

# posts

avigoz

January 29, 2015

## Contents

### 1 Screen - unique logs for each run

Screen is a little wrap around linux shell that enables detaching and logging out while the session you created is still running. It could be used for lots of different purposes, and running an intensive computation on a remote computer is an obvious example.

You would normally do :

```
1 screen -md -L -S session_name your_program
```

-md = detach immediately after running, and return to the current terminal session -L = create a log file -S = create meaningful name for your session

To check the stat of your sessions you will :

```
1 screen -ls
```

To have different log files with unique names for different sessions, you need to create a `~/.screenrc` file, with the following single line

```
logfile screenlog-%Y%m%d-%c:%s
```

### 2 matlab : regridding unequally spaced sampled field, and plotting an imagesc

```
[x1,y1]=ndgrid(x,y); I = TriScatteredInterp(x1(:),y1(:),z(:)); x1 = linspace(min(x),max(x),5);  
% Define X-grid y1 = linspace(min(y),max(y),5); [x1,y1]=ndgrid(x1,y1);  
z1=I(x1,y1); myimagesc(x1(1,:),y1(:,1),z1,0.55,0.95,0.05);
```

### 3 matlab : save a plot in png, eps, and fig formats

```
%% in the parameters section prints=struct('suff',{'png','eps','fig'},... 'func',{@(x)
print('-dpng',x),@(x) print('-depsc2',x), @hgsave}); n_printfuncs=length(prints);
%% after the plot commands filename='stam.'; for i_printfunc=1:n_printfuncs
% fig, png, and eps files prints(i_printfunc).func([filename,'.',prints(i_printfunc).suff]);
end % for i_printfunc=1:n_printfuncs
    this is now incorporated in /home/avigoz/Dropbox/oct_scripts/plotting/savefigs.m
```

### 4 matlab general slicing syntax :

```
idx.type='()'; % indices structure idx.subs={':',':',':',':'}; idx.subs{dim_face}=1;
z=subsref(z,idx);
```

### 5 bash shell execute the same program on all the files in the current directory

```
find -exec prog_name flags {} +
    {} represents the list of filenames that will be appended by find. it must
    be the last argument since the "+" syntax tells "find" to create a large list
    and send them all at the same time.
    if we want "find" to run prog_name for each of the files separately the
    correct syntax is : find -exec prog_name '{}' \;
    in this case {} doesn't need to be the last argument
    if we want a more complicated sequence of commands
    find -printf "zcat %p | agrep -dEOE 'grep'" | sh
```

### 6 cvs create new repository, add a new directory, and begin working:

<http://www-mrsl.stanford.edu/~brian/cvstutorial/>

1. create a new repository in ~/cvsroot:

```
cvs -d ~/cvsroot init
```

1. env variables used by cvs:

```
export CVSROOT=$HOME/cvsroot export CVSEDITOR=emacs
```

1. backup original directory:

```
mkdir cvsexample2 cp -r cvsexample/* cvsexample2/
```

1. remove the original files:

```
rm -r cvsexample/*
```

1. add the empty directory to cvs:

```
cd ~/cvsexample cvs import -m "dir structure" cvsexample yourname start
```

this adds a directory cvsexample in the repository, so one can have several projects in the same repository, and checking out only the particular project of interest.

1. remove the created directory and check it out from repo (i'm not sure this is necessary...)

```
cd .. rm -r cvsexample cvs checkout cvsexample
```

1. add a subdirectory

```
cd ~/cvsexample mkdir cartilage cvs add cartilage
```

1. add a file :

create a file (or copy from your backups), and then :

```
cvs add *.tex cvs commit -m "original files" *.tex
```

without -m "blahh", cvs will just launch emacs for your log message

1. download updates from repo:

```
cvs update
```

1. see difference between current version and repo version:

```
cvs diff sample.tex
```

1. submit a modified file:

```
cvs commit sample.tex
```

1. read log messages :

cvs log sample.tex

1. when you have a working version, tag it:

```
cvs tag Clinical-Release-1.0
```

now the tagged version can be restored in a new directory if we wish:

```
mkdir tempstuff cd ~/tempstuff cvs checkout -r Clinical-Release-1.0 cv-  
sexample
```

## 7 wget :

```
wget -r -ll -H -nd -np -A.txt -w5 -erobots=off -i ~/list.txt
```

-r recursively -H follow links that point away from the website -ll only go one level deep -np "no parent" -nd save every thing in one directory -A.txt tells wget to only download files that end with the .txt extension. -i ~/list.txt - if we have a list of websites. otherwise we can just add the URL of a specific website -w5 wait 5 seconds between downloads -erobots=off ignore site policy

