical levels):

1. `cookie.C.p_worpl1.NONE`
– a single polar super-continent. (Note potential ‘l’ and ‘l’ confusion in ‘p_worpl1’.)
2. `cookie.C.p_worpl2.NONE`
– one continent at each pole.
3. `cookie.C.p_woreq1.NONE`
– a single Equatorially-centred super-continent. [this is the current configuration (that you used previously)]
4. `cookie.C.p_woreq2.NONE`
– two continents straddling the Equator.

You can use the given *user-config* file (LAB.1.1.EXAMPLE) again as an experiment template file, and any of the alternative configurations can be run very similarly to as per before, i.e.:

```
$ ./runcookie.sh cookie.C.p_xxxxxx.NONE LABS LAB.1.1.EXAMPLE 100
```

Note that you are using a different *base-config* file name: `cookie.C.p_xxxxxx.NONE` where `xxxxx` is one of: `worpl1`, `worpl2`, `woreq1`, or `woreq2`.

Also note that no re-starts are provided for any of these configurations. You may (or may not) want to create some (you will need to judge for yourselves how long to run the *re-start* experiments to achieve as close to steady-state as you think is ‘sufficient’). Recall again, that *re-starts* are just ‘normal’ experiments that have already been run. Be careful that when changing from one *base-config*

to another, the model re-compiles. Simply running the new configuration briefly (for even just a single year) is sufficient to ensure this. Experiments can then be safely submitted to a cluster queue, i.e. do not try and submit an experiment using a different *base-config* straight to the cluster queue without having run it (or a short version of the experiment you want) interactively first (to ensure the model is re-compiled). This is also good practice – checking that a new sort of experiment and/or model configuration works as you intend and without hiccups.

1.2.3 Continental configuration (2)

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

The *base-config* is named: cookie.C.fm0635cb.NONE

The *user-config* is: cookie.C.fm0635cb.NONE.SPIN

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

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

```
$ wget --no-check-certificate
http://www.seao2.info/cgenie_output/cookie.C.fm0635cb.NONE.SPIN.tar.gz
```

and unpack by:

```
$ tar xfzv cookie.C.fm0635cb.NONE.SPIN.tar.gz
```

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

To run (e.g. for 100 years), following on from its *re-start* (and copying (/renaming) the *user-config* to the LABS directory):

```
$ ./runcookie.sh cookie.C.fm0635cb.NONE LABS cookie.C.fm0635cb.NONE.SPIN 100
cookie.C.fm0635cb.NONE.SPIN
```

(all on one line)⁴

1.2.4 Geothermal heat input

Finally, **cookie** will fairly happily build up sea-ice, apparently without limit (with the remaining wet ocean becoming progressively colder and more saline). In the real world, one might expect some sort of limit to the maximum thickness achieved as the heat diffusion across a progressively greater thickness of sea-ice approaches the heat input at the bottom of the ocean from geothermal energy. Different modes of ocean circulation are also possible if one considers heating from the bottom as well as cooling (and brine rejection) from the top and which might affect the entry into or exit from a snowball state.

⁴NOTE that the *base-config* and *user-config* filenames start cookie.C. whereas the *re-start* filename starts cookie.CB. ... just to try and trip you up ...

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

```
bg_par_Fgeothermal=100.0E-3
```

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

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

1.2.5 Seasonality

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

```
ea_dosc=.true.  
go_dosc=.true.  
gs_dosc=.true.
```

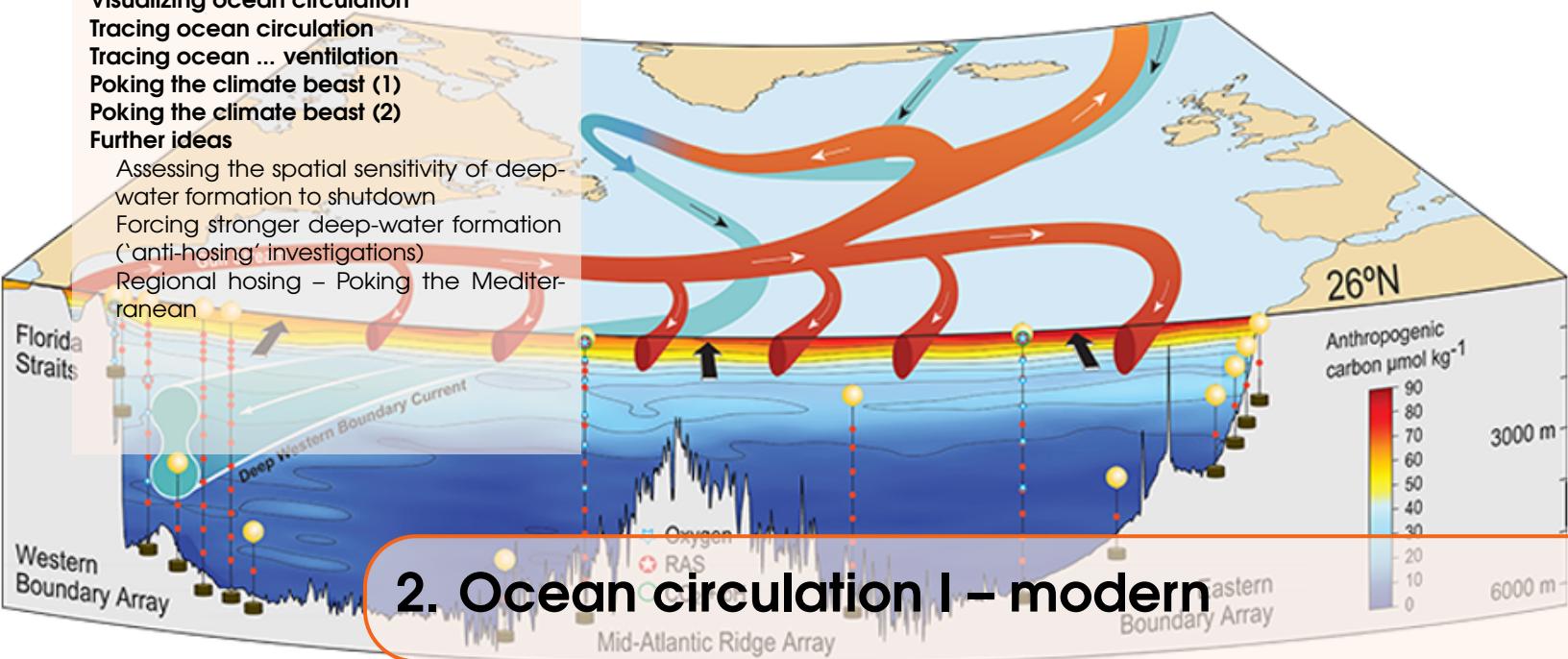
The scientific question here in trying this would be whether or not taking into account a seasonally-varying climate substantially affects the entry (and/or exit) thresholds for a snowball climate state. (At least, whether it is important in the context of the resolution and physics of the model you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

You can also save the data seasonally if you like – see Section 12.2.3 in the **cookie** User Manual (this document!).⁵

⁵For reference, your configuration has 24 time-steps per year set for the **BIOGEM** module.

Simulating ocean circulation
Visualizing ocean circulation
Tracing ocean circulation
Tracing ocean ... ventilation
Poking the climate beast (1)
Poking the climate beast (2)
Further ideas

Assessing the spatial sensitivity of deep-water formation to shutdown
Forcing stronger deep-water formation
(‘anti-hosing’ investigations)
Regional hosing – Poking the Mediterranean



2. Ocean circulation I – modern

Stuff to keep in mind:

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

Background reading (Atlantic circulation and stability in **muffin**):

- Hargreaves et al. [2004] (Climate Dynamics 23, 2004, Pages 745 – 760)
→Simple assessment of the likelihood of AMOC collapse.
- Marsh et al. [2004] (Climate Dynamics, 23 2004, Pages 761 – 777)
→Characterization of thresholds of AMOC collapse.
- Singaray et al. [2008] (GRL 35, doi:10.1029/2008GL034074)
→Role of changing ocean circulation in atmospheric radiocarbon variability during the Younger Dryas.

Background reading (Miscellaneous (model) Atlantic circulation and stability):

- IPCC [2007] (e.g., Section 10.3.4)
→Future predictions of AMOC strength.
- Schmittner [2005] (Nature 434, 628– 633)
→Impacts on marine ecosystems and carbon cycling.
- Obata [2007] (J. Clim. 20, 5962–5976)
→Climate-carbon cycle model response to freshwater discharge.

READ.ME

You will need to download new *re-start* files prior to embarking on the experiments with modern ocean circulation. To fetch them: change to the cgenie_output directory, and type (or copy and paste carefully from the PDF ...):

```
$ wget --no-check-certificate http://www.seao2.info/cgenie_output/
  cookie.C.p_worjh2.r.SPIN.tar.gz
$ wget --no-check-certificate http://www.seao2.info/cgenie_output/
  cookie.C.p_worjh2.rb.SPIN.tar.gz
```

(single line ... no space between line fragments ...) This downloads archived/compressed copies of the 10,000 year *spin-up* experiments: cookie.C.p_worjh2.r.SPIN and cookie.C.p_worjh2.rb.SPIN. Extract the contents by typing:

```
$ tar xfzv cookie.C.p_worjh2.r.SPIN.tar.gz
$ tar xfzv cookie.C.p_worjh2.rb.SPIN.tar.gz
```

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

2.1 Simulating ocean circulation

Important.

1. Note that the configurations provided for this exercise do not have a carbon cycle, but do include a numerical 'age' tracer to help diagnose ocean circulation and rate of ventilation of the deep ocean.
 2. Make use of *control* experiments in order to create anomaly maps (see later). An ideal control experiment is one that is run for as long as your 'real' (perturbation) experiments, but has no changes made to the *user-config* (compared to the *re-start*).
-

To start off – run a *control* experiment using the *re-start* experiment that you have just downloaded as the initial state of your experiment. A *control* experiment is basically one in which you do not change any parameter settings and is used top quantify any residual drift (that you might otherwise interpret as the result of a parameter change in your actual experiment). The template *user-config* file you are provided for, for this, is: LABS.2.1.EXAMPLE

Run the model for ... whatever, 10 years will do for now::

```
$ ./runcookie.sh cookie.C.p_worjh2.r LABS LABS.2.1.EXAMPLE 10  
cookie.C.p_worjh2.r.SPIN
```

where cookie.C.p_worjh2.r.SPIN is the *re-start* experiment name.

You will analyze the results of this in the next couple of Sections.

NOTE: For the initial steps, you will be using a seasonally-forced modern continental configuration of cGENIE with 16 vertical levels in the ocean. This is indicated by the code 'p_worjh2' in the *base-config* file-name. The grid and definition of the vertical levels are illustrated in Figure 2.1) and (Figure 2.2), respectively.

36	10	09	08	09	10	11	11	10	08	07	07	07	07	07	07	07	11	13	14	14	12	12	93	93	91	91	09	05	05	08	10	11	11	11	12	12	11	11
35	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94
34	94	94	94	94	94	92	92	08	07	10	15	12	09	93	93	94	94	94	91	91	91	06	05	05	06	09	12	93	93	94	94	94	94	94	94	94	94	
33	94	91	91	91	16	09	04	03	04	05	03	03	06	93	93	91	91	91	91	91	09	05	03	04	06	93	93	93	94	94	94	94	94	94	94	94		
32	94	91	91	91	10	04	01	01	01	01	01	01	04	11	93	91	91	91	91	91	12	05	03	03	03	11	93	93	94	94	94	94	94	94	94	94		
31	94	91	91	11	05	01	01	01	01	01	01	01	03	08	93	93	91	91	91	91	12	06	03	03	02	09	93	93	94	94	94	94	94	94	94	94		
30	91	91	91	07	02	01	01	01	01	01	01	01	02	05	93	91	91	91	91	91	09	03	01	01	04	02	93	10	94	94	94	94	94	94	94	94		
29	91	91	91	08	02	01	01	02	01	01	01	01	01	03	93	93	92	92	92	92	06	01	01	02	04	03	02	12	10	08	08	11	94	94	94	94	94	
28	91	91	16	05	01	01	01	01	01	01	01	01	01	10	93	92	92	92	92	92	05	01	01	03	01	03	01	93	93	12	11	94	94	94	94	92	92	
27	91	91	09	03	01	01	01	01	01	01	01	01	02	02	07	91	13	09	04	01	01	02	01	01	05	93	93	94	94	94	94	92	92	92				
26	91	91	05	02	02	01	01	01	02	01	01	01	02	05	91	08	07	04	01	01	02	01	01	09	93	93	94	94	94	91	16	09	92	92	92			
25	91	08	02	01	02	01	01	02	02	01	01	01	02	04	91	09	08	05	02	01	02	01	12	93	93	94	94	94	91	14	06	92	92	92				
24	93	06	02	01	02	01	01	02	01	01	01	01	02	03	06	91	10	06	03	01	03	01	02	12	93	93	94	94	94	91	09	03	92	08	11			
23	91	05	03	01	02	01	01	01	01	01	01	01	01	02	02	04	06	91	05	04	01	02	01	10	93	93	94	94	94	91	05	02	11	05	09			
22	93	06	04	01	02	01	01	01	01	01	01	01	01	02	02	03	03	04	07	93	08	03	01	01	07	93	93	94	94	94	91	03	02	09	04	08		
21	91	07	05	02	03	03	02	01	01	01	01	01	02	03	03	03	07	91	91	07	02	03	02	05	93	93	93	93	91	91	03	02	06	03	08			
20	93	09	04	02	02	03	02	01	01	02	02	02	02	03	03	03	05	91	91	11	04	02	02	03	05	08	93	93	91	11	02	02	04	03	07			
19	91	91	04	03	03	03	02	01	01	01	02	02	02	02	04	04	05	91	91	91	05	03	02	02	04	04	93	93	91	07	02	02	03	02	04			
18	93	91	07	08	07	05	03	01	01	01	02	02	02	02	03	04	05	91	91	91	08	04	01	02	03	03	93	93	91	04	02	02	02	01	02			
17	09	91	07	10	10	07	04	01	01	01	01	02	02	02	03	03	03	91	91	91	91	05	01	02	02	02	93	93	91	03	03	04	02	01	01			
16	04	93	07	14	10	05	03	01	01	02	01	01	02	02	03	03	02	02	91	91	91	91	05	01	03	02	01	93	91	91	03	03	04	01	01	01		
15	01	03	92	94	10	04	03	03	02	01	02	02	03	03	02	02	91	91	91	91	04	01	03	02	01	93	91	91	04	03	04	01	02	01				
14	01	02	93	91	09	04	04	04	04	01	02	03	03	03	03	02	92	91	91	91	03	01	03	01	01	93	91	91	07	03	03	02	02	01				
13	01	04	93	91	13	06	04	04	04	01	02	03	03	03	02	02	04	93	91	91	03	01	03	01	01	93	93	91	09	03	03	01	01	01				
12	01	08	93	91	91	08	06	04	03	01	02	02	03	04	02	02	02	93	91	91	03	01	03	01	02	93	93	10	08	02	03	02	01	01				
11	01	10	93	93	91	07	05	03	02	01	02	02	03	04	03	03	03	93	91	91	09	02	01	02	10	93	07	05	01	02	02	01	02	01				
10	01	09	93	92	91	05	06	03	01	01	02	02	03	04	03	03	03	93	91	06	02	01	03	02	02	07	93	05	04	01	02	03	03	02				
09	02	08	92	91	04	06	04	01	01	01	02	02	03	03	03	03	93	91	04	02	03	02	02	06	93	04	03	02	02	03	04	03	02					
08	01	03	03	05	92	03	05	07	01	01	01	02	03	03	03	03	04	93	09	02	02	03	02	01	03	05	02	03	02	03	02	03	02					
07	02	01	01	02	07	02	04	10	02	01	01	01	02	03	03	03	06	93	05	01	01	02	04	02	01	01	02	02	04	02	02	03	03	03				
06	03	02	01	01	04	02	04	12	03	02	01	01	02	04	03	02	03	06	93	03	01	01	02	03	02	01	03	01	05	02	02	04	03	03				
05	03	03	02	02	04	02	05	07	02	01	01	02	03	04	02	02	08	93	04	01	01	03	04	03	02	01	02	04	03	04	06	03	03					
04	02	02	02	03	03	04	03	01	01	02	03	03	03	02	01	02	05	05	04	03	02	02	04	04	03	02	02	01	03	05	03	02						
03	02	01	01	02	03	03	01	01	02	03	03	02	02	02	02	03	05	09	07	04	02	01	01	02	01	01	01	01	01	02	04	03	03					
02	94	10	09	09	08	07	06	03	04	03	03	02	02	02	02	03	05	09	07	04	02	01	01	02	03	03	04	06	06	06	08	08	04					
01	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94					

Figure 2.2: The cookie ocean vertical level definitions for a modern 16-level ocean grid.

k	mid depth (m)	base of layer (m)
16.00	38.91	60.84
15.00	126.04	174.75
14.00	227.26	283.85
13.00	344.84	410.58
12.00	481.44	557.80
11.00	640.12	728.83
10.00	824.45	927.51
9.00	1038.59	1158.31
8.00	1287.35	1426.43
7.00	1576.33	1737.90
6.00	1912.04	2099.73
5.00	2302.02	2520.05
4.00	2755.05	3008.34
3.00	3281.33	3575.57
2.00	3892.71	4234.52
1.00	4602.92	5000.00

2.2 Visualizing ocean circulation

Visualizing the 3D flow (/transport) of the ocean, much less the rate at which this occurs, is no trivial matter. Even with the aid of a model. (Or rather, the problem then becomes: how to visualize ocean circulation in a model.) We'll consider two different ways of analysing model velocity fields first – simply utilizing the *restart* that you have downloaded and unpacked, and then later take a more pro-active approach in subsequent Sections with some new experiments.

The first approach we can take is to simply visualize the raw velocity fields, but plotted as ocean currents.

