CLIO's nested grids, NetCDF and Ferret

or

The Good, the Bad and the Ugly

(in arbitrary order)

Peter Herrmann

Peter Herrmann scientific computing Kiel

March 2004

Contents

List of Figures

1 Introduction

This document describes the interplay of the revised NetCDF–output routines of CLIO with Ferret. CLIO (Coupled Large-scale Ice Ocean model) is part of the coupled climate system model ECBILT/CLIO and is documented in detail (*Campin, 1997; Goosse et al., 2000*). CLIO utilizes a global grid consisting of a rotated subgrid with the North Pole shifted to the Equator for the Arctic and North Atlantic, connected to a conventional longitude–latitude grid for the rest of the globe in the equatorial Atlantic. In contrast with the atmospheric component, the pristine CLIO model does not use NetCDF–output on the native model grid, but interpolates state variables onto a regular longitude–latitude grid and dumps them to machine–dependent binary files, suitable, e.g., for GrADS. Unfortunately, the interpolation to the regular grid prevents the user from further analysis, because algebraic calculations suffer from the loss of accuracy.

In a previous work, NetCDF–output has been implemented into the CLIO–model. NetCDF is a data model for scientific data access based on a machine–independent format. It is freely available at *http://www.unidata.ucar.edu/packages/netcdf/*. Because the NetCDF–approach is a very general concept, it had to be taken care of some conventions to assure that other tools (like Ferret) can access the data without problems. Basically, the NetCDF–output implemented into the CLIO model adheres to the COARDS conventions (please refer to *ftp://ftp.unidata.ucar.edu/pub/netcdf/Conventions/COARDS*). The main objective of that previous work was to store data on the native model grid and make them accessible to Ferret, an interactive computer visualization and analysis package, which is available for free at *http://ferret.pmel.noaa.gov/Ferret/*. This has been achieved by mapping the combined grids to a regular grid of pseudo–longitudes and –latitudes (reaching virtual latitudes of 120°N). Even though this type of mapping is also found in the CLIO documentation (*Campin, 1997*, Fig. II.4, pages 70 ff.), it is somewhat unusual, and partly misleading, especially in combination with vector plots.

Thus, the NetCDF–output implemeted into CLIO has been revised to allow utilization of the so called 3–argument shade command of Ferret, and extended to assist the user in vector rotation. The next section details the revised subroutines of CLIO, then the use of Ferret to shade/contour data in a more conventional projection is demonstrated, before a way to display vector data in a correct way is described. It should be mentioned here that pointing to apparent flaws, quirks or inconveniences with the software in the following sections should not be regarded as serious criticisms (it can't be acknowlegded enough when people offer their work for free), but may instead reflect the author's poor knowledge of CLIO, NetCDF and Ferret, or even models, data management and visualization concepts in general.

2 Revision of the NetCDF–output in CLIO

The previous implementation of NetCDF–output in CLIO is based on a mapping of the two grids in pseudo–coordinates. In routine *defgrid.f*, arrays of true gridpoint longitudes and latitudes have been defined for both tracer type variables and momentum to reflect the B–grid layout of the model. Following the NetCDF–conventions, edges of the respective gridcells have been defined, such that each gridpoint (i,j), located at (lon(i,j),lat(i,j)) is surrounded by a gridcell with edges (lon_edges(i,j,4), lat_edges(i,j,4)). In this particular case of the B–grid layout, edges of the tracer type gridcells are given by momentum points and vice versa. In the entire (i,j)–domain, there are a lot of undefined coordinates, the so–called "outer space", that do neither correspond to the rotated North Atlantic/Arctic grid, nor to the regular longitude–latitude grid of the rest of the globe. When doing simple shade commands in pseudo–coordinates, Ferret ignores the gridcell edges (lon_edges(i,j,4),lat_edges(i,j,4)) and places cell boundaries just in the middle of the gridpoints. In general, state variables are flagged as missing in the outer space in *outave.f*, and there arises no problem with the plots despite the fact that the true model geometry does not follow lines of constant (pseudo–) longitude and latitude in the North Atlantic/Arctic, and the unusual map projection.

More recent versions of Ferret are able to process so–called 3–argument shade commands `shade temp,tlons,tlats`. Even though all the edges have been correctly defined in (lon_edges(i,j,4), lat_edges(i,j,4)), Ferret ignores this information, and instead operates in one of two modes:

1. When the dimensions of `tlons` and `tlats` are the same as those of `temp`, Ferret shades boxes around gridpoint (tlon(i,j),tlat(i,j)), estimating the rims of the gridcell from the averages of the surrounding gridpoints. Thus, the size and shape of a single tracer cell temp(i,j) is determined by nine coordinate pairs in the range i-1 to i+1 and j-1 to j+1.

2. When the dimensions of both `tlons` and `tlats` exceeds those of `temp` by one in each dimension, `tlons` and `tlats` are assumend to represent the corners of the cell, not the position of the individual gridpoint. Four coordinate pairs in the range i to i+1 and j to j+1 are used with this method.

The first method represents a quick and dirty mapping method, the second one the realistic representation of the model geometry. It will be demonstrated in the next section that none of the above methods is sufficient to correctly render an image of a horizontal field in case of non–monotonic coordinates in a global model, but as an extension of the first implementation of the NetCDF–output in CLIO, additional coordinates tlonsp, tlatsp, ulonsp and ulatsp have been defined in *defgrid.f*, the p denoting the "plus one" in each dimension. With respect to the B–grid layout, the tracer grid corners are given by the momentum gridpoints and vice versa.

Ferret seems to have problems with undefined coordinates in the "outer space". These regions have thus been filled with very slim cells adjacent to the rim cells of the realistic grids. The state variables in these cells will nevertheless be flagged as missing in *outave.f*.