## 8 python ginput:

example from <http://glowingpython.blogspot.co.il/2011/08/how-to-use-ginput.html>

```
from pylab import plot, ginput, show, axis  
axis([-1, 1, -1, 1]) print "Please click three times" pts = ginput(3) # it  
will wait for three clicks print "The point selected are" print pts # ginput  
returns points as tuples x=map(lambda x: x1,pts) # map applies the func-  
tion passed as y=map(lambda x: x2,pts) # first parameter to each element  
of pts plot(x,y,'o') axis([-1, 1, -1, 1]) show()
```

## 9 export from libreoffice :

(source <http://www.commandlinefu.com/commands/view/11692/commandline-document-conversion>)  
libreoffice --headless --convert-to odt:"writer8" somefile.docx

---

<sup>1</sup>DEFINITION NOT FOUND.

<sup>2</sup>DEFINITION NOT FOUND.

## 10 mitgcm alternating checkpoint :

```
in data, parm03: pickupSuff='ckptA'
```

## 11 python split filename to file+extension

```
(source: http://stackoverflow.com/questions/541390/extracting-extension-from-filename-in  
»> import os »> fileName, fileExtension = os.path.splitext('/path/to/somefile.ext')  
»> fileName '/path/to/somefile' »> fileExtension '.ext'
```

## 12 python equivalent for importdata

```
(source http://stackoverflow.com/questions/1057666/using-python-to-replace-matlab-how-to  
import numpy imported_array = numpy.loadtxt('file.txt',delimiter=',')  
# assuming tab-delimiter print imported_array.shape
```

## 13 latex reference ranges of images (other stuff)

```
(source: http://tex.stackexchange.com/questions/7624/how-to-reference-ranges-rather-than  
http://www.howtotex.com/packages/automatic-clever-references-with-cleveref/  
)  
cleveref figureFig.Figs.  
winter,fall,christmas,summer,pentecost
```

## 14 matlab

```
filenames=fill_sprintf(index_array,filename_pattern) :
```

```
1 % syntax: filenames=fill_sprintf(index_array,filename_pattern)  
2 % fill_sprintf is meant to extend sprintf to dealing with cell  
3 % arrays of strings (e.g. filenames with running indices).  
4 % the function loops through all indices in index_array, and calls  
5 % sprintf(filename_pattern,ind). filenames is a cell array of all  
6 % filenames.  
7 %  
8 % see also : sprintf  
9 %  
10 % no special dependencies  
11
```

```

12 % $Log$
13 function filenames=fill_sprintf(index_array,filename_pattern)
14 % some input checking
15 if(length(index_array)<1)
16     filenames={};
17     return;
18 end
19 [s,er]=sprintf(filename_pattern ,index_array(1));           % this check doesnt work in oc
20 if(~isempty(er))
21     error('wrong filename pattern');
22 end                                                         % if(~isempty(er))
23 index_array=num2cell(index_array);
24 filenames=cellfun(@(x)sprintf(filename_pattern,x),index_array, ...
25                  'uniformoutput',false);

```

creates a cell array of filenames with running indices

## 15 latex small horizontal space between figs

(source <http://tex.stackexchange.com/questions/41476/lengths-and-when-to-use-them>)

## 16 latex code snippets

(source <http://stackoverflow.com/questions/3175105/how-to-insert-code-into-a-latex-doc>)

in the header : listings color

```

frame=tb, language=Java, aboveskip=3mm, belowskip=3mm, show-
stringspaces=false, columns=flexible, basicstyle=, numbers=none, number-
style=, keywordstyle=, commentstyle=, stringstyle=, breaklines=true, breakatwhitespace=true, tabsize=3

```

in the body text : // Hello.java import javax.swing.JApplet; import  
java.awt.Graphics;

```

public class Hello extends JApplet public void paintComponent(Graphics
g) g.drawString("Hello, world!", 65, 95);

```

## 17 extract data from csv (in non trivial cases):

(source : <http://stackoverflow.com/questions/1641519/reading-date-and-time-from-csv-file>)

```

fid = fopen(filename, 'rt'); a = textscan(fid, '%f/%f/%f %f:%f %f
%f', ... 'Delimiter',',', 'CollectOutput',1, 'HeaderLines',4); fclose(fid);