1. In the 3D **netCDF** file, the three components of ocean velocity are represented by the variables: ocean velocity – u (Eastwards), ocean velocity – v (Northwards), and ocean velocity – w (upwards). 2. Open up velocity – u. Choose ‘lon-lat’.
2. Select/highlight velocity – v. and click on the ‘Combine Plot’ icon (as per before).
3. Rather than a difference map, which is what you get by default, i.e., ‘Array 1 – Array 2’ – from the drop-down menu (next to the ‘Interpolate’ button) select ‘Vector Magnitude’.
4. You should have a color contoured (or not if you prefer plotting without contouring on) map of ocean current speed, with velocity vectors (direction and magnitude) overlain. You'll need to re-scale the velocity vectors to properly see them – from the ‘Contours and Vectors’ tab – change the ‘Scale Length’ to e.g., 0.1. (On a Mac, look under the ‘Vectors’ tab for a ‘Reference Value’ to something like 0.1.) When fresh-water hosing – look out for impacts on the N. Atlantic current system associated with the AMOC.
5. You can repeat this for deeper depth levels in the ocean – e.g., between about 1500 and 2000 m is a good place to go looking for the Western boundary current (and AMOC return) in the model (such as it exists at this low resolution) but you'll need to re-scale the velocity vectors again (e.g., to 0.01 to less).

An example plot (using **Panoply** for visualizing surface ocean current fields) is given in Figure 2.3.

The second approach is to visualize the large-scale ocean transport in terms of the meridional overturning circulation (‘stream-function’) (e.g. see background literature).

Two example plots (using **Panoply**) are shown in Figure 2.4 for the Atlantic basin, and Figure 2.5 for the Pacific. In the In the 2D **netCDF** file, relevant fields (netCDF variables)¹ are:

```
phys_opsia == global overturning stream-function
phys_opsip == overturning in the Atlantic
```

¹Note that these fields are only meaningful for the modern arrangement of the continents and a continuous separation of the Atlantic from the Pacific from high northern latitudes down to the tip of South America. Different (e.g. paleo) arrangements of the continents may not have recognisable (or definable) Atlantic and/or Pacific basins and it may only be possible to define and visualize the global meridional overturning circulation – variable: phys_opsi.

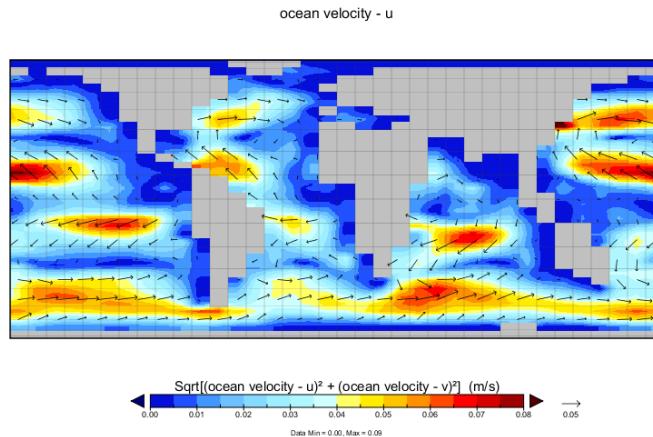


Figure 2.3: Example plot of (normal/default modern) ocean current fields (3D netCDF file). Again scaling has been set manually to create an easy-to-interpret axis scale. On the left is the surface field, and on the right an intermediate depth (illustrating what approximates the Deep Western Boundary current in the model in the Atlantic).

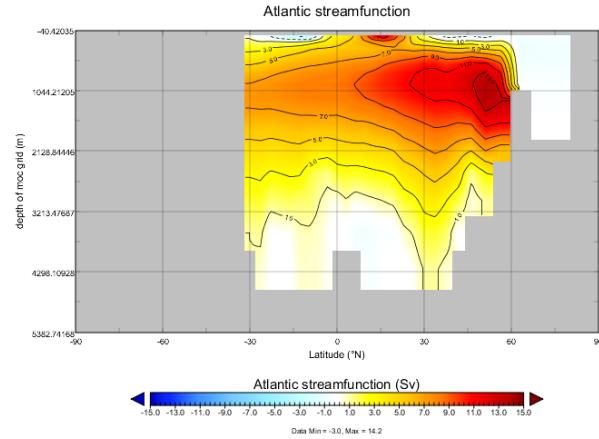


Figure 2.4: Example plot of (normal/default modern) overturning stream-function (2D netCDF file). (e.g., for Atlantic: **netCDF** parameter name: phys_opsia, long-name: Atlantic stream-function). Note that auto-scaling has been turned off and the min and max plotting limits set manually. By convention, stream-functions are plotted with their scale symmetrical around zero, giving red and ‘warm’ colors for positive value and clockwise overturning, and blues and ‘cold’ colors for negative values and anti-clockwise overturning. (The plot has been tart-ed up by overlaying solid contours plus contour labels.) It may be necessary in **Panoply** to re-orient (invert) the vertical grid.

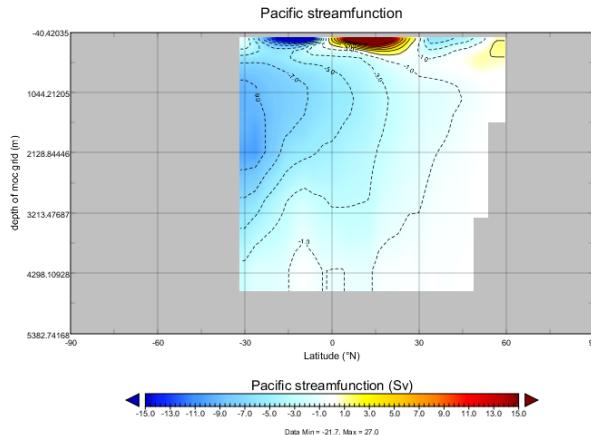


Figure 2.5: Pacific meridional overturning circulation (PMOC).

2.3 Tracing ocean circulation

The ocean biogeochemistry module (**BIOGEM**) in **cookie** provides a framework for applying time- and spatially-variable ‘forcings’ of the Earth system² – fluxes or ‘restored-to’ boundary conditions that can be prescribed for any gas, dissolved substance (including temperature and salinity), or particulate matter. Examples include freshwater input (== a negative salinity flux forcing) of the North Atlantic to alter ocean circulation, fossil fuel CO_2 emissions to the atmosphere (== a CO_2 gas flux forcing), or aeolian iron supply to the surface ocean (a 2-D dust flux forcing).

For example: view the *user-config* file: LAB.2.3.colortracer – you will see the following lines (under the heading: '# -- FORCINGS --')

```
bg_par_forcing_name="pyyyyz_Fblue"
bg_par_force_point_i=22
bg_par_force_point_j=33
bg_par_force_point_k=8
bg_par_ocn_force_scale_val_49=1.0E12
```

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

There are then three lines (bg_par_force_point_i=20, ...) that specify the location in the ocean of the geochemical forcing that is going to be applied. The point sources are specified in (i,j,k) coordinates, which in this case is (22,33,08). For the ocean model resolution we are using, the grid is 36x36x16, and in which: longitude (i) is counted from left-to-right (1 to 36); latitude (j) is counted from bottom-to-top (1 to 36); level depth (k) is counted from downwards top-to-bottom (16 down to 1). Thus, (22,33,08) is a release of tracer in the North Atlantic, a little south of Greenland, and intermediate depth (level = 8 out of 16). Refer to the Figures for how the horizontal (Figure 2.1) and vertical (Figure 2.2) grid is specified.

Finally, there is a scaling parameter (bg_par_ocn_force_scale_val_49) which modifies the magnitude of the flux to be applied³ – the default value in the forcing definition itself is zero $0.0\text{mol}\text{yr}^{-1}$ (so you will need to set something here!).

²Refer to the ‘force the system’ HOW-TO in the cookie manual for further details on *forcings*.

³Flux *forcings* in **cookie** are in units of molyr^{-1} per model grid point.

You are going to run a brief experiment in which you will be injecting a conservative ‘dye’ tracer into the ocean. The **BIOGEM** module has two tracers that can be defined for this purpose – ‘blue’ and ‘red’ – you will be using the blue one here. You can control the flux of blue dye by opening the *user-config* file: LAB.2.2.colortracer and editing the flux scaling parameter:

```
bg_par_ocn_force_scale_val_49=1.0E12
```

The *base-config* you will be using is different from previously: cookie.C.p_worjh2.rb – this specifies a 16 vertical levels ocean and also includes seasonality of solar insolation.⁴

Run the model for ... whatever, 20 years will do. Use the *re-start* experiment that you have just downloaded to start from:

```
$ ./runcookie.sh cookie.C.p_worjh2.rb LABS LAB.2.3.colortracer 20
cookie.C.p_worjh2.rb.SPIN
```

View the results – for instance how the blue tracer distribution evolves with time – in the *time-slice* files (full ocean/atmosphere) properties saved in the **netCDF** format (.nc) files). You can follow the progress of the dye (and hence diagnose the properties of ocean circulation in the model) by plotting vertical and/or horizontal slices that go through (or near) the cell location in which you inject the dye tracer in the 3D **netCDF** file. Note that **Panoply** appears to ‘count’ the ocean layers in the opposite direction to the way in which the ocean model is actually counting them – the correct definition is with ‘1’ being very deepest level possible (and as displayed in the figure) and ‘16’ is the surface.

You can also view the tracer distributions in terms of a water-column integrated tracer inventory (**netCDF** variable name: ocn_int_colb; long name: colb water-column integrated tracer inventory) in the 2D **netCDF** output. (See: Sabine *et al.* [2004] for the use of water column integrals in the context of the distribution of anthropogenic CO₂ uptake and storage.) Changes in tracer inventory with time can be tracked in the time-series file: biogem_series_ocn_colb.res

Spend a little while altering the flux (bg_par_ocn_force_scale_val_49) and/or location (bg_par_force_point_i, bg_par_force_point_j, bg_par_force_point_k) of tracer input. Overall – note how you can use numerical ‘tracers’ to help diagnose (and better understand) the circulation of the ocean.

Ignore the ‘red’ tracer for now ... we’ll look at that shortly.

⁴Note that because the *base-config* is different from that used in the previous chapter, you need to force a re-compile of the model code before any experiments can be submitted as cluster jobs. The easiest way to do this is to run an experiment interactively at the command line.

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

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

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

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

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

If you try this, you should (hopefully) find much less (or no) negative tracer concentrations occur. However, also note that by changing the physics of ocean mixing, you also slightly alter the large-scale circulation of the ocean (and e.g. the AMOC might change slightly in strength).⁵

Lastly, an interesting (honest!) and illustrative exercise is to use the dye tracer to pick out the path taken by Mediterranean Intermediate Water. Despite the low resolution of the **cookie** ocean circulation model component and the highly restricted representation of the Mediterranean, the model does project a salty Mediterranean as a consequence of P-E across this basin (and its catchments) being negative and this higher density water makes its way out in the subsurface into the Atlantic.

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

1. View the dye-tagged plume of Mediterranean Intermediate Water by plotting a lat-lon slice (from the 3D **netCDF** file). This will give you the depth of the plume. How does this compare with salinity observations (salinity observations and appropriate global data-sets can be found on the web with a little patience)? You can also view the water-column integrated distribution (2D **netCDF**).
2. Try viewing the plume via a lat-depth slice. Refer to the figure to determine the 'i' value up the Atlantic that will just graze the edge of what passes for Spain at this low model resolution. Which direction does it head after exiting the Mediterranean? Is this 'realistic'?

⁵You might contrast the overturning stream-functions for experiments run both with and without horizontal/vertical mixing.

2.4 Tracing ocean ... ventilation

Yet another way to think about global ocean circulation is through considering the connection (rate of mass exchange) between surface and deep ocean – ‘ventilation’.

cookie has the capability to employ/simulate a ventilation age tracer⁶ – a numerical tracer carried in the ocean circulation model that tracks the time since a parcel of water last ‘saw’ the surface. The older the ‘age’ of the parcel of water, the longer the time since it last saw the surface.

We can use the second numerical tracer (red’) to keep track of age, but rather than apply a flux forcing to the surface, we let the model automatically restore the tracer value at the surface to zero and everywhere else (in the ocean interior) increase the age each time-step (by the duration of the time-step) such that a parcel of water away from the surface ages by 1 year, each year.

The *base-config*, and *restart*, provided for the (‘blue’) circulation tracing, already has the (‘red’) age tracer included and activated. As a result, all of your experiments on ocean circulation which you have conducted so far already have simulated ocean ventilation age and you do not need to run a new experiment (unless you want to!).

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

Explore the distribution of water mass age and think about the physical ocean circulations reasons for this. Are all the modelled distributions reasonable? Are there indicators of facets of the simulated circulation that are not particularly realistic? Try plotting both lon-lat and lat-depth slices through the ocean. How does the distribution of water mass age relate to the overturning stream-function for Atlantic or Pacific basins (more on this in the next Section)?

For the remainder of the sections in this chapter, we’ll ditch the ‘blue’ color tracer and use a configuration (and different *base-config*) with just the ‘red’ one: `cookie.C.p_worjh2.r`

⁶Under the ‘screw with and/or diagnose the climate system’ HOW-DO – see ‘Add a water mass age tracer’ (and the ‘easy’/automatic method described towards the end of that section).

2.5 Poking the climate beast (1)

The ocean biogeochemistry module (**BIOGEM**) in **cookie** provides a framework for applying time- and spatially-variable ‘forcings’ of the Earth system⁷ – fluxes or ‘restored-to’ boundary conditions that can be prescribed for any gas, dissolved substance (including temperature and salinity), or particulate matter. Examples include freshwater input (== a negative salinity flux forcing) of the North Atlantic to alter ocean circulation, fossil fuel CO_2 emissions to the atmosphere (== a CO_2 gas flux forcing), or aeolian iron supply to the surface ocean (a 2-D dust flux forcing). In this exercise you are going to add fresh water to the ocean surface to assess the sensitivity of the Atlantic Meridional Overturning Circulation (AMOC) to collapse, in a classic ‘hosing’ experiment.

The *user-config* file for this is called: LAB.2.5.hosing. This includes a prescribed transport of salt to the North Atlantic (north of ca. 30°N) from the result of the global surface ocean (for mass balance). This is specified by the line:

```
bg_par_forcing_name="p_worjh2.Fsal_SUR_NAtl"
```

This is paired with a scaling parameter for ease of changing the magnitude of the flux forcing:

```
bg_par_ocn_force_scale_val_2=0.0
```

which by default is zero (i.e., no flux forcing).

To orientate you in freshwater forcing space: `bg_par_ocn_force_scale_val_2=-1.1E18` – equivalent to 1Sv – should be sufficient to make ‘stuff happen’ and quickly. BUT, this is a pretty extreme flux (see overleaf for a rough conversion between salinity forcing units ($mol\ yr^{-1}$) and fresh water flux (in $m^3\ s^{-1}$ or Sv). Much more than this and the model may crash or at the very least, you might be left with a large freshwater pond in the North Atlantic. Don’t forget the minus sign if you want a net freshwater flux to the N Atlantic rather than a salinity one ...

To run the model for 10 years using the same *re-start* as previously:

```
$ ./runcookie.sh cookie.C.p_worjh2.r LABS LAB.2.5.hosing 10  
cookie.C.p_worjh2.r.SPIN
```

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

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

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

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

⁷Refer to the ‘force the system’ HOW-TO in the cookie manual for further details on *forcings*.

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

You might plot current velocity fields and visualize how these change in response to the fresh water forcing, and if you ran much longer than 10 years, see impacts on ocean ventilation ages.

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

1. First open a dataset, e.g., `atm_temp` (surface air temperature) in the 2D **netCDF** file. Double-click the variable name, or, with the variable name highlighted, click the ‘Create Plot’ icon.
2. Now, with the `atm_temp` still selected (and the first plot window still open), click on the ‘Combine Plot’ icon. A dialogue box will appear and ask you to select a plot to combine the new one with. Make sure the name of your first plot window is selected/highlighted. Click ‘Combine’. OR, simply drag a second dataset into the plot window of the first dataset.
3. You now have a plot window that by default it is showing you the difference between two identical (in time) slices. The two slices are labeled Array 1 (LH side) and Array 2 (RH side).

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

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

Try and answer the question: **How much fresh water flux (in Sv) is required to collapse the AMOC?** and how non-linear is the response of overturning to fresh water flux?

You will need to submit multiple experiment jobs to the cluster queue, with each experiment differing in the assumed radiative forcing parameter value. Plot the strength of the AMOC that is reached at the end of each experiment (from the time-series file: `biogem_series_misc_opsi.res`) vs. radiative forcing.

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

Start, by thinking of a value of `bg_par_ocn_force_scale_val_2` of -34.9 as equivalent to taking all the salt out of $1kg$ of freshwater (since the mean global salinity is $34.9PSU$). Or equivalently, since the ocean volume is fixed, an applied forcing value of -34.9 is equivalent to adding $1kg$ of freshwater to a (surface) box. So, a value of `bg_par_ocn_force_scale_val_2` of -3.49×10^4 ($-3.49E04$) would be a flux of $1m^3 yr^{-1}$ ($1000kg m^{-3}$) of freshwater.

So, in the example earlier (`bg_par_ocn_force_scale_val_2=-1.0E18`), the freshwater flux is $1.0 \times 10^{18} / 3.49 \times 10^4 = 2.8653 \times 10^{13} m^3 yr^{-1}$.

The literature invariably gives freshwater fluxes in units of Sv ($10^6 m^3 s^{-1}$). So in the example, the freshwater flux is: $9.0797 \times 10^5 m^3 s^{-1}$ ($365.25 \times 24 \times 3600 = 31557600 syr^{-1}$). Or $0.9Sv$.

Or or ... $1.0Sv$ is equivalent to a model (`bg_par_ocn_force_scale_val_2`) total forcing flux of $-1.0E18 / 0.90797 = -1.1E18$

Read the literature ... but generally, fluxes of ca. $0.05Sv$ and larger (and to quite specific places) are applied in models to induce an AMOC collapse.

2.6 Poking the climate beast (2)

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

You can explore the transient response of ocean circulation to warming by simply adjusting the radiative forcing parameter as per used in the snowball Earth experiments: `ea_radfor_scl_co2`. By default in the modern continental configuration, this has a value of 1.0, corresponding to 278 ppm atmospheric CO_2 . A value of 2.0 would reflect warming equivalent to 556 ppm CO_2 . And 3.0 would be more like an end-of-the-century warming. Note that you are applying the warming instantaneously by manipulating the climate system in this way and hence the changes will be more extreme than those occurring over the time-scale of this century. Also note that a cooling (a radiative scaling smaller than 1.0) could be applied instead.

