

## Integrating Pecube and Move: A brief runthrough

Compiled by Victoria Buford

[Vmb21@pitt.edu](mailto:Vmb21@pitt.edu)

Updated: October 30, 2018

Overview: This writeup discusses the following:

- Creation of point cloud in MATLAB
- Import, deformation, and export of point cloud in MOVE
- Exporting of topo lines and structure lines in MOVE
- Folder & File Structure for Pecube & Plotting in MATLAB
- Setting up all Pecube files
- Connecting to the server using Fetch (Ftp program) and Terminal/Command Prompt
- Running Pecube model
- Downloading & Analyzing the Pecube Model in MATLAB

Summary of files:

- Required input files for Pecube
  - Grids (\*.dat)
    - Tab or space separated (space is best)
    - No header
    - In kilometers
    - [X      Y      Z      Colour ID      UID]
  - Topos (\*.dat)
    - Tab or space separated (space is best)
    - No header
    - In kilometers
    - [X      Y      Z      ID]
  - Velocity (\*.txt)
    - 4 header lines followed by ages and grid files
  - Pecube.in
    - Edited to reflect your model timing, thermal parameters, topography file names, and which thermochronometers to model (and more)
  - Run\_pecube.sbatch
    - Doesn't need to be edited (other than to put your email address on it the first time), unless you want to run on a specific node
- Pecube output files produced
  - Vel\_new
    - Download to analyze if the velocity produced is correct. Can help identify problems with Pecube or problems with the grid itself
  - Vel\_top
    - We don't really use this one
  - Pecube\_errors
    - Will be 0kb if nothing is wrong, but if the model crashes, will list a (sometimes cryptic) reason for the crash
  - \*\*.log

- will grow the entire time the model is running, detailing what pecube is computing when. Lists the time to complete when the model is done. Is helpful for understanding where in the model run pecube crashed.
- temps\_tec\*\*.dat
  - for each model step, this lists the temperatures for the full modeled space along the cross-section (so both x and z)
- ages\_tec\*\*.dat
  - for each model step, this lists the ages for the modeled thermochronometers and along-section (x) coordinates; the spacing of these points is determined by the pecube spacing you selected, not by your Move grid spacing.
- Required files for MATLAB Plotting with DLG script
  - Vel\_new (\*.dat downloaded from Pecube)
  - Ages\_tec (\*.dat downloaded from Pecube)
  - Temps\_tec (\*.dat downloaded from Pecube)
  - Structure lines (\*.dat exported from Move)
    - Space-separated
    - In kilometers
    - With a Header row
    - [X    Y    Z    ID]
- Optional Files for MATLAB plotting with DLG script
  - Thermochronometer data (\*.txt)
    - Tab-separated
    - With header which contains: (Method Age Error xdist)
      - With that capitalization
      - Methods available: AFT, AHe, ZFT, ZHe, Mar
      - Age and Error in Ma
      - xdist in km, corresponding to location matching the section

#### Recommended programs

- Text Editors
  - Mac: TextEdit (Default), Xcode, Smultron (\$)
  - Windows: Notepad ++ (NOT NOTEPAD), Atom
- SFTP Programs
  - Mac: Fetch
  - Windows: WinSCP
- Command Line Interface
  - Mac: Terminal (default)
  - Windows: Command Line (default), PuTTY
- Plotting/Analysis Software
  - MATLAB
- Image Viewers
  - Mac: Preview (default)
  - Windows: Photos (default), Windows Photo Viewer

## 1) Setup

### 1a) Creating the Point Cloud (grid)

Create a 0.5km x 0.5 km grid in MATLAB such that the grid extends from the basal décollement up to the surface. Make sure that the grid exists over 0km in places that form basin fill, (eg if the expected basin is 4.5km deep, make sure in the initial setup your grid extends up to 4.5km elevation in the basin area).

