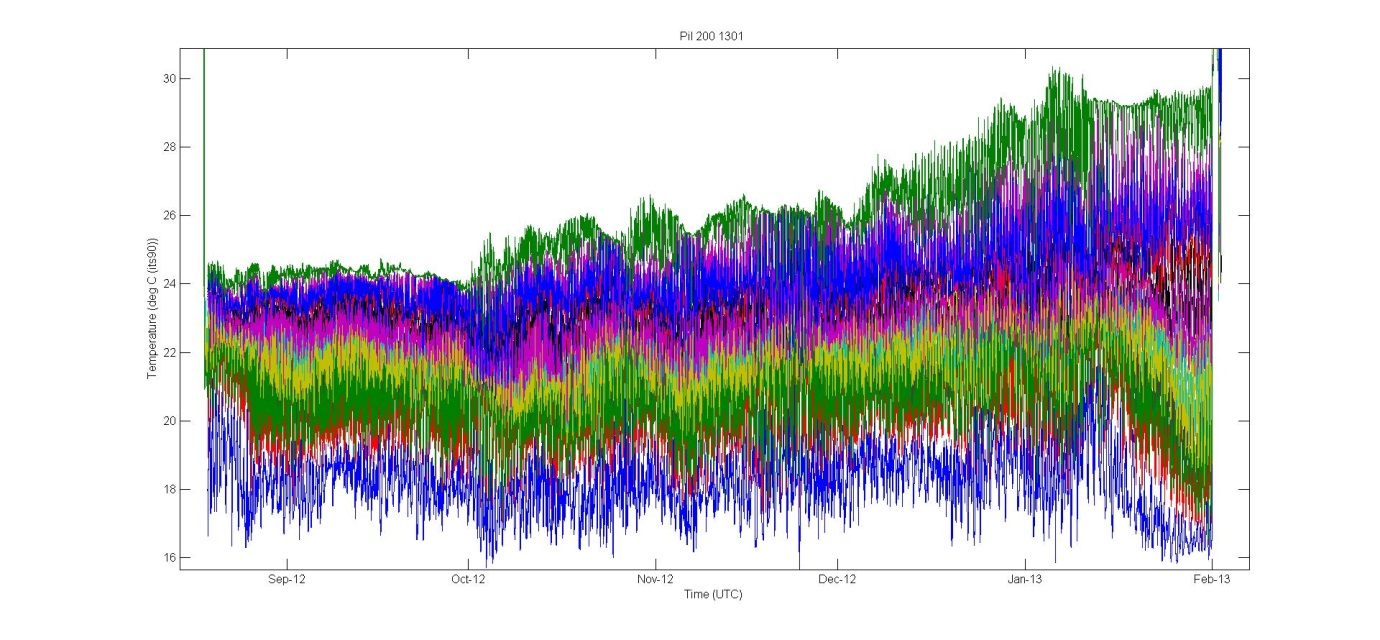
Og tech easy plot

May 13



Mederic MAINSON

Introduction.

Deliver an easy to plot and compare every instrument we can get across and need to quickly diagnostic in the lab or in the file.  
The idea is not only being able to plot on instrument data, but also to compare instrument between them allowing a more accurate diagnostic.  
To do so, we take 3 different steps:

* Importing various data file into Matlab workspace
* Plot Matlab workspace
* Edit plot

Ps: I write those scripts on my spare time, so please be indulgent. They have to and will be improved but for the moment are a bit messy and very simple. The benefit of this is probably that they are really easy to understand and will make it easy to get familiar with Matlab.

