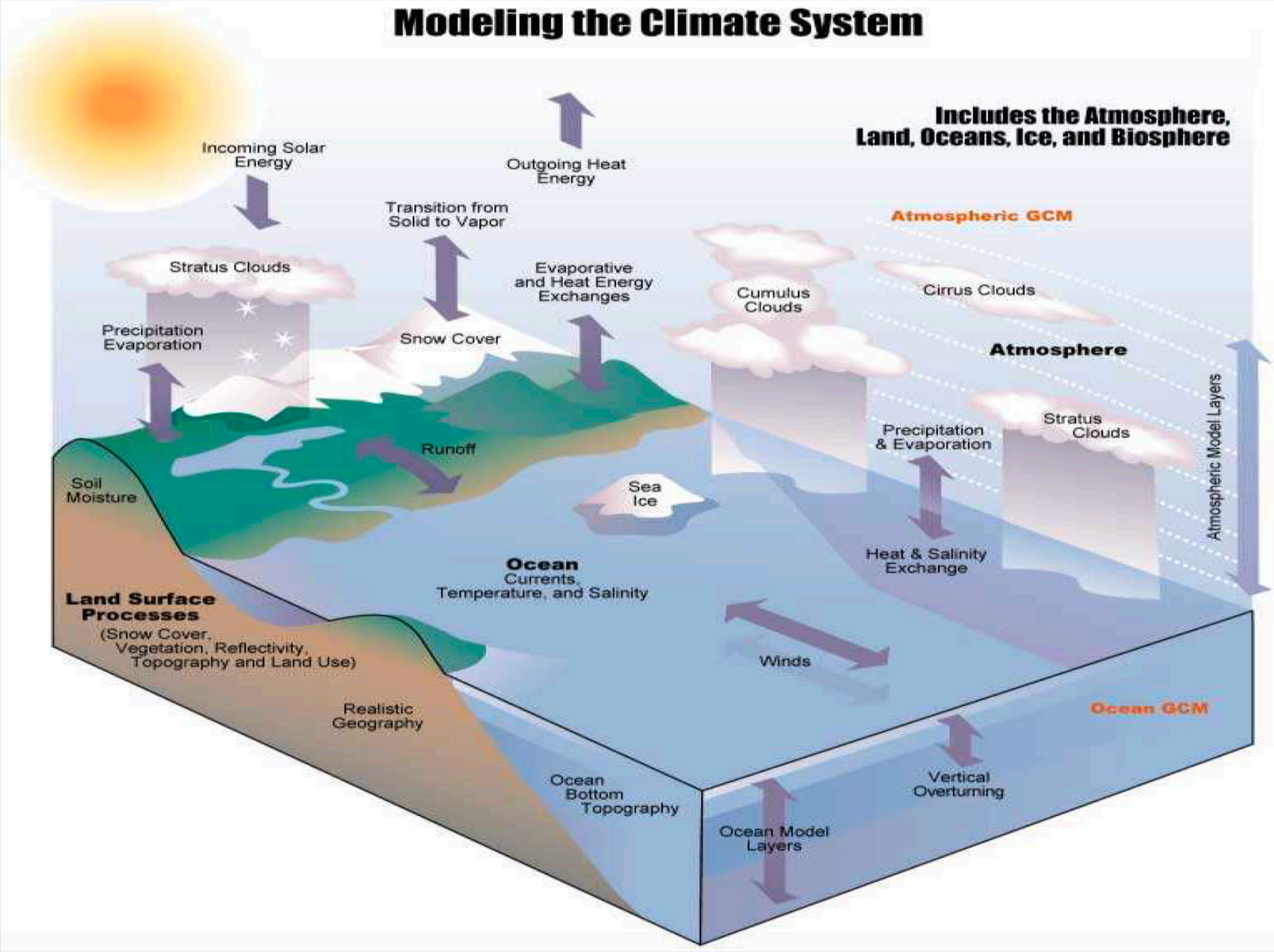
Using the CSIRO Mk3L climate system model

CleX Winter School
June 2019

climate extremes
ARC centre of excellence


Modeling the Climate System
Includes the Atmosphere, Land, Oceans, Ice, and Biosphere
Incoming Solar Energy
Outgoing Heat Energy
Transition from Solid to Vapor
Atmospheric GCM
Stratus Clouds
Evaporative and Heat Energy Exchanges
Cumulus Clouds
Cirrus Clouds
Precipitation Evaporation
Snow Cover
Atmosphere
Atmospheric Model Layers
Precipitation & Evaporation
Stratus Clouds
Runoff
Soil Moisture
Sea Ice
Heat & Salinity Exchange
Ocean
Currents, Temperature, and Salinity
Land Surface Processes
(Snow Cover, Vegetation, Reflectivity, Topography and Land Use)
Winds
Ocean GCM
Realistic Geography
Ocean Bottom Topography
Vertical Overturning
Ocean Model Layers