```

```
t=datetime(a{1}(:,3)+2000, a{1}(:,2), a{1}(:,1), a{1}(:,4), a{1}(:,5), zeros(length(a{1}(:,1)),1));
directions=a{1}(:,6); speeds=a{1}(:,7);
```

## 18 python argument line parser

```
(source : http://docs.python.org/dev/library/argparse.html)
import argparse parser = argparse.ArgumentParser(description='create
encoded longitude-latitude list') parser.add_argument('lon_file', help='longitudes
file') parser.add_argument('lat_file', help='latitudes file') parser.add_argument('out_file',
help='out file') args = parser.parse_args()
the different fields are in a data structure args.lon_file args.lat_file
args.out_file
```

## 19 svn sourceforge username not recognized :

```
(source http://highlevelbits.com/2007/04/svn-over-ssh-prompts-for-wrong-username.html)
```

```
just include the file config in ~/.ssh with the following content: Host
svn.code.sf.net User youruser
```

## 20 checking out from sourceforge :

```
(note the +ssh in the protocol prefix)
svn -username avigdev checkout svn+ssh://svn.code.sf.net/p/panet/code
./
```

## 21 gdb mode of emacs 24 has a bug. a way around it :

```
(clue from http://stackoverflow.com/questions/13959747/using-gdb-i-mi-integration-in-ema)
M-x gdb gdb -i=mi -annotate=0 PANet
```

## 22 awk multiple types of delimiters:

```
awk -F[_.] '{print $3}'
```

## 23 mitgcm numeric stability criteria

The stability criterion for the horizontal laplacian friction is  $4 \Delta h \Delta t / \Delta x^2 < 0.3$  (pp. 123 in the manual) Stability for inertial oscillations (although we don't expect such a thing)  $f^2 \Delta t^2 < 0.5$  (pp. 123 in the manual) Advective Courant-Friedrichs-Lewy criterion (pp. 123 in the manual)  $\max_u \Delta t / \Delta x < 0.5$

## 24 compiling large array :

```
FFLAGS="$FFLAGS -g -convert big_endian -assume byterecl -mcmmodel=large"
```

## 25 sync folders to hd

```
rsync -force -ignore-errors -delete -exclude home/avigoz.opera/*cach* -  
backup-dir='date +%Y-%m' -avb home/avigoz /media/linux_part/backups/home_64
```

## 26 setting up a (mac) computer checklist

- ☐ d/l home directory from external hd
- ☐ make .profile speak with .bashrc
- ☐ echo "logfile screenlog-%Y%m%d-%c:%s" > ~/.screenrc
- ☐ d/l homebrew
- ☐ d/l and setup Dropbox, Ubuntu one
- ☐ d/l skype
- ☐ d/l XCode
- ☐ for compilers - enter xcode->preferences->components->command line tools->install
- ☐ d/l (using the command "brew install") cvs,git ??
- ☐ d/l latest version of emacs (brew install -cocoa emacs)
- [ ] see <http://stackoverflow.com/questions/10171280/how-to-launch-gui-emacs-from-command-line>
- [ ] >link it to Applications :



- [ ] n -s /opt/boxen/homebrew/Cellar/emacs/24.3/Emacs.app /Applications
- [ ] > prepare a bash script somewhere with the following script :
- [ ] □
- [ ] !/bin/sh
- [ ] Applications/Emacs.app/Contents/MacOS/Emacs -Q "\$@"
- [ ] □
- [ ] >include
- [ ] (setq mac-function-modifier 'control) in .emacs (to avoid ctrl-space problems)
- [ ] □
- [ ] □ to d/l xmgr , first d/l xquartz (<https://xquartz.macosforge.org>). afterwards use "brew install grace" .
- [ ] □
- [ ] □ to d/l octave run (see [http://wiki.octave.org/Octave\\_for\\_MacOS\\_X](http://wiki.octave.org/Octave_for_MacOS_X)):
- [ ] brew tap homebrew/science
- [ ] brew update && brew upgrade
- [ ] brew install gfortran
- [ ] brew install octave
- [ ] brew install gnuplot
- [ ] n -s /usr/local/Cellar/gnuplot/4.6.3/bin/gnuplot /Applications/gnuplot
- [ ] □
- [ ] > edit /usr/local/share/octave/site/m/startup/octaverc to be :
- [ ] □
- [ ] # System-wide startup file for Octave.