# Get MATLABS

Hurrah guys, here is the download link, I suggest you install asap on your machine:

Login Information   
User ID:  [j.luetchford@aims.gov.au](mailto:j.luetchford@aims.gov.au)   
Temporary Password:  M2rYulA2

Download Your Licensed Products   
You can download your licensed products immediately at: <http://www.mathworks.com/download/>   
  
Manage Your Licenses   
You can manage your licenses at: <http://www.mathworks.com/licensecenter/>

# Prepare data file

Do as we usually do to get data from an instrument and convert them as we also usually do. Below is the description on how output variable should be setup. It’s important output variables are setup right (see table 1) to avoid error when importing data into Matlab. User can also look at comments in importOnexxx function to know what matlabs is expecting.

|  |  |  |
| --- | --- | --- |
| Instrument | Data file type expected | Output variable order |
| SB16plus | .cnv | 'Conductivity' 'Pressure' 'Temp' ' time elapsed in second' 'Flag' |
| CTDSB19plus | .cnv | 'Pressure' 'Temp' 'Conductivity' ' time elapsed in second' 'Flag' |
| SB37 | .cnv | 'Pressure' 'Temp' 'Conductivity' ' time elapsed in second' 'Flag' |
| SB39 | .asc | NA. eazy plot auto detect p or pt |
| SB56 | .cnv | %setting for export are: file type: .cnv  % date format: julian days,  % miscelleanous: output informational header. |
| WQM | .dat | WQM processed file setup:  WQM header  SN  Date  Time  Temperature  Salinity  DO(ml/l)  Chl user coef  NTU |
| TR1060 | .txt | Use Ruskin 1.7.19, open your dataset .Hex file. Right click on the dataset in the navigator window and export as Rtext using engineering format. |
| TDR2050 | .txt | Use Ruskin 1.7.19, open your dataset .Hex file. Right click on the dataset in the navigator window and export as Rtext using engineering format. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Tableau 1 - expected format.

# Importing Data file.

In Matlab, set your ‘CURRENT FOLDER’ to wherever you store ogTechMatlabScript folder. A copy can be found on \\PEARL\ocean\OGTECH\Software\Matlab eazy plot  
You will find a set of function call importxxx. xxx being one of the instrument we commonly use.   
To import data into matlabs, one has to click onto the matlabs function related to the instrument he’s looking for and press F9. A window will show up and ask to browse for your file.  
Let the program run (shouldn’t be long…) and you’ll see your data in the workspace window.

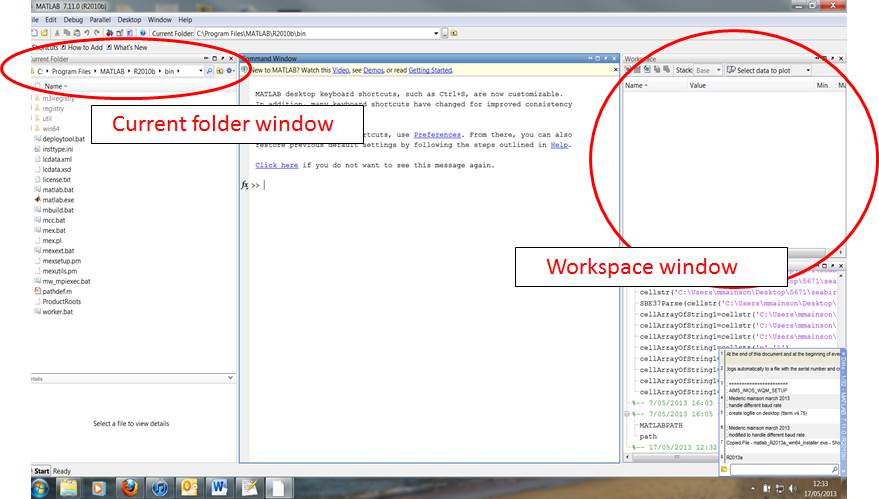
Then you’re ready for the next step  
  


Figure 1 - Default Matlab window

Note:

* Time series loaded from data file into the workspace always have the same structure, it’s a 2 colunms \* n rows matrice. The first columns are the actual measurement, and the second column shows the time in Julian days at which the same line measurement has been taken.
* Also, I tried to automatically generate a variable name from data file header, including instrument type and serial number…. It’s not very consistent, it will be improved in the future
* If this step is generating an error, it very likely the instrument data file hasn’t been converted like Matlab expects it to be converted.

# Plot workspace variables

Simply select the plotWS function in the current folder windows and click F9. All-time series previously loaded shows up on a graph with time and legend.  
If you have too much too plot, it will be unreadable, luckily the next step is editing plot!

