HopsGui, a Graphical User Interface for HOPS model packages

By

Pedro Vélez Belchí, (UIB - IEO) July 1999

Hopsgui is a set of MATLAB (v5.X) routines to assist in the visualization of HOPS model outputs as well as in the necessary previous steps to set up the model. The package allows the user to visualize the data as depth slices at fixed depths, horizontal plots at fixed sigma level, vertical slices at fixed i,j index, timeseries, profiles, ...There is also the possibility of doing animations and save the images into postscript files. The package is composed by three different applications: *hopsgui*, to visualize model outputs; *oaggui*, to visualize the results of the objective analysis package, and *peigui* to visualize the initialization fields for the HOPS model.

1.1 Installing the software:

1.2 Starting Up hopsgui

1.1 Installing the software

Once you have downloaded the compress file, decompress it (gunzip or zip) and , if it is the case, get the files out form the tar file (the command is *tar -xvf* gui.tar). Right after the decompression you will have a directory structure that looks like:

```
Gui_hops_oag/
      ./Hopsgui
      ./Peigui
      ./Oaggui
       ./additional
            ./netcdf
            ./matlab_netcdf_4x5
```

Then, you have to add the path to the matlab, the command is :

path(path,'/Gui_hops_oag/');
path(path,'/Gui_hops_oag/Hopsgui');
path(path,'/Gui_hops_oag/Oaggui');
path(path,'/Gui_hops_oag/Peigui');
path(path,'/Gui_hops_oag/additional');
path(path,'/Gui_hops_oag/additional/matlab_netcdf_4x5');
path(path, fullfile('/Gui_hops_oag/additional', 'netcdf', ''));
path(path, fullfile('/Gui_hops_oag/additional', 'netcdf', 'nctype', ''));
path(path, fullfile('/Gui_hops_oag/additional', 'netcdf', 'ncutility', ''))

and **read the following section** of this user guides.

1.2 Starting Up hopsgui

From within the MATLAB environment execute hopsgui:

>>hopsgui

Once hopsgui starts up, you should see a matlab window.

The first thing you should do is go to *file menu* and open a valid HOPS NetCdf output file, usually *pe_out.nc*. Right after the file has been opened, selected a variable to work with. The Variable menu brings up a list of choices, this list of choices depends on the netCDF file you are working with, therefore for understand the names of the variables you should look to the ascii version of the NetCdf File. By default temperature (*temp*) is selected. Once the appropriate variable has been selected, the data model can be visualized in four different ways or trough an animation:

1.2.1 Depths slices

Horizontal plots at fixed depth can be visualized, to do that go to the Depth slider and select the depth where you want the horizontal slide, this will take some time as hopsgui have to interpolate from sigma levels to pressure levels. Once the plot have been done there are several actions that can be done over the plot:

- To Overlay the vector field just press the radiobutton that said *'Overlay Arrows'*. There are some values that can be modified (*menu: Default Values>Velocity Settings*) to make the vector field as easy to read as possible: the frequency of the arrows, the size of the reference vector, the scale of the vectors and the color of the arrow, of course the field that is visualized can be modified, by default is the total one but can be changed to the baroclinic or the geostrophic depending of the NetCdf File.
- To Overlay the land mask just press the radiobutton that said *'Overlay Land Mask'*.
- To Overlay the topography just press the radiobutton that said *'Overlay Topography'*.
- To add grid just press the radiobutton that said *'Grid'*.
- To add contour lines between the contours jut press the radiobutton *'Contour Lines'*.
- To add labels to the contour lines push the radiobutton *'Contour Labels'*, the labels will be draw at the automatic selected contour levels, although this can be modified, see below.
- To modify the levels at which are drawn the contours, from automatic to just manual, first select the minimum, the maximum and the interval between levels and after that push the radiobutton *'Automatic Contour'* and the automatic contouring will be disable and the new contour plotted.
- To zoom the graph push the radiobutton *'zoom'* and click inthe graph as the normal matlab zoom
- To modify the color palette select in the Color menu the one you want or just write in the prompt; >>colormap(*palette*)

All this actions have memory, what it means that further plots will keep the selection you have already done, in case you want disable an action just click again in the corresponding radio button.