- [ ]#
- [ ]# This file should contain any commands that should be executed each
- [ ]# time Octave starts for every user at this site.
- [ ]etenv ("GNUTERM", "X11")
- [ ]nuplot\_binary("/Applications/gnuplot")
- ☐
- [ ]> create a small shell script with :
- [ ]!/bin/sh
- ☐
- [ ]C\_CTYPE="en\_US.UTF-8"
- ☐
- ☐ Replace the following line with the result in step 3 (where your octave is located)
- [ ]usr/local/bin/octave
- ☐
- [ ]> in .bash\_aliases : alias octave="path\_to\_your\_file"
- ☐
- ☐ for python scientific packages (and upgrading python):

sudo easy\_install pip brew install swig sudo pip install scipy

-> run "brew doctor" to see whether anything wrong is going on.

->put the following in .bashrc: export PATH=/usr/local/bin:\$PATH

export PATH=/usr/local/share/python:\$PATH

-> continue with python . . . following <http://iknownothingaboutcoding.blogspot.co.il/2012/04/mac-os-x-lion-install-of-python-numpy.html>

:  
 brew install readline sqlite gdbm pkg-config --universal brew install  
 python --framework --universal cd /System/Library/Frameworks/Python.framework/Versions

sudo rm Current sudo ln -s /usr/local/Cellar/python/\*\*\*version\*\*\*/Frameworks/Python.framework/Ve

Now install pip, by using:

? \$ easy\_install pip To test the installation of pip type:

? \$ which pip and you should see the following returned:

? /usr/local/share/python/pip Next use pip to install virtualenv and virtualenvwrapper:

? \$ pip install virtualenv \$ pip install virtualenvwrapper \$ source /usr/local/share/python/virtualenvwrapper.sh Install Numpy via:

? \$ pip install numpy Install SciPy also using pip - the “green room” link installs SciPy using the github.egg however, they’ve fixed things now so you can use the method below. The first command gets the required Fortran compiler:

? \$ brew install gfortran \$ pip install scipy Pip Install Matplotlib

?

(i had to also do : \$ sudo pip install --upgrade six)

\$ pip install -e git+https://github.com/matplotlib/matplotlib.git#egg=matplotlib-dev iPython, Pandas, SciKits, & Nose Pip Install iPython

? \$ pip install ipython then:

? \$ brew install pyqt append your ~/.bash\_profile with the appropriate statement given to you at the END of the pyqt installation, for me it was:

? export PYTHONPATH=/usr/local/lib/python2.7/site-packages:\$PYTHONPATH

Then:

? \$ brew install zmq \$ pip install pyzmq \$ pip install pygments Install Pandas:

? \$ pip install pandas Install Scikits.Statsmodels

? \$ pip install scikits.statsmodels Lastly, to ensure that we have the necessary testing suites to check the packages that we’ve just installed. The testing suite that (conveniently) all of these packages is called nose.

? \$ pip install nose And we are finished with the installation!

Installation Testing Numpy Testing First, let’s check the installations of Numpy and SciPy, as is provided on their documentation

In terminal, here is what to type, along with the output that I get back:

? \$ python Python 2.7.3 (default, Apr 20 2012, 17:20:12) [GCC 4.2.1 Compatible Apple Clang 3.1 (tags/Apple/clang-318.0.58)] on darwin Type "help", "copyright", "credits" or "license" for more information.

>> import numpy >> numpy.test('full') ... [lots of text] ... [final lines]

---

Ran 3552 tests in 35.886s

FAILED (KNOWNFAIL=3, SKIP=1, failures=9) Although it’s not perfect with 0 failures, I’ll definitely take it. One issue of many that prompted

me to reinstall Python and these libraries is that when I would run this test, my Terminal would crash and quit (for both Numpy and Scipy)... yeah, not good.

SciPy Testing Now let's test SciPy.

```
? »> import scipy »> scipy.test() ... [lots of text] ... [final lines]
```

---

Ran 5101 tests in 56.231s

FAILED (KNOWNFAIL=12, SKIP=42, failures=9) Again, not batting 1000, but I'm definitely satisfied.

