NAME

emtools_data_plot − tool for plotting data from geophysical electromagnetic (EM) input/output files. Currently supported EM data file formats:
1) ModEM - ModEM is a modular system of computer codes for inversion of EM geophysical data developed over the past decade at Oregon State University. A stable version of the code tailored to inversion of 3D magnetotelluric data has been made freely available for academic research. See also: Egbert, G.D., Meqbel, N., and Ritter, O. (2014). Implementing novel schemes for inversion of 3D EM data in ModEM, the OSU modular EM inversion system. Soc. Expl. Geophys. Tech. Prog. Exp. Abstr.

emtools_data_plot is part of the EMTOOLS suite.

SYNOPSIS

emtools_data_plot [OPTIONS] FILE1 FILE2 ... FILEN

DESCRIPTION

A set of N data sets is given by the input files *FILE1 FILE2 ... FILEN* . Currently, only the ModEM datafile format is supported. ModEM data files contain positional EM data of various field types.

emtools_data_plot is a C-Shell frontend and uses the standard Linux tools
csh: main command interpreter calling the following components,
gawk: GNU-version of AWK used for input file processing,
sed: stream editor for filtering and transforming text,
gnuplot: for plot creation (gnuplot version >= 5.0 is required),
ps2pdf: for conversion of postscript output to PDF-format and final PDF output, containing all plot pages.
Note that gawk is the GNU Project's implementation of AWK (pattern scanning and processing tool).
On some OS, gawk=awk, which is no problem for emtools_data_plot, as long as gawk is present.
Gnuplot uses the postscript output terminal. Postscript output is converted to PDF after plotting.

emtools_data_plot automatically bins and sorts all unique data lines of the
first data set (FILE1) according to the following default source/transmitter
attribute tuples:

ModEM data:
MT-data: (Period,Component)
CSEM-data: (Tx_Dipole,Tx_Period,Tx_Azi,Tx_Dip,Tx_X,Tx_Y,Tx_Z,Component)

The resulting set of attribute tuples is assumed identical for all other input files (FILE2,...,FILEN).

All data records (i.e., text lines in an input file) pertaining to a given attribute tuple
form one data subset to be plotted. Each subset generates a separate plot page.
For multiple input files (data sets), the same subset is plotted for each input file
FILE1,...,FILEN, so that each plot page shows N data sets.
Attribute tuples can be further refined by the profile (P) option given with the option
"-r N:P" (see below). The P-specifier enlarges an attribute tuple by a profile coordinate
(x, y, or z) that is common for all receivers of one tuple.

In the following, "plot" will refer to a single plot page.
Each plot contains two subplots:
1) Upper subplot for the RE-data (real field component).
2) Lower subplot for the IM-data (imaginary field component).
Each subplot is a two-dimensional graph with position-field data pairs mapped
onto the x-axis (plot-x-axis) and y-axis (plot-y-axis), respectively.

emtools_data_plot produces the following output files:

*_plot.pdf: PDF output of all plot pages.

*_plot.gnu: Text file with all Gnuplot commands.

*_plot.dat: All plot data formatted for Gnuplot and as plotted in *_plot.gnu.

The default for "*" is the fileroot of the first input file (FILE1). It can be changed using the "-o" option.

OPTIONS

Options and its arguments can appear anywhere within the command line. In the following, capitalized strings (or characters) denote option arguments. Option arguments or parts thereof can be optional, which will be denoted by brackets [*] below.

−h Print this help screen and exit. Use "q" to exit help screen.

−col W:C

Set background colors. The background color can be set for the two subplots (RE, IM) or for the whole plot page. The placeholder W (What?) can be r (RE-subplot), i (IM-subplot), or b (background of entire page). The placeholder C denotes a colorname. More than one comma-separated W:C entries can be specified. To get a list of all possible color names, start gnuplot and type "show colornames":
gnuplot> show colornames

EXAMPLE:
emtools_data_plot -col r:light-blue,i:pink MyData.dat
assigns a light-blue background to the RE-data subplot and a pink background to the IM-data background.

−d[a] Plot differences d=d1-d2 between two data sets (FILE1 - FILE2). Absolute differences are plotted if (the optional) "a" is specified. The following subcommands are available:
−d%[a] Percentage differences.
−dr[a] Relative differences.

