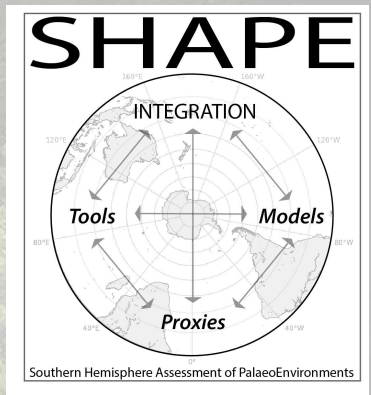
Modelling using CSIRO Mk3L
Part 2: Design your own experiment

Steven J. Phipps
ARC Centre of Excellence for Climate System Science
Climate Change Research Centre
University of New South Wales

SHAPE Training Workshop
24–25 February 2015



1. Working with model output

Output files

- When the model runs, it generates output. This is what you want!
- The model generates two types of output:

output files save the state of the model *during* a simulation
restart files save the state of the model at the *end* of a simulation

- The output files contain the simulated climate.
- In common with almost all climate models, CSIRO Mk3L saves its output in a format called netCDF.
- netCDF is a self-describing, machine-independent data format. For further information see:
  - `http://www.unidata.ucar.edu/software/netcdf/`

Ferret

- A free data visualisation and analysis package.
- Specifically designed for visualising climatic data.
- Makes it a breeze to visualise, analyse and manipulate the contents of netCDF files.
- Very powerful and easy-to-use averaging, interpolation and re-gridding capabilities.
- Your new best friend!
- For further information see:
  - `http://ferret.pmel.noaa.gov/Ferret/`

Basic Ferret commands

`use <file>`	Load the netCDF file `<file>`
`show data`	List the data which is available
`list <variable>`	List the values of `<variable>`
`plot <variable>`	Produce a line plot of `<variable>`
`shade <variable>`	Produce a shade plot of `<variable>`
`fill <variable>`	Produce a filled plot of `<variable>`
`contour <variable>`	Produce a contour plot of `<variable>`
`exit` or `q`	Exit

Basic Ferret transformations

- If the variable `tsc` contains surface air temperature as a function of longitude and latitude, then you can slice and dice the data using these expressions:

`tsc[i=10,j=8]`	Temperature at gridpoint (10, 8)
`tsc[x=140e,y=35s]`	Temperature at 140°E, 35°S
`tsc[x=90e:180e,y=45s:0]`	Temperature within 90–180°E, 45-0°S
`tsc[i=@ave]`	Zonal-mean temperature
`tsc[i=@ave,j=@ave]`	Global-mean temperature
`tsc[i=@max,j=@max]`	Global-maximum temperature
`tsc[i=@min,j=@min]`	Global-minimum temperature

Exercise 1: Ferret and model output

- First, get the course material:

```
cd ~/<NAME>
tar zxvf /srv/scratch/z3210932/material.tar.gz
```

- These commands create a new directory, `material`, which contains some material for this course. This includes some typical output from a CSIRO Mk3L simulation.
- Change to this directory by entering the command:

```
cd material
```

Exercise 1: Ferret and model output

- Now, load and run Ferret:

```
module load ferret
ferret
```

- Within Ferret, load the sample atmosphere model output:

```
yes? use stsc_spi62.nc
```

- This file contains data for surface air temperature.

Exercise 1: Ferret and model output

- Try commands such as:

```
show data
fill tsc[k=1,l=1]
fill tsc[k=@ave,l=@ave]
fill tsc[i=@ave,k=@ave]
fill tsc[k=@max,l=@max]
plot tsc[i=@ave,j=@ave,k=@ave]
plot tsc[i=@ave,k=@ave,l=@ave]
plot tsc[x=140e,y=35s,l=@ave]
list tsc[i=@ave,j=@ave,k=@ave,l=@ave]
show transform
```

Exercise 1: Ferret and model output

- A sample ocean model output file is also provided. Within Ferret, load this file:

```
yes? use com.spi62.00001.nc
```

- Try commands such as:

```
shade/lev=1d temp[k=1,l=1]
fill/lev=1d temp[i=@ave,l=@ave]
fill/lev=2dc motg[l=@ave]
plot mota[y=30n:60n@max,k=@max]
```

2. Configuring CSIRO Mk3L

Running and configuring CSIRO Mk3L

- The three basic steps involved in running the model are:
  - Create a run directory
  - Copy everything that you need to this directory
  - Run the model
- The "everything" in this second step consists of:
  - The model itself (the "executable")
  - All the input files needed to run the model
- To configure the model for a particular experiment, we need to modify one or more of these input files.

Input files

- The model requires three types of input files:

control file	configures the model for a particular simulation
restart files	initialise the model at the *start* of a simulation
auxiliary files	provide the boundary conditions *during* a simulation

- The model may be configured for a particular scenario by modifying one or more of these files.
- Auxiliary files provide the boundary conditions that the model cannot simulate itself e.g. topography.
- See Chapters 4 and 5 of the Users Guide for further information.