The MATLAB script is titled Create\_MOVEPointCloud.m and consists of the following:

```
function [MOVE_PointCloud] =...
    Create_MOVEPointCloud(x_max, x_min, z_max, z_min, resolution)
% INPUT:
% x_max, x_min, z_max, z_min: integers of bounding box in [km]
% resolution: MOVE point cloud resolution in [km]

% OUTPUT:
% columns: x,y,z, color ID, unique ID
% Paul R. Eizenhofer, PhD
% University of Pittsburgh, peizen@pitt.edu

[X,Z] = meshgrid(x_min:resolution:x_max, z_min:resolution:z_max);
MOVE_PointCloud(:,1) = X(:);
MOVE_PointCloud(:,2) = 0;
MOVE_PointCloud(:,3) = Z(:);
MOVE_PointCloud(:,4) = 0;
for i = 1:length(MOVE_PointCloud(:,3))
    MOVE_PointCloud(i, 5) = i;
end
```

The resolution for our MOVE models is 0.5km.

Then, open the variables section\_ptcld and basin\_ptcld in MATLAB, copy them into a text editor, and save as a .txt file. (You can paste them into the same text file, or separately, if you prefer to keep them as separate point clouds.)

*Note:* If you wish to create a non-rectangular grid, you can do so by creating the two clouds separately, but starting the unique IDs (UID) of the second cloud at the max UID of the first plus one. Eg.

```
section_ptcld=Create_MOVEPointCloud(500,0,0,-15,0.5);
basin_ptcld=Create_MOVEPointCloud(650,500.5,5,-15,0.5);
basin_ptcld(:,5)=basin_ptcld(:,5)+max(section_ptcld(:,5));
```

### 1b) Importing the Point Cloud into MOVE

Open your MOVE Model in cross-section view, use File>Insert (Ctrl+I, Cmd+I) and select the .txt file(s) of your point cloud(s). Change “Select Object Type” to “Points”, ensuring the units are in kms. Ensure the columns read: X Y Z Colour ID UID. If you don’t have them, right click on the column and select Colour ID and then click “Create New” (for UID). Title it *UID*, Type: *Integer*, and Category: *Dimensionless*.

In the Model Browser, click “Clear Filters”, and under “Object Types” find “Point Data” and select your point cloud. Go to Project> To Section, click “Add” or “Collect” and click “Apply”. You can do the majority of your work with the visibility toggled off, so that MOVE will run faster.

### 1c) Deforming the Point Cloud in MOVE

Deform, Load, Erode, Unload, (and sediment load, if required) as normal, but be sure to add the point cloud into the objects being deformed/loaded/unloaded.