EXAMPLE:
emtools_data_plot -d%a MyData1 MyData2
plots the difference d=|d1-d2| in %, where d1=MyData1, d2=MyData2.

−fn FACE,SIZE

Set the font to be used for title, axis labels and plot annotation:
FACE: Font name given by the standard fonts "Times-Roman", "Helvetica", "Courier", and "Symbol". Default is FACE=Courier.
SIZE: Font size, default is SIZE=13.

Font handling for gnuplot's postscript output is relevant for emtools_data_plot. More information is available under gnuplot: "gnuplot> help fonts postscript". From gnuplot: PostScript font handling is done by the printer or viewing program. Gnuplot can create valid PostScript or encapsulated PostScript (*.eps) even if no fonts at all are installed on your computer. Gnuplot simply refers to the font by name in the output file, and assumes that the printer or viewing program will know how to find or approximate a font by that name. All PostScript printers or viewers should know about the standard set of Adobe fonts "Times-Roman", "Helvetica", "Courier", and "Symbol". It is likely that many additional fonts are also available, but the specific set depends on your system or printer configuration. Gnuplot does not know or care about this; the output *.ps or *.eps

files that it creates will simply refer to whatever font names you request.

EXAMPLE:
emtools_data_plot -fn Arial,12 MyDataObs.dat MyDataPre.dat -k Observed,Predicted

−k K1,K2,...,KN
Set key (plot annotation) for each data set (FILE1,...,FILEN). K1 corresponds to FILE1, K2 to FILE2, etc. Default: K1=FILE1 (name of input file FILE1), K2=FILE2, etc. The key appears in both (RE and IM) subplots. If you want to disable the key for all plots, use
−k 0
However, in case you have only one input data set (FILE1) and want the actual key to be the string "0", use
−k 0,K2
where K2 is an arbitrary string.

EXAMPLE1:
emtools_data_plot MyDataObs.dat MyDataPre.dat -k Observed,Predicted
Assigns annotations
"Observed" to data set# 1 (MyDataObs.dat), and
"Predicted" to data set# 2 (MyDataPre.dat).

EXAMPLE2:
emtools_data_plot -k 0,whatever MyData.dat
Unsets the key, so the plot annotation for MyData.dat is an empty string.

−keep Keep plot data (*_plot.dat) and gnuplot-script (*_plot.gnu) output files for further editing and adjusting. Useful for the fine-tuning of publication-quality plots.

−logo PNGFILE
Insert a logo image at the bottom of each plot page. PNGFILE is the file path of an image file in PNG format. Currently in experimental phase, because image file inclusion into gnuplot graphs is still maturing with later versions.

EXAMPLE:
emtools_data_plot -logo ~/${EMTOOLS}/shared/pics/Logo.png MyData.dat

−o FOUT
Set the output fileroot to FOUT. Default: FOUT="out". Will write files: FOUT_plot.dat, FOUT_plot.gnu, FOUT_plot.pdf, where FOUT is a string of your choice.

−r N[:P]
Select the receiver data columns, which defines a plot's x-axis (plot-x-axis). N can be: n,x,y,z (alternatively: 0,1,2,3)
-r n Plot data against the receiver number (Rx#). For each data subset, Rx# is defined by the input order of the respective data lines.
-r x Plot data against the receiver x-coordinates.
-r y Plot data against the receiver y-coordinates.
-r z Plot data against the receiver z-coordinates.

The specifier *P* adds a common profile coordinate to the tuple of source attributes that defines a data subset. For a given source attribute, the number of plots then corresponds to the number of unique profile (x, y, or z) coordinates (otherwise, it is one). P can be: x,y, or z
-r N:x When a plot's x-axis is the y- or z- coordinate (N=y or N=z): Group all receivers that share

a unique x-coordinate into a subset.
-r N:y When a plot's x-axis is the x- or z- coordinate (N=x or N=z): Group all receivers that share a unique y-coordinate into a subset.
-r N:z When a plot's x-axis is the x- or y- coordinate (N=x or N=y): Group all receivers that share a unique z-coordinate into a subset.

The specifier *P* invokes an ascending-order sorting of all plot-x-axis coordinates within one subset. For plots with data style lines, non-consecutive coordinates may otherwise cause crossing lines.