Pandas Testing And lastly, let's make sure that Pandas is working properly.

```
? »> exit() $ nosetests pandas
```

```
..... [lots of periods, S's and other things] ... Ran 1509 tests in 70.357s
```

```
OK (SKIP=11)
```

- ☐ to install gmt : brew install gmt
- ☐ to install maxima : brew install maxima
- ☐ d/l MITgcm
- ☐ d/l ferret
- ☐ d/l AUTO

## 27 take a column of numbers and put them in a row with a "+" delimiter :

paste -sd+ on a mac os x : paste -sd+ - (where the last dash indicates that we take standard input instead of a filename)

## 28 installing emacs on MAC

(after getting brew, XCode etc.) » brew install emacs create a text file with the following :

```
#!/bin/sh /Applications/Emacs.app/Contents/MacOS/Emacs -Q "$@"  
and PATH it.
```

remove previous vers from *usr/bin*

## 29 MITGCM recipee for building a package (the name of the example package is diffus2):

1. prepare an empty package that does nothing

the minimal list of files (which can be copied, with necessary name changes of files/variables/parameters/functions, from MYPACKAGE) is:  
diffus2\_calc.F diffus2\_diagnostics\_init.F DIFFUS2\_OPTIONS.h DIFFUS2\_PARAMS.h  
DIFFUS2.h diffus2\_output.F diffus2\_routines.F diffus2\_check.F diffus2\_init\_varia.F  
diffus2\_readparms.F  
their description :

file	description
headers	
DIFFUS2.h	define pkg variables, and their common blocks
DIFFUS2_OPTIONS.h	package specific MACRO option defs
DIFFUS2_PARAMS.h	package parameters and their common block (read from data.diffus2)
code	
diffus2_calc.F	interface for mitgcnv (this is what the model's core calls)
diffus2_check.F	check dependencies/conflicts with other packages
diffus2_diagnostics_init.F	define diagnostics related to the package
diffus2_init_varia.F	initialize DIFFUS2 parameters and variables
diffus2_output.F	create diagnostic outputs
diffus2_readparms.F	parse data.diffus2
diffus2_routines.F	routines that implement double diffusion parametrization schemes

they should be under a new directory of the rootdir (in diffus2 case  
~/MITgcm/model/pkg/diffus2 )

the input file data.pkg should include the entry "useDiffus2=.TRUE.,"  
under the namelist "&PACKAGES"

this parameter should be declared (with the type LOGICAL), and in-  
cluded in the common block *PARAM\_PACKAGES* under ~/MITgcm/model/inc/PARAMS.h  
. it should also be included under the namelist "PACKAGES" in ~/MIT-  
gcm/model/src/packages\_boot.F , and its default value should usually de-  
clared in this file to be .FALSE..

1. parse user parameters

in diffus2\_readparms - create a separate NAMELIST for each namelist  
that should appear in data.diffus2 . then give the parameters default

conditions. (e.g. `diffus2_scheme = 'kunze'` ) then try to read them (e.g. `READ(UNIT=iUnit,NML=DIFFUS2_SCHEME,IOSTAT=errIO)` ) and monitor events where `errIO<0` :

```
READ(UNIT=iUnit,NML=DIFFUS2_SCHEME,IOSTAT=errIO) IF (
errIO .LT. 0 ) THEN WRITE(msgBuf,'(A)') & 'S/R INI_PARMS' CALL
PRINT_ERROR( msgBuf , 1) WRITE(msgBuf,'(A)') & 'Error reading nu-
merical model ' CALL PRINT_ERROR( msgBuf , 1) WRITE(msgBuf,'(A)')
& 'parameter file "data.diffus2"' CALL PRINT_ERROR( msgBuf , 1)
WRITE(msgBuf,'(A)') & 'Problem in namelist DIFFUS2_SCHEME' CALL
PRINT_ERROR( msgBuf , 1) STOP 'ABNORMAL END: S/R DIF-
FUS2_INIT' ENDIF
```

```
CLOSE(iUnit)
```

finally tell `STDOUT.*` that you're finished `WRITE(msgBuf,'(A)') ' DIF-  
FUS2_INIT: finished reading data.diffus2'`

declare these variables in `DIFFUS2_PARAMS.h`

these subroutines are run from the model file `"packages_readparms.F"`.

these are the needed lines in `packages_readparms.F`:

```
C- Initialize Diffus2 parameters IF (useDiffus2) CALL DIFFUS2_READPARAMS(
myThid ) #endif
```

### 30 ssh tunnel through proxy :

in: `.ssh/config`:

```
Host tsia Hostname tsia.boker User avigoz ForwardAgent yes Port 22
```

```
ProxyCommand ssh avigoz@sansana.bgu.ac.il nc %h %p
```

to make it passwordless :

on the local machine : `» ssh-keygen -t rsa`

on the remote machine : `» mkdir -p .ssh`

on the local machine : `cat .ssh/id_rsa.pub | ssh b@B 'cat » .ssh/authorized_keys'`

repeat these for logging to `a->b->c` , for the pairs `a->b` , `a->c` .