When taking into account the rotation of the grid in the North Atlantic/Arctic region, it must be considered that in CLIO this is only a transformation of the grid, not of the variables. Zonal momentum is directed eastward on whatever subgrid, and the third section of this document does demonstrate that simple `vector u,v` commands in Ferret fail to correctly represent the flow field without rotating the vectors. To assist the user in this rotation, the angle of inclination of the grid against a simple longitude–latitude grid is also computed in *defgrid.f*. It should be mentioned here that when rotating a grid, it is also possible to transform the variables as well. This is done in the SPEM–/SCRUM/–ROMS–family, where velocities in the model are directed parallel and normal to curvilinear coordinate lines.

Changes applied to *defgrid.f* are fairly well documented by comments beginning with `cph`. Additional arrays have been defined in *bloc.com*. *outave.f* has been extended to handle the additional coordinate fields. The routine is a bit lengthy due to the administrative nature of NetCDF, but the coding is straight forward and easy to understand. It is assumed that routines *defgrid.f*, *outave.f* and the corresponding include file *bloc.com* distributed with this document replace their respective earlier versions in the *clio/sources* directory in the EMIC source tree, and that at least one year has been simulated with the new code.

3 Horizontal mapping with the 3–argument shade command

Let's fire up ferret and have a look at sea surface height in pseudo–coordinates as a starting point:

```
bash: ferret
        NOAA/PMEL TMAP
        FERRET v5.53
        Linux(g77) 2.4.20 - 10/16/03
        18-Mar-04 20:09

yes? use CLIO3_NEWcdf_ann_avrg_0001.nc
yes? set region/l=1
yes? shade ssh
yes? cont/ov/col=1/lev=(0.,420.,30.)/nolab tlon
yes? cont/ov/col=1/lev=(-90.,90.,30.)/nolab tlat
```

The resulting figure 1 resembles the aforementioned figures in *Campin, 1997*, but is overlayed here with contours of true gridpoint longitudes and latitudes. More advanced Ferretist will certainly manage to produce fancier figures, but the true coordinates were plotted to illustrate the fact that the coastline in the North Atlantic/Arctic sector does not represent the true geometry of the model, and that the crowding of contours east (in pseudo–longitude) of the

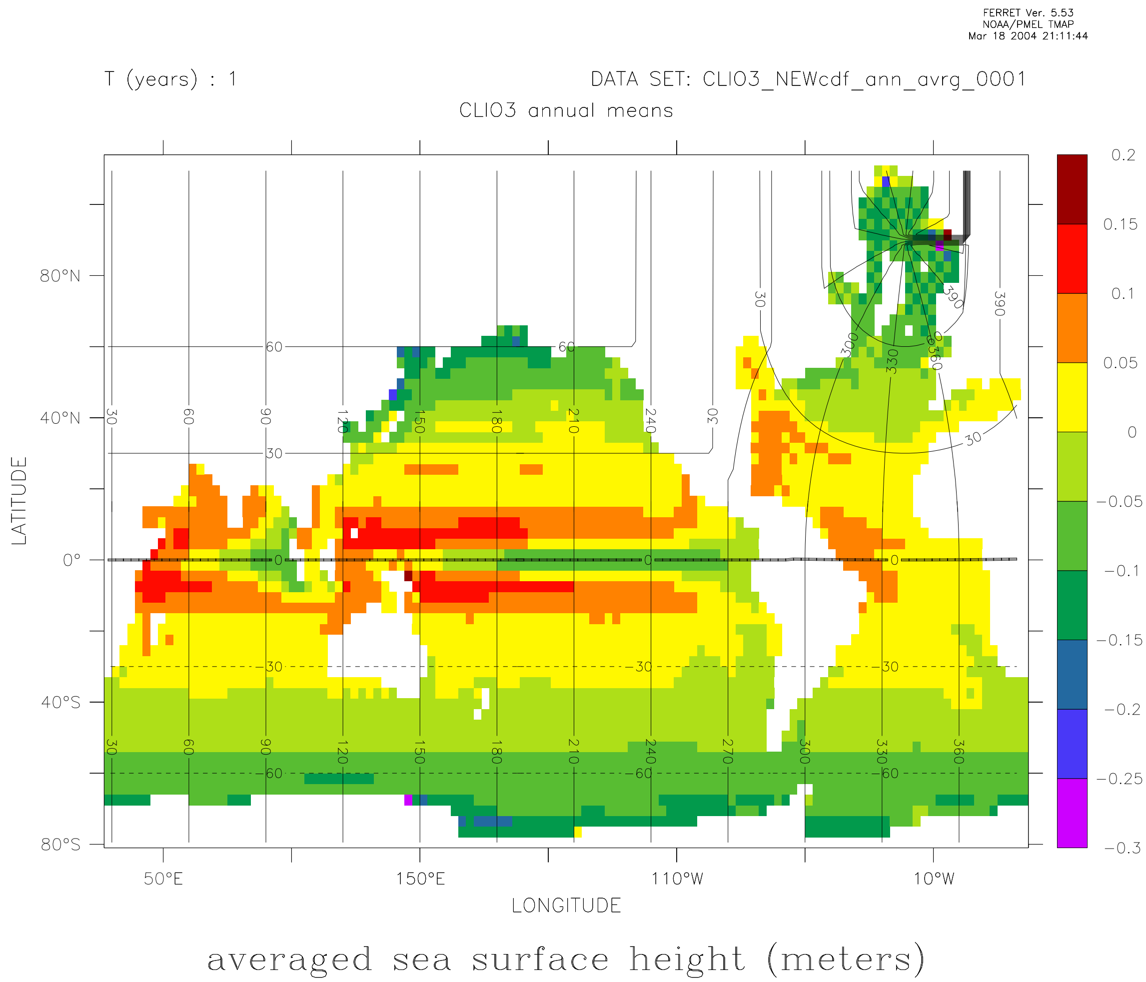

Figure 1: Annual mean field of sea surface height [m] plotted in pseudo–coordinates. Overlayed by lines of constant longitude and latitude in 30° intervals. Note the non–monotonicity of true longitude near the North Pole.