EXAMPLE1:
emtools_data_plot -r x MyData1
or
emtools_data_plot -r 1 MyData1
Plots the data of file MyData1 against the receiver x-coordinate.

EXAMPLE2:
emtools_data_plot -r x:y MyData1
Same as before, where each data subset described by a set of unique source attributes will be further split up into n subsets, and n is the number of unique y-coordinates in a given subset.

−sep Write a separate plot output file for each data subset (=plot page). Currently not available.

−t Input test run without plot creation. Useful for reporting minimum and maximum data values. Will write out *.gnu and *.dat output files and skip gnuplot plot creation.

−v Set verbose mode from normal to high. Reports minimum and maximum data values for each data subset.

−view[,PDFVIEWER]
Launch the PDF-file viewer PDFVIEWER after plot generation. The entry PDFVIEWER is optional. If not specified, emtools_data_plot tries to find PDFVIEWER in the emtools-configuration file, if the latter is present. The setting via the "-view" option takes precedence over the setting via the emtools-configuration file.

EXAMPLE:
emtools_data_plot Data_obs.dat -view,xpdf
Launches the PDF-file viewer xpdf after plot generation.

−w S1[:T1],S2[:T2],...,SN[:TN]
Select the data style (lines, point-symbols, and corresponding lines and points attributes). Each style-string S[:T] can have additional options, the latter given by the optional string T and separated from S by ":". In gnuplot, data may be displayed in one of a large number of styles, where the "with"-keyword provides the means of selection. Under gnuplot, "help plot with" will show all the possibilities of choosing your favorite data style. In order to retain all of gnuplot's options, the S-string after "-w" (S1,S2,...) follows the gnuplot syntax after "with", except that blank-characters need to be replaced by "_", for example, in gnuplot:
gnuplot> plot sin(x) with linespoints linetype 1 linewidth 2
will plot with lines and point symbols of a specific linetype and linewidth. In gnuplot, commands can also be abbreviated, leading to the same result:
gnuplot> plot sin(x) w lp lt 1 lw 2
Hence, in emtools_data_plot, displaying data with this same style, replace all blanks by "_":

EXAMPLE1:
emtools_data_plot MyData.dat -w linespoints_linetype_1_linewidth_2
which is equivalent to
emtools_data_plot MyData.dat -w lp_lt_1_lw_2

Individual data styles:
An individual data style can be selected for each input data set (FILE1,FILE2,...,FILEN). Multiple data styles are separated by ",": S1,S2,... or S1[:T1],S2[:T2],...

EXAMPLE2:
emtools_data_plot Data_obs.dat Data_pre_iter10.dat -w p_pt_5_ps_0.4,l
Plots the first data set (Data_obs.dat) using data style "points pointtype 5 pointsize 0.4". The second data set (Data_pre_iter10.dat) uses lines. Since no line attributes are specified, gnuplot's default settings will be used.

Data-error bars:
Gnuplot also has the option of selecting a style that plots data points with vertical errorbars (yerrorbars). The style-string "yerr" invokes this style.

EXAMPLE3:
emtools_data_plot Data_obs.dat Data_pre_iter10.dat -w yerr_pt_4_ps_0.35,l_lw_1.5
Plots the data contained in Data_obs.dat with the data style "yerrorbars pointtype 4 pointsize 0.35". The second data set Data_pre_iter10.dat will be plotted against the error bars using data style "lines linewidth 1.5".

emtools_data_plot automatically calculates the lower and upper end of the vertical yerrorbars from the data errors of each input data set (column of errors, i.e. standard deviations). The error-columns usually follow the data-columns in EM-data files. For a given data value y and its associated error e>0, the corresponding yerrorbar is then [y1,y2]=[y-e,y+e]. Instead of using a data set's data errors (if error-columns are not present), one can select a constant error value, using the optional T-string that follows each style string S.

EXAMPLE4:
emtools_data_plot Data_obs.dat -w yerr_pt_4_ps_0.3:1e-4
Plots the data set Data_obs.dat using data style "yerrorbars pointtype 4 pointsize 0.3", where the yerrorbar of a given data point y extends over the range [y1,y2]=[y-1e-4,y+1e-4], thus the total yerrorbar length y2-y1 is 2e-4.

If error-columns are not present, one can further select a percentage error value, which is also specified in the optional T-string.