The same parameter as before appears at the start of the *user-config* provided for this Section (LAB.2.6.EXAMPLE) for you to make this radiative forcing adjustment:

```
# scaling for atmospheric CO2 radiative forcing, relative to 278 ppm
ea_radfor_scl_co2=1.0
```

Remember that it is best practice to copy-rename-edit LAB.2.6.warming to create a new and unique experiment name for a new experiment.

You are running your new experiment from the same *re-start* state as before, e.g.:

```
$ ./runcookie.sh cookie.C.p_worjh2.r LABS LAB.2.6.warming 10
cookie.C.p_worjh2.r.SPIN
```

Potentially interesting properties of the Earth system to look at include sea-ice extent and AMOC strength (in the ASCII time-series files), and the overturning stream-function (2D with **netCDF** variable `phys_opsia`) and sea-ice extent in the 2-D **netCDF** output. With the age tracer included you can also make projections of how patterns of ventilation age change with transient warming.

Finally – is the spatial pattern of warming even? You will need to create a difference map of surface temperature – variable `atm_temp` in `fields_biogem_2d.nc` – to be able to see this well.

Try and answer the question: **How much radiative forcing is required to collapse the AMOC?** (and how much atmospheric CO_2 does this approximately correspond to?) How non-linear is the response of overturning to fresh water flux?

You will need to submit multiple experiment jobs to the cluster queue, with each experiment differing in the assumed radiative forcing parameter value. Plot the strength of the AMOC that is reached at the end of each experiment (from the time-series file: `biogem_series_misc_opsi.res`) vs. radiative forcing.

2.7 Further ideas

2.7.1 Assessing the spatial sensitivity of deep-water formation to shutdown

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

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

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

2.7.2 Forcing stronger deep-water formation (‘anti-hosing’ investigations)

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

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

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

2.7.3 Regional hosing – Poking the Mediterranean

It is also possible to apply freshwater fluxes in a specific pattern/region, rather than just at a single location.

The experiment configuration (*user-config*) – LAB.2.7.saltymed – is almost identical to all the other ones in this Chapter, except that it prescribes a net salt transport into the Eastern Mediterranean (5 cells on the model grid) with a balancing freshwater flux distributed across the remainder of the global ocean surface. This is implemented by the forcing:

```
bg_par_forcing_name="p_worxx2.Fsal_SUR_EMed"
```

This is paired with a scaling parameter for changing the magnitude of the flux forcing:

```
bg_par_ocn_force_scale_val_2=0.0
```

which by default is zero (i.e., no flux forcing). Setting the value of this parameter to:

`bg_par_ocn_force_scale_val_2=1.1E18` would create a net freshwater removal from the Eastern Mediterranean of 1Sv (probably the largest you want to try). Note that you need to apply a value that is positive so that we get a salt addition rather than freshwater addition in the Eastern Mediterranean.

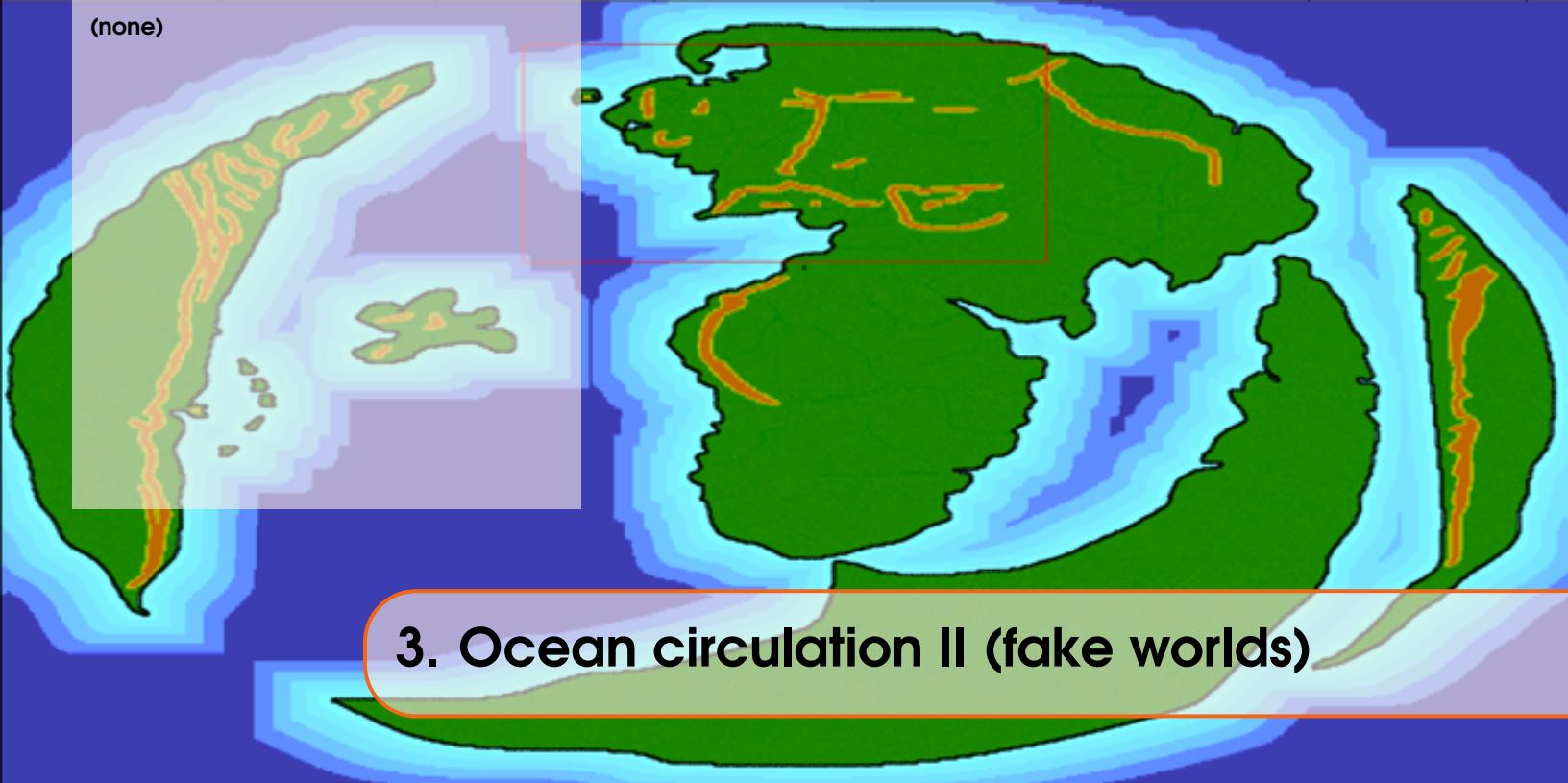
Try running the experiment for 20 years to start with. i.e.,

```
$ ./runcookie.sh cookie.C.p_worjh2.r LABS LAB.2.7.saltymed 20  
cookie.C.p_worjh2.r.SPIN
```

(again running on from a *re-start*).

You should find a rich response to adding salt to the Eastern Med. Initially, the AMOC weakens because you are taking salt away from everywhere including the N. Atlantic to feed into the Med. After year ca. 15, something interesting happens ... I am not going to spoil the surprise – you need to run the experiment and then work out 'why' the model does what it does (by repeating out some of all of the analyses you have carried out previously).

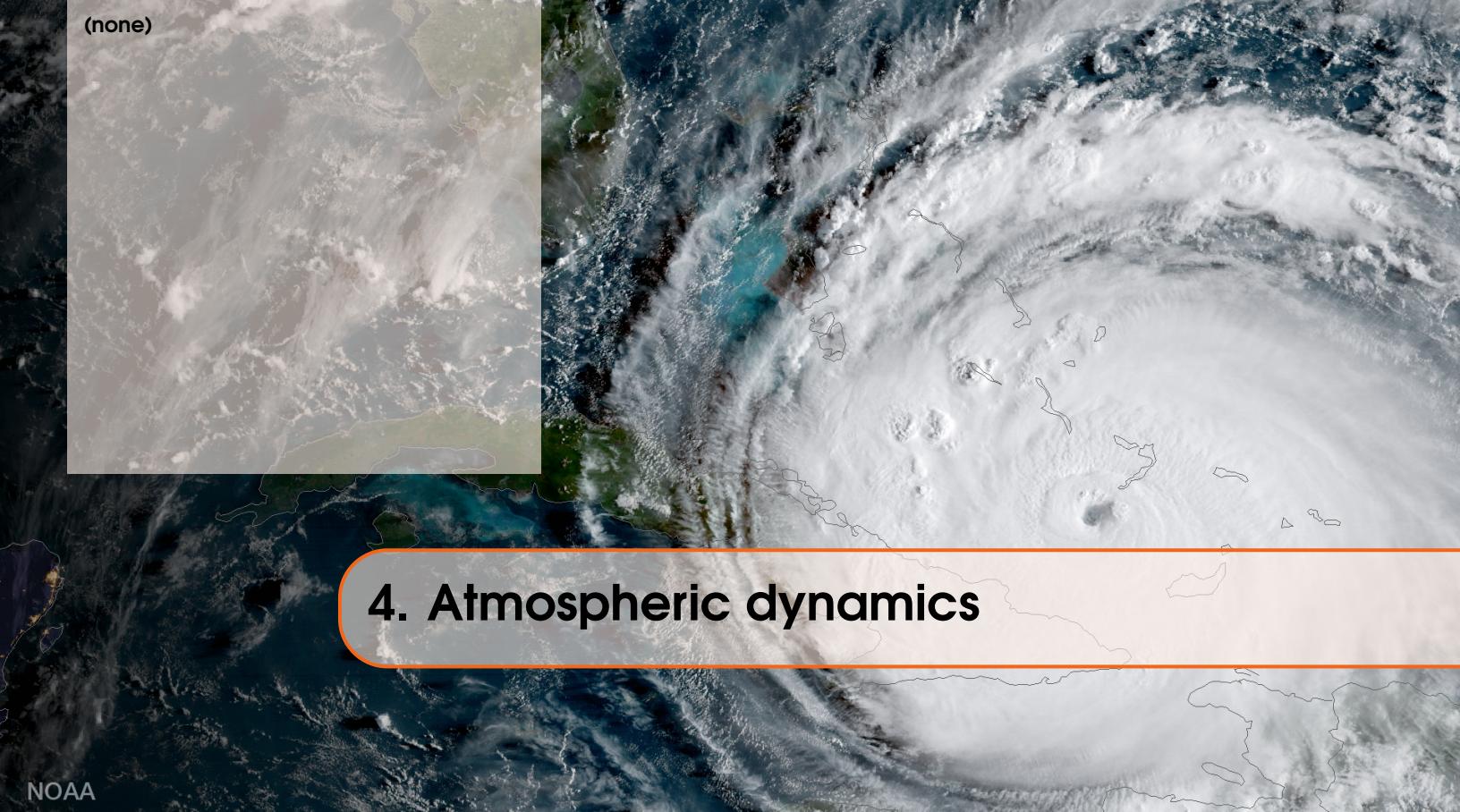
(none)



3. Ocean circulation II (fake worlds)

3.1 (none)

(none)



4. Atmospheric dynamics

NOAA

4.1 (none)

Exploring the consequences of fossil fuel

CO_2 emissions

Idealized emissions forcing

Where has my carbon gone???

Historical (real-world!) emissions forcing

Assessing future carbon emissions impacts

Further ideas

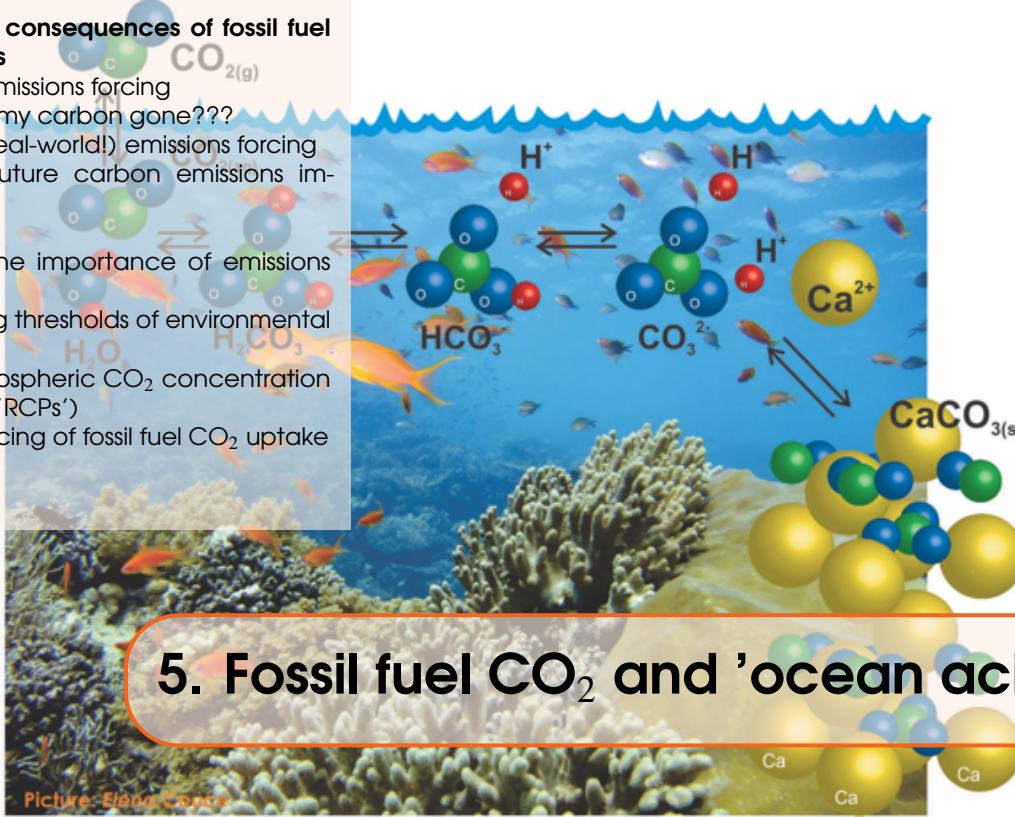
Assessing the importance of emissions rate

Determining thresholds of environmental impact

Future atmospheric CO_2 concentration pathways ('RCPs')

Isotopic tracing of fossil fuel CO_2 uptake

Appendix



Picture: Elana Correa

Readme

You will need to download a new *restart* file prior to embarking on the experiments. This pre-industrial spin-up includes a basic ocean (-atmosphere) carbon cycle. It follows the physical climate configuration of *Cao et al.* [2009] but to simplify your introduction of carbonate chemistry, does not include biology (or associated e.g., nutrient tracer).

To fetch this: change to the cgenie_output directory, and type (or copy-paste with care¹):

```
$ wget --no-check-certificate  
http://www.seao2.info/cgenie_output/cookie.CB.p_worjh2.rCARB.SPIN.tar.gz
```

Extract the contents of this archive by typing:

```
$ tar xfzv cookie.CB.p_worjh2.rCARB.SPIN.tar.gz
```

(and then change directory back to genie-main to run the model)

¹ALL one line with a space before the URL.

5.1 Exploring the consequences of fossil fuel CO₂ emissions

For the next experiment(s) you can chuck CO₂ into the atmosphere, just for the hell of it. As much as you want! Apparently, humans are actually doing this now. Imagine that!

A new *user-config* is provided – LAB.5.1.EXAMPLE – which unlike experimental setups in previous Chapters, is configured with climate being responsive to any changes in atmospheric CO₂ (i.e., it takes account of CO₂-climate feedback). (We can make this choice because now we are going to calculate atmospheric CO₂.) The setting (near the top of the *user-config* file) that does this is:

```
# set CO2-climate feedback  
ea_36=y
```

(Previously it was disabled (ea_36=n) and climate was fixed at a given radiative forcing equivalent, set by parameter ea_radfor_scl_co2.)

Additional (netCDF) output has also been prescribed, via the *user-config* parameter setting: bg_par_data_save_level=10 so that more data relevant to assessing ocean acidification is saved.

First, you should start with a control experiment to provide a baseline response (or hopefully, lack of response) to which you can contrast your perturbation experiments. It is also a good experiment to interrogate and explore new output and concepts (i.e., carbonate chemistry). For this, copy and rename² the example *user-config* for this section – LAB.5.1.EXAMPLE.

```
$ ./runcookie.sh cookie.CB.p_worjh2.rCARB LABS LAB.5.1.control 10  
cookie.CB.p_worjh2.rCARB.SPIN
```

Now, run an actual experiment. The *user-config* LAB.5.1.emissions is almost identical to your control experiment, except it specifies a release of CO₂ to the atmosphere, which by default is set to a value of just 1 PgC (cf. current emissions are ca. 10 PgCyr⁻¹) and over an interval of just a single year. (Releasing CO₂ just over a single year is obviously rather unrealistic and many impacts will decay away rapidly, but represents a useful idealized experiment for assessing the time-scale(s) of fossil fuel CO₂ uptake by the ocean.)

Run for e.g., 10 years (or for somewhat longer if you like but make sure that you have a *control* experiment of the same length) and also starting from the preindustrial *re-start* experiment cookie.CB.p_worjh2.rCARB.SPIN, i.e.:

```
$ ./runcookie.sh cookie.CB.p_worjh2.rCARB LABS LAB.5.1.emissions 10  
cookie.CB.p_worjh2.rCARB.SPIN
```

A summary of relevant *time-slices* and *time-slice* output is given in the Appendix to this Chapter.

What specific model results variables to consider ... ? Obviously atmospheric CO² to start with. Think about the climate change and ocean acidification literature and which environmental (physical and geochemical) properties are considered either critical for ecosystems or are simply helpful and/or illustrative. In the 3-D netCDF *time-slice* file remember, for instance, that ocean surface waters in which aragonite becomes under-saturated (OHMEGA < 1.0) is regarded as a critical threshold for organisms making aragonite shells and skeletons and spells TROUBLE for some poor calcifying marine organism somewhere. For climate change ... the variables of particular interest should be obvious. Remember that there are both *time-series* outputs, as well as 2D and 3D fields, any or all of which might be helpful for elucidating impacts.