North Pole represents a discontinuity, the most problematic region to deal with when transforming the figure to true geographic coordinates. A third point to mention is related to the representation of vectors, and will be dealt with in detail in the following section. Nevertheless, the user can scroll horizontally or extract arbitrary regions with the set region command when using pseudo–coordinates.

As already mentioned above, recent versions of Ferret are able to transform the figure with the 3–argument shade command:

```
yes? shade ssh,tlon,tlat
```

Unfortunately, Ferret does not utilize the full edges information provided within the NetCDF–dataset, but operates in one of the two modes outlined in the previous section. Obviously (but not depicted here), Ferret fails to represent the sea surface height field, when coordinates are not monotonic. But a simple trick helps:

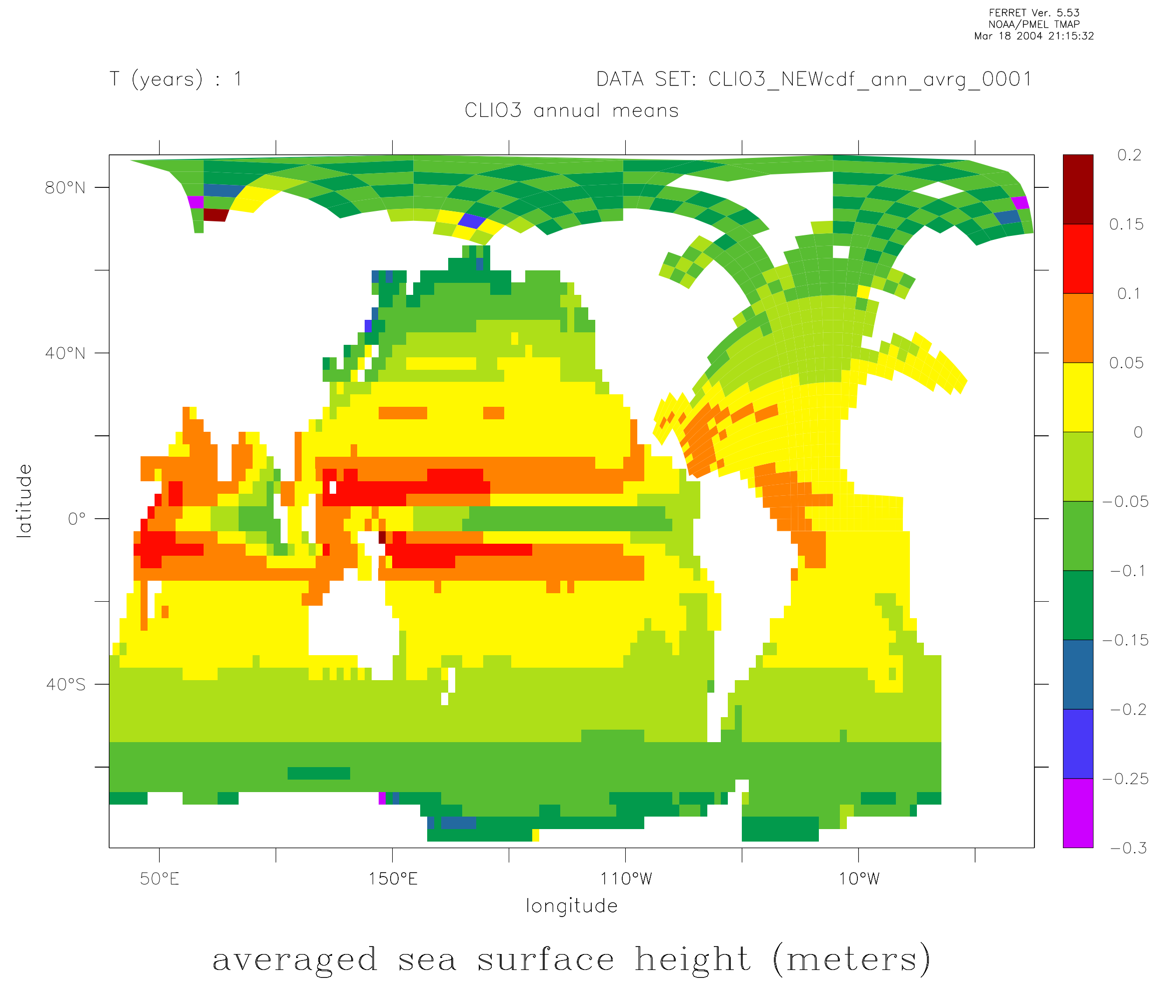

Figure 2: Annual mean field of sea surface height [m] plotted in a conventional longitude–latitude mapping. Note that the nine point algorithm is used to estimate the gridcell boundaries.

```
yes? shade/lev=(-0.3,0.2,0.05)/j=1:57 ssh,tlon,tlat
yes? shade/lev=(-0.3,0.2,0.05)/j=57:65/ov/nolab ssh,mod(tlon,360.),tlat
```

The resulting figure 2 deserves some attention:

- The modulo function applied to the longitudes in the non–monotonic region does all the trick. Instead of a single shade, you will need two to render the picture correctly. Not as simple as stated in the Ferret User's Guide, but worth the effort.
- The entire globe is plotted, even though in the lower right corner something seems to miss. It's all there, you will find Cape Agulhas in the bottom left, and the Southern Ocean extends to the "white space" in the bottom rigth.
- The Arctic wraps around in a cyclic way: starting at Spitzbergen in the top right of the figure, move two boxes to the right, and reenter the figure at the top left.

