IPSL BootCamp: Ferret*

Institute Pierre Simone Laplace, IPSL BootCamp

The content of the BootCamp can be found in:
`https://forge.ipsl.jussieu.fr/igcmg_doc/wiki/Train`

March 24, 2016

Contents

[*] Author of this chapter: Marco van Hulten, Laboratoire des Sciences du Climat et de l'Environnement (LSCE).
This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.

1 Introduction

After months of struggling with your model, you finally got it running, and apparently you even have some output: large files ending with `.nc`. You now want to easily visualise them. You are able to use `ncview(1)`, but you want to save the figure to disk, or change the colour scale or spatial region, or want to plot your output on a regular grid (damn those crazy irregular grids)!

Some things to know about Ferret:

- Good for visualising gridded data, typically in NetCDF format;
- you have more features than ncview, and more control over the data;
- it understands irregular grids;
- quick visualisation of model output is easy;
- high-quality plots are also possible, albeit with some effort.

I do almost all visualisation within Ferret. One may choose to do the automation (like loops and post-processing) outside of Ferret, e.g. from *Bash* or *Python*.[1]

2 Basics

You can load the Ferret environment and run it on `ada` with these commands:

`$ ssh -X ada`	login on `ada` with X11 forwarding
`$ module load ferret/6.96`	load the Ferret environment
`$ ferret`	start Ferret

Now there is also *PyFerret*, a python wrapper for Ferret with a minimal GUI, but that is not installed on `ada`.

`$ ferret`	plain old ferret (no GUI)

`$ pyferret`	a python wrapper for Ferret with minimal GUI

Enter these Ferret commands:

`yes? use levitus_climatology`	loads the Levitus climatology
`yes? show data`	shows the variables in the dataset
`yes? shade temp[k=1]`	visualise the sea surface temperature

The result is in Figure 1. Say, we want to have more detail on the equatorial Pacific Ocean:

`yes? set window/aspect=.5`	sets aspect ratio
`yes? shade temp[x=120:300, y=-30:30, z=0]`	shows only the equatorial Pacific
`yes? shade/levels=(-inf)(20,30,.2)\`	
`...?  temp[x=120:300, y=-30:30, z=0]`	more detail on temperatures over 20 °C

The result is in Figure 2. If you want to interpolate the data to get smooth lines, you can use the command `fill` instead of `shade`.

`yes? list temp[i=@max,j=@max,k=1]`	show maximum SST

[1] That said, it is possible to do fairly complicated scripting in Ferret. See e.g. my comPlot data–model visual comparison package; this package attempts to integrate structured handling of model output and observations, with high-quality plotting, all within Ferret.

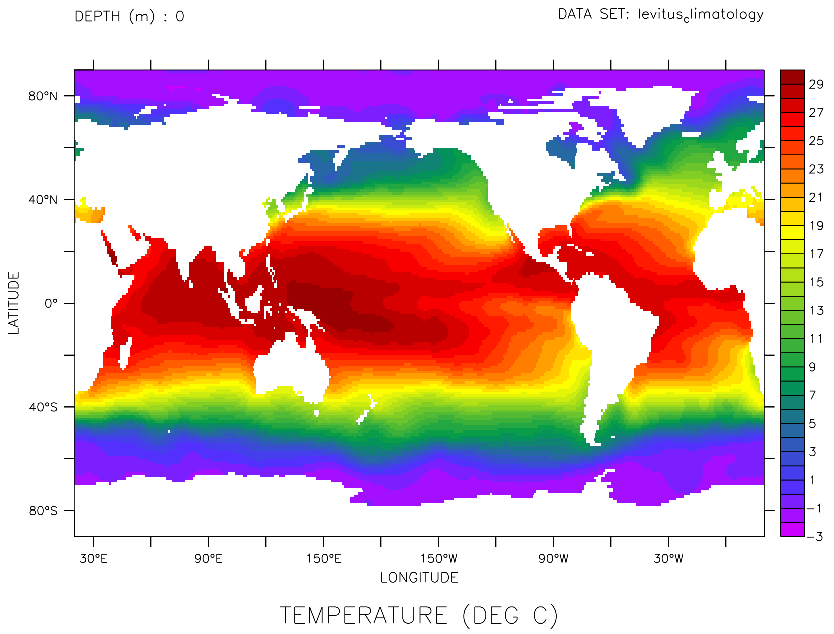

Figure 1: Sea surface temperature (SST) from the World Ocean Database. To get this result, execute this command: `shade temp[z=0, d=levitus_climatology]`