²e.g., LAB.5.1.control

5.1.1 Idealized emissions forcing

The default settings in the experiment that you have just run was for 1 PgCyr^{-1} over a single year (i.e., an idealized pulse of unit size and duration). You can easily modify the experimental design to release more/less CO₂ very much as you did for the red dye tracer. In the *user-config* file, the line:

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

scales the time-history of the CO₂ flux, given in the forcing file:

```
biogem_force_flux_atm_pCO2_sig.dat
```

... which can be found in the directory:

```
cgenie.cookie/genie-forcings/pyyyz.FpCO2_Fp13CO2
```

The format of this file is:

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

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

Pause ... and note briefly how the final CO₂ flux is arrived at. **cookie** calculates it by multiplying the value in the forcing file (1.0) by a modifying parameter in the *user-config* file (8.3333e+13). The total flux is hence: $1.0 \times 8.333 \times 10^{13} = 8.333 \times 10^{13} \text{ molCO}_2\text{yr}^{-1}$. If you set both values as 1.0, you'd get very little carbon released (a single mol!). If you screw up and multiply 8.3333e+013 and 8.3333e+013 together as the total flux ... you'll soon know it as you cook the Earth ... But it does not matter which parameter has value 1.0 and which scales the units (8.3333e+013). For now, it is simply more convenient to be able to edit the *forcing* file with 'simple' numbers (and leave the large numbers and units conversion in the *user-config* file).

Together, the scaling and forcing value gives a CO₂ release of 1 PgCyr^{-1} for just a single year compared to current emissions are about 10 PgCyr^{-1} . So, do not expect anything exciting if all you emit to the atmosphere is a single measly 1 PgC (over 1 year). (The parameter: *bg_par_atm_force_scale_val_4=-27.0* specifies the carbon isotopic composition of fossil fuel carbon and can be ignored for now.)

If you want modify files in a forcing directory, it is good practice to copy the entire directory (in the example here: pyyyz.FpCO2_Fp13CO2 and then copy and rename it (within the *forcings* folder –cgenie.cookie/genie-forcings). And then edit the file(s) you want modified. The intention is that you always retain a copy of the original, unmodified *forcing*.

Having created a forcing folder with a different name, you point to it by setting the parameter:

```
bg_par_forcing_name="MYFORCING"
```

where MYFORCING is whatever you called the forcing directory
(it was originally: *bg_par_forcing_name="pyyyz.FpCO2_Fp13CO2"*).

The importance of the control experiment which you have already run, is that ‘accidents can happen’ and the global environmental changes induced by the massive fossil fuel CO₂ release can obscure mistakes made in the experiment configuration (parameter values) and/or the *re-start* used. The template experiment provided that you used as a control – LAB.5.1.EXAMPLE is designed to be identical to that of the actual experiment itself (LAB.5.1.emissions) with the exception of the scaling of the CO₂ emissions, which is set to zero.³ i.e.:

```
bg_par_atm_force_scale_val_3=0.0
```

If everything is OK with the control experiment, atmospheric CO₂ (and climate) following on from the *re-start* should be stable and there should be little (or no) drift in any of the output variables (because the *spin-up* you are re-starting from should have been run to an equilibrium state and you have not changed anything in the control experiment, right?).

It is good practice (i.e., always do it!) to always run a control experiment for each different type of experiment – e.g., you only need to run one control experiment for a set CO₂ emissions experiments differing only in total carbon release or the time-history of that release. When you have run both the real and control experiment, compare the results. View (or plot) both relevant *time-series* output, and create anomaly maps of key *time-slice* variables in **Panoply** or **MATLAB**, using a corresponding *time-slice* from the control experiment to create the experiment anomaly with.

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

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

you might have:

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

for a total of 1000PgC is released over a single year. Now you should see some policy-relevant impacts occur :o)

You can control the shape of the emissions profile as well as its magnitude. Between the start and end ‘tags’ in the text *forcing* file, the data is arranged into 2 columns: the first contains a series of tie-points for defining the timing of changes in emissions, and the 2nd column contains flux information (units of PgCyr⁻¹ when scaled by the parameter parameter bg_par_atm_force_scale_val_3 in the *user-config*). At each time-step of the model, the CO₂ flux to be applied to the atmosphere is interpolated between these time points.

³For completeness, the isotopic value of emitted CO₂ is also scaled to zero, but because there are no emissions in the control experiment, this does not actually achieve anything in practice.

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

```
0.0 1.0
1.0 1.0
1.0 0.0
999999.9 0.0
```

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

```
0.0 1.0
10.0 1.0
10.0 0.0
999999.9 0.0
```

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

In contrast:

```
0.0 0.0
10.0 1.0
10.0 0.0
999999.9 0.0
```

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

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

```
0.0 0.0
10.0 1.0
20.0 0.0
999999.9 0.0
```

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

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

```
# specify forcings
bg_par_forcing_name="pyyyzz.FpCO2_Fp13CO2.NEW"
```

if you, for instance, called your new *forcing* directory (in genie-forcings): pyyyzz.FpCO2_Fp13CO2.NEW.⁴

⁴Refer to the directory map in Figure 1.1. if in doubt here.

5.1.2 Where has my carbon gone???

In any of the emissions experiments you have tried out, in the time-series of atmospheric pCO_2 (file: biogem_series_atm_pCO2.res) you'll undoubtedly see atmospheric pCO_2 initially rise, but then once the emissions cease, start decaying back down again. Where is it 'going'?

Well obviously the ocean. D'uh! (At least, this is true in this particularly configuration of **cookie** without a terrestrial biosphere.) A better question would be: 'where in the ocean has it gone?', and even better: 'why there?'.

In the 3D netCDF, the variable ocn_DIC is the total dissolved carbon inorganic concentration (*DIC*). Open this up ... and by slicing horizontally (e.g. start at the surface, and then slice downwards), or vertically (up through the middle of the Atlantic would be a good latitude-vertical section to create), can you 'see' where the carbon (as *DIC*) is going? If not ... why not? Try making the same data sections from the same year of the control experiment. *DIC* is everywhere in the ocean in the control, with a highly spatially variable distribution. It could be then that the carbon taken up from the atmosphere in your experiment, simply overprints too small a pattern of (fossil fuel *DIC*) to tell background + fossil fuel from just background. (A similar situation arose when looking for the surface temperature impact of a weakening AMOC.)

To resolve this, create difference maps of a slice (lon-lat, or lat-depth) from your experiment at time t minus the control, also at time t . Now re-evaluate whether you can tell where the fossil fuel CO_2 is going. Why (there)? (We'll also consider this question in the next exercise.)

What about the other carbonate system parameters? Can you track identify patterns of uptake and ocean circulation transport. For instance $CO_{2(aq)}$? What about CO_3^{2-} (before looking, and remembering your basic carbonate chemistry, what would you expect)? (HCO_3^{2-} should primarily track *DIC*.)

Are there any other carbonate system impacts you can discern? Some fields to consider might include:

- pH – variable misc_pH

(There is also a field for the hydrogen ion concentration if you really want to see it.)

- Calcite saturation state ($\Omega_{(cal)}$) – variable carb_ohm_cal

- Aragonite saturation state ($\Omega_{(arg)}$) – variable carb_ohm_arg

In looking at the different fields and based on your reading of the literature (you did read the background papers, right ... ?), think about what organisms live where and what environmental (carbonate system) variables might affect them (or their prey).

Also see the subsection – "Isotopic tracing of fossil fuel CO_2 uptake".

5.1.3**Historical (real-world!) emissions forcing**

Historical and future (e.g. IPCC 'SRES') emissions scenarios can be prescribed explicitly and simply in **cookie**. An example is given as *user-config* file: LAB.5.1.historical. In this, a historical emissions forcing (technically: a prescribed concentration profile of *pCO₂* is specified by the *forcing*):

```
bg_par_forcing_name='pyyyzz.RpCO2_Rp13CO2.historical2010'
```

In contrast to before, no additional scaling is needed because the forcing specification directly follows the observed change in atmospheric concentration with time (in units of atm *CO₂*).

Note that an additional line appears in the *user-config*. This is because the historical *pCO₂* transient starts in the 1700s (for which a nominal date of 1765 is often used) rather than year zero. To start **cookie** counting from year 1765 rather than year zero, a start year parameter value is specified:

```
bg_par_misc_t_start=1765.0
```

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

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

specifies a series of time points at which data is saved that aligns with historically relevant years. Try viewing the contents of these (text) files:

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

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

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

Running a transient historically-forced experiment looks like this (all one line!):

```
$ ./runcookie.sh cookie.CB.p_worjh2.rCARB LABS LAB.5.1.historical 245  
cookie.CB.p_worjh2.rCARB.SPIN
```

The 245 parameter value for experiment duration (years) arises on the basis of the start year being 1765 (as specified by `bg_par_misc_t_start=1765.0`) and to give an experiment end at year 2100.

Note that from year 1765 onward, changes in atmospheric *CO₂* only rise very slowly initially. Don't expect to see anything happen in 10 seconds flat because relatively few people and countries in the 1800s could be bothered to burn much more than a little local coal. You could potentially start your experiment at year 1850, changing the value of `bg_par_misc_t_start` and specifying shorter experiment duration (150 years) if you are desperate for the End of the World to come.⁵

Given that there is observationally-based information on the distribution of anthropogenic *CO₂* taken up by the ocean (e.g., *Sabine et al.* [2004]) ... and you are running a historical transient experiment with the model driven by observed increases in atmospheric *pCO₂* ... you are in a position to critically evaluate the models ability (or lack of) to represent the future-critical process of oceanic fossil fuel *CO₂* uptake and transport by large scale ocean circulation.

⁵Don't forget: you could submit this experiment to the cluster and do more (idealized emissions) 'playing' which it runs.

In the 2D **netCDF** output, there is a variable for the water column integrated inventory of DIC – equivalent to the Sabine map except you will need to subtract the preindustrial background of DIC first, i.e., to create a DIC anomaly map representing only the added fossil fuel CO₂ component of ocean DIC. The data in the Sabine paper clusters around 1994. A *time-slice* centered on this year (1994.5) has been configured in the model exactly for this purpose. Your baseline state can either be from prior to CO₂ emissions commencing at any significant rate (e.g., 1750.5) or (better), from a control experiment. Note that similar comparisons could be (and are regularly) made with other tracers such as CFCs, which provide additional insights into the patterns and time-scales of trace gas update and ocean circulation. (See: *Cao et al.* [2009])

Observational data, re-gridded to the **cookie** grid and in netCDF format can be downloaded from: <http://www.seao2.org/mymuffin.html> (and under the ‘got data?’ box on the left). You could for instance, compare horizontal or vertical slices (3D netCDF) and create difference (anomaly) maps. Somewhat more representative of the entire ocean is to compare (or calculate difference maps) of zonal average profiles. Unfortunately, the observations are not in the form of water column integrals and hence you cannot create difference maps of model as per the *Sabine* paper ... unless you use the 3D **BIOGEM MATLAB** plotting scripts. Examples of **MATLAB** plotting of the model *vs.* observed anthropogenic anomaly are shown in Figure 5.1.

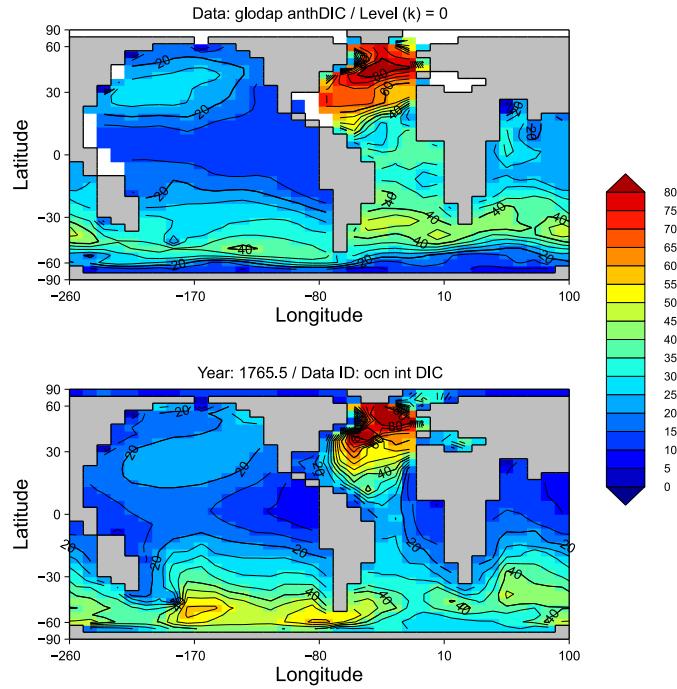


Figure 5.1: Observed (top) *vs.* Model (bottom) anthropogenic CO₂ inventories. Data and model water column integrals in units of mol CO₂ m⁻² and are nominally with respect to year 1994.

Note that model and data are not strictly comparable in the particular configuration used here, as we do not have any biology or a ‘biological pump’ (more on this later) in the ocean. (The example in the figure uses the with-biology biogeochemical configuration described by *Cao et al.* [2009].)

5.1.4**Assessing future carbon emissions impacts**

Finally, and the closest to being slightly interesting: rather than applying highly idealized pulses of CO₂ emissions, the IPCC 'SRES' emissions scenarios⁶ can be used to make future projections with. An example forcing of this sort is provided and can be selected by changing the name of the forcing selection parameter (`bg_par_forcing_name`) to any one of the following:

- pyyyyz.FpCO2_Fp13CO2.A1_AIM
- pyyyyz.FpCO2_Fp13CO2.A1G_MINICAM
- pyyyyz.FpCO2_Fp13CO2.A1T_MESSAGE
- pyyyyz.FpCO2_Fp13CO2.A2 ASF
- pyyyyz.FpCO2_Fp13CO2.B1 IMAGE
- pyyyyz.FpCO2_Fp13CO2.B2_MESSAGE

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

Again, as these forcings have units of $PgCyr^{-1}$ in the time-series files, you will need to ensure that the scaling parameter in your *user-config* file is set to turn units of $PgCyr^{-1}$ into $molCyr^{-1}$.⁷

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

You will want to run your experiment starting from the end of the historical transient experiment you have just run rather than the original steady-state *re-start*:⁸

```
$ ./runcookie.sh cookie.CB.p_worjh2.rCARB LABS LAB.5.1.future 90 LAB.5.1.historical
```

and to run for 90 years from 20120 to 2100, set the start year to the year that the previous historical transient finished on (2010):

```
bg_par_misc_t_start=2010.0
```

You can also easily replace the details of the emissions with other SRES scenarios – simply find the year *vs.* emissions rate information from the interweb⁹ and edit or copy-and-paste the flux values for each decade into the file `biogem_force_flux_atm_pCO2_sig.dat` in the forcing directory. `cookie` will then automatically interpolate between the decadal tie-points to give a continuous change in emissions. Now you are able to make a rather more realistic/plausible assessment of when and where potential ecological impacts (via assumed ocean chemistry criteria) might occur.

Try running (e.g. as jobs submitted to the cluster queue) some other actual or made up CO₂ emissions scenarios.

⁶Note that the past few IPCC assessment reports have switched to 'using 'RCP's – Representative Emissions Pathways, rather than the SRES emissions scenarios – see later Section.

⁷For completeness also add in the specification of the isotopic composition of the carbon emissions – refer back to the idealized experiments.

⁸Note that the *user-config* `LAB.5.1.future` is not provided for you – you will need to create this (or a file named whatever you like) by copying e.g., `LAB.5.1.emissions` and making the parameter changes described above (forcing specification parameter, emissions scaling parameter, and start year parameter).

⁹e.g., http://sres.ciesin.columbia.edu/final_data.html

5.2 Further ideas

Some further possibilities for investigations that build on the basic previous ones.

5.2.1 Assessing the importance of emissions rate

By editing the flux magnitude and/or timing (i.e. the years that are assigned to the different forcing time-points) information of the idealized emissions *forcings*, you can control the CO_2 emissions trajectory as well as the total of fossil fuel carbon burned. Explore some different assumptions about CO_2 release rate but for the same total carbon emitted, and note their differing impact on climate and ocean (carbonate) geochemistry. For example rather than $1000 PgC$ over a single year (which you tested earlier), in terms of emissions rate, more realistic would be e.g., 10 or $20 PgCyr^{-1}$ but spread over a longer interval (order of 100 years), for the same total carbon release. For instance, one might try and address the question: “For a given total release of fossil fuel CO_2 , is it safer to burn it slower?” The answer is maybe not completely obvious, as burning carbon resources slower will result in a small global impact, but perhaps one that persists for longer(??). You could conceive of an ensemble (related set) of model experiments, maybe one of $100 PgCyr^{-1}$ for 1 yr, one of $10 PgCyr^{-1}$ for 10 years, and one of $1 PgCyr^{-1}$ for 100 years.¹⁰

Because the experiments are getting longer to run in real time . . . remember to make appropriate use of the cluster queuing facility – i.e., think about whether you want to sit around starting at the screen for 15 minutes waiting for a new line of numbers appear – if not: submit to the cluster queue. (Don’t forget the control experiment! (configured the same, except with $0 PgCyr^{-1}$ of emissions for 100 years).)

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

5.2.2 Determining thresholds of environmental impact

There are various concerns about the impacts of continuing fossil fuel CO_2 emissions and a number of proposed climatic (e.g., the $1.5^\circ C$ paris protocol global warming limit often mentioned in policy documents) and ecological ‘tipping points’. You could assess the maximum allowable CO_2 emissions to remain within particular global environmental limits in the model. For example:

- What is the maximum total CO_2 release that can be made without inducing aragonite undersaturation at the ocean surface anywhere (or any season – see Section 5.2.3 in the User Manual for seasonal time-slice data saving)? How important is the time-scale of emissions in determining this? For total emissions above this: where in the ocean does the surface first become under-saturated and what sort of (calcifying) organisms might be impacted there? How large would the emissions have to be in order to start to induce under-saturation with respect to aragonite at the surface in the tropics (home to socio-economically important reef systems)? These are questions that can be addressed with simple CO_2 release experiments in

¹⁰These all represent rather unrealistically small total CO_2 releases and you may want to consider a total more like $1000 PgC$ or rather more. You may also want to think about more realistic shapes rather than pulses, such a some sort of ramp up and then down in the emissions rate (but for the same total emissions).

ocean carbon cycle models and everyone seems to get a GRL paper out of it each and every time!