Choosing the right model for you

- A model is a tool - the type that you use depends upon the question that you want to answer.
- Which components, processes or quantities of the climate system do you need to model?
- Do you need a regional or a global model?
- How much spatial resolution do you need?
- How long do you need to run the model for? (for example, it isn't feasible to run a high-resolution global model for 10,000 years)
- No model is a perfect representation of the real world.

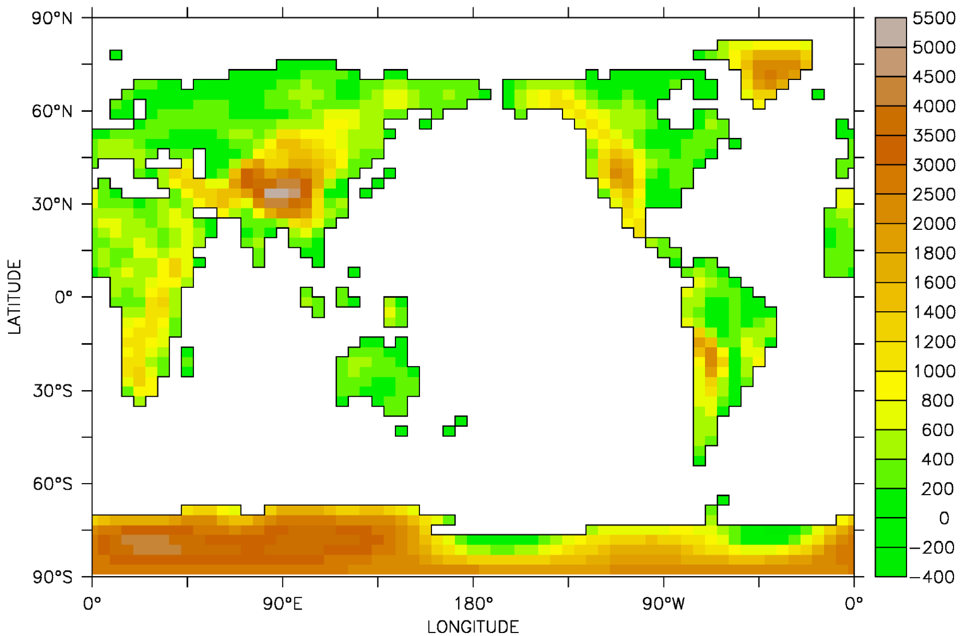
The CSIRO Mk3L climate system model

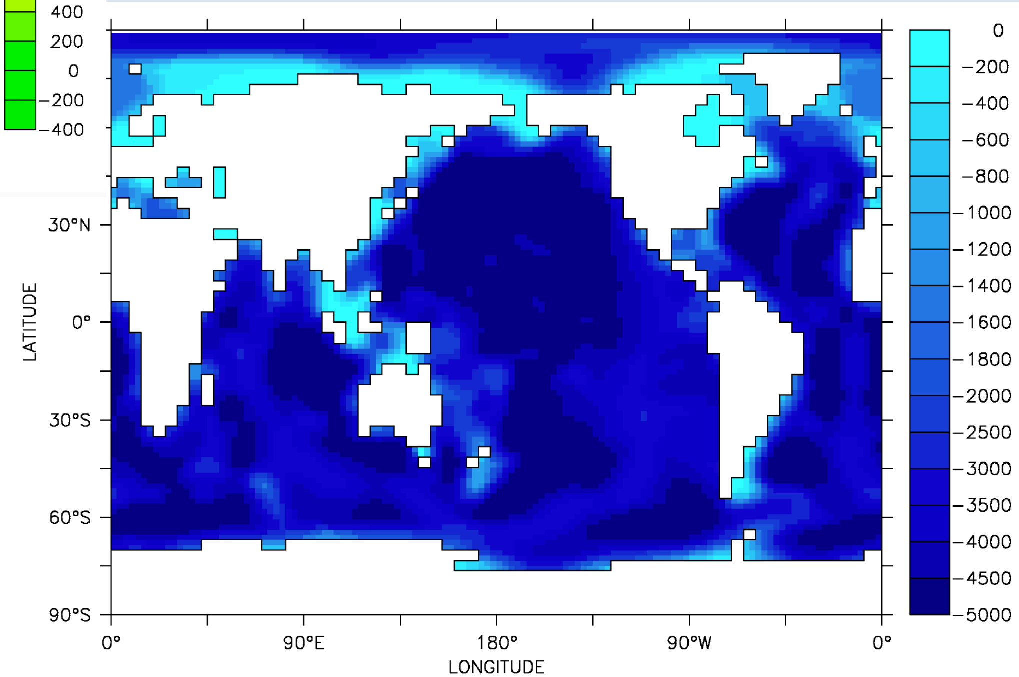
- Low-resolution version of the CSIRO climate system model (e.g. IPCC 1st, 2nd, 3rd, 4th and 5th Assessment Reports).
- Coupled atmosphere-land-sea ice-ocean general circulation model.
- Designed to enable millennial-scale simulations of climate variability and change
- Can simulate 1000 years in around a month.
- Community model.