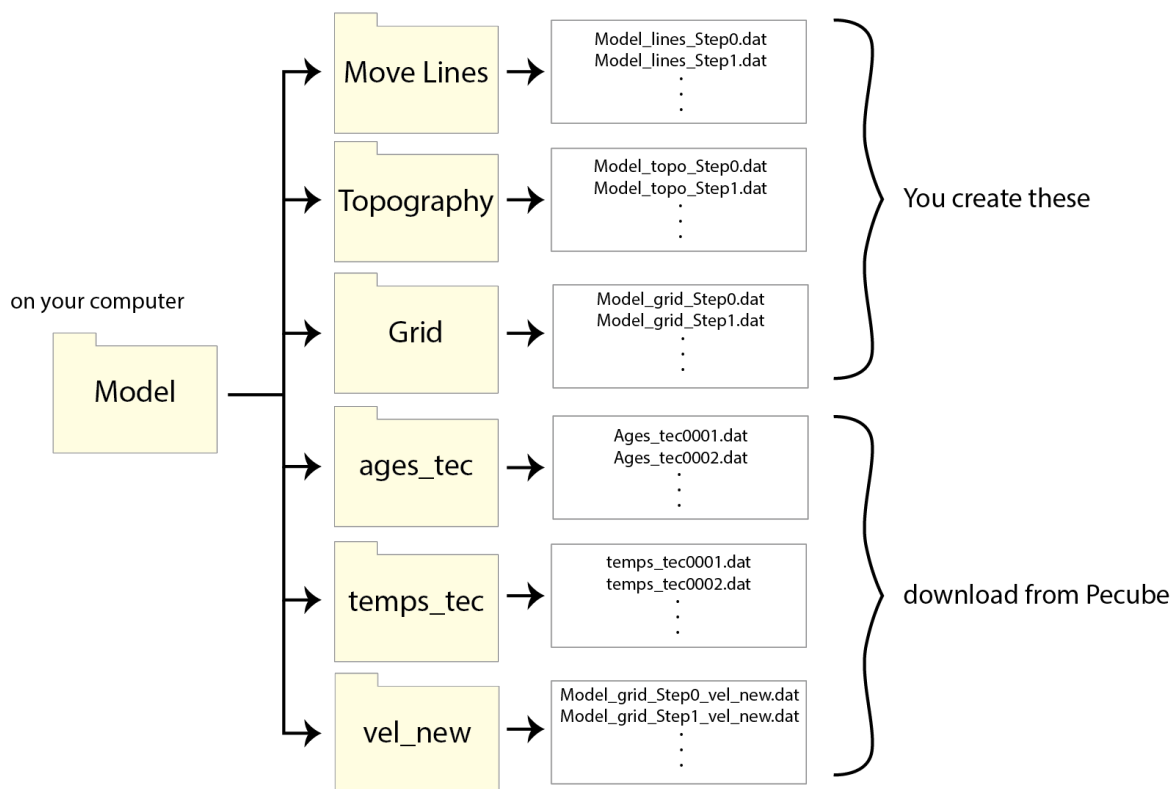
*Note:* Having the point cloud visible, particularly when performing a step that involves deformation or movement, makes MOVE run slower because it has to render the grid. Thus, you can do this with visibility off (the little checkbox) and just select the point cloud, and click “Add” in the Move on Fault or Decompaction Modules. After you have performed the movement, you should toggle the visibility of the point cloud on to make sure that the cloud has actually moved like it should.

### 2) Exporting the Point Cloud

*Note on Naming Schemes:* The simplest naming scheme is the best. Generally, we prefer to have the total deformation in kms in the name, but this creates problems for MATLAB later on. You can either name with the deformation amounts in the name, and rename them for MATLAB plotting later, or use the scheme suggested below if you don’t want to rename it. If you decide to have step identifiers other than step #, be careful to not make the name too long as Pecube/Cascade can cut it off if it’s too long.

*Model\_Step#\_(object type) is best for MATLAB plotting later on (Object type = lines, grid, topo)*

*Folder Structure Setup:*



At the end of each step (DLEU), export the grid, topography, and the Move Lines.

*Grid:* Make sure (by toggling visibility on) that the grid has actually deformed, loaded, and unloaded. properly. To export the grid (Even with visibility off) select it in the model browser, and go File > Export (Ctrl+Shift+E Cmd+Shift+E). Change File Format > ASCII, and click “Match Model Selection”, then click Next. Select X, Y, Z, Colour ID, and UID as attributes, and change Separator to “Space”. On the Next screen, change the XY and Z units to kilometers, and do not write a header. Select where, and what, to name the file (keeping in mind the *note* above).

*Topo:* Select the single topography line, and Export. You need X, Y, Z, and ID, space-separated, in kilometers, with no header.

*Lines:* Select all Move structural and bedding lines (ctrl+A or cmd+A, ctrl or cmd click the section trace and any lines you don’t want) and export. You want X, Y, Z, ID, space-separated, in kilometers, WITH A HEADER.

### 3) Model Setup

You can create all three of the following files using the Matlab script

*Pecube\_CreateInputFiles.m*. This will allow you to select your grid and topo files, input the ages, thermal paramters, and model names, and will create the velocity.txt file, pecube.in file, and run\_pecube.sbatch file.