EXAMPLE5:
emtools_data_plot Data_obs.dat -w yerr_pt_4_ps_0.3:3%
Plots the data set Data_obs.dat using data style "yerrorbars pointtype 4 pointsize 0.3", where the half-yerrorbar of a given data point y is 3% of its absolute value and thus extends over the range [y1,y2]=[y-|y|*3/100,y+|y|*3/100].
Note that for a log y-axis, the range is
[y1,y2]=[|y|-|y|*3/100,|y|+|y|*3/100].
Thus, each yerrorbar half-length equals 2*|y|*0.03.

For plots with a log y-axis, the magnitude of data errors can lead to yerrorbars that cover multiple decades, or sometimes the whole y-axis. To limit the yerrorbar-size, use the "lf" (letters L+F) option, specified in the T-string. The number before or after "lf" is a factor FAC, so that for a given data value y=|y| and its error e:

if y1=y-e < y/FAC: set [y1,y2] = [y-y/FAC,y+y/FAC]
This limits the yerrorbar size [y1,y2] to the fraction 2*y/FAC of the actual data value y. Reasonable values for FAC may be between 3 and 10.

EXAMPLE6:
emtools_data_plot Data_obs.dat -w yerr:lf3
Plots with yerrorbars (using gnuplot's default data style) which are size-limited to
[y1,y2]=[|y|-|y|/3,|y|+|y|/3].
For a data value of y=1, the yerrorbar would then extend over [y1,y2]=[0.666,1.333].

An alternative way of limiting the yerrorbar-size uses the "lc" (letters L+C) option, to be specified in the T-string. The number before or after "lc" is a constant number CON, so that for a given data value y=|y| and its error e:
if y1=y-e < y-CON: set [y1,y2] = [y-CON,y+CON]
This limits the yerrorbar size [y1,y2] to the 2*CON.

EXAMPLE7:
emtools_data_plot Data_obs.dat -w yerr:1e-4lc
Plots with yerrorbars (using gnuplot's default data style) using the data errors of Data_obs.dat. The yerrorbars are size-limited to
[y1,y2]=[|y|-1e-4,|y|+1e-4].
For a data value of y=0.001, the yerrorbar would then extend over [y1,y2]=[0.0009,0.0011].

Note that both the yerrorbar limit-options lf and lc may not reflect actual data errors. They are only intended for improving the appearance of errorbars, as they can span multiple decades in log plots.

−xr X1,X2[:P]
Set global x-axis range (interval [X1,X2]) for the receiver coordinate axis (plot-x-axis). The optional number P defines an axis-stretching percentage, with the interval-enlarging effect [X1,X2] -> [X1*(1-P/100),X2*(1+P/100)].

EXAMPLE1:
emtools_data_plot -r 2 -xr 1000,2000 Iter0.dat
Sets the global plot-x-axis range to [X1,X2]=[1000,2000]. Note that the receiver y-coordinates define the plot-x-axis (set by "-r 2").

EXAMPLE2:
emtools_data_plot -r x -xr 15000,25000:15 Iter0.dat
Sets the global plot-x-axis range to [X1,X2]=[15000,25000]. Note that the receiver x-coordinates define the x-axis. The actually shown range is enlarged by 15%, thus [X1,X2]=[12750,28750].

−yr Y1[,Y2,Y3,Y4:P1,P2]
Set global plot-y-axis range defined by a range [Ymin,Ymax] for the RE and IM subplots. Without the "-yr" option, gnuplot sets the plot-y-axis ranges individually for each plot, according to the min and max values [min_RE,max_RE] and [min_IM,max_IM] of the corresponding data subset. With the "-yr" option, gnuplot sets both the RE-plot-y-axis and IM-plot-y-axis ranges to predefined ranges, the latter given by the number pairs [Y1,Y2] and [Y3,Y4]. These ranges apply globally (for each data subset); hence the "-yr" option is useful if one wants to compare data amplitudes on a common y-axis range. In the most specific case:
-yr Y1,Y2,Y3,Y4
Y1,Y2 defines the RE-plot-y-axis range (interval of the RE-data).
Y3,Y4 defines the IM-plot-y-axis range (interval of the IM-data).