### 31 get a list of links from a website, using the textual web browser lynx :

(source : <http://tips.webdesign10.com/general/lynx-browser> )

```
lynx -dump -listonly "http://www.example.com/"
```

## 32 define a remote directory

```
in fstab : sshfs#avigoz@132.64.144.245:/data/avigoz /data1 fuse defaults,allow_other
0 0
    in /etc/fuse.conf , uncomment : user_allow_other
```

## 33 to umount sshfs directory :

```
fusermount -u data_sedeboker
```

## 34 sshfs on mac :

(source : <http://superuser.com/questions/134140/mount-an-sshfs-via-macfuse-at-boot>  
)

```
brew install sshfs brew install fuse4x sudo bin/cp -rfX /usr/local/Cellar/fuse4x-
kext/0.9.2/Library/Extensions/fuse4x.kext /Library/Extensions sudo chmod
+s /Library/Extensions/Support/load_fuse4x
```

```
sudo mkdir -p /mnt/tsia sudo chown avigoz /mnt /mnt/tsia sudo chmod
a+rwX /mnt /mnt/tsia
```

now you should be able to manually mount the remote drive: sshfs  
tsia:/home/avigoz /mnt/tsia -oreconnect,allow\_other,volname=tsia,sshfs\_debug

so now /mnt/tsia includes files from the remote source. unmount it:  
umount /mnt/tsia

the following does not work properly for me. I do see the files but I don't  
have permissions to change them

if this works, pursue :

```
mkdir -p progs/sshfs/ cat «END > progs/sshfs/sshfs-authsock #!/bin/bash
export SSH_AUTH_SOCKET=$( ls -t /tmp/launch-*/Listeners | head -1)
/usr/local/bin/sshfs $* END
```

check the location of sshfs in the last line, since it might vary between  
versions of OS X .

```
chmod a+rwX progs/sshfs/sshfs-authsock
sudo emacs /Library/LaunchAgents/tsia.home.plist
and therein :
```

```
<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE plist PUBLIC
"-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.
com/DTDs/PropertyList-1.0.dtd"> <plist version="1.0"> <dict> <key>Label</key>
<string>tsia.home.sshfs</string> <key>ProgramArguments</key> <ar-
ray> <string>/Users/avigoz/progs/sshfs/sshfs-authsock</string> <string>avigoz@tsia:</string>
```

```
<string>/mnt/tsia</string> <string>-oreconnect,allow_other,volname=tsia</string>
</array> <key>RunAtLoad</key> <true/> </dict> </plist>
    with the obvious modifications of directory/file/user/host names .
    launchctl load /Library/LaunchAgents/tsia.home.plist % launchctl start
tsia.home.sshfs -> does not seem relevant
```

### 35 perl command line arguments :

```
(source : http://stackoverflow.com/questions/3515877/how-to-print-program-usage-in-perl)
    use Getopt::Long::Descriptive;
    my ($opt, $usage) = describe_options( 'diff_entire_directory.pl file_pattern
reference_directory', [ 'help|h', "print usage message and exit" ], );
    print($usage->text), exit if $opt->help;
```

### 36 sollution to matlab blurry imagesc :

```
eps2eps in_fig.eps out_fig.eps
```

### 37 mac os x : halt and resume processes :

```
kill -STOP PID kill -CONT PID
```

### 38 remove a huge buggy directory with a lot of files that just refuse to be removed (source : <http://serverfault.com/a/215766>) :

```
<?php $dir = '/directory/in/question'; $dh = opendir($dir) while (($file =
readdir($dh)) !== false) { unlink($dir . '/' . $file); } closedir($dh); ?>
```

### 39 xmgr different types of plots :

```
xmgrace -settype xysize
```

where the type may be :

XY 2 An X-Y scatter and/or line plot, plus (optionally) an annotated value  
XYDX 3 Same as XY, but with error bars (either one- or two-sided) along X axis  
XYDY 3 Same as XYDX, but error bars are along Y axis  
XYDXDX 4 Same as XYDX, but left and right error bars are defined separately