*Note: All files created will have .txt extension. You MUST CHANGE the Pecubein.txt extension to Pecube.in for it to run. If you don’t change the sbatch extension, use the command sbatch run\_pecube.txt to start the model.*

*To change the extension:*

- On Windows, Use Notepad ++ (NOT NOTEPAD); open Pecubein.txt: save as, Pecube.in
- On Mac, right click on Pecubein.txt, click “Get Info”. In the Name & Extension: blank, change the name to Pecube.in; when a dialog pops up, click “Use .in”

**velocity file:** list the ages and corresponding grid for n+1 deformation steps (all def steps+starting); include 10 5 10 1.0 at beginning.

```
10
5
10
1.0
age_start model_grid_step0.dat
.
.
.
0 model_grid_steplast.dat
```

**Note: DO NOT COPY AND PASTE FROM ANY MICROSOFT PROGRAMS INTO THE TEXT EDITOR.**

It makes pecube angry. Pretty much any text editor works, some of use XCode, Smultron, Mac’s TextEdit, Atom (Windows).

*Note:* the first four lines correspond to: (1) the y dimension in km, (2) # nodes in y-direction, (3) interpolation window for averaging over the MOVE grid, and (4) x step-size for interpolation (of topo).

### The pecube.in file

*This is where you enter the names of the topography files exported from Move, select thermal parameters, select which thermochronometers to plot, and more. n=#def steps.*

*-Check the box folder for an example pecube.in file*

**(input #2):** # of topo files (n+2)

**(input #4):** Names of topo files (n+2) (step 0 listed twice).

Model\_topo\_step0.dat

Model\_topo\_step0.dat

Model\_topo\_step1.dat

.

.

.

Model\_topo\_steplast.dat

**(input #6):** initial, undeformed width, (in node spacing): take the width of your model (eg, it runs from -65 to 600 (so, 665km wide), and you want two nodes per km; so, you type 1330 5).

The 5 indicates width in the along strike direction, and we don't change it.

**(input #7):** spacing of nodes in meters (so, if you did two nodes per km: 500 1000) (we don't change the second 1000 bc we don't really care about the along strike direction in this instance)

**(input #9):** starting location, in meters, for the pecube grid; (example above: -65000 0.0)

**(input #10):** # of deformation steps (n+1)

**(input #12a):** change column (a) to correspond with time steps

**(input #14):** where you alter thermal parameters: atmospheric lapse rate (i), e-folding depth (k), thermal heat production (j)

**(input 17):** which Tchron systems you want to be calculated (0=no, 1=yes)

**(after input 22):** at the end of the pecube.in file, type the name of the velocity file (.txt).

### The run\_pecube.sbatch file

```
#!/bin/bash -l
```

```
## Run script for pecube monte carlo on esd slurm
```

```
## General configuration options
```

```
#SBATCH -J Pecube_W
```

```
#SBATCH -e Pecube_Error%j
```

```
#SBATCH -o Pecube_Screen%j
```

```
#SBATCH --mail-user=email_id@pitt.edu
```

```
#SBATCH --mail-type=ALL
```

```
## Machine and CPU configuration
```

```
## Number of tasks per job:
```

```
#SBATCH -n 1
```

```
## Number of nodes:
#SBATCH -N 1
#SBATCH -w u-017-s133
```

```
module load pecube
```

```
pecube
```

*Note: if you want to run your model on a less busy node, check which nodes are busy using squeue. (See section 6 for more details on how to implement Command Prompt commands.)*

*Include this in the .sbatch file in file: the highlighted area:*

```
#SBATCH -w esd_node1 (to run on #38)
#SBATCH -w esd_node2 (to run on #39)
#SBATCH -w argand (to run on #40)
#SBATCH -w u-017-s133
#SBATCH -w u-017-s134
#SBATCH -w u-017-s135
#SBATCH -w u-017-s136
```

#### 4) Setting up SFTP connections using Fetch (or another ftp program; WinSCP for windows)

Download and install Fetch.

File > New Connection

Hostname: 134.2.5.40

SFTP

Username: whatever Willi assigns you

Password: whatever willi assigns you

Initial folder: /esd/esd01/data/username

Port: 6307

Click the Heart to SAVE

Check add to keychain

In your personal folder, you can do what you want.