Equalizing the plot-y-axis range:

You will probably find the "equalize"-options most useful, realized by replacing a number pair by "e". The "e" option will employ the global min and max values:
[min_RE,max_RE]: min, max values over all RE-data, and
[min_IM,max_IM]: min, max values over all IM-data.
-yr e
Sets [Y1,Y2] = [Y3,Y4] = [min,max], where
[min,max] = [min(min_RE,min_IM),max(max_RE,max_IM)]
are the global min and max values over all input data.
-yr e,e
Sets [Y1,Y2] = [min_RE,max_RE]; [Y3,Y4] = [min_IM,max_IM].
One can omit arguments, the following number combinations are supported:
-yr Y1 same min: RE: Y1,max(RE); IM: Y1,max(IM).
-yr Y1,Y2 same min,max: RE: Y1,Y2; IM: Y1,Y2.
-yr Y1,Y2,Y3 individual min, same max: RE: Y1,Y2; IM: Y3,Y2.
-yr Y1,Y2,Y3,Y4 individual min,max: RE: Y1,Y2; IM: Y3,Y4.
P1,P2 defines axis-stretching percentages, with the interval-enlarging effect
[Y1,Y2] -> [Y1*(1-P/100),Y2*(1+P/100)]
If both P1 and P2 are given:
P1 enlarges the RE-plot-y-axis interval: [Y1,Y2] -> [Y1*(1-P1/100),Y2*(1+P1/100)]
P2 enlarges the IM-plot-y-axis interval: [Y3,Y4] -> [Y3*(1-P2/100),Y4*(1+P2/100)]
One can omit P2:
-yr Y1[,Y2,Y3,Y4]:P1 sets P2=P1=P for both the RE- and IM-plot-y-axis.

EXAMPLE1:
emtools_data_plot -yr 1e-4,1e-3:10
Sets the global plot-y-axis range for both RE and IM data to [Y1,Y2]=[1e-4,1e-3]. The actually shown range is enlarged by 10%, thus [Y1,Y2]=[9e-5,0.0011].

EXAMPLE2:
emtools_data_plot -yr 1e-4,1e-3,1e-5,1e-2:10,25
Sets the RE-plot-y-axis range to [Y1,Y2]=[1e-4,1e-3].
Sets the IM-plot-y-axis range to [Y3,Y4]=[1e-5,0.01].
The actual RE range is enlarged by 10%, thus [Y1,Y2]=[9e-5,0.0011].
The actual IM range is enlarged by 25%, thus [Y3,Y4]=[7.5e-6,0.0125].

EXAMPLE3:
emtools_data_plot -yr 1e-4,1e-3,1e-5,1e-2:10
Same as EXAMPLE2, where both RE and IM plot-y-axis ranges are enlarged by 10%.

FILES

$HOME/.emtools - the emtools configuration file. Entries relevant for emtools_data_plot:
AWK: gawk executable.
GNUPLOT: gnuplot executable.
LOGO: PNG-file to be included as image into page-bottom of each plot page.
PDFVIEWER: PDF-file viewer.
Entries for AWK, GNUPLOT and PDFVIEWER can be made active if these tools are not present as standard tools, i.e. their executables are not found within $PATH.
Lines beginning with "#" are inactive.
EXAMPLE file .emtools:
$ cat $HOME/.emtools
#AWK: /usr/local/bin/gawk # Gnu-AWK executable, line commented out -> inactive
GNUPLOT: /usr/local/bin/gnuplot_5.4 # gnuplot executable >= version 5.0 needed
#LOGO: /home/micha/Pictures/mylogo.png # line commented out -> inactive
PDFVIEWER: xpdf # PDF-file viewer, launched if option "-view" specified

AUTHOR

Michael Commer @ CMC GeoConsulting

emtools_data_plot is a derivative of Michael Commer's EMGeo datafile plotting tool em3d_dataplot, developed at Lawrence Berkeley Nat'l Lab, California. Data plot examples can for example be seen in:

Commer, M., and Newman, G.A., (2008). New advances in three-dimensional controlled-source electromagnetic inversion, Geophys. J. Int., 172/2, 513–535.

The emtools_data_plot extension for ModEM data files (1st version: 2023.10) was developed at Observatorio Nacional, Rio de Janeiro.

BUGS, COMMENTS, ISSUES?

micha@on.br

LAST UPDATE

Tue Aug 20 00:41:27 CEST 2024