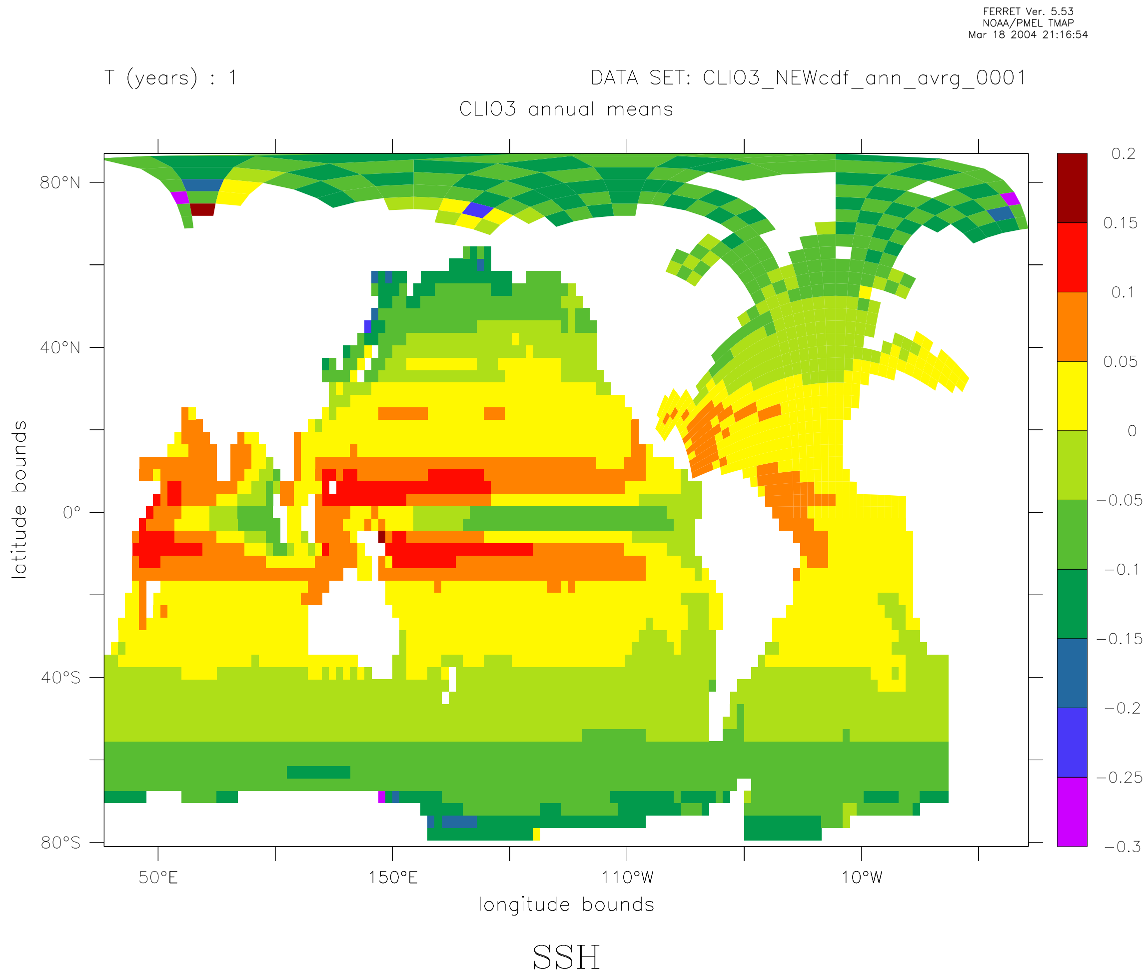

Figure 3: Annual mean field of sea surface height [m] plotted in a conventional longitude–latitude mapping. Note that the four point algorithm is used to determine the gridcell boundaries, showing a closed Bering Strait.

- Note that this way the Mediterranean is not wrapped around from the middle right to the middle left.
- Setting zonal limits and then transforming the figure will fail, you can only depict the entire globe in the way described above. Latitudinal limits seem to work, however.
- This form of the shade command utilizes the nine point interpolation to find the gridcell edges. Note that Bering Strait seems to be open, even though it is definitely closed and the flow through it is parameterized.
- The four point algorithm can be employed with the newly defined boundaries tlonp, tlatp (figure 3).
- State variables on the momentum grid can be plotted accordingly using the respective set of coordinates: ulon, ulat and ulonp, ulatp.

- shade of sea surface height is a wonderful means to show the structure of the grid due to the 2 dx structures inherent to the numerical scheme; the user may overlay this with a final fill (figure 4). But astonishingly only the nine point algorithm works with fill!

```
yes? shade/lev=(-0.3,0.2,0.05)/j=1:57 ssh,tlonp,tlatp
yes? shade/lev=(-0.3,0.2,0.05)/j=57:65/ov/nolab ssh,mod(tlonp,360.),tlatp
yes? fill/lev=(-0.3,0.2,0.05)/j=1:57/ov/nolab ssh,tlon,tlat
yes? fill/lev=(-0.3,0.2,0.05)/j=57:65/ov/nolab ssh,mod(tlon,360.),tlat
```