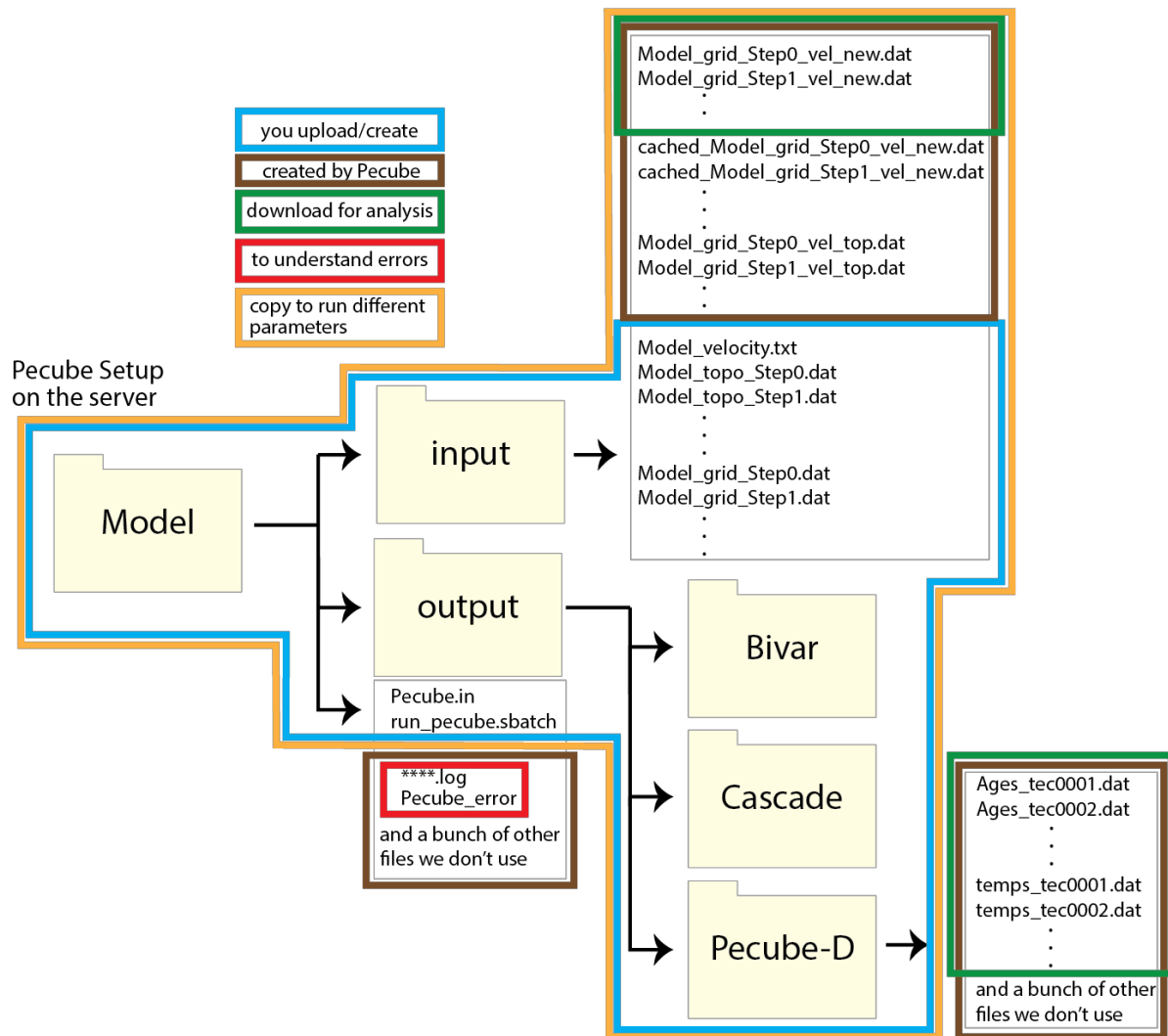
You are given the following folders:

*data\_large*

*model\_runs*: for important models; is backed up

*scratch*: for testing, less important runs; is NOT backed up

#### 5) Upload your model according to the folder structure:



## 6) Running your model

- Using Terminal (Mac) or Command Prompt (Windows; or Putty) use `ssh -l username -X -p 6307 134.2.5.40` to logon to the server (replace `username` with yours). Then type in your password. It won't be visible.
- Use `cd /esd/esd01/data/username/` to switch to your folder, and then navigate to your specific model (eg `cd /model_runs/Model`).
- Then, type in `sbatch ./run_pecube.sbatch` (for pecube). It should confirm the job and give you a number.
- Check to make sure it's still running with `squeue` OR `squeue -u username`. If it computes the velocity files (found in the input file), you're most likely good (this takes 1-4 hrs).