- How important are CO₂-climate feedback in amplifying or diminishing future climate and ocean carbonate chemistry changes – e.g., is the same atmospheric pCO₂ value reached with and without climate feedback (and surface warming) – if not, why?

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

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

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

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

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

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

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

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

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

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

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

Also:

- How much carbon can be burned (and how quickly) such that atmospheric pCO_2 does not increase any further and remains at some specific value (you might pick $\times 2$ or $\times 4$ preindustrial pCO_2 (ca. 278 ppm))? This is difficult to determine well because to hold atmospheric pCO_2 constant, continuing emissions are required (as CO_2 continues to be taken up by the ocean).

One possible approach to tackling this, might be:

1. Run one or more (SRES) emissions scenarios and find the year at which your pCO_2 threshold value is crossed.
2. Re-run the same emissions experiment, but only for as many years as the threshold-crossing year you had identified, is reached. This will become your new *re-start*.
3. Create a new experiment, with zero emissions, and run on from your new re-start. You should see atmospheric pCO_2 start close to the value you reached at the end of the last experiment, but then decay away as the ocean continues to absorb carbon from the atmosphere. The decline in atmospheric pCO_2 each year tells you something about the yearly emissions needed to keep atmospheric pCO_2 constant.
An approximate rule-of-thumb, is that 1 ppm in atmospheric pCO_2 is equivalent to 2 PgC (actually, a more exact conversion is 1 ppm = 2.123 PgC). So you could calculate a yearly time-series of how much (in ppm) atmospheric pCO_2 falls by each year ... convert this to PgC, and then create a new *forcing*, with this as the annual emissions.
4. Create and run a new experiment, from the same *re-start* experiment you created, and apply the new emissions forcing you created. See if atmospheric pCO_2 is approximately maintained constant(?)
5. If not sufficient constant to your satisfaction, you could also carry out a second iteration – calculating from your latest experiment, the ppm change in pCO_2 each year, converting to an emissions rate, and running a further experiment ...
Note that if you 'overheat' and pCO_2 rises above the threshold value rather than falls below, you are allowed to have a negative carbon emission to the atmosphere in your forcing.
6. The sum of the carbon emissions up to the threshold being reached, is your answer to how much more carbon can be burned and not cross that threshold ... but you will see that after that, further emissions are allowed without pCO_2 exceeding the threshold. So the answer e.g. for year 2100 (or 2200) will be larger.

There are automated ways provided in the model framework of achieving this, and you can for instance simply tell **cookie** to maintain a specific (or changing) value of atmospheric pCO_2 , and from this diagnose the emissions rate that was required. For instance, you'll see in the original *re-start user-config* for this chapter – cookie.CB.p_worjh2.rCARB.SPIN – the *forcing* used is pyyyyz.RpCO2_Rp13CO2. If you go look in that forcing directory at:

biogem_force_restore_atm_pCO2_sig.dat

you will see a unit forcing. This is then scaled by the parameter bg_par_atm_force_scale_val_3 to achieve a constant atmospheric pCO_2 value of 278 ppm – $0.278 \times 10^{-6} atm$. To specify a different pCO_2 value, simple change the scaling factor (as per e.g., for the flux forcing).

The equivalent carbon emissions required to do this, are diagnosed and provided as a *time-series* output (in units of $molCyr^{-1}$) in file:

biogem_series_diag_misc_specified_forcing_pCO2.res

- Similarly – how much more carbon can be burned but still keep global mean surface air temperature from rising beyond the 'Paris' limit of 2°C (as compared to preindustrial)? This is also difficult to determine well because there are significant climate lags in the system with warming continuing even if atmospheric *pCO₂* was held constant.
- Trial-and error would be one approach ...

5.2.3 Future atmospheric CO₂ concentration pathways ('RCPs')

The more recent/current incarnation of IPCC future scenarios revolves not around making projections of future greenhouse gas emissions rates, but rather future greenhouse gas concentration pathways. (The reasoning is partly to cut out the differences in carbon cycle feedback between models, where for the same unit *CO₂* release, different climate/Earth system models might project a different atmospheric *CO₂* concentration and hence climate change (in addition to differences between models in climate sensitivity).) In **cookie**, these work pretty well much like in the historical forcing scenario, where the observed change in atmospheric *CO₂* concentration in the atmosphere with time, is prescribed and the model 'forced' to conform to this trajectory.

A series of RCP scenarios for how atmospheric *CO₂* concentrations may evolve with time are provided. Each actually starts at year 1765 and hence incorporates the historical transient. They can hence be used with an experiment starting at 1765 and *re-starting* from a steady-state preindustrial *spin-up*. Or they can be jumped into at any point, and e.g. experiments started at year 2010, when only the year 2010 onwards part of the *CO₂* restoring *forcing* is utilized. The RCP *forcings* currently provided are:

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

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

```
bg_par_forcing_name='pyyyzz.RpCO2_Rp13CO2.RCP8p5'
```

which would select the RCP8.5 scenario.

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

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

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

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

```
biogem_series_diag_misc_specified_forcing_pCO2.res
```

5.2.4 Isotopic tracing of fossil fuel CO₂ uptake

The experiments on fossil fuel CO₂ emissions to the atmosphere include an assumed isotopic composition of the emitted carbon. For any of the carbon emissions experiments you have run (including the historical transient) – explore in the 3D output how the isotopic composition of fossil fuel carbon is propagated from the atmosphere into the ocean and through the ocean via its large-scale circulation. The variable you want to plot is: ocn_DIC_13C¹¹. In this, it is helpful to also take a control experiment (you did run one, right ... ?) and create a difference map to better visualize how the $\delta^{13}\text{C}$ patterns in the ocean evolve through time.

By default, fossil fuel carbon is tagged with a mean fossil fuel isotopic signature of $-27\text{\textperthousand}$. This is set by the parameter:

```
bg_par_atm_force_scale_val_4=-27.0
```

which scales the value of 1.0 specified in the file biogem_force_flux_atm_pCO2_13C_sig.dat in the *forcing* directory, which in the previous fossil fuel release experiments was the genie-forcing sub-directory: pyyyyz.FpCO2_Fp13CO2

To create a more pronounced 'tag' (or tracer) of fossil fuel carbon, you could, for instance, make the assumed value of the CO₂ more negative, e.g. $-60\text{\textperthousand}$ would be the signature of methane (natural gas). You could also push the value even more negative and consider it an idealized numerical tracer of fossil fuel carbon (note that the lowest value you are allowed to set is $-999\text{\textperthousand}$).

¹¹Also calculated and saved are the isotopic compositions of the 3 different aqueous carbonate chemistry components – carb_d13C_CO2, carb_d13C_CO32, and carb_d13C_HCO3.

5.3 Appendix

Relevant CO₂ and carbonate chemistry time-series data

Filename	Data	Application
biogem_series_*.res		
atm_pCO ₂	Global inventory (<i>mol</i>), mean concentration (<i>atm</i>) of atmospheric CO ₂ .	Drivers of and feedbacks with climate. Diagnostic of response to carbon emissions (and removal).
atm_pCO ₂ _13C	¹³ C inventory (<i>mol</i>) and δ ¹³ C of atmospheric CO ₂ .	Diagnostic of carbon emissions (and removal). Comparison with (terrestrial) proxy δ ¹³ C data.
carb_sur_conc_*	Carbonate chemistry components (mean surface) (mol kg ⁻¹).	Not generally useful.
carb_sur_H	Surface ocean mean [H ⁺] (mol kg ⁻¹).	More useful is pH – reported under misc (see below).
carb_sur_ohm_arg	Mean surface aragonite saturation.	Ocean acidification impacts of CO ₂ release. Weathering impacts. Relates to carbonate production by (modern) corals, pteropods.
carb_sur_ohm_arg	Mean surface calcite saturation.	Ocean acidification impacts of CO ₂ release. Weathering impacts. Carbonate production by foraminifera and coccolithophorids.
diag_misc_specified_forcing_*	Applied flux <i>forcings</i> (mol yr ⁻¹).	Whenever a <i>restoring</i> , or <i>flux forcing</i> is specified, the actual flux employed, is saved here. Useful for diagnosing the flux associated with a restoring forcing (e.g. allowing emissions flux associated with RCP (<i>restoring forcing</i>) scenario to be diagnosed.)
misc_surrH	Mean surface <i>pH</i> .	Ocean acidification.
ocn_DIC	Global inventory (<i>mol</i>), mean global, surface, benthic concentrations (mol kg ⁻¹) of DIC.	Carbon release and removal.
ocn_DIC_13C	Global inventory (<i>mol</i>), mean global, surface, and benthic δ ¹³ C.	Carbon release and removal.
ocn_ALK	Global inventory (<i>mol</i>), mean global, surface, benthic concentrations (mol kg ⁻¹) of ALK.	'Ocean Alkalinity Enhancement' CDR.

Table 5.1: Summary of the main (useful, plus notes on a few less used) *time-series* output for (bio)geochemistry (non ecological) investigations.

Relevant CO₂ and carbonate chemistry time-slice data

variable	variable (long name)	Description	Application
carb_ben_ohm_arg		Benthic aragonite and calcite saturation.	Impacts of ocean acidification of distribution of benthic organisms. Indicator of sediment preservation.
carb_sur_ohm_cal			
carb_sur_ohm_arg		Ocean surface aragonite and calcite saturation.	Impacts of ocean acidification of distribution of planktic organisms.
carb_sur_ohm_cal			
fseaair_pCO ₂	pCO ₂ : net sea->air gas exchange flux density	Air-sea CO ₂ gas exchange.	Indicator of air-sea gas disequilibrium, regions of out-gassing/in-gassing.
misc_pH	ocean pH	Ocean surface pH.	Ocean acidification.
ocn_int_DIC	DIC water-column integrated tracer inventory	Pattern of water column integrated ocean DIC (i.e. dissolved carbon storage) ($mol m^{-2}$).	Indicator of CO ₂ emissions storage and transport when used in difference/anomaly maps and calculations.

Table 5.2: Summary of the main (mostly useful) 2D time-slice output for (bio)geochemistry.

Basic controls on biological productivity
Comparing model vs. observations
Temperature and the biological pump
Iron co-limitation of biological productivity

Further ideas

Further analysis ideas
Further experiment ideas
Other thoughts and suggestions
Biological pump background information
Advanced experiment ideas ...

6. Ocean biogeochemical cycles

READ.ME

You will need the following *re-starts* files prior to embarking on the experiments in any of the listed Sections in this Chapter.

6.1 This is the basic single (PO_4) nutrient biological export scheme configuration for playing with.

```
$ wget --no-check-certificate
http://www.seao2.info/cgenie_output/cookie.CB.p_worbe2.BASES.ridgwelletal.SPIN.tar.gz
```

6.2 A single (PO_4) nutrient biological export scheme but in a higher (vertical) resolution ocean.

```
$ wget --no-check-certificate
http://www.seao2.info/cgenie_output/cookie.CB.p_worjh2.BASES.caoetal.SPIN.tar.gz
```

World Ocean Atlas modern (PO_4) climatology (re-gridded/interpolated observations), re-gridded to the same ocean grid as EXAMPLE.worjh2.Caoetal2009.SPIN

```
$ wget --no-check-certificate
http://www.seao2.info/cgenie_output/worjh2.p_an.200709.nc
```

World Ocean Atlas modern (O_2) climatology (re-gridded/interpolated observations), re-gridded to the same ocean grid as EXAMPLE.worjh2.Caoetal2009.SPIN

```
$ wget --no-check-certificate
http://www.seao2.info/cgenie_output/worjh2.o_an.200709.nc
```

6.3 This is a paired T-dependent and non-T-dependent export and remineralizaton configuration.

```
$ wget --no-check-certificate
http://www.seao2.info/cgenie_output/cookie.CB.p_worjh2.BASES.crichtonetal.STND.SPIN.tar.gz
```

```
$ wget --no-check-certificate
http://www.seao2.info/cgenie_output/cookie.CB.p_worjh2.BASES.crichtonetal.TDEP.SPIN.tar.gz
```

6.4 This is a scheme with T-dependent biological export (but with a non T-dependent, fixed remineralization profile) and with Fe co-limitation alongside P.

```
$ wget --no-check-certificate
http://www.seao2.info/cgenie_output/cookie.CB.p_worjh2.BASESFe.FeMIP.SPIN.tar.gz
```

6.5 (same *restart* as per for 6.4)

Extract the results in the usual way and in the usual place ... and return to genie-main in the usual way ... all ... as usual.

The ocean's biological pump

In this Chapter we'll step through some of the facets of the cycle of carbon (and nutrients) in the ocean – the 'biological pump'. And then in a later section we'll look at the same processes but in a different way – through the lens of 'geoengineering', and hopefully learn something further about how everything 'works' regardless of the desirability and effectiveness (or not) of geoengineering.

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

Overall, biological production and net export of particulate (POM) and dissolved (DOM) organic matter plus calcium carbonate ($CaCO_3$) are directly determined by the availability of nutrients (phosphate and/or total dissolved iron and/or NO_3^{2-} (and/or NH_4^+) and/or dissolved silica) together with the degree to which physical conditions, particularly light and temperature, are conducive to growth. DOM is remineralized (transformed back to inorganic dissolved constituents) relatively rapidly (a ca. annual time-scale) and hence mostly at or close to the ocean surface, while POM is assumed to sink down into the ocean interior. See: *Hülse et al. [2017]*¹ for a comprehensive review (esp. of implementation in models).

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

¹Hülse, D., S. Arndt, J.D. Wilson, G. Munhoven, and A. Ridgwell, Understanding the causes and consequences of past marine carbon cycling variability through models, *Earth-Science Reviews* 171, dx.doi.org/10.1016/j.earscirev.2017.06.004 (2017).

What should you be thinking of looking at in terms of model output to understanding something about the role of the oceans biological pump in the Earth system (and climate dynamics)? Each subsequent exercise will direct you to some of the outputs to analyze/visualize that are directly relevant to the question (model experiment). In addition to that, what follows is a brief general over-view of relevant fields.

- Firstly, it is worth noting, if you have not already, the summary text file:
biogem_year_yyyy_yyy_diag_GLOBAL_AVERAGE.res where yyyy_yyy is the mid-point of the annual average year saved. Hence for the provided 10,000 year spin-up cookie.CB.p_worbe2.BASES.ridgwelletal.SPIN, the file is:
biogem_year_09999_500_diag_GLOBAL_AVERAGE.res.

Each of these files contains summary information associated with each *time-slice*. Model properties that you might pay particular attention to includes:

– ATMOSPHERIC PROPERTIES

Atmospheric pCO ₂	:	278.000 uatm
Atmospheric pCO ₂ _13C	:	-6.500 ‰

is a useful place to find e.g. the final (annual average) value of atmospheric *pCO₂* (rather than searching to the end of the relevant *time-series* file).

– BULK OCEAN PROPERTIES

Ocean DIC :	2211.435 umol kg ⁻¹ <->	0.2977295E+19 mol
-----------	---------	------------------------------------	-------------------

reflecting how much carbon is stored in the ocean,

Ocean P04 :	2.154 umol kg ⁻¹ <->	0.2899897E+16 mol
-----------	---------	---------------------------------	-------------------

the ocean nutrient (here: phosphorous) inventory,

Ocean O2 :	221.288 umol kg ⁻¹ <->	0.2979245E+18 mol
----------	---------	-----------------------------------	-------------------

is the average concentration of dissolved oxygen in the ocean (and its inventory).

– SURFACE EXPORT PRODUCTION

Export flux POC	:	204.253 umol cm ⁻² yr ⁻¹ <->	0.7505843E+15 mol yr ⁻¹
Export flux CaCO ₃	:	28.624 umol cm ⁻² yr ⁻¹ <->	0.1051875E+15 mol yr ⁻¹

fluxes reflecting the global biological export of particulate organic matter and calcium carbonate, and then

SURFACE EXPORT PRODUCTION

Export flux POC	:	204.253 umol cm ⁻² yr ⁻¹ <->	0.7505843E+15 mol yr ⁻¹
Export flux CaCO ₃	:	28.624 umol cm ⁻² yr ⁻¹ <->	0.1051875E+15 mol yr ⁻¹

is the residual flux that actually reaches the sea-floor.

Finally, there is a summary of the summary, which is often the most useful of all:

– SURFACE EXPORT & SEDIMENT DEPOSITION (RAIN) FLUX SUMMARY

Total POC export	:	0.7505843E+15 mol yr ⁻¹ =	9.007 PgC yr ⁻¹
------------------	---	--------------------------------------	----------------------------

Total CaCO ₃ export	:	0.1051875E+15 mol yr ⁻¹ =	1.262 PgC yr ⁻¹
--------------------------------	---	--------------------------------------	----------------------------

Total POC rain	:	0.8654062E+14 mol yr ⁻¹ =	1.039 PgC yr ⁻¹
----------------	---	--------------------------------------	----------------------------

Total CaCO ₃ rain	:	0.5191799E+14 mol yr ⁻¹ =	0.623 PgC yr ⁻¹
------------------------------	---	--------------------------------------	----------------------------

where carbon fluxes are also given in helpful and literature-friendly units of PgCyr⁻¹.

- Secondly, there are a number of biological pump related *time-series* outputs, e.g.:

biogem_series_fexport_POC.res
biogem_series_fexport_POPres
biogem_series_fexport_CaCO3.res

are the *time-series* of the carbon and phosphorous of particulate organic matter export, and that for $CaCO_3$, and

biogem_series_ocnsed_POC.res
biogem_series_ocnsed_POPres
biogem_series_ocnsed_CaCO3.res

are the corresponding fluxes at the sea-floor. The *time-series* output:

biogem_series_ocn_PO4.res

will tell you how what the global annual average nutrient concentration is (left) at the surface, and

biogem_series_ocn_O2.res

the average concentration dissolved oxygen in the ocean and at the sea-floor.

- Thirdly, as always – **netCDF** outputs.

- In the 2D **netCDF**:

bio_export_POC
bio_export_POP
bio_export_CaCO3

are spatial fields of the export flux of *POC*, *POP*, $CaCO_3$, and variables with *focnsed* in place of *bio_export* in its name the flux distributions to the sea-floor.