In the title of the plot is always the title of the run, the path of the file, the depth, kindex, index or jindex of the slice and the time of the run. The time of the run can be referenced to an absolute time, by default is zero, but can be modified in the Menu: *Default Values>Absolute time*.

1.2.2 k Sections

These sections are realized trough a constant sigma level (selected in the *Kindex slider*), they are not very useful for physical purposes, but they are useful for debugging. Once the plot is done is possible to: *'Overlay Arrows'*, *'Contour Labels'*, *'Contour Lines'*, *'Overlay Land Mask'*, change the *'Automatic Contour'* and *'zoom'* in the same way as in depth sections.

1.2.3 I and J Sections

These sections are realized trough a constant i(j) index in the model domain (selected in the *i(j)index slider*), and correspond, in case of a non-oblique domain, to latitudinal and longitudinal sections. Here is also possible to add labels to the contour levels (*’Contour Labels’*), modify the levels you want to be contoured (*’Automatic Contour’*) and do a zoom.

1.2.4 Time Series and profiles.

The software allows you to visualize time series, profiles and time-profiles (a contour plot with the x-axis been the time and the y-axis the depth) at specific positions. To do so, go to the menu: *Others* and chose timeseries, profiles or timeprofiles. Just after that a ’cross’ will allow you to select the geographic localization of the timeseries/profile/timeprofile. You have to select two points as the timeseries/profile/timeprofile will be plotted for two different localizations. Of course, the variable is the current one and the depth/time will be the current one.

1.2.5 Adding the coast line.

Just write hopgui_realcoast in the prompt,

```
>>hopsgui_realcoastline
```

although first you have to be sure that the path for the correct coastline file is correct in the hopsgui_realcoast.m file.

1.2.6 Animations

Animations in time for depth-slices, i-slices, j-slices and k-slices can be done. First you have to select the depth,k-i-j index as well as the options you want: *’overlay vectors’*, *’contour labels’*, ’overlay land mask’, *’contour lines’*, *’automatic contouring’* and *’zoom’*. After, open the menu: *Animation>Make New* and push the ’*Make animation*’ button in the animation window. You will see that the graphs for each time step are visualized in the main window and then storage, after the animation is done you can play it (first select the Frame per second ’*FPS*’ and the number of times the animation is played ’*Times*’ in the animation window) pushing the button *’Play animation’*. There is also the possibility of save the animation into a file and reproduce it later under the menu: *Animation>Play saved*.

1.2.7 Printing Graphs

All the different slices can be plotted towards the printer or to a file, either postscript (.ps) or postscript encapsulated (.eps), to do so go to the file menu and choose the corresponding option.

1.2.8 Getting the data

The sofware also permits to obtain the data that is actualy used to make the plot. Just write:

```
>> [field,x,y,z]=hopsgui_getdata;
```

And the data will be saved as weel as the coordinates. This option can be used for whatever variable is selected, for instance, for depth slices the depth is the actual one (saved in z) and the latitude and longitude will be stored at x,y. This option do not work with the data plotted using the *timeseries/profiles/timeprofiles* option, but in that case you can easily make a matlab scripts with the corresponding hopsgui_*functions* (*hopsgui_extprofile.m*, *hopsgui_extimeseries.m*, *hopsgui_eximeprofiles.m*) to get the data.

1.2.9 Plotting drifters trayectories

It is also possible to plot (not in the main window) the trayectories for drifters saved in pe_trk.nc (In case the trayectories are saved in other file, please edit hopsgui_drifters.m and modify there the input file). To do so, write:

>>hopsgui_drifters

A small menu inthe prompt will ask you what floats do you want to plot, once you have answer, the program will open a window for each drifter and plot the trayectories (there is a square evey day overimposed to the solid line), and the end it will create a new window with all the trayectories togheter.

1.3 Starting up *peigui* and *oaggui*

Together with the *hopsgui* application, there are two other applications: *Peigui*, to visualized the output files from the pe_initial step in the HOPS package; and *Oaggui,* to visualize the output files from the oag step in the HOPS package. To use then just write:

>> oaggui
or
>>peigui

These two applications also works in case on assimilation. As both applications have been build from *Hopsgui* the menus, contouring and other instructions are very similar to hopsgui and the previous notes are enough.

1.4 The MATLAB code

Here there is a brief explanation of the way the *hopsgui* code works, in case you are interested in modify it, the clues here are enough. As the code for *hopsgui*, *peigui* and *oaggui* are based in the same philosophy, this brief explanation will be enough for understand the three packages.