### Terminal commands

|                                       |  |
|---------------------------------------|--|
| ssh -l username -X -p 6307 134.2.5.40 | log on to the server   |
| squeue                                | find out what's running  |
| squeue -u username                    | what are you running   |
| cd ..                                 | go up a directory  |
| cd /esd/esd01/data/username/          | change directory to your   |
| sbatch run_pecube.sbatch              | run a pecube file  |
| scancel job#                          | cancel a job   |
| ls                                    | display contents of folder, indicates what other folders are there<br>(Can use dir also) |
| scp -r source/folder                  | copy a folder (source) and its contents to destination                                   |
| destination/folder                    |  |
| exit                                  | log off  |

*Note: if you want to run your model on a less busy node, check which nodes are busy using squeue. Include this in the run\_pecube.sbatch file:*

```
#SBATCH -w u-005-s038 (to run on #38)
#SBATCH -w u-005-s039 (to run on #39)
#SBATCH -w u-005-s040 (to run on #40)
```

### 7a) Analyzing your Pecube Model output

Download the vel\_new (from input folder), temps\_tec (from output folder), and ages\_tec (from output folder). Save them according to the folder structure in in section 2.

Remove temps\_tec0000.dat from the other temps\_tec folder on your computer (otherwise you'll have too many files).

Ensure that in addition to the three folders listed above, you also have the move structures /lines exported as listed in section 2.

Place PecubeOutputPlot\_batch\_DLG\_v\*\*\*\*.m and inputsdlg.m in your MATLAB folder. Open PecubeOutputPlot\_batch\_DLG\_v\*\*\*\*.m and in Editor, click Run. This will pop up a dialog box where you can select the folders for your lines, velocities, ages, temperatures, and where to save the output.

The **Model Name** is very important, and must be the phrase by which you named your lines and grids (and thus velocities). This is how MATLAB searches for which files to plot.

**Model ID** is just a catch-all text blank for you to identify which model you are plotting. It gets plotted in the title of the plot, and included in the save name for the figure. It is not used for searching for files.

**Deformation front on:** right OR left. This controls where the legend is plotted so as to not obscure the data due to the deformation front.

**Move Lines Folder:** click in this blank and select where your lines/structures are stored

**Velocity Folder:** click in this blank and select where your (vel\_new) velocities are stored

**Ages:** click in this blank and select where your ages\_tec\*\*\*\*.dat are stored

**Pecube Temperatures Folder:** click in this blank; select where your temps\_tec\*\*.dat are stored

**Save Folder:** Select where to save the figures.

**XRange:** Type in what xrange you want plotted (this is the along cross-section distance; positive to the right, like a normal plot)

**ZRange:** Type in what depth/elevation range you want to plot

**Age Range:** Type in what ages you want to plot (in Ma; generally, model start age + time you put in to equilibrate) (0-100Ma is the default)

**Figure Format:** the extension for the figure to be saved in (eg. .png, .epsc, .fig)

**Tchron Smoothing Method:** using MATLAB's `smoothdata` function, this will average the data using the *Method* you select over the *window size* you select. Using a window size of 1 will not average/smooth the data. If you used a 0.5km grid and node spacing (in Pecube), then a window size of 3 will average over the nearest 0.5km on both sides of the point it is computing at. Window size of 5 = nearest 1km. Window size of 7 = nearest 1.5km. If you put in an even number, it will do compute the average over a trailing window: eg, with a window size of 4, MATLAB will smooth over two behind and one in front. If you aren't sure which method to use, `movmean` or `movmedian` are a good place to start.

**Timesteps to plot:** Will either plot All OR Range. For Range, you must enter something in the *From X:Y*. If you just want to plot one timestep, just enter the number by itself (eg 12) or a range (23:25).