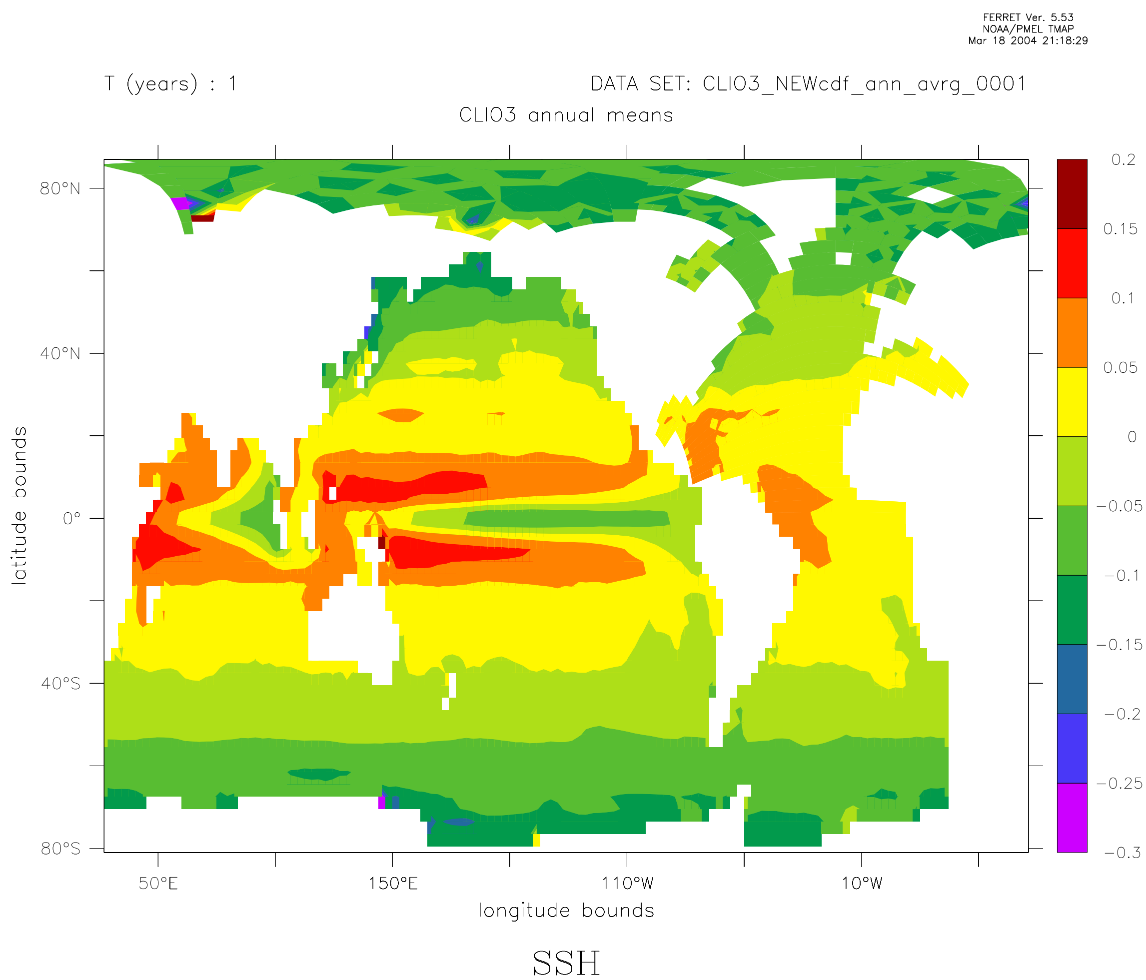

Figure 4: Annual mean field of sea surface height [m] plotted in a conventional longitude–latitude mapping. Note that the four point shade is overlayed with a nine point fill.

The author of this document thinks that future releases of Ferret could do a better job using the true cell edges provided in the NetCDF–dataset, because this does no longer require monotonicity of the coordinates and bounds. The fact that fill only works with the nine point algorithm clearly illustrates that the present implementation of the 3–argument shade command in Ferret can only be regarded as a first step towards a more general handling of curvilinear coordinates. Nevertheless, using two shade commands instead of one and the simple trick with the modulo function applied to the non–monotonic coordinates requires only minor additional effort to render fieleds correctly. It should be stated here that the user does not need to redo all his previous numerical experiments to plot his data in the new form. Just open the ancient dataset after this new one, and use coordinates from [d=1], but state variables from [d=2].

4 Rotating vectors

As mentioned earlier, CLIO partly rotates its grid, but not the vectors. Zonal windstress is always directed eastward, as is zonal velocity. To figure out what both the mapping and Ferret do to the vectors, let's simply define a purely zonal windstress of unity and draw the vectors in pseudo–coordinates:

```
yes? let taux if wsx then 1
yes? let tauy if wsy then 0
yes? set region/x=120W:10e/y=20S:120N
yes? shade ssh
yes? vec/ov/len=2/xs=1/ys=1 taux,tauy
yes? cont/ov/col=3/lev=(0.,420.,30.)/nolab tlon
yes? cont/ov/col=3/lev=(-90.,90.,30.)/nolab tlat
```