The main function is *hopsgui*, where all the initial variables are defined. All those initial values (and successive) are defined in a cell array called hopsgui_*obj{1}* All the parameters are passed to the subroutines trough this cell array, except the data it self that is never stored due to Memory problems at the moment of this work.

The basic *hopsgui_obj{1}* structure fields is here:

General purpose variables

- *hopsgui_obj{1}*.filein - Name of the Mexcdf file
- *hopsgui_obj{1}*.variable - Variable to plot
- *hopsgui_obj{1}*.time - Actual Time

- *hopsgui_obj{1}*.depth - Actual depth
- *hopsgui_obj{1}*.iindex - Actual iindex for slices
- *hopsgui_obj{1}*.jindex - Actual jindex for slices
- *hopsgui_obj{1}*.kindex - Actual jindex for slices
- *hopsgui_obj{1}*.titlrun - Title of the Run
- *hopsgui_obj{1}*.action - The last action performed
- *hopsgui_obj{1}*.timemax - Maximum value for time
- *hopsgui_obj{1}*.timeinterval - Interval of time between plotted time steps in hours

Values for contour levels

- *hopsgui_obj{1}*.cautomatic - Label to do or not automatic levels in contour plots
- *hopsgui_obj{1}*.cmin - Min value to be ploted
- *hopsgui_obj{1}*.cmax - Max value to be plotted
- *hopsgui_obj{1}*.cint - Interval of the contouring
- *hopsgui_obj{1}*.labelcontours - Flag to put or not labels in the contour plot
- *hopsgui_obj{1}*.linecontours - Flag to put or not contour lines in the contour plot

Default values for drawing vectors

- *hopsgui_obj{1}*.drawvectors - Flag to Overlay or not the vectors
- *hopsgui_obj{1}*.fac - Scale factor for vectors in the screen
- *hopsgui_obj{1}*.isub - Sampling interval for vectors
- *hopsgui_obj{1}*.refvector - Size of the reference vector in units of the read velocity
- *hopsgui_obj{1}*.vectorcolor - Default color for the vectors, k is black,w white,b blue,g green, r red, y yellow, c cyan.
- *hopsgui_obj{1}*.vectorvar - Is the vector field of vector to be draw

Land mask

- *hopsgui_obj{1}*.drawmask - Flag Overlay or not the land mask

Animation

- *hopsgui_obj{1}*.nframes - The lat time step of the movie
- *hopsgui_obj{1}*.fps - Frames by second
- *hopsgui_obj{1}*.nta - Number of times the movie is repeated

Default values for the absolute time

- *hopsgui_obj{1}*.absyear
- *hopsgui_obj{1}*.absmonth
- *hopsgui_obj{1}*.absday
- *hopsgui_obj{1}*.abshour - Define the absolute reference time

Handles of the differences objects used

- *hopsgui_obj{1}*.hfigure - Handle for the main figure
- *hopsgui_obj{1}*.haxes1 - Handle for the Main axes
- *hopsgui_obj{1}*.haxes2 - Handle for the Colorbar axes
- *hopsgui_obj{1}*.htextaxes - Handle for the Upper Text in main axis

- *hopsgui_obj{1}*.hedit_iindex - Handle for the edit iindex box
- *hopsgui_obj{1}*.hslider_iindex - Handle for the Slider iindex
- *hopsgui_obj{1}*.hedit_jindex - Handle for the edit jindex box
- *hopsgui_obj{1}*.hslider_jindex - Handle for the slider jindex
- *hopsgui_obj{1}*.hedit_kindex - Handle for the edit jindex box
- *hopsgui_obj{1}*.hslider_kindex - Handle for the slider jindex
- *hopsgui_obj{1}*.hedit_depth - Handle for the edit depth box
- *hopsgui_obj{1}*.hslider_depth - Handle for the slider depth
- *hopsgui_obj{1}*.edit_timestep - Handle for the edit time box
- *hopsgui_obj{1}*.hmenuvar - Handle for the menu of the variables
- *hopsgui_obj{1}*.hvelocitysettingsmenu - Handle for the menu used to change some default values in the vectors
- *hopsgui_obj{1}*.hautominedit - Handle for the edit min box
- *hopsgui_obj{1}*.hautomaxedit - Handle for the edit min box