**Tchron data file [txt]:** a text file containing a tab-separated list of thermochronometer data. You must have a header row with: Method, Age, Error, and xdist (with that capitalization).

*Method options:* (as of v 1.4) MAr, ZFT, ZHe, AFT, AHe. Must be tab separated

**Thermochronometers to plot:** Check which thermochronometers to plot. "Modeled" pulls from the pecube output, while "data T=end" pulls from the Tchron data file.

The function is setup to remember your last inputs as long as you don't clear MATLAB's working memory (clear). To rerun subsequent times, either "Run Section" (Cmd+Enter) or comment out (use % at the beginning of the line) the `clear; close all; clc;` line (on or about line 44).

## 7b) Ensuring no vertical exaggeration

Currently, this relies on the MATLAB command `daspect`. However, sometimes this resizes the cross-section view of the subplot so that it is narrower than the Age plot. So, either, turn off the `daspect` command at the end of the plotting, or use the following description to set it up so it works for the ranges you want.

As this depends on the monitor, XRange, and ZRange, the simplest way is to setup your MATLAB runs as follows:

Comment out the `saveas(***)` and `close` lines at the very end.

Make sure the `daspect` command is NOT commented out (at the end of plotting).

Run the script for just one timestep

Adjust the figure window to the size/shape you like.

In the command window, type `get(gcf, 'Position')`

Replace the `set(gcf, 'Position', [1 5 1280 700])` with the numbers from the command window.

You can now uncomment the `saveas(***)` and `close` lines and run the entire script for your entire model (for this specific XRange and ZRange; it may or may not work for other ones) and it will plot with no vertical exaggeration.

### 7c) Troubleshooting the MATLAB script

#### *Files appear to be missing!*

This means that in the folders you've included, there aren't an equal number of files. The command window should display the number of files it is finding in each folder. Sometimes, this is a problem of leaving `temps_tec0000.dat`, and sometimes it is a problem with a storage volume in that MATLAB is seeing hidden files. If you know you have the correct number of files in the folders, and you know how many total files that is, you can override this error by typing in the command window: `TotalFiles= ##`, where `##` is the number of files/steps you have.

*Warning: Duplicate data points have been detected and removed -corresponding values have been averaged.*

You can basically ignore this.

#### *Exceeds matrix dimensions.*

This is basically the least helpful error ever, but may be caused by not selecting the exact thermochronometers for `pecube` to create that this file was written for (AHe, AFT, ZHe, ZFT, and MAR). In the `%%% Plot ages %%%` section (approx. line 470), you'll have to change which column `Age_Avg` is pulling the data from. Open an `ages_tec` file (in excel, or a text editor), and look at the header row. Match up which column goes with which thermochronometer, and alter the y value for each thermochronometer to plot. For example, if you choose for `Pecube` to create AHe, AFT, ZHe, and MAR, then the script should read as follows (the changes have been highlighted):

```
% MAR
if MAR==1
plot(Age_Avg(:,4), Age_Avg(:,10),...
    'o-',...
    'MarkerEdgeColor',[0.6350 0.0780 0.1840],...
    'MarkerFaceColor',[0.6350 0.0780 0.1840],...
    'MarkerSize',4);
leg(end+1)=[ 'MAR' ]};
end

% ZFT
if ZFT==1
plot(Age_Avg(:,4), Age_Avg(:,10),...
    'd-',...
    'color',[0.4940, 0.1840, 0.5560],...
    'MarkerFaceColor',[0.4940, 0.1840, 0.5560],...
    'MarkerSize',4);
leg(end+1)=[ 'ZFT' ]};
end

ZHe
if ZHe==1
plot(Age_Avg(:,4), Age_Avg(:,9),...
    'v-',...
    'color',[0.8500, 0.3250, 0.0980],...
    'MarkerFaceColor',[0.8500, 0.3250, 0.0980],...
    'MarkerSize',4);
leg(end+1)=[ 'ZHe' ]};
end

% AFT
if AFT==1
plot(Age_Avg(:,4), Age_Avg(:,8),...
    's-',...
    'color',[0, 0.4470, 0.7410],...
    'MarkerFaceColor',[0, 0.4470, 0.7410],...
    'MarkerSize',4);
leg(end+1)=[ 'AFT' ]};
end

% AHe
if AHe==1
plot(Age_Avg(:,4), Age_Avg(:,7),...
    'p-',...
    'color',[0.4660, 0.6740, 0.1880],...
    'MarkerFaceColor',[0.4660, 0.6740, 0.1880],...
    'MarkerSize',4);
leg(end+1)=[ 'AHe' ]};
end
```