Note:

* If this generate an error, it’s probably because a system variable sneak into your workspace, Check all your variable are time series, if you find some that are not delete them and execute the plotWS function again.

# Editing plot

Here is where Matlab get handy by using its built in feature the ‘PLOT BROWSER’.  
In the figure window that just open, go onto the view tab and click ‘plot bowser’. You can then select easily show or hide the curve you like to see!!!

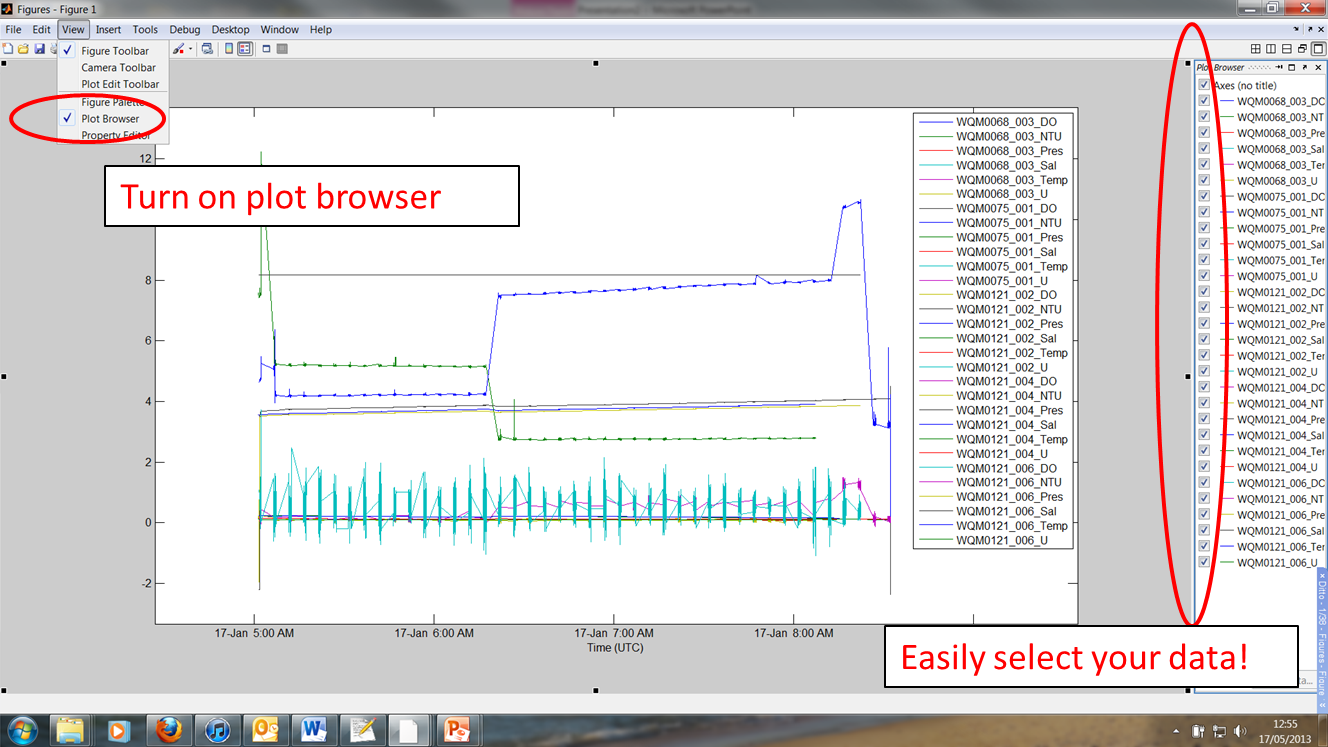


Figure 2 - Matlab figure window

Unfortunately, legend doesn’t update with the plot browser, so u can use the function ‘showSelectedLineSerieLegend’, select and click F9, then turn off the legend using the button ‘legend in the top ribbon of the figure window, and then turn it back on. The legend is up to date. I know it’s a bit clunky for the moment; I will work on it…

Another nice feature is the property editor, have a go at it….  
  
Conclusion

I’m sure everything will evolve pretty fast, typically I want incorporate imox toolbox parser into this plotting solution, but we have here a nice way to get started.