Once the different default values are defined the main frame is done in the *hopsgui_main.m* file, here all the windows are defined except those for velocity setting (done in *hopsgui_velocity_settingsmenu.m*), absolute time (*hopsgui_abstimegui.m*) and animations (*hopgui_animationsaved.m* and *hopsgui_animatiogui.m*). All the callbacks for the difference interfaces in hopsgui are evaluated in four different functions:

hopsgui_menu_command is for general purposes and mainly gui, the commands are:

- *openfile,*. Open the input file
- *print_eps, print_lp, print_ps_tall, print_ps,*. Print in different ways
- *new_variable,* set the current variable
- *settime, timeincrement,*. timedecrement set the current time
- *rbvectoroverlay,* Radio button to set or not overlay vetors
- *rbmaskoverlay,* Radio button to set or not overlay mask
- *rblabelcontours*, Radio button to set or not label at contours
- *rbautomaticcontours,* Radio button to set or not automatic contouring
- *getmaxauto,getminauto,*
- *getintauto,* Get information in case of manual contours
- *velocity_settingsmenu,setarrowsampling,*
- *setarrowscale,setarrowcolor,*
- *setarrowreference,* Set different values for drawing vectors
- *fc_edit_depth,fc_slider_depth,*
- *fc_edit_iindex, fc_edit_jindex,*
- *fc_slider_iindex, fc_slider_jindex,*
- *fc_edit_kindex,*
- *fc_slider_kindex*, Manage the sliders for kindex,iindex and jindex
- *setarrowrvariable,* Set the vector field to be plotted
- *getabsyear,getabsmonth,getabsday,getabshour,* Get absolute time

hopsgui_contour_command for the differents contouring, the commands are:

- *depth_contour,* Do depth contours
- *jindex_contour,* Do jindex contours
- *iindex_contour,* Do iindex contours
- *kindex_contour,* Do kindex contours
- *maskoverlay,* Overlay land mask

hopsgui_vector_command, for overlaying vectors, the commands are:

- *animation* for operations just related with animation
- *gnframes,getfps,gettimes,* Get setting for doing and playing the animation
- *sanimation,* Save an animation
- *loadanimation,* Load a previoulsy done animation
- *playloadanimation,* Play a previoulsy done animation
- *panimation,* Play an animation just done
- *manimation,* make the animations

An additional set of subroutines: *hopsgui_2dslice, hopsgui_depthslice, hopsgui_depthsliceu, hopsgui_depthslicev, hopsgui_islice, hopsgui_isliceu, hopsgui_islicev, hopsgui_jslice, hopsgui_jsliceu, hopsgui_jslicev, hopsgui_kslice, hopsgui_ksliceu, hopsgui_kslicev* extract the data from the NetCdf file to produce the sections, for further information just refer to the specified subroutines as all of them are well documented.

1.5 Frequently asked Questions

In this section I have written some question that can help you to modify the code in case you need:

Who can I change the font size of the contourlabels?: Just go to the hopsgui_contour_command.m file and modify the Fontsize field on the extcontour command under the appropriate action (depth_contour, jindex_contour, iindex_contour, kindex_contour).

Who can I produce unfilled contours? Just go to the hopsgui_contour_command.m file and modify the excontour command under the appropriate action (depth_contour, jindex_contour, iindex_contour, kindex_contour). OR, go the colorscale menu, select the *white* one and after make invisible the colorbar: *set(hopsgui_obj{1}.haxes2, 'visible','off');*.

Who can I modify the color of the topographic lines? Go to the hopsgui_contour_command.m, look the section adding topography , and the modify the corresponding contour commad.

1.6 Getting help:

All the subroutines are well documented, and in all of them there is a matlab help that you can read typing:

>> help *subroutine_name*

For a short introduction to hopsgui (or oaggui, peigui) type:
>> help *hopsgui*

Also in each directory there is a file call Contents.m that explain the files contained in the directory, this file is also accessible trough the usual matlab help.
In case you want to send me an e-mail, it will be welcome, althought I do have the purpose, by the moment, to further update the code.

1.7 Additional Notes

You should know that the present software used a couple of subroutines that are not currently installed in matlab (among others the netcdf interface), those subroutines are in the subdirectory ./additional supply in the downloaded file.