The CSIRO Mk3L climate system model

- Atmosphere:
  - Three-dimensional general circulation model
  - Horizontal resolution of 5.6 3.2 with 18 vertical levels
- Ocean:
  - Three-dimensional general circulation model
  - Horizontal resolution of 2.8 1.6 with 21 vertical levels
- Sea ice:
  - Dynamic-thermodynamic sea ice model
  - Three layers (two ice, one snow)
- Land surface:
  - Soil-canopy scheme (13 land surface/vegetation types, 9 soil types)
  - Six soil layers, three snow layers

The topography and bathymetry within CSIRO Mk3L





Using Raijin

- Launch Terminal
- Connect through secure shell into Raijin while enabling for pop up figures

  In the terminal, enter userid@raijin.nci.org.au

  Type yes to allow login

  Type your password
- Familiarise yourself with the basic Linux commands (see the next slide).

Basic Linux commands

`ls`	list the contents of a directory
`ls -l`	create a long listing
`mkdir <directory>`	create the directory <directory>
`cd <directory>`	change to the directory <directory>
`cp <file1> <file2>`	copy the file <file1> to <file2>
`mv <file1> <file2>`	move the file <file1> to <file2>
`rm <file>`	delete the file <file>
`rmdir <directory>`	delete the directory <directory>
`man <command>`	display the manual page for <command>

- For some more Linux commands see:

  www.dummies.com/how-to/content/linux-for-dummies-cheat-sheet.html

Getting CSIRO Mk3L

- Mk3L is available via "subversion" but we're not going to use this today.
- To save time, there's a copy of the model distribution set as a module on Raijin.
- Install the version 1.2 of CSIRO Mk3L module by entering these commands:

```
module use /g/data/hh5/public/modules

module load mk3l/1.2
```

Running CSIRO Mk3L

- The basic command which runs Mk3L is simply:

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

- `model` is the executable. This is the “model".
- `input` is the control file. This contains the instructions which tell
- the model what to do.
- `output` contains diagnostic information generated by the model.

- The above command executes the model and feeds it the information contained within the control file and redirects the diagnostic information to an output file.

Running CSIRO Mk3L

- Production jobs can take weeks or months to complete, we therefore need to use a queueing system.
- Raijin uses the Portable Batch System (PBS).
- Run the model by entering this command:

```
qsub qsub_test_cpl
```

- This runs the model for one day using the `qsub_test_cpl` script which launches the model for a 1 day run.
- The `qsub` command submits a job to the queueing system.
- The `file qsub_test_cpl` tells the queueing system what to do.
- Use the command `nqstat` to check the progress of your job:

```
nqstat
```

Running CSIRO Mk3L

- The file `qsub_test_cpl` is called a script. The instructions contained within this file describe how to run the model.

- Using the `vi` command, examine the contents:

```
vi qsub_test_cpl
```

- Hint: type ESC, then `q!` to exit vi.

- The lines beginning with `#` are comments.

- The lines beginning with `#PBS -l` tell the queueing system which resources are required to run the job.

Requesting resources

- When using a queueing system, you need to request sufficient resources to run your job.

- The script that you just ran uses three different options to do this:

`nodes`	The number of nodes to run on
`vmem`	The total amount of memory required
`walltime`	The expected run time

- It's important to request sufficient resources, but not too much.

- For further information see:
  https://nci.org.au/user-support/training/training-exercises/parallel-programming/

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.

- See Chapter 6 of the Users Guide for further information.