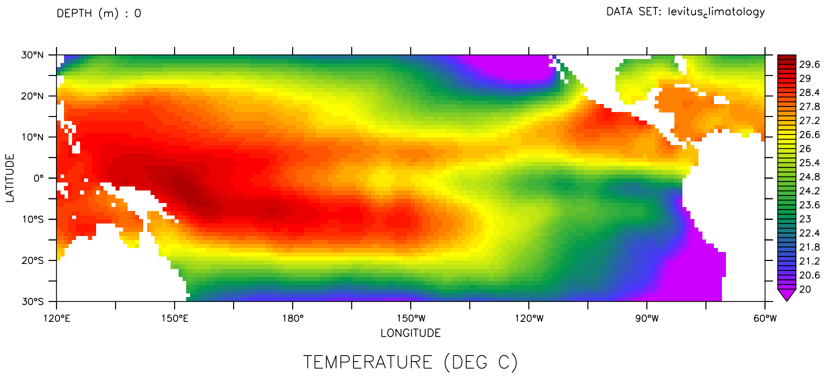

Figure 2: `shade/levels=(-inf)(20,30,.2) temp[x=120:300, y=-30:30, z=0]`

2.1 Irregular grids

Let's now look at output from the IPSL model, the NEMO component, ocean dynamics. Put `piControl2pm01_SE_2100` in `$FER_DATA` (which typically contains the current directory). You need to scp(1) it from `ipcours` to `ada`. In Ferret we do the following to plot the simulated SST of March (Figure 3/left):

```
yes? use piControl2pm01_SE_2100_2109_1M_grid_T   load the simulation output
yes? show data                                   show the variables from the loaded dataset
yes? set window/aspect=.7
yes? shade/title="Modelled SST (^oC)"\           plot the SST, and override caption title
...?  votemper[k=1, l=3]
```

The projection of the land is not what you expected. Model configurations are often defined on irregular grids, and often they output the data on that grid. From the `show` command you can see that this dataset is defined on a grid of 182 times 149 points. If you plot such data using the model grid, you'll get distortions.

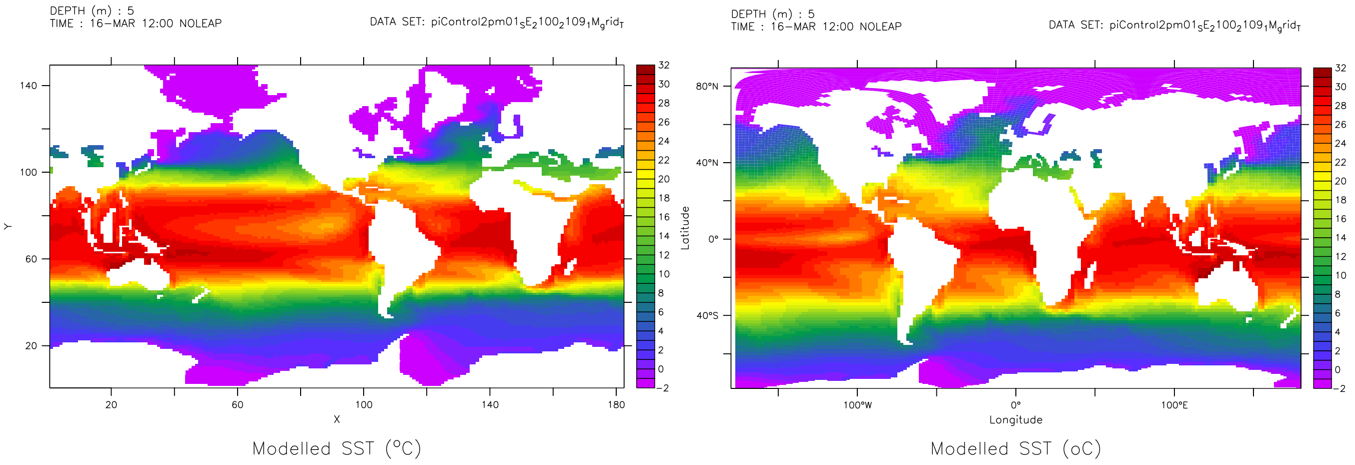

Figure 3: SST from the model: **(left)** on the grid, **(right)** transformed to regular lon-lat grid.

E.g., this model, NEMO, uses the ORCA configuration. More specifically, this simulation uses the ORCA2 grid that is nominally 2^o in resolution, but its resolution is increased at several locations, and it has two North Poles over land. There are two options to plot your data that are on an irregular grid on a regular grid (Figure 3/right):

- convert your data before visualisation (or other manipulation);
- use a visualisation tool that can do coordinate transformations based on your metadata.

You can use SOSIE to regrid your data.[2] For direct plotting (so on-the-fly regridding) usually libraries can be found for lots of visualisation software. I've never been able to make any of those work with my output. Luckily, Ferret has built-in capabilities for this:

```
yes? set window 2
yes? set data piControl2pm01_SE_2100_2109_1M_grid_T
yes? shade/title="Modelled SST (^oC)" votemper[k=1, l=3], nav_lon, nav_lat
yes? go land
```

This sets the second window and then sets the default dataset. Then, the model output is plotted on a regular grid. Finally, the main coast lines are displayed with the `land` script that is in the default Ferret installation. You can inspect the variables `nav_lon` and `nav_lat` that define the model grid, e.g. by listing them:

```
yes? list nav_lon, nav_lat
```

You can do this with votemper and other variables as well, but do specify a range as they are defined on many gridboxes!