Derived from these:

misc_sur_rCaCO3toPOC
misc_sur_rPOCtoPOP

are the ratios of $CaCO_3/POC$ and POC/POP , respectively.

There are then some fields for surface and benthic tracer concentrations, such as of phosphate and dissolved oxygen.

- In the 3D **netCDF**, *bio_** are the 3D spatial distributions of particles setting down through the water column – *bio_fpart_** is the particulate flux density, *bio_fparttot_** the total flux associated with each grid point, and *bio_fparnorm_** are the fluxes in the water column normalized to the export flux out of the base of the surface ocean layer.

Also see spatial distributions of dissolved carbon, nutrients, and oxygen etc – *ocn_**.

From just the *spin-up* experiment results you can explore and/or plot some of these fields. Or better – construct a control experiment following on from the provided *restart* and analyze/explore that.

6.1 Basic controls on biological productivity

The following exercises will utilize a very basic (but relatively uncomplicated and fast) representation of the biological pump – one with only a single nutrient (PO_4) potentially limiting to biological export, considered. The scheme is described in full in *Ridgwell et al.* [2007]². Note that this specific configuration of the (modern) ocean model accounts only for 8 layers in the ocean (but spanning the same 0 – 5000 water depth range as the 16-level model configuration used in evaluating AMOC stability). Although the limited vertical resolution of the 8-level model places additional constraints on the ability of the model to reproduce features such as oxygen minimum zones in the ocean, it does enable a much shorter run-time (via a longer time-step in the model as well as calculations only needing to be carried out for about 50% of the total number of 16-level ocean cells). Hence we are making trade-off between model fidelity (in reproducing observed distribution of tracers in the modern ocean) and speed (and hence the ability to more effectively 'play' with the Earth system in a relatively short amount of real time).

We will consider the following scientific questions as a starting point, and then devise some model experiments to address them:

1. What would the global carbon cycle look like with no biological production in the ocean? Or alternatively: how important is the biological pump in controlling atmospheric $p\text{CO}_2$ and how much higher would $p\text{CO}_2$ be today in the absence of an active marine biosphere?
2. Conversely, is the biological pump operating at its maximum efficient in the ocean today, and if not, how much lower would atmospheric $p\text{CO}_2$ be if virtually all nutrients at the surface were consumed?
3. Associated with #1 and #2 – how does changing the biological pump affect the distribution and hence availability of dissolved oxygen in the ocean? (How more oxygenated would the ocean be with no export of organic matter into the ocean interior, and how much more severe would oxygen depletion be if plankton were able to utilize all the nutrients at the ocean surface?)
4. If carbon and nutrients are returned back to solution (remineralized) much closer to the ocean surface, does atmospheric $p\text{CO}_2$ increase or decrease??? (The faster return of DIC to the ocean surface will tend to increase $p\text{CO}_2$ while the return of nutrients will tend to enhance biological productivity and decrease $p\text{CO}_2$... so the answer is not necessarily intuitive and hence why we build and run computer models.)
5. Conversely, what if carbon and nutrients are much more efficiently exported into the very deepest parts of the ocean – does $p\text{CO}_2$ increase or decrease (and what happens to e.g. $[O_2]$)?
6. What role does the calcium carbonate (CaCO_3) 'counter pump' play? Would $p\text{CO}_2$ be higher or lower (and by how much) in the absence of calcifying organisms and the production of CaCO_3 in the ocean?
7. How sensitive is the cycling of carbon and nutrients and hence the pattern of ocean oxygenation to changes in the large-scale circulation of the ocean?. For instance, would $p\text{CO}_2$ be higher or lower (and by how much) with a weaker AMOC?
8. What about feedbacks with climate – how does changing global surface temperature (and hence ocean circulation patterns and sea-ice extent) affect biological productivity and the biological pump in the ocean? (Is this a positive or negative feedback on climate change?)

²Ridgwell, A., et al., Marine geochemical data assimilation in an efficient Earth System Model of global biogeochemical cycling, *Biogeosci.* 4, 87-104 (2007).

Experiments modifying the biological pump in the ocean would ideally be run for a few thousand and perhaps as long as 5000 years in order to obtain a complete new (quasi) steady state of carbon and nutrient cycling in the ocean. However, the main impacts in the ocean and on atmospheric pCO_2 often tend develop relatively rapidly (decades). Hence model experiments could be run for a few 10s or perhaps a few 100 years to see something 'happen'. A full 5000 years can also be run, and you can periodically download and viewing the results as the experiment progresses and not necessarily wait until the end of the experiment to determine a good answer. As always – if possible – run a representative (or perhaps the most extreme, such as shutting off the biological pump entirely) experiment for 5,000 or even 10,000 years, and answer for yourself how long different key facets of the system (ocean circulation, global export production, mean ocean $[O_2]$, etc.) take to adjust. Don't forget to submit your jobs to the cluster!

All the experiments in this first section will start from the same *re-start*³ and are based on the same *user-config*⁴. To run an e.g. 100 year experiment (but it need not be this – see above) using the template *user-config* and provided *re-start* would look like:

```
./runcookie.sh cookie.CB.p_worbe2.BASES LABS LAB.6.1.EXAMPLE 100
cookie.CB.p_worbe2.BASES.ridgwelletal.SPIN
```

(When you run new experiments based on this, remember to copy and rename the provided *user-config*LAB.6.1.EXAMPLE in order to create new and unique experiments each time.)

Whatever you plan to do re. perturbation experiments to explore the working of the ocean's biological pump, the first thing to do, is to run a **control experiment** for however long you have decided to run all the actual experiment experiments for. (You need not wait for this to finish, but can submit it as a *job* to the cluster and get on with running some real experiment experiments.) The control experiment can simply be derived (copied) from LAB.6.1.EXAMPLE, with no further alterations in parameter values needed. In this *user-config*, the value of atmospheric pCO_2 is not prescribed (there is no *forcing* defined) and hence pCO_2 is free to respond to any change in parameters. In the case of the control – there are no changes in the biological pump parameters compared to the *spin-up*, so you should see no (or little) drift in pCO_2 occurring. A substantive drift in the control experiment (e.g. 10s of ppm of CO_2 in the atmosphere over 100 years) is your way of knowing that you have either used the wrong *re-start* or have accidentally modified key parameters in your control experiment *user-config* compared to those used to generate the *re-start*.

Plan in advance the output you want to view and make sure it is going to appear(!) In the *user-config* – check that the results/output select 'level' is going to give you the output you are expecting. In the example *user-config*, this is specified by:

```
bg_par_data_save_level=7
```

(Refer to Chapter 13 for more details on how different (2D and 3D) *time-slice* variable fields and *time-series* are selected to be included in the model output.)

Also note that in the example *user-config* provided, netCDF time-slices are only saved at the very end (final annual average) of the model experiment, regardless of how long that is:

```
bg_par_infile_slice_name='save_timeslice_NONE.dat'
```

You may want to request more frequent saving of the spatial fields – see Section 13.2 (and 13.3).

In the example *user-config*, *time-series* saving is set to every year.

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

⁴The same *user-config* as used to generate the *re-start* (with the exception that the atmospheric CO_2 concentration is no longer prescribed).

In terms of the specific 'questions' at the start (the numbering system is the same in the following), the parameters and parameter values to change and well as what to 'look for' are:

1. How to kill the biological pump in the ocean?

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

```
# maximum rate of conversion of dissolved PO4 into
# organic matter by phytoplankton (mol kg-1 yr-1)
bg_par_bio_k0_P04=1.9582242E-06
```

(in units of $\text{mol PO}_4 \text{ kg}^{-1} \text{ yr}^{-1}$). Setting this to zero would 'turn off' completely the biological pump, leaving you with an abiotic ocean. Alternatively you could initialize the ocean with a zero phosphate concentration.

What to look for? The value of atmospheric $p\text{CO}_2$. You might also confirm that biological export really is zero (so check the export fluxes).

To obtain these values you can refer to the respective time-series results files. Or, the summary text file: biogem_year_yyyy_yyy_diag_GLOBAL_AVERAGE.res where yyyy_yyy is the mid-point of the annual average year saved. Each of these files contains summary information associated with each *time-slice*, including mean atmospheric composition, mean ocean composition, biological export fluxes. The same information can be found in various *time-series* files, but sometimes it is simply easier to find it here in one place!

2. Maxing out the biological pump?

Similar to above – nutrient uptake and the strength of the biological pump can be enhanced by increasing the value of the parameter `bg_par_bio_k0_P04` (maybe 10 times larger?). The question is then what does the ocean and atmospheric $p\text{CO}_2$ look like if the biological pump is operating at its maximum rate and (almost) all nutrients are consumed at the surface.

You may find regions of nutrients that still never get fully consumed ... why?

3. For the question on dissolved oxygen ($[O_2]$) availability in the ocean interior – the experiments are the same as #1 and #2 (above), except you are looking for how the mean and distribution of dissolved oxygen in the ocean changes. Mean ocean $[O_2]$ values can be obtained from time-series of the summary file. Spatial distributions are recorded in the *netCDF* output. You might view horizontal or vertical slices (from 3D), or benthic distributions (2D). Think about where animals tend to live and where they could potentially be impacted if export from the ocean surface was much higher than 'today' (i.e., compared to the *spin-up*, or ideally, the same year in your control).

4. What would the ocean look like if organic matter was remineralized much closer to the ocean surface?

The parameter controlling the depth-scale (and vertical distribution) at which particulate organic matter (POM) is remineralized in the ocean interior is controlled by:

```
bg_par_bio_remin_P0C_eL1=550.5195
```

(units of m). Reducing the value of this parameter forces a greater proportion of sinking POM to be remineralized closer to the surface, while a larger value pushes particulate organic matter (and associated carbon and nutrients) deeper down on average into the ocean interior.

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

See Meyer *et al.* [2016]⁵ for an example of a study testing these sort of model changes and its implications (and hence ideas of what else to look for, and for what a 'reasonable' parameter value to test might be).

5. Conversely, you can also increase the value of the depth scaling parameter in order to force a deeper mean depth of POM remineralization.

Alternatively – you can partition more of the total export into a POM form that is assumed to be resistent to degradation and is transported to the ocean floor completely unchanged (see: Ridgwell *et al.* [2007]):

```
bg_par_bio_remin_POC_frac2=6.4591110E-02
```

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

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

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

The scaling parameter controlling the molar ratio of $\frac{CaCO_3}{POC}$ is:

```
bg_par_bio_red_POC_CaCO3=0.044372
```

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

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

Because (rightly or wrongly) in the model, $CaCO_3$ depends on surface ocean Ω (w.r.t. calcite) you might explore what happens to $CaCO_3$ production and export under a release of (fossil fuel) CO_2 – simplest here is to use the same re-start, and create and add a forcing to the *user-config* to implement a large pulse or continuous CO_2 flux to the atmosphere

⁵Meyer, K.M., A. Ridgwell, and J.L. Payne, The influence of the biological pump on ocean chemistry: implications for long-term trends in marine redox chemistry, the global carbon cycle, and the evolution of marine animal ecosystems, Geobiology, DOI: 10.1111/gbi.12176 (2016).

⁶See Ridgwell et al. 2007] for a description of how this all works.

7. While it is possible to add a freshwater 'hosing' forcing (as used in an earlier Chapter/exercise) to modify large-scale ocean circulation, there is a simpler way just to 'turn off' the AMOC in the standard modern continental configuration of **cookie**.

The consequence of using the simplified **EMBM** atmosphere in conjunction with no topography over land, is that the net moisture transport from the Atlantic to the Pacific in the real world, is not reproduced. (This net transport is primarily a consequence of the blocking of Westerly transport moisture across the North American continent by e.g. the Rockies, while low latitude Trade Winds travel relatively topographically unrestricted through Central America, meaning that moisture transport from the Pacific to North Atlantic is partly blocked, while the reverse lower latitude transport is now.) To correct for this, there is a built-in prescribed moisture transport (technically, a negative salinity transport), principally from the North Atlantic to the North Pacific. In the model, there is a scaling parameter for the magnitude this transport ... which can be set to zero ... hence removing the prescribed moisture transport and likely leading to a collapse of the AMOC.

To kill (collapse) (hopefully) the AMOC in the modern continental configuration of **cookie**, set:

`ea_28=0.0`

Stuff to look for – in addition to how the AMOC changes (e.g. plot the stream-function) how do the patterns of $[PO_4]$ and $[O_2]$, particularly in the deep North Atlantic change? What impact does this have on global export production, and ultimately on atmospheric pCO_2 ?

8. Finally, you might explore how in some (or all) of the above experiments, a changing climate in response to changing atmospheric pCO_2 , in turn modulates the impacts through carbon-climate feedback.

By default in the *user-config*, climate- CO_2 feedback is disabled:

`ea_36=n`

i.e. changing atmospheric pCO_2 does not influence the climate system (which remains implicitly forced with a preindustrial radiative forcing value). This allows us to avoid complications arising due to carbon-climate feedback and hence obtain a the underlying response of the system to changing the biological pump in the ocean.

You can re-enable climate feedback by setting:

`ea_36=y`

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

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

6.2 Comparing model vs. observations

Key to having any 'confidence' in both past and future biogeochemical (e.g. carbon, nutrient, oxygen) cycling and climate dynamics and sensitivity to perturbation in a model, is having confidence in the model's ability to adequately reproduce the relevant features of the modern ocean in some direct comparison made with observations. Note that it is not necessarily critical that every single feature in the modern ocean is faithfully accounted for – the degree to which the model needs to match observations is going to depend on the question. For example, if a model is only needed for making first order projections of changes in atmospheric pCO_2 or pO_2 , the details of the structure of biogeochemical cycling in the ocean may be less important, as long as the gross partitioning of carbon between deep ocean vs. surface and atmosphere, or bottom water oxygenation and carbon flux to the sediments, respectively, is reasonable. Other questions such as involving denitrification may require more explicit details of the distribution and intensity of oxygen minimum zones (OMZs) to be able to be reproduced in the modern ocean. However, even in the latter case, if structurally in the model OMZs tend to be structurally (e.g. as a property of the fundamental ocean grid and physics and/or biases in circulation) too weak (too high $[O_2]$), it is always possible to adjust denitrification (in this example) to occur at the 'correct' rate by changing parameter values (e.g. of a minimum $[O_2]$ value at which denitrification occurs). As long as the same structural biases are present in paleo simulations, there is no reason to believe that projected past ocean denitrification rates would be incorrect. (In fact, this adjusting of parameter values to correct for some bias and achieve an appropriate rate of some process, is ubiquitous throughout all Earth system modelling, including atmospheric physics and particularly cloud formation.)

As an example/practice in model-data comparison, the output of a higher vertical resolution ocean model simulation (16 vs. the previous 8 levels in the ocean) – cookie.CB.p_worjh2.BASES.caoetal.SPIN – is provided here (see **READ.ME** downloads) along with observations of the distributions of $[PO_4]$ and $[O_2]$ in the modern ocean. The latter (observations) are interpolated and re-gridded to a high (1 degree in lon and lat) resolution grid and available as part of the World Ocean Atlas series of ocean climatologies (both physics and geochemical), and then re-gridded to the same **cookie** ocean grid as the provided (16 ocean level) experiment (which is the spin-up of *Cao et al.* [2009]).

While the comparison can be made more formally and statistically (with a little **MATLAB** or **python** code), much can be gained visually (and indeed this is typically how model-data comparisons are made in the literature). Fortunately, this visual comparison can be made in **Panoply** and both re-gridded observations and model output, being **netCDF** format, can be loaded into **Panoply**. When making visual comparisons, make sure that you use the same scale limits in both plots (these can be set manually). You can also create difference plots of model minus observations. For difference plots – be careful when changing depth or longitude slices, that you change to the same slice in both model and observations. Note that for difference plots the convention is to use a color-scale that goes from blue (negative) to red (positive) through white (or sometimes, a yellow). Set the magnitude of the scale limits equal so that zero (no difference between model and data) maps onto white (or yellow).

You may (and ideally should!) have ideas/thoughts on 'why' the model and observations diverge where they do. These thoughts may well involve processes in the ocean, e.g. is the dissolved oxygenation feature x due to insufficient or excessive organic matter remineralization in the ocean, or is the surface dissolved phosphate concentration in region y excessive/over-depleted due to insufficient/excessive biological productivity? Perhaps the most fundamental purpose or advantage

of models is that they provide you a way to test and attempt to answer such questions (which are basically questions of understanding global biogeochemical cycles in the first place and how tracer patterns in the ocean arise). So given a hypothesis for model-data mismatch, you might return to the previous section and see if a model parameter is listed that could be adjusted in value to test your hypothesis (e.g., changing remineralization depth or changing a control on biological export might improve or worsen the model-data fit – often it can improve in one location and degrade elsewhere).

Note that the first section of this chapter uses a faster 8-level ocean model configuration, whereas the model-data comparisons are on the basis of a 16-level ocean and so the results of testing parameter changes in an 8-level ocean should be regarded as qualitatively rather than quantitatively comparable.

If you wish to play with (modify and test hypotheses etc.) this configuration, and/or run a control experiment to make the comparisons with observations with, an example *user-config* (with fixed climate and no CO_2 -climate feedback) is provided as: LAB.6.2.EXAMPLE and can be run"

```
./runcookie.sh cookie.CB.p_worjh2.BASES LABS LAB.6.2.EXAMPLE 100  
cookie.CB.p_worjh2.BASES.caoetal.SPIN
```

6.3 Temperature and the biological pump

So far, both 8- (non-seasonal) and 16- (seasonally forced) level configurations have not account for the role of temperature in biological (metabolic) activity and rates of carbon transformation.

A pair of example configurations are provided for you as per published in *Crichton et al. [2021]*⁷.

1. The *user-config* for the non temperature-dependent ('standard') control configuration of *Crichton et al. [2021]*:

LAB.6.3.STND

and the resulting *spin-up* which you can use as a *re-start*:

cookie.CB.p_worjh2.BASES.crichtonetal.STND.SPIN

Note that this configuration is basically the same as in *Cao et al. [2009]*.