XYDYDY 4 Same as XYDXDX, but error bars are along Y axis XYDXDY  
 4 Same as XY, but with X and Y error bars (either one- or two-sided) XY-  
 DXDXDYDY 6 Same as XYDXDY, but left/right and upper/lower error  
 bars are defined separately BAR 2 Same as XY, but vertical bars are used  
 instead of symbols BARDY 3 Same as BAR, but with error bars (either  
 one- or two-sided) along Y axis BARDYDY 4 Same as BARDY, but lower  
 and upper error bars are defined separately XYHILO 5 Hi/Low/Open/Close  
 plot XYZ 3 Same as XY; makes no sense unless the annotated value is Z  
 XYR 3 X, Y, Radius. Only allowed in Fixed graphs XYSIZE 3 Same as XY,  
 but symbol size is variable XYCOLOR 3 X, Y, color index (of the symbol  
 fill) XYCOLPAT 4 X, Y, color index, pattern index (currently used for Pie  
 charts only) XYVMAP 4 Vector map XYBOXPLOT 6 Box plot (X, median,  
 upper/lower limit, upper/lower whisker)

## 40 xmgr

produce eps file without gui

```
gracebat -settype xydy gyre_anticyc_yz_year_1_season_1_exp23acont.txt
gyre_cyc_yz_year_1_season_1_exp23acont.txt -param ../vert_gyres.par
-printfile vert_gyres_exp23a.eps
```

## 41 matlab slice mat - file without reading all of it :

(source : )

```
file=matfile(filename); r=file.r(1:4,200,8); sz_q=size(file,q); vars=fieldnames(file);
plot(file.r(1:3,5));
etc...
```

when indexing a variable in matfile (e.g. file.r(1:3,1)) it is important

## 42 number of threads matlab uses for calculations

:

(source : <http://stackoverflow.com/questions/20648360/how-can-i-determine-the-number-of->  
 )

```
maxNumCompThreads
```

## 43 linux number of threads used by a program :

(source : <http://stackoverflow.com/questions/20648360/how-can-i-determine-the-number-of->  
)  
ps uH p <PID> | wc -l

## 44 checking a paper:

- spell check
- read abstract
- general look at figures
- format of references
- order of references
- structure :

abstract intro: general view, problem, several people who tackled it, new approach, outline of the paper methods results discussion acknowledgement  
refs

- graphs : good captions
- graphs : good legends, and axis labels that include units
- graphs : big fonts (around 16), big line widths (around 2), big symbols, grid lines

## 45 matlab cycle through colors when plotting in a loop

(source : <http://www.mathworks.com/matlabcentral/answers/25831-plot-multiple-colours-aut>  
use "hold all" instead of "hold on"

## 46 emacs assign file suffix to certain mode (here I use cuda in c++ mode):

(source : <http://stackoverflow.com/questions/8632325/start-c-syntax-highlighting-for-cu->

```
(add-to-list 'auto-mode-alist '(  
.cu  
'" . c++-mode))
```

## 47 emacs put backupfile in a dedicated directory.

```
(source : http://www.emacswiki.org/emacs/BackupDirectory)  
(setq backup-by-copying t ; don't clobber symlinks backup-directory-alist  
'(("." . "~/ .save")) ; don't litter my fs tree delete-old-versions t kept-new-  
versions 6 kept-old-versions 2 version-control t) ; use versioned backups
```

## 48 c++ precision of operator« :

```
std::cerr.setf(std::ios_base::scientific, std::ios_base::floatfield); cerr.precision(4);  
"scientific" can be replaced by "fixed"  
another possibility:  
cerr<<"stam mashehu"<<std::scientific<<somedouble<<endl;  
to always show signs : cerr<<std::showpos;
```

## 49 org mode inline code switches:

```
http://orgmode.org/org.html#session
```

## 50 mitgcm convergence criteria:

```
inertial oscillations:  
 $f^2 \Delta t^2 < 0.5$   
ACFL :  $u \Delta t / \Delta x < 0.5$ 
```

## 51 matlab modulo (almost) symmetric around zero :

```
mod(x+L/2,L)-L/2
```

## 52 youtube download an entire list with automatic numbering :

```
youtube-dl -i PLNiWLB_wsOg5urbUQZHdnRXw7KEO-FTie -o "earth%(autonumber)s.%(ext)s"
```

## 53 libreoffice openoffice change formatting of all sheets :

(source : <http://www.oooforum.org/forum/viewtopic.phtml?t=49217>)  
right click on a sheet, select all sheets, and change whatever you want