The control file

- To run the model, you use a command such as:

```
./model < input > output
```

- The file `input` is the *control file*.
- This file contains a number of `namelist` groups.
- The parameters contained within these groups specify:
  - the duration of a simulation
  - the physical configuration of the model
  - which model variables are to be saved

namelist groups

- A namelist group looks like this:

```
&control
  lcouple=T
  locean=F
  mstep=20
  nsstop=0
  ndstop=1
  lastmonth=0
  months=0
  nrad=6
&end
```

nano

- nano is a simple Linux text editor.
- To edit a file, enter the command:

```
nano <file>
```

- Some basic nano commands are:

Ctrl-G	Get Help
Ctrl-O	Write (Save)
Ctrl-X	Exit

Exercise 2: Editing control files

- Change to this directory, which contains some sample control files:

```
cd ~/<NAME>/version-1.2/core/control
```

- Create a copy of one of the control files, using a command such as:

```
cp input_cpl_1day input_copy
```

- Use `nano` to examine and edit this file.

Basic `namelist` options

`nsstop, ndstop, lastmonth, months`

These determine the duration of the simulation:

`nsstop`	Stop after `nsstop` timesteps
`ndstop`	Stop after `ndstop` days
`lastmonth`	Stop at the end of calendar month `lastmonth` (1=January, 2=February, ..., 12=December)
`months`	Stop after `months` months

The first of these to have a non-zero value is the one that takes effect.

Basic `namelist` options

`bpyear, csolar`

- `bpyear` specifies the epoch, in years before present (where the "present" is the year 1950 CE)
- `csolar` specifies the solar constant, in Wm^{-2}

`runtype`

- `runtype` specifies the name of the experiment

Exercise 3: Basic `namelist` options

- Look at the control files in the following directories:

```
~/<NAME>/version-1.2/core/control
~/<NAME>/material/exp01
~/<NAME>/material/exp02
~/<NAME>/material/exp04
```

- Find the following parameters, and see how the values differ:

```
nsstop, ndstop, lastmonth, months
bpyear, csolar
runtype
```

3. Using auxiliary files

The control file versus auxiliary files

- Using the control file, we can specify:
  - the duration of a simulation
  - the physical configuration of the model
  - which model variables are to be saved
- However, to configure other aspects of the model we need to modify the auxiliary files. Examples include:
  - Topography
  - Bathymetry
  - Albedo
  - Vegetation and soil types
  - CO_2 transmission coefficients
  - Ozone mixing ratios

Changing the atmospheric CO_2 concentration

- The CO_2 transmission coefficients are read from an auxiliary file.
- These files are generated by the utility `radint`.
- To compile and initialise this utility, change to the directory:

```
cd ~/<NAME>/version-1.2/pre/co2
```

- Then enter the commands:

```
make
./pset -n 18
```

Changing the atmospheric CO_2 concentration

- To generate the auxiliary file for an atmospheric CO_2 concentration of `<concentration>` ppm, enter the command:

```
./radint -c <concentration>
```

- For example, for a CO_2 concentration of 280 ppm:

```
./radint -c 280
```

- This generates a file called `co2_data`.

Exercise 4: Changing the atmospheric CO_2 concentration

- Compile and initialise `radint` by entering the commands:

```
cd ~/<NAME>/version-1.2/pre/co2
make
./pset -n 18
```

- Now generate auxiliary files for CO_2 concentrations of 280, 560 and 1120 ppm e.g.

```
./radint -c 280
```

- Remember to rename the auxiliary file each time e.g.

```
mv co2_data co2_data.280ppm
```

Applying freshwater hosing

- To apply freshwater hosing, use these `namelist` parameters:

`hosing_flag`	If `T`, apply freshwater hosing
`hosing_rate`	The freshwater hosing rate (Sv)

- You must also supply the auxiliary file `hosemask`.
- A sample auxiliary file is provided with the model:

```
~/<NAME>/version-1.2/core/data/atmosphere/hosing/hosemask
```

Exercise 5: Design your own freshwater hosing mask

- Change to the directory containing the sample freshwater hosing mask and create your own copy e.g.

```
cd ~/<NAME>/version-1.2/core/data/atmosphere/hosing
cp hosemask hosemask_new
```

- Now use `nano` to edit this file:

```
nano hosemask_new
```

- The number 7 indicates land: don't change these values!
- Put 1 where you want the water to go, and 0 everywhere else.

4. Design your own experiment

Exercise 6a: Pre-configured experiments

- Now it's time to design and run your first experiment!
- If you're new to climate modelling, then I recommend that you choose one of the following experiments:

exp01	Control simulation
exp02	Mid-Holocene (6,000 years BP)
exp03	Last Glacial Maximum (21,000 years BP)
exp04	Snowball Earth
exp05	$2\times CO_2$
exp06	Water hosing

- Each of these experiments has already been set up for you.

Exercise 6a: Pre-configured experiments

- For your experiment, change to the appropriate directory e.g.

```
cd ~/<NAME>/material/exp01
```

- Now start your experiment e.g.

```
qsub qsub_exp01
```

- Look at the script which carries out each experiment.
- How does it differ from the control simulation (`exp01`)?
- What would you change if you wanted to run your experiment for 50 years, rather than 10 years?

Exercise 6b: Design your own experiment

- If you're a climate modelling guru, then design and run your own experiment instead.
- Ideas:
  - Change the epoch: simulate the past or future
  - Change the solar constant: $\pm 5\%$, $\pm 10\%$, more?
  - Change the CO_2 concentration: $\times \frac{1}{2}$, $\times 2$, $\times 4$?
  - Freshwater hosing: melting of the Greenland or Antarctic ice sheets
- Tips:
  - Use one of the pre-configured experiments as a basis.
  - Unless you're feeling extremely confident, get me to check your experiments before you run them.
  - Once you've started your experiment, use `qstat` to monitor progress.