2.2 Scripting

There are many `go scripts` part of the standard Ferret distribution. In the previous section we saw the script `land.jnl`. This file can be found in the directory `${FER_DIR}/go/`. In this directory are a number of demo scripts, like:

[2] *Regridding* is climate modeller talk for coordinate transformation (of a discretised metric).

```
yes? go mp_demo
```

run a script

which presents the map projection capabilities of Ferret. You can show the content of the script, often serving as a help, by adding the option `/help` to the `go` command:

```
yes? go/help mp_demo
```

show the script

You can also put your own commands in a `.jnl` file and execute it in the same way:

```
yes? go myscript.jnl
```

execute script

provided that `myscript.jnl` is in the `${FER_GO}` path.

If you use vim(1), you may want to use Ferret highlighting: `http://www.gfdl.noaa.gov/~atw/ferret/ferret.vim`. Put this file in `~/.vim/syntax/`.

```
$ mkdir -p ~/.vim/syntax
$ cd ~/.vim/syntax/
$ wget \
> http://www.gfdl.noaa.gov/~atw/ferret/ferret.vim
```

make vim configuration directory if not existing

download vim descriptor file for Ferret

Now you only need to tell vim to use `ferret.vim` when a Ferret script is opened:

```
$ echo "au BufRead,BufNewFile *.jnl set filetype=ferret" »  /.vimrc
$ echo "au!  Syntax ferret source  /.vim/ferret.vim" »  /.vimrc
$ vim -R ${FER_DIR}/go/mp_demo.jnl
```

Future vim sessions should now recognise `.jnl` files as Ferret scripts. As a result, the opened Ferret script will be syntax-highlighted.

2.3 Saving figures

From PyFerret you can save your plots through the *Image* menu.

From plain old Ferret you can save a vector figure in this way:

```
yes? set mode meta
yes? shade votemper[k=1,l=3],nav_lon,nav_lat
yes? can mode meta
```

set metafile mode
make a figure
exit metafile mode

This creates a Metafile[3] that you need to convert to a usable graphics file format, like this (in a Unix shell):

```
$ gksm2ps -p portrait -l cps -d \
> cps -o out.eps metafile.plt
$ epstopdf -outfile=out.pdf out.eps
$ convert -density 300 out.pdf out.jpeg
```

makes PostScript file
(\ and > are line continuation marks)
makes a PDF file (still vector), see `epstopdf(1)`
makes a JPEG, see `convert(1)`

With some Bash scripting you can make this more user-friendly. There is no PDF reader installed on `ada`, but there is on `ipcours`. Copy the file from `ada` to `ipcours` and open the file:

```
ipcours $ scp ada:out.pdf ./
ipcours $ evince out.pdf
```

[3] A *GKS Metafile* is a device and application independent graphics data description, similar to DVI for LaTeX.

3 Exercises

If you have problems, ask one of the teachers or look it up in the Ferret documentation.

1. Load `piControl2pm01_SE_2100_2109_1M_grid_T` and plot the January salinity at 3000 m depth.
2. Plot the January salinity at 5500 m depth. Replot with a useful `/levels=()` range.
3. Plot the yearly average interpolated (using `fill` rather than `shade`) meridional transect through the Atlantic Ocean of the salinity (use `l=@ave` for the yearly average).
4. Plot the northern-hemisphere ($j > 74$) average seasonal cycle of the SST (using `plot`).
5. Do the exercises from the official tutorial aka *Ferret Tour*: `http://ferret.pmel.noaa.gov/Ferret/documentation/ferret-tutorial/`. While the first half of this tutorial focuses on analytical functions, the second half may be more useful as that is about real data.

4 Useful links

- Ferret homepage: `http://ferret.pmel.noaa.gov/Ferret/`
- Ferret documentation: `http://ferret.pmel.noaa.gov/Ferret/documentation/`
- Ferret tutorial: `http://ferret.pmel.noaa.gov/Ferret/documentation/ferret-tutorial`
- E-mail archive: `http://ferret.pmel.noaa.gov/static/Ferret_search/ferret_mail_archives.html`[4]
- PyFerret: `http://ferret.pmel.noaa.gov/Ferret/documentation/pyferret`
- IPython-Ferret extension: `https://github.com/PBrockmann/ipython-ferretmagic`
- *FAST-ATLAS* Ferret library and related info on climate data: `http://dods.ipsl.jussieu.fr/fast/`
- *comPlot*, a visual model–data comparison Ferret library: `http://klimato.org/complot/` (alpha version – feedback welcome)

[4] Tip: some web browsers add a search engine when adding `https://www.google.fr/search?q=%s&sitesearch=http://www.pmel.noaa.gov/maillists/tmap/ferret_users/` to your bookmarks (including `%s` does the trick).