2. The *user-config* for the temperature-dependent (both biological uptake and water column remineralization) configuration of *Crichton et al. [2021]*:

LAB.6.3.TDEP

and the corresponding spin-up:

cookie.CB.p_worjh2.BASES.crichtonetal.TDEPSPIN

For both, the *base-config* you need is the same: cookie.CB.p_worjh2.BASES

One thing to do is simply to explore how the biogeochemical cycling (export flux magnitudes and patterns, nutrient and dissolved oxygen distributions at the surface and down through the water column) differs. You might then apply a warming perturbation by adjusting the scaling value for atmospheric pCO_2 in the prescribed restoring *forcing*:

```
# *** FORCINGS ****  
...  
bg_par_atm_force_scale_val_3=280.0E-06
```

and see what 'happens' (contrast the response of non T-dependent vs. T-dependent configurations under the exact same perturbation). **And then ... try and deduce why whatever has happened, has happened ...**

Note that in these *user-configs*, atmospheric pCO_2 is restored to a given value and this controls climate (rather than was the case with *ea_36=n*).

⁷Crichton, K. A., J. D. Wilson, A. Ridgwell, P. N. Pearson, Calibration of temperature-dependent ocean microbial processes in the cGENIE.muffin (v0.9.13) Earth system model, GMD 10.5194/gmd-14-125-2021 (2021)

You could also swap out the fixed prescribed atmospheric pCO_2 restoring forcing:

```
# specify forcings -- generic forcing of atmospheric pCO2 and d13C
# NOTE: the original paper used 280 ppm
bg_par_forcing_name="pyyyz.RpCO2_Rp13C02"
bg_par_atm_force_scale_val_3=280.0E-06
bg_par_atm_force_scale_val_4=-6.5
```

with an emissions forcing as per you user previously exploring the impact of fossil fuel CO_2 release.

For example, from Chapter 5, the basic emissions flux forcing was:

```
# specify forcings -- flux of CO2 to atmosphere
# NOTE: bg_par_atm_force_scale_val_3 == scaling of units from mol yr-1 to PgC yr-1
# NOTE: bg_par_atm_force_scale_val_4 == carbon isotopic composition of CO2 emissions
bg_par_forcing_name="pyyyz.FpCO2_Fp13C02"
bg_par_atm_force_scale_val_3=8.3333e+013
bg_par_atm_force_scale_val_4=-27.0
```

which you could paste into the *user-config* (in place of the restoring forcing) and modify as before for a more exciting perturbation of the system.

Experiments to configure and run that you might like to consider are:

1. Drive both 'standard' and T-dependent configuration with the same emissions scenario.
This contrasts the response, including climate feedback, of 'standard' and T-dependent scheme and gives you some quantitative sense of the important of T-dependent processes in the ocean and the strength and sign of their associated feedbacks.
2. Run both the above pair of experiments with fixed climate (radiative forcing), which for the assumption made in the paper of 280.0 ppm pCO_2 would require:

```
ea_36=n
ea_radfor_scl_co2=1.007
```

(where $280/278 = 1.007$)

This would help you isolate the strength and sign of the feedback of both systems (and pCO_2) with climate.

3. Strictly ... you would run 4 controls – one for both 'standard' and T-dependent and for both fixed and CO_2 -responsive climate. All 4 control experiments would be run with no carbon emissions.

What to look out for?

- Changes in pCO_2 between standard and T-dep configurations and with and without carbon-climate feedback (ea_36) enabled.
- Changes in biological export between standard and T-dep configurations in response to.
- Latitude-vertical slices of dissolved PO_4 and O_2 for the T-dep experiments would reveal how nutrient cycling and the intensity and depth of OMZs might respond to a warming climate.

6.4 Iron co-limitation of biological productivity

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

As with macronutrients such as phosphate (PO_4^{2-}) and nitrate (NO_3^-), supply of Fe to the surface ocean occurs through up-welling and mixing of ocean waters from below. However, dissolved Fe has a short lifetime in the oxygenated seawater environment. Fe^{II} , the most soluble state, is rapidly oxidized to Fe^{III} , which is highly insoluble and tends to precipitate out and be removed (scavenged) by particulate matter settling through the water column. The result is that in up-welling water, the ratio of dissolved Fe to that of highly soluble PO_4^{2-} is lower than the ratio required by phytoplankton cells to grow and divide. Consequently, phytoplankton in surface waters cannot fully utilize the abundant up-welled phosphate (or nitrate) unless more Fe is brought into the system.

Dust is important because mineral aerosols contain iron, primarily in the form of Fe oxides such as hematite and oxide-hydroxides such as goethite (found as coatings on other mineral grains). However, the present-day flux of aeolian Fe is not everywhere sufficient to correct the relative nutrient imbalance. For instance, dust fluxes to the Southern Ocean are among the lowest anywhere on Earth, and the aeolian Fe supply is too small to compensate for the depleted Fe relative to PO_4^{2-} . Consequently, phytoplankton cannot fully utilize available macronutrients, and dissolved PO_4^{2-} exists year-round at the ocean surface. Similar reasoning applies to the presence of excess PO_4^{2-} in the Eastern Equatorial Pacific. Although dust supply to the North Pacific often appears moderately high in global model simulations, fertilization experiments in the Northwest Pacific suggest that this region is still iron-limited. Hence, the geography of complete nitrate utilization at the surface will be controlled to a first order by the mass distribution of dust deposition."^{8 9}

⁸Ridgwell, A., The Global Dust Cycle, in Surface Ocean–Lower Atmospheres Processes, Eds. C. Le Quéré and E. S. Saltzman, AGU Geophysical Monograph Series, Volume 187, 350 pp.

⁹Also see: Jickells, T. D., Z. S. An, K. K. Andersen, A. R. Baker, G. Bergametti, N. Brooks, Cao J. J., P. W. Boyd, R. A. Duce, K. A. Hunter, H. Kawahata, N. Kubilay, J. laRoche, P. S. Liss, N. Mahowald, J. M. Prospero, A. J. Ridgwell, I. Tegen, and R. Torres, Global Iron Connections Between Desert Dust, Ocean Biogeochemistry and Climate, *Science*, 308, p. 67 (2005).

All experiments in this section start from the same *re-start* and are based on the same *user-config*¹⁰:

```
./runcookie.sh cookie.CB.p_worjh2.BASESFe LABS LAB.6.4.EXAMPLE
100 cookie.CB.p_worjh2.BASESFe.FeMIP.SPIN
```

This is effectively the iron cycle configuration used and evaluated in *Tagliabue et al.* [2016]¹¹.

The *forcing* used in both the experiment includes a prescribed dust flux to the ocean surface (the Mahowald part of the directory name string). This is necessary because the model configuration you are using includes a co-limitation of biological productivity by iron (*Fe*) in addition to phosphate (*PO₄*). (The files associated with the dust forcing are: *biogem_force_flux_sed_det_sig.dat* and *biogem_force_flux_sed_det_SUR.dat* but you do not need to edit these files.) **BUT** – note that no prescribed value of atmospheric *pCO₂* is given in the *forcing* for this *user-config* (although it was for generating the *re-start*) – this is so that any change you make to the marine iron cycle and hence to the biological pump in the ocean can be seen as an impact on atmospheric *pCO₂*. This also helps make for a better control experiment, because if the *re-start* was somehow mis-matched with the *user-config* or you accidentally changed something you did not mean to (or notice), you could expect to immediately see atmospheric *pCO₂* starting to drift. (Also note that the experiments need not be run for 100 years ... you may want longer or shorter, depending on how you are perturbing the model and what exactly you are 'looking for'.)

What to change? (aka, what experiment/playing around to do?)

1. Well ... firstly, you might check that biological productivity is in fact limited anywhere. You can try 2 different things:

- (a) Reduce the half-saturation constant for *Fe* limitation to zero (so phytoplankton growth is never limited by iron, at least, up to the point where it completely runs out):


```
# [Fe] M-M half-sat value (mol kg-1)
bg_par_bio_c0_Fe=0.10E-09
```

 and set this parameter to zero (or something very very small).

- (b) Flood the ocean with so much iron that it simply cannot be limiting anywhere.

The dust field itself is sort of difficult to do anything with (in terms of manipulations), so instead, you can change the assumed solubility of iron in dust – i.e. what fraction of iron delivered to the surface ocean in dust is assumed to dissolve and hence become bio-available. The parameter controlling solubility¹² is:

```
# aeolian Fe solubility
bg_par_det_Fe_sol=0.00291468
```

And change to ... ? Maybe try an order of magnitude (or more) higher value.

(The second modification is likely to have more impact and come closer to answering the question: what would export production and atmospheric *pCO₂* look like if there was no iron limitation on biological productivity? In the first option, you might still completely run out of iron somewhere in the ocean and hence limit biological productivity.)

¹⁰Remember to copy and rename the file LAB.6.4.EXAMPLE in order to create new and unique experiments each time.

¹¹Tagliabue, A., O. Aumont, R. DeAth, J.P. Dunne, S. Dutkiewicz, E. Galbraith, K. Misumi, J.K. Moore, A. Ridgwell, E. Sherman, C. Stock, M. Vichi, C. Völker, and A. Yool, How well do global ocean biogeochemistry models simulate dissolved iron distributions?, GBC DOI: 10.1002/2015GB005289 (2016).

¹²Note that this is a fractional solubility, not a % solubility, so a value of 0.001 equates to a 0.1 wt% solubility.

2. A second question/investigation (also utilizing a copy of *user-config LAB.6.4.EXAMPLE*) might be to explore the internal cycling of iron and hence the importance of dissolved iron supplied 'from below' via ocean up-welling and mixing, rather 'from above' and directly from dust.

The dissolved *Fe* content of the ocean interior is set by: ocean transport (iron from elsewhere), iron released through the remineralization of organic matter, and iron scavenged from solution and removed onto sinking particles. Of these, you have direct control over *Fe* scavenging, via:

```
# modifier of the scavenging rate of dissolved Fe
bg_par_scav_Fe_sf_POC=1.338130
```

that scales the rate at which *Fe* is removed from solution (this rate also depends on the sinking flux of particulate organic matter as well as the concentration of 'free' iron (not bound to organic ligands and hence assumed protected from scavenging).

Increasing this value will increase the loss rate of *Fe* from the ocean interior (and surface), while decreasing it will reduce the rate of loss. Consider an order of magnitude¹³ change in value (in either direction) to create meaningful changes to the marine iron cycle.

What to look for? Obviously, start with *pCO₂* (*time-series*) and *POC* export (which can be viewed as both *time-series* and spatially in the 2D **netCDF** output). (All of these are also included in the summary output files.) Also 2D distributions of *PO₄²⁻* and *Fe* are highly relevant and useful. For more involved outputs, also in the 2D **netCDF** output you can find the pattern of iron solubility (*misc_sur_Fe_sol*) and the flux of dissolved iron to the ocean surface (*misc_sur_fFe_mol*).

Also, in the 2D **netCDF** output – *misc_sur_PO4Felimbalance* – is a simple attempt to illustrate where *PO₄²⁻* (positive values) rather than *Fe* limitation of biological productivity occurs. For example, high positive values (on scale up to 1.0) can be found in the Equatorial Atlantic and Indian Oceans, suggesting plenty of dissolved iron but no phosphate. The Southern Ocean is more iron than phosphate limited, while the Pacific gyres (most negative values) are dominantly iron limited.

¹³It need not be any more than this. In fact, you might try a slightly smaller change, e.g. factor-5.

6.5 Further ideas

6.5.1 Further analysis ideas

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

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

6.5.2 Further experiment ideas

By default, the update and export in organic matter, of carbon, occurs in a fixed 'Redfield' ratio, with PO_4^{2-} . The classical value is 106 (i.e., for every mole of PO_4^{2-} taken up from the ocean, 106 moles of CO_2 are removed (and fixed into organic matter). The parameter determining this is:

`bg_par_bio_red_POP_POC=106.0`

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

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

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

`bg_ocn_dinit_8=1.0E-6`

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

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

```
# [PO4] M-M half-sat value (mol kg-1)
bg_par_bio_c0_P04=2.1989611E-07
```

As set, this specifies that at a PO_4^{2-} concentration of about $0.2 \mu mol kg^{-1}$, growth (net export) is half the maximum possible.

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

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

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

```
bg_par_bio_red_DOMfrac=0.66
```

which specifies 66% of organic matter production is diverted into dissolved form. See *Ridgwell and Arndt [2014]*¹⁴. And/or, you might adjust its mean lifetime in the ocean, which by default is 0.5yr:

```
bg_par_bio_remin_DOMlifetime=0.5
```

6.5.3 Other thoughts and suggestions

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

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

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

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

- If you are having doubts that your experiment is actually ‘doing’ anything (different from the control) – remember to create anomaly maps (plots) to look for specific changes in e.g. saturation state, pH, or the water column inventory of anthropogenic CO_2 . Even before this – plot anomalies of the flux you think you have applied, looking specifically at the region you think you have applied it to. For this, **cookie** saves the 3D distributions of dissolved Fe and PO_4 . See Figures below.
- Always be aware of the caveats regarding this specific model (and models in general) – how much does it different form the ‘real world’ for the modern ocean, particularly in terms of patterns of carbonate saturation? Does it even simulate anthropogenic CO_2 uptake adequately in the first place?

¹⁴Ridgwell, A., and S. Arndt, Why Dissolved Organics Matter: DOC in Ancient Oceans and Past Climate Change, in: Biogeochemistry of Marine Dissolved Organic Matter Eds. Hansell, D. A., and C. A. Carlson, Elsevier (2014).

6.5.4**Biological pump background information**

A few notes and background information on controls on the parameterized controls on the biological pump in **cookie**.

Remineralization depth

In the model configuration that you have been using, the degradation of particulate organic matter sinking in the water column proceeds according to a fixed profile of flux with depth (there is no e.g. temperature control on the rate of bacterial degradation of sinking organic matter) with CO_2 and PO_4 released back to the seawater as the particulate flux decreases. The parameter that controls the (*e*-folding) depth scale of particulate organic matter is:

```
bg_par_bio_remin_POC_eL1=589.9451
```

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

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

There is also an associated parameter: `bg_par_bio_remin_POC_frac2`, which sets a fraction of organic matter that is assumed to settle through the water column completely un-altered (currently assigned a value of 0.045 == 4.5%), but this is arguably less useful to change than the remineralization length-scale of the more labile fraction (the other 95.5% of particulate organic carbon exported from the ocean surface).

 $CaCO_3$:POC rain ratio

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

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

```
bg_par_bio_red_POC_CaCO3=0.0485
```

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

6.5.5 Advanced experiment ideas ...

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

As an alternative, we could re-configure the model to read in a field specifying the spatial pattern of iron solubility, set the values of iron solubility to a uniform value (0.1%), and then simply edit the values in the file to create whatever pattern of modified iron input you want.

To do this, you need to add the following lines to the *end* of the *user-config* file¹⁵:

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

which tells **cookie** to use a prescribed field of solubility values rather than calculating it, and then directs **cookie** to the file: *worjh2.det_Fe_sol.MahowaldUNIFORMsol.dat* containing the spatial pattern of solubility values.

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

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

Lastly, we need to modify the iron scavenging rate:

```
# modifier of the scavenging rate of dissolved Fe
bg_par_scav_Fe_sf_POC=0.1
```

to match the change in iron solubility.

A little care will be needed in running experiments. The change in pattern of iron solubility and scavenging rate means that the iron cycle will be slightly different as compared to the *re-start* and when you start running it, the distribution of dissolved iron and hence iron availability to biology will start to change, in turn changing the biological pump and atmospheric pCO_2 . You could either run a control and subtract the drift in whatever parameters you are interested in from the real experiment, or better, spin the adjusted configuration up. For the latter, you'd take the *re-start* as before, and maybe run e.g. 1000 years under the new iron parameter values and see whether the system is re-equilibrating. If this looks 'good', simply use this (e.g. 1000 year) experiment as your new *re-start*.

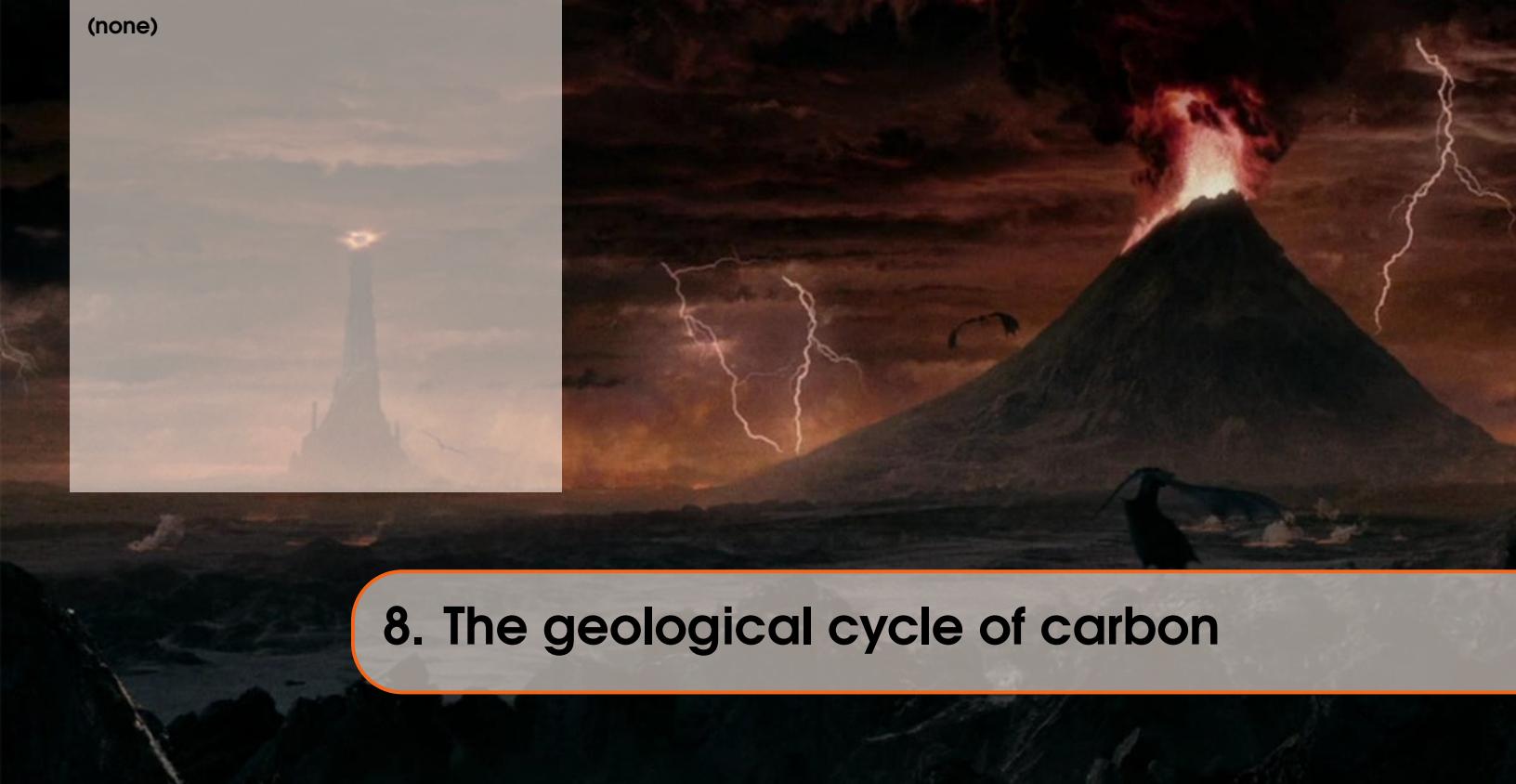
Increasing or decreasing the iron input to different regions of the ocean surface is simply then a matter of editing the values in *worjh2.det_Fe_sol.MahowaldUNIFORMsol.dat* in some pattern (hopefully informed by some hypothesis that you have formulated about iron limitation and the biological pump).