## 54 mac os x libreoffice calc , switch between sheets

(source : <http://ask.libreoffice.org/en/question/470/what-keyboard-shortcuts-are-used-to>  
cmd+pageup (or on a laptop : Fn + Command + up arrow / down arrow)

## 55 GMT pen attributes:

width,color,style

width = faint default thinnest thinner thin thick thicker thickest fat fatter  
fattest obese

this can also be indicated in numbers in the range [0 18p]

The color can be specified using:

1. Gray. Specify a gray shade in the range 0–255 (linearly going from black <sup>1</sup> to white

## 56 xclip equivalent in mac os x:

(source : <http://stackoverflow.com/questions/3482289/easiest-way-to-strip-newline-charac>  
pbcopy  
so to remove , and send to clipboard we'd do : alias xcn="tr -d " |  
pbcopy"

## 57 grep with or operator :

```
grep "hist\\|frac_larg"
```

## 58 to know which temp files are opened by a program :

`sudo opensnoop -n Emacs`

## 59 extract page range from a pdf file :

(source : <http://www.linuxjournal.com/content/tech-tip-extract-pages-pdf>)  
`pdftk A=100p-inputfile.pdf cat A22-36 output outfile_p22-p36.pdf`

## 60 make emacs work with an octave shell :

(source <http://stackoverflow.com/questions/24971756/emacs-stops-responding-when-i-run-r>)  
insert:  
`PS1("» ")`  
to your `.octaverc`

## 61 mitgcm, phihyd and phihydlow units:

(taken from <http://mitgcm.org/pipermail/mitgcm-support/2004-August/002438.html>)

$\frac{\partial \phi}{\partial r} = b \text{ is the } SCALED \text{ density } \rho / \rho_{\{0\}}$ . (In fact, it's the scaled density anomaly  $g(\rho_{\text{rho}_{\{0\}}}) / \rho_{\{0\}}$ ).

So when you backout pressure from phiHyd, you have to multiply by  $\rho_{\{0\}}$

For the full pressure, you'll have to add the constant density contribution  $-g\rho_{\{0\}}z$ .

$$P_{\{b\}} = \text{phiHydLow} * \text{rhoConst} + g * \text{rhoConst} * H$$

## 62 python read mat files (using the hdf5 capabilities)

(source: <http://stackoverflow.com/questions/17316880/reading-v-7-3-mat-file-in-python>)  
`import h5py f = h5py.File('test.mat')`  
`f.keys()` should give you the names of the variables stored in 'test.mat'.  
you can access `f['s']1` etc..  
for mat files that were not saved with the option 'v7.3' :  
`from scipy.io import loadmat mat = loadmat('measured_data.mat')`

## 63 echo without new line

(source : <http://www.unix.com/unix-for-dummies-questions-and-answers/88784-echo-without-newline-character.html>)  
echo -n "text "

## 64 diff between multiple files

(source : <http://unix.stackexchange.com/questions/33638/diff-several-files-true-if-all-n>  
/usr/bin/diff -qs --from-file ../code/packages.conf\_cont40 ../code/packages.conf\_cont40\_0\*

## 65 slurm number of cpus ("allocated/idle/other/total")

sinfo -o "%C"

## 66 cvs adopt the repo version (revert to repo version and discard your own's)

(resource : <http://stackoverflow.com/questions/15704945/how-to-revert-the-file-in-cvs>)  
cvs update -C utils/matlab/rmdms.m

## 67 missing libraries in compilation :

(source : <http://prefetch.net/articles/linkers.badldlibrary.html>)  
to deal with this kind of error : \$ curl ld.so.1: curl: fatal: libgcc\_s.so.1:  
open failed: No such file or directory Killed  
run : ldd curl  
and add the missing libraries to ld\_library\_path

## 68 emacs orgmode bibliography

in .emacs : (custom-set-variables ... .. '(org-latex-pdf-process '("latexmk  
-pdflatex='pdflatex -interaction nonstopmode' -pdf -bibtex -f %f"))  
in the org file :

## 69 blogofile basics :

(source : <http://docs.blogofile.com/en/latest/index.html>)

### **69.1 Initialize a blog site in a directory call mysite:**

» `blogofile init mysite blog`

### **69.2 Build the site:**

» `blogofile build -s mysite`

### **69.3 Serve the site:**

» `blogofile serve -s mysite`

### **69.4 help**

» `blogofile help`