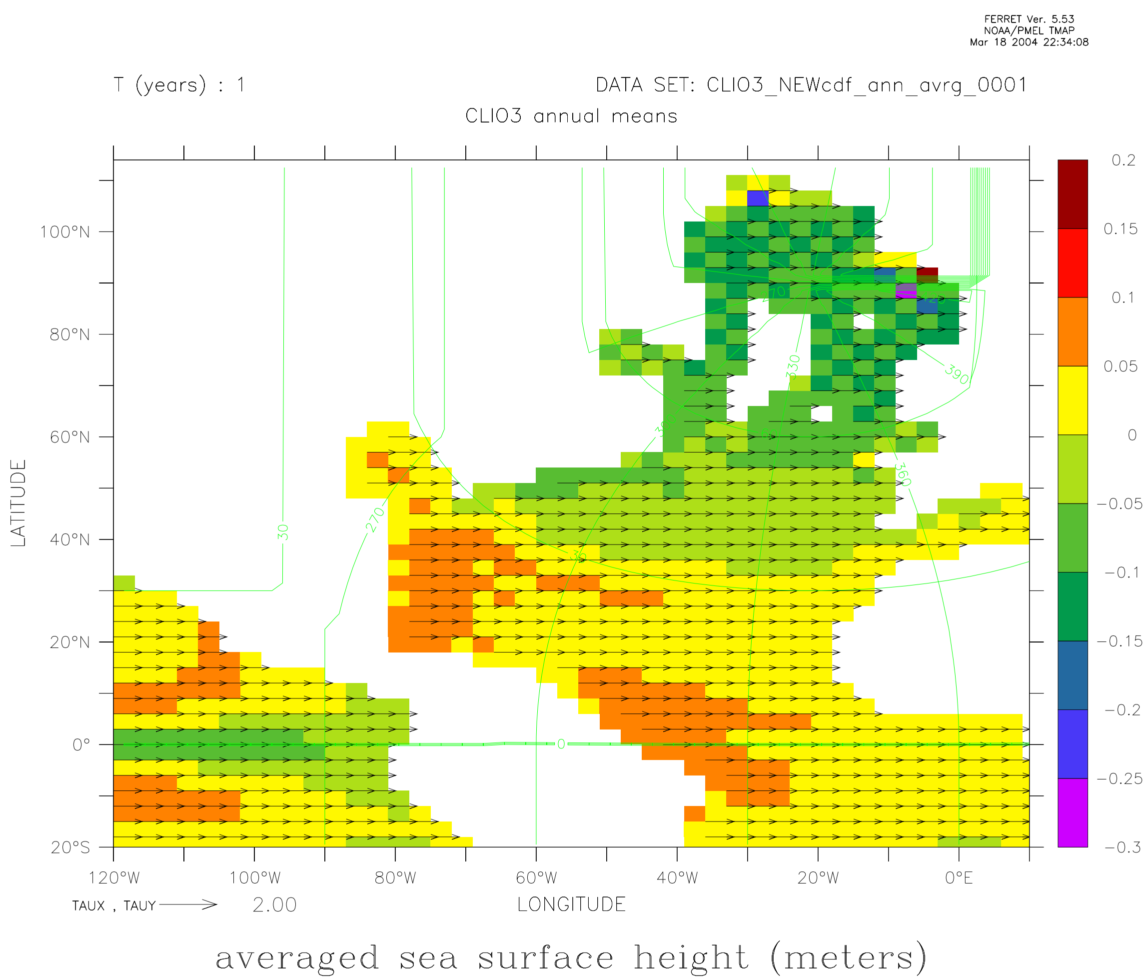

Figure 5: Vectors of purely zonal windstress of unity over shades of sea surface height in pseudo–coordinates. Vectors are not rotated with the grid.

Obviously, the vectors in figure 5 point from left to right, but do not correctly represent a zonal windstress in the polar region. Instead, a rotation of the vectors is required. The angle of inclination of the grid has been computed in *defgrid.f* and stored in the NetCDF–dataset by *outave.f*:

```
yes? let stressx=taux*cos(angle)+tauy*sin(angle)
yes? let stressy=-1.0*taux*sin(angle)+tauy*cos(angle)
yes? shade ssh
yes? vec/ov/len=2/xs=1/ys=1 stressx,stressy
yes? cont/ov/col=3/lev=(-90.,90.,30.)/nolab tlat
yes? cont/ov/col=3/lev=(0.,420.,30.)/nolab tlon
```

Figure 6 demonstrates that the rotated vectors are parallel to lines of constant (true) latitude, correctly representing the purely zonal windstress field.

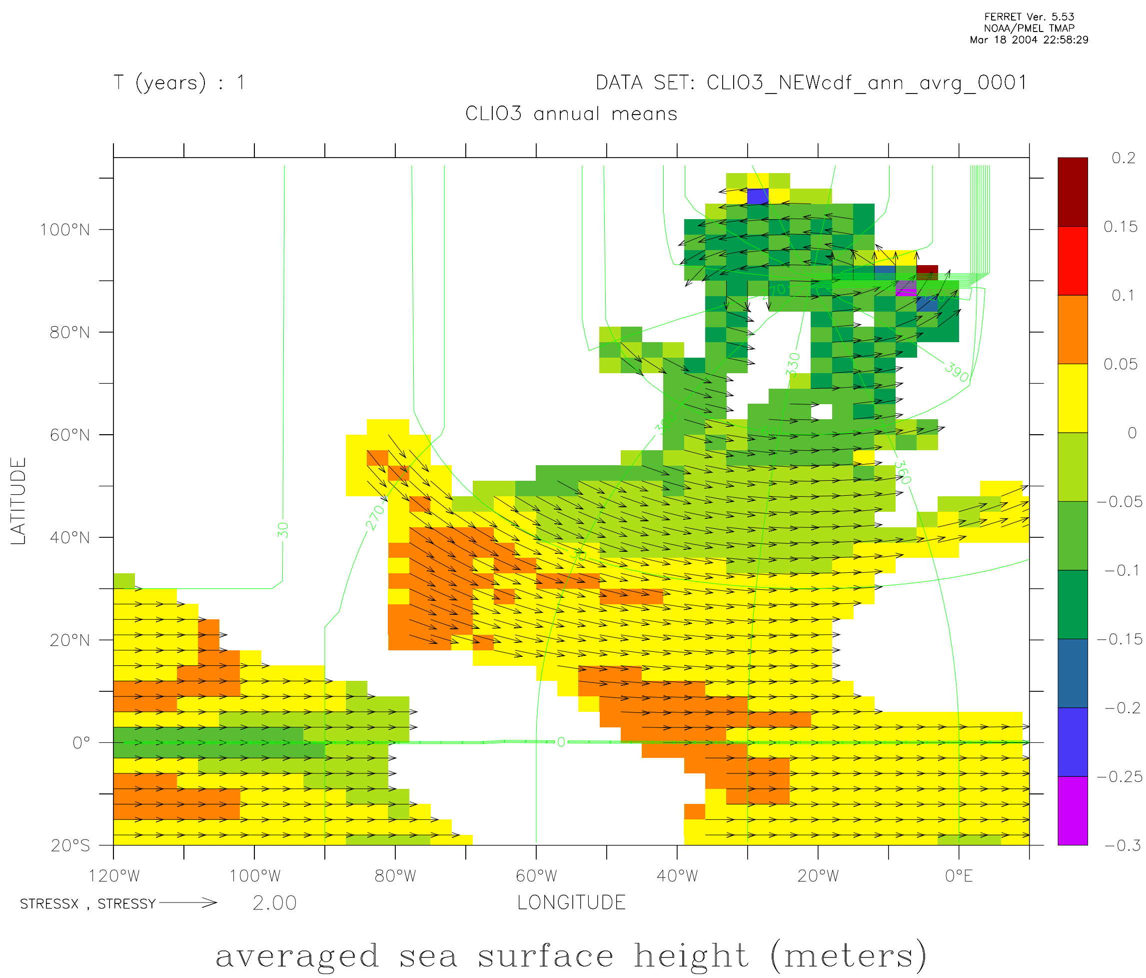

Figure 6: Vectors of purely zonal windstress of unity over shades of sea surface height in pseudo–coordinates. Vectors are rotated with the grid.