#### *TchcronData.Method does not exist.*

Oops, either you selected the wrong file to import, or something isn't importing correctly. Sometimes MATLAB is finicky with the importation of tables. Check `TchcronData` by double clicking it in the Workspace. Are the table headers what you typed in, or are they

something generic like “Var4”? If they are generic, you can rename them in MATLAB and comment out the line `TchcronData=readtable(Answer.TchcronDataFile,...`  
`'Delimiter','\t');` so that MATLAB will just use the edits you just made rather than reimporting the file.

If they aren’t generic, you may have not used the capitalization scheme: Method, Age, Error, xdist. Fix that in your txt file, and it should import fine.

#### 7d) Troubleshooting login via Command Line.

On mac, if they update the server security settings, you may need get a warning like this:

```
[cl2-wifi-10-215-50-50:~ victoriabuford$ ssh -l vbuford -X -p 6307 134.2.5.40
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:iEarESpjKBgAt0Alss/7nKmYGFMM2myBLJcYrM0f8A0.
Please contact your system administrator.
Add correct host key in /Users/victoriabuford/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /Users/victoriabuford/.ssh/known_hosts:1
ECDSA host key for [134.2.5.40]:6307 has changed and you have requested strict checking.
Host key verification failed.
cl2-wifi-10-215-50-50:~ victoriabuford$
```

To fix this, ensure you are in `/Users/your_user_name/` (by `$ pwd`), and then

```
$ rm ~/.ssh/known_hosts
```

to remove the old authentication keys. Then, log back into the server, and you’ll probably have to confirm the new authentication.

```
[cl2-wifi-10-215-50-50:/ victoriabuford$ pwd
/
[cl2-wifi-10-215-50-50:/ victoriabuford$ cd /Users/victoriabuford/
[cl2-wifi-10-215-50-50:~ victoriabuford$ rm ~/.ssh/known_hosts
[cl2-wifi-10-215-50-50:~ victoriabuford$ ssh -l vbuford -X -p 6307 134.2.5.40
The authenticity of host '[134.2.5.40]:6307 ([134.2.5.40]:6307)' can't be established.
ECDSA key fingerprint is SHA256:iEarESpjKBgAt0Alss/7nKmYGFMM2myBLJcYrM0f8A0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[134.2.5.40]:6307' (ECDSA) to the list of known hosts.
[vbuford@134.2.5.40's password:
Creating directory '/home/vbuford'.
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-38-generic x86_64)
```

#### 8) Notes on operating system of the server, if you’re curious.

Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-38-generic x86\_64)

- \* Documentation: <https://help.ubuntu.com>
- \* Management: <https://landscape.canonical.com>
- \* Support: <https://ubuntu.com/advantage>

System information as of Tue Oct 30 15:44:21 CET 2018  
System load: 2.02 Processes: 342  
Usage of /: 5.8% of 915.40GB Users logged in: 4  
Memory usage: 9% IP address for eno1: 134.2.5.40  
Swap usage: 0%

- \* Security certifications for Ubuntu!  
We now have FIPS, STIG, CC and a CIS Benchmark.  
- [http://bit.ly/Security\\_Certification](http://bit.ly/Security_Certification)

\* Want to make a highly secure kiosk, smart display or touchscreen?  
Here's a step-by-step tutorial for a rainy weekend, or a startup.  
- <https://bit.ly/secure-kiosk>

\* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
<https://ubuntu.com/livepatch>

0 packages can be updated.  
0 updates are security updates.

The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.