¹⁵Ideally, we would replace or delete a few parameter settings in the *user-config* file in order to avoid confusion or mistakes, but this way is much simpler.

7. Marine ecosystems and dynamics

7.1 (none)

(none)



8. The geological cycle of carbon

8.1 (none)

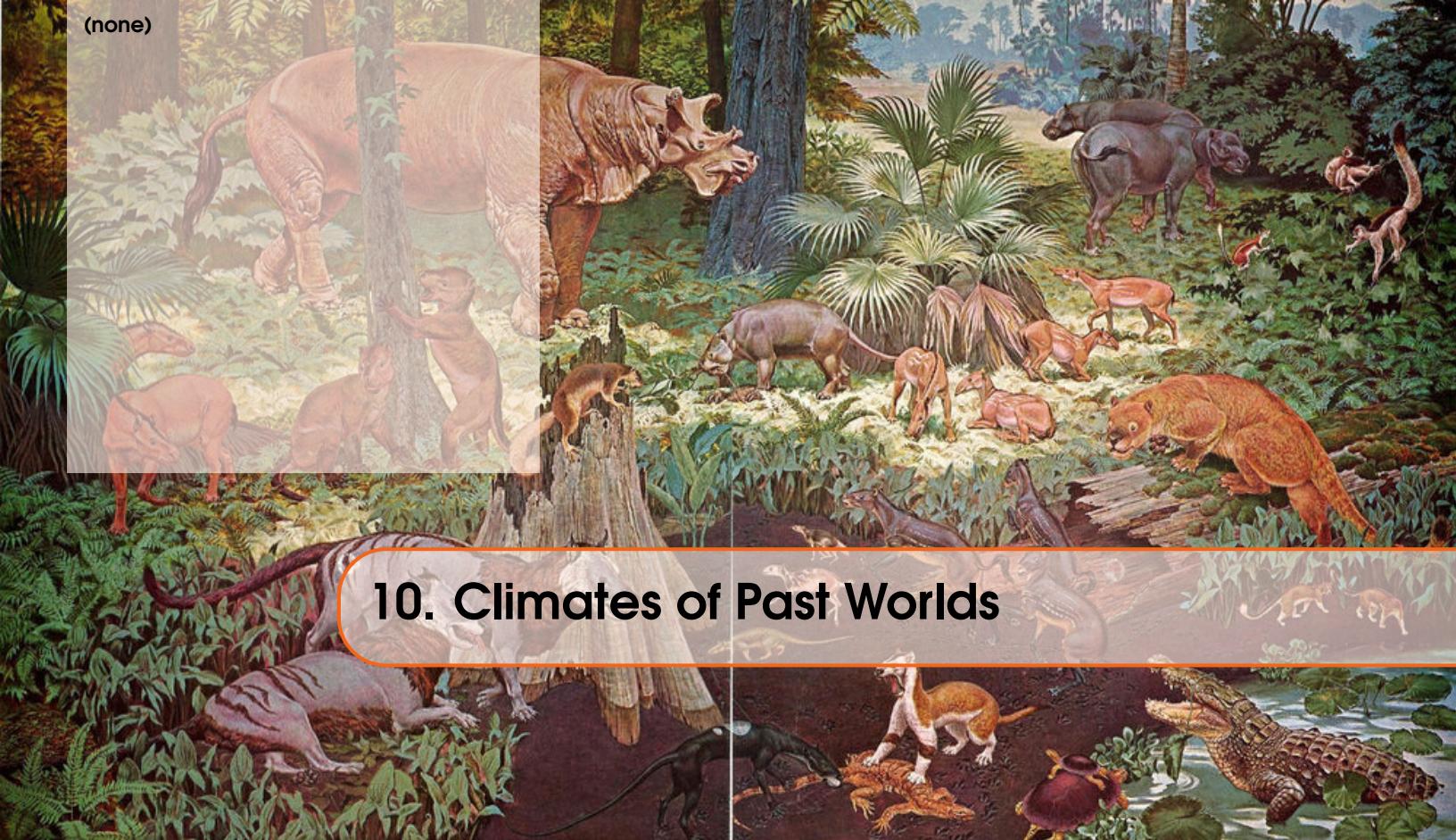
(none)



9. The land surface and carbon-climate feedbacks

9.1 (none)

(none)



10. Climates of Past Worlds

10.1 (none)

11. Proxies and Reconstructing the Past

11.1 (none)

(none)

12. Synthesis: carbon in the Earth system

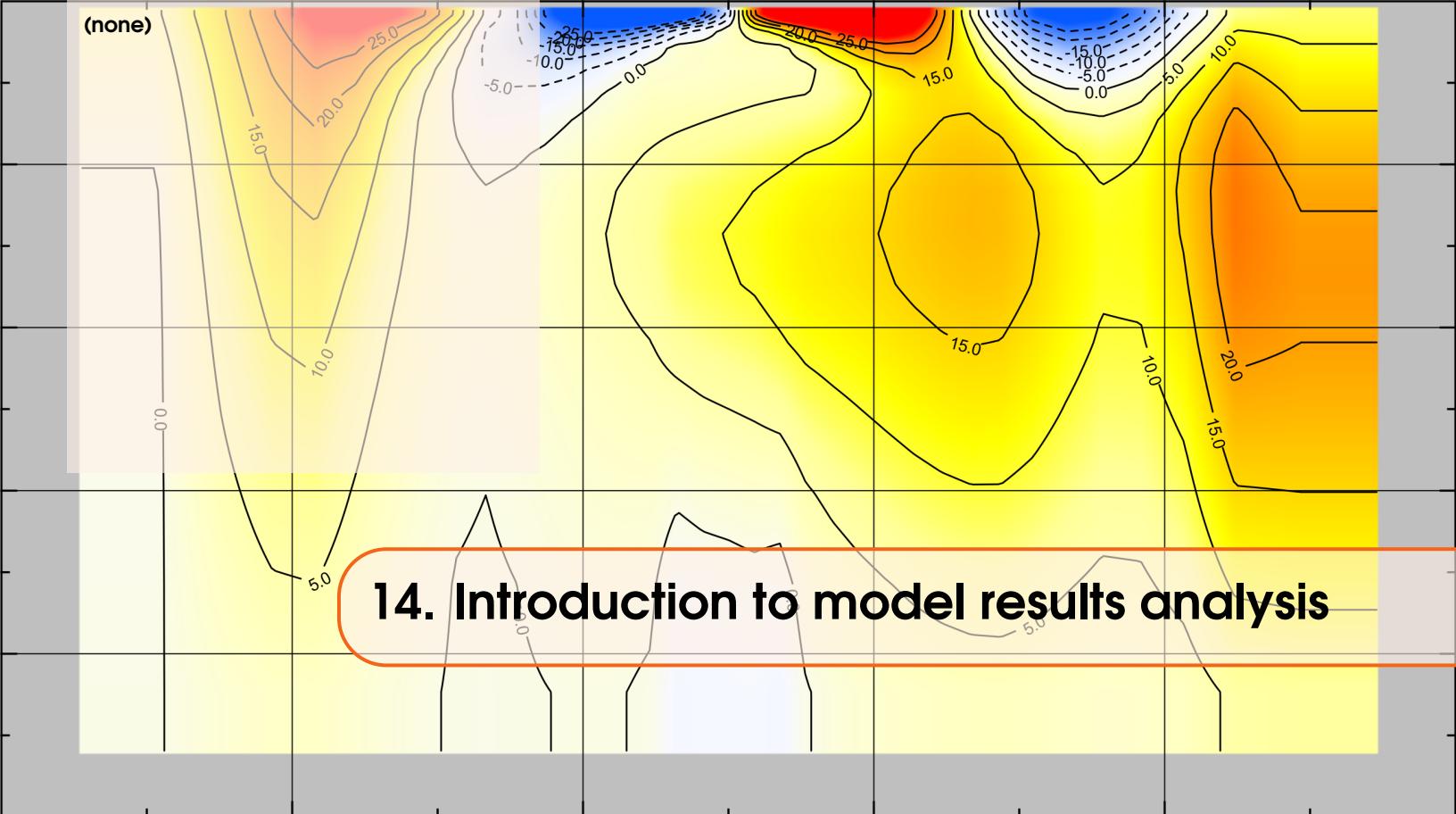
12.1 (none)

(none)



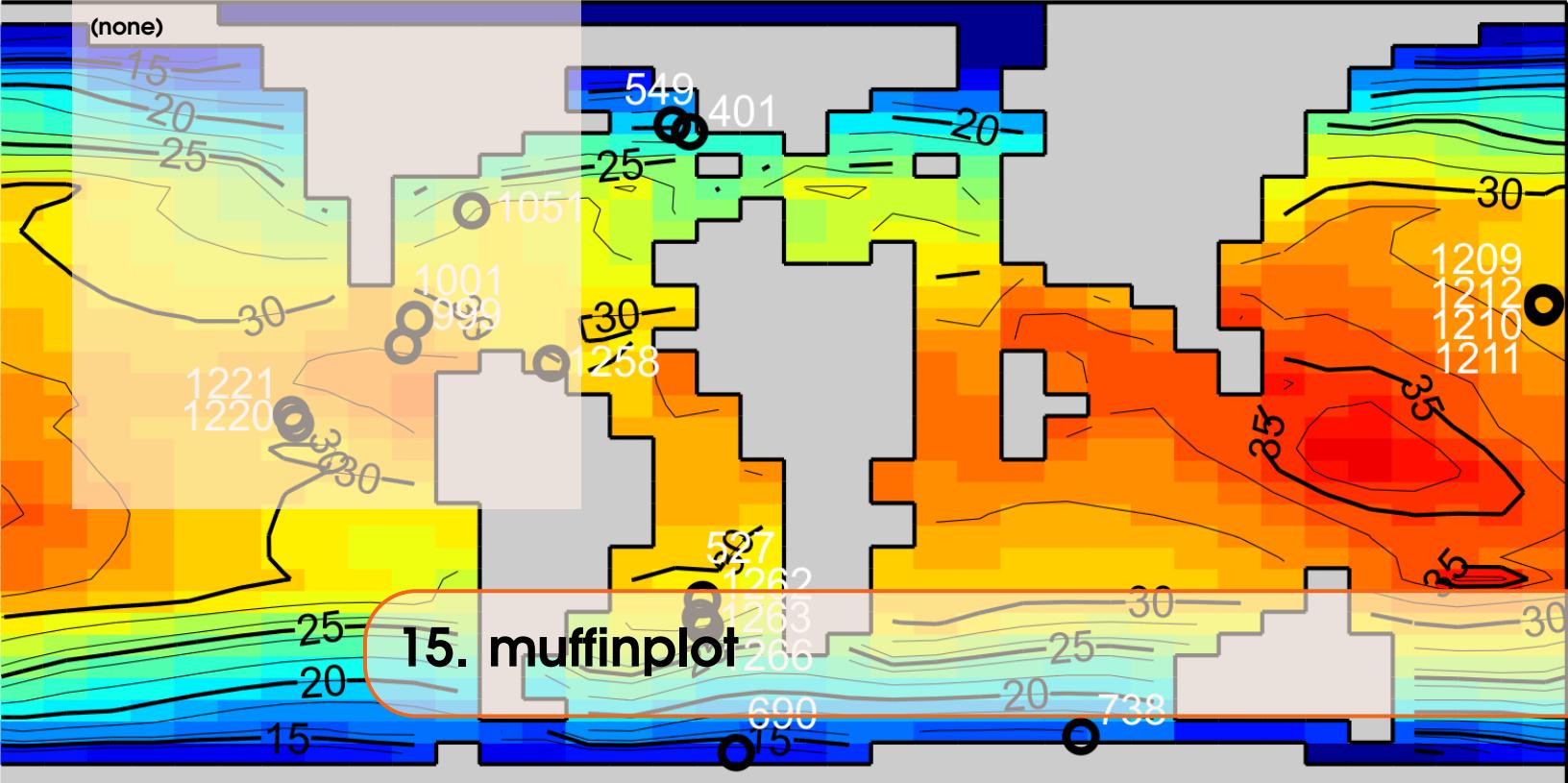
13. cookie model output

13.1 (none)



14. Introduction to model results analysis

14.1 (none)



15.1 (none)

(none)

16. muffindata

16.1 (none)

(none)

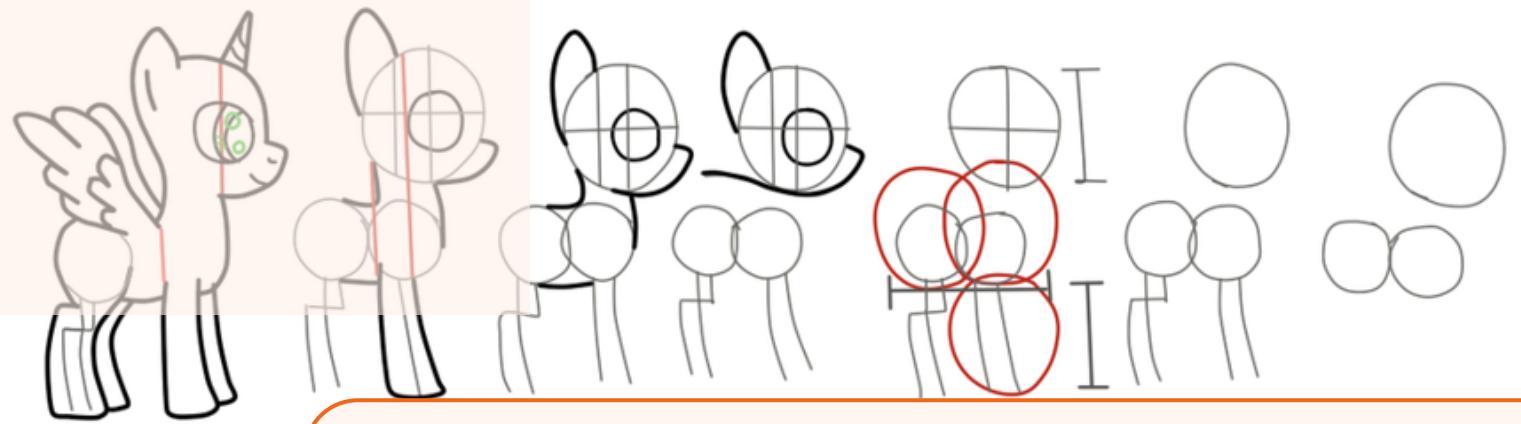
17. muffingen

17.1 (none)



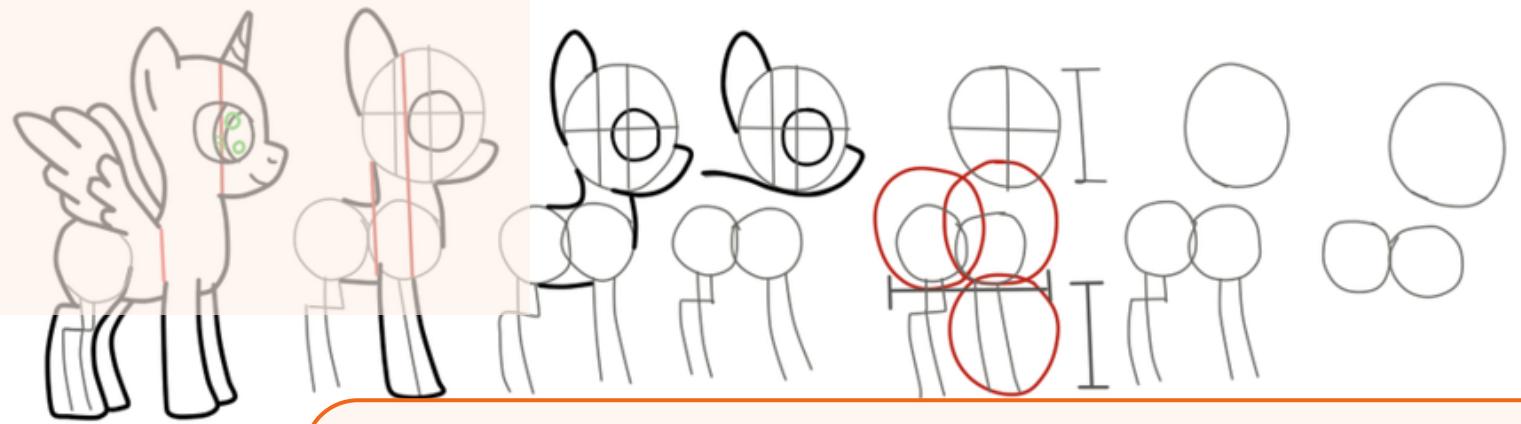
18. FAQ

18.1 (none)



19. HOW-TO (technical)

19.1 (none)



20. HOW-TO (experiment)

20.1 (none)

(none)

21. cookie Development

21.1 (none)