Let us try to find out how Ferret operates with unrotated vectors in true coordinates:

```
yes? cancel region/all
yes? shade/lev=(-0.3,0.2,0.05)/j=25:57 ssh,tlonp,tlatp
yes? shade/lev=(-0.3,0.2,0.05)/ov/j=57:65/nolab ssh,mod(tlonp,360.),tlatp
yes? vec/ov/len=2/xs=1/ys=1 taux,tauy,ulon,ulat
```

The resulting figure 7 demonstrates that on the rotated grid the vectors are parallel to coordinate lines, hus point into the wrong direction and need to be rotated in the same way as with pseudo–coordinates:

```
yes? shade/lev=(-0.3,0.2,0.05)/j=25:57 ssh,tlonp,tlatp
yes? shade/lev=(-0.3,0.2,0.05)/ov/j=57:65/nolab ssh,mod(tlonp,360.),tlatp
yes? vec/ov/len=2/xs=1/ys=1 stressx,stressy,ulon,ulat
```

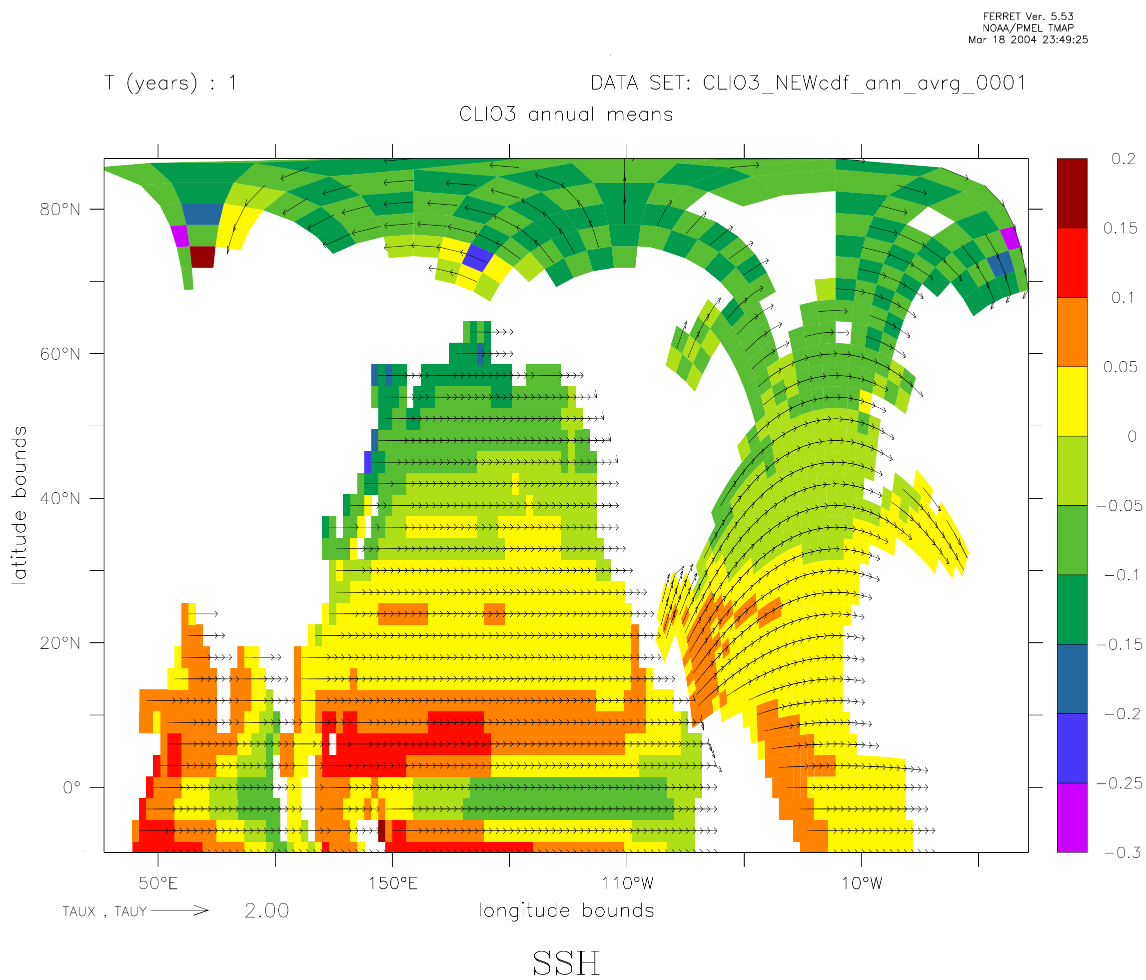

Figure 7: Vectors of purely zonal windstress of unity over shades of sea surface height in true coordinates. Vectors are not rotated with the grid.