- In common with almost all climate models, CSIRO Mk3L saves its output in a format called netCDF.

netCDF

- network Common Data Form.
- A self-describing, machine-independent data format.
- Probably the most common data format in the climate sciences.
- The names of netCDF files usually end with .nc.
- For further information see:

  http://www.unidata.ucar.edu/software/netcdf/

netCDF

- Load netCDF by entering the command:
  module load netcdf

- Use ncdump to examine the contents of the sample atmosphere model output file, stsc_spi62.nc. Try commands such as:

```
ncdump -h stsc_spi62.nc
ncdump -c stsc_spi62.nc
```

- What can you see?

Ferret

- A free visualisation and analysis package designed for visualising climatic data.
- For further information see:

  http://ferret.pmel.noaa.gov/Ferret/

Using Ferret

- 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.

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 commands

- 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 140E, 35S
`tsc[x=90e:180e,y=45s:0]`	Temperature within 90-180E,45-0S
`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

Using Ferret

- 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[x=140e,y=35s,l=@ave]
list tsc[i=@ave,j=@ave,k=@ave,l=@ave]
show transform
```

Using Ferret

- A sample ocean model output file, com.spi62.00001.nc, is also provided. Examine the contents of this file using ncdump and Ferret.
- Within Ferret, try commands such as:

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

- Try commands such as:

```
fill temp[i=@ave,l=@ave]
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]
```

- Table 6.1 of the Users Guide will be useful here.

Using Ferret

- The atmosphere data are output every pressure level in a different file, so we provide a script to combine them, in a single file:

```
comb_atm
```

- Look at the combined atmospheric outputs:

```
use su_exp01_all_atm.nc
fill u[i=39,l=1]

use sv_exp01_all_atm.nc
fill v[j=42,l=1]
```

Using Ferret

- The ocean data are output every year in a different file, so we provide a script to combine them, in a single file:

```
comb_ocn
```

- Look at the combined atmospheric outputs:

```
use com.ctl01.res_all_yr.nc
fill/LEVELS=(-2,22,1)(inf) temp[i=@ave,l=@ave]
```

Running a 50 years simulation

- The script `qsub_raijin_ctl01` runs the coupled model for 50 years. It is the script that was used to run the control experiment.

- Using the vi command, examine this script.

- How does it differ from the script which runs the model for one day?

  Hint: to compare two files, you can use vimdiff

  `vimdiff qsub_raijin_ctl01 qsub_test_cpl`

  To exit, ESC then `:q!` twice (once for each file)

Running a 50 years simulation

- Set up of the experiment part of the script

```
# Set name of run
set run = ctl01

# Set duration of run, in years
set DURATION = 50

# Set stack sizes
limit stacksize unlimited
setenv KMP_STACKSIZE 16M

setenv OMP_NUM_THREADS 10

# Create a directory for the outputs of this experiment
# if it already exists, delete the content
set TMP_DIR = /g/data/hh5/WS2019/$USER/$run/
if (-e $TMP_DIR) /bin/rm $TMP_DIR/*

# Copy the model executable to the run directory
cp $Mk3LHOME/core/bin/convert_averages $TMP_DIR

# Copy the control input file to the run directory
cp input_ctl01 $PBS_JOBFS/input
```

Running a 50 years simulation

- The script that you just examined includes the following lines:

```
#PBS -l walltime=35:00:00
#PBS -l mem=2GB
#PBS -l ncpus=10
#PBS -l jobfs=2GB
```

- These request the resources needed to run the job.
- The job is expected to take up to 35 hours (walltime).
- The job will require up to 2 GB of memory (mem).
- We want to run on 10 cpus.
- We allocate 2GB of memory for the run directory
- When you design your own experiments, walltime is the only option that you might need to change. It takes ~24h for a 50 years run.

Experiment setup

- To launch your experiment submit the script in the queue e.g.

```
qsub qsub_raijin_exp01
```

- To setup an experiment, you need to modify the input files taken by this script and the name of the experiment (run = ctl01)

- Remember that the three steps involved in running the model are:

  create a run directory

  copy model executable and input files to this directory

  run the model

Setup your own experiment

- 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.

Setup your own experiment

- Bottom boundary conditions:

  topography

  bathymetry

  albedo

  vegetation and soil types

- Radiative boundary conditions:

  CO_2 transmission coefficients

  ozone mixing ratios

- Applying a perturbation:

  freshwater hosing mask

Specific experiments

1. Magnitude of CO_2 effect on stormtracks
2. Reversibility of atmospheric CO_2
3. Different epoch (mid-Holocene)
4. Difference between the melting of Greenland and Antarctic icesheets
5. Reflecting sunlight away through geoengineering
6. Effect of forests on rainfall