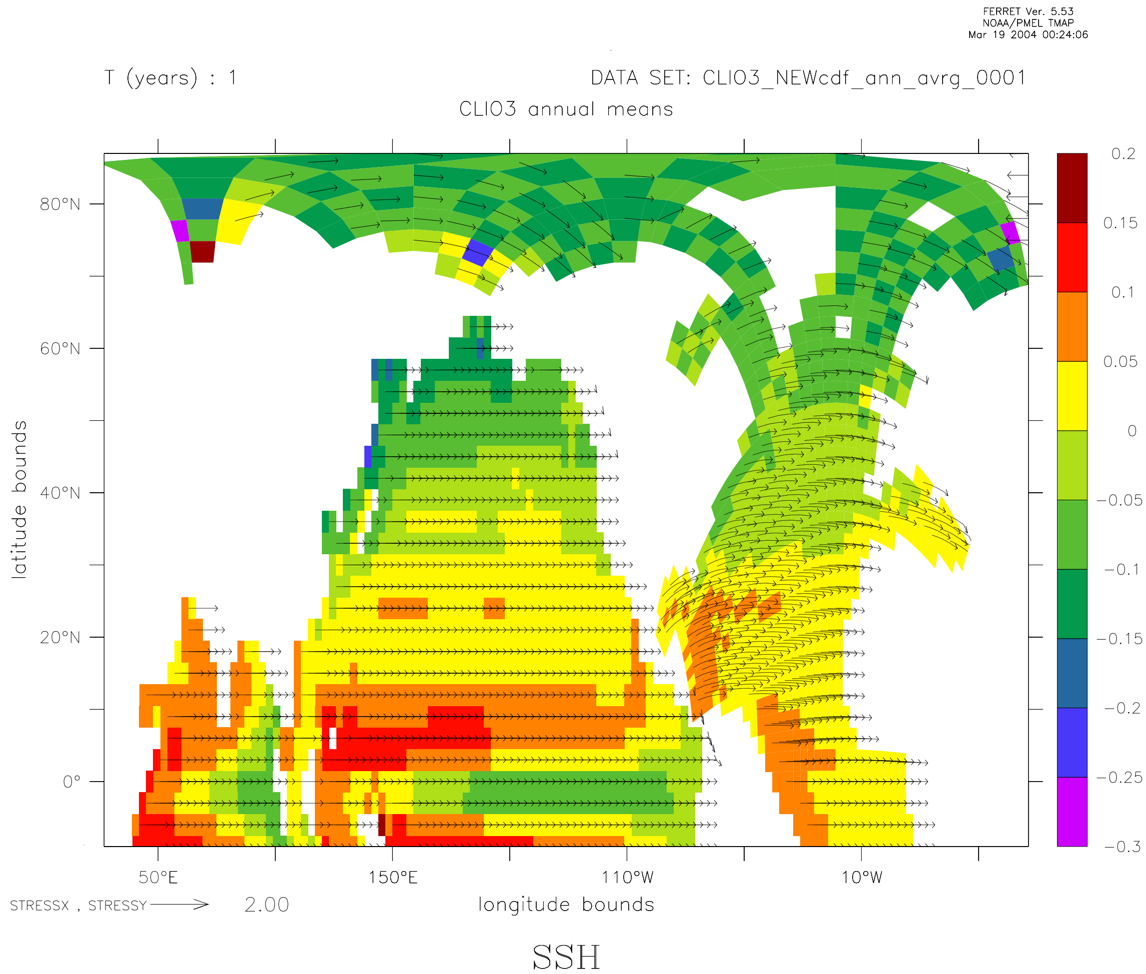

Figure 8: Vectors of purely zonal windstress of unity over shades of sea surface height in true coordinates. Vectors are rotated with the grid.

Note how the vectors in the Arctic change their directions and now point from left to right as intended. Nevertheless, there is still a slight curvature north of 70°N and close to land masses. This may be due to

1. the fact that we attempt to draw a simple longitude–latitude map up to 90°N, which is generally impossible;

2. computational errors in rotating the vectors;

3. some averaging of coordinates and angles Ferret seems to apply when plotting. Note that the four point shade algorithm has been used; each square should thus represent a single tracer cell. Accordingly, the vectors should arise from the gridcell corners. Instead, they reside in the middle of a box side. Ferret seems to perform some averaging, and since the angle is zero over land, this may explain the relatively strong curvature in the Carribean.

5 Summary and conclusions

The NetCDF–output of CLIO has been revised to allow for more conventional longitude–latitude maps of state variables from model snapshots or time–averages. Unlike the original binary dumps, history files are not interpolated to a regular grid, but still reside on the model grid, allowing further analysis of the data.

With Ferret, arbitrary regions can be defined for analysis and visualization in pseudo–coordinates. The resulting maps reveal an unusual projection, however, and in the area of the rotated grid, the representation of the model geometry is not exact. Vectors need to be rotated to point into the right direction.

Utilizing the 3–argument shade command allows a more conventional map projection, even though a true equidistant lon–lat map up to the North Pole is geometrically impossible. There is a minor loss of flexibility in the selection of zonal limits, the user always needs to plot a full longitudinal circle. A simple workaround using the `modulo` function overcomes the problems arising from the discontinuity of true longitudes in the rotated subgrid. Two `shade` commands instead of one are an affordable price for much nicer plots. As in pseudo–coordinates, vectors need to be rotated to correctly represent the corresponding flow field. Minor uncertainties remain how Ferret exactly averages coordinates and angles, and finally places vectors – this is a drawback when using precompiled binaries without exploring the sources in detail. This would be a task far beyond the scope of this work.

There is hope however that future releases of Ferret will provide means to operate with more generalized curvilinear coordinates.

6 References

Campin, J.M., 1997: Modélisation tridimensionelle de la circulation générale océanique lors du dernier maximum glaciaire. PhD Thesis, Université Catholique de Louvain, Louvain–la–Neuve, Belgium, 342pp.

Goosse, H., J.M. Campin, E. Deleersnijder, T. Fichefet, P.P. Mathieu, M.A. Morales–Maqueda, and B. Tartainville, 2000: Description of the CLIO model – version 3.0. *ftp://ftp.astr.ucl.ac.be/pub/IGL/clio30.pdf*, 49pp.

Hankin, S., J. Callahan, A. Manke, and K. O'Brien, 2004: Ferret User's Guide Version 5.53. *http://ferret.pmel.noaa.gov/Ferret/Documentation/Users_Guide/current/fer_html.htm*

UNIDATA, 2001: NetCDF–conventions. *http://www.unidata.ucar.edu/packages/netcdf/conventions.html*