

LitMod2D_2.0 manual

Ajay Kumar *¹

¹Group of Dynamics of the Lithosphere, GEO3BCN-CSIC
Barcelona, Spain , Email: ajay6763@gmail.com

October 2020

*SUBITOP (<http://www.subitop.eu/home/>)

1 Introduction

LitMod2D_2.0 is a finite element code which combines potential field, geochemical and seismological data to work out thermo-chemical structure of the crust and upper mantle (Fig. 1).

This document is a manual for a python based GUI to build the model where the user draws geometry of the bodies in the cross-section and associate physical properties to those bodies. It also includes various post-processing schemes incorporated in the LitMod2D_2.0 to incorporate addition datasets (e.g., passive seismological) to constraint the modelling.

2 Installation

There are two ways in which you can use LitMod2D_2.0: i) through GitHub and installing all required libraries and ii) through docker and downloading a Linux image of LitMod2D_2.0 allowing it to be used in any operating system. If you are comfortable with the hassle of installing required python libraries then go for GitHub or if you do not want to get into this haste and save time, then I highly recommend using the docker option.

2.1 git hub way

User can download or clone the package from https://github.com/ajay6763/LitMod2D_2.0_package_dist_users.git. You will have following directory structure:

```
LitMod2D_2.0_package_dist_users
├── Generator_Linux -- to generate material file
├── GUI -- includes the GUI in python
├── manual -- Manual for GUI use
└── Post_processing -- packages for post-processing
    ├── flexure_tao
    ├── Phase_diagrams
    ├── RF
    ├── Surface_wave_dispersion
    └── Synthetic_Seismic_tomography
REAME.md
```

To setup Generator, follow the instructions in REAME.md file in the Generator_Linux directory.

Now we need to setup the GUI which essentially means we need to install python libraries. Generally, Linux comes with installed python2.7 but in Windows you might have to install python2.7 (<https://www.python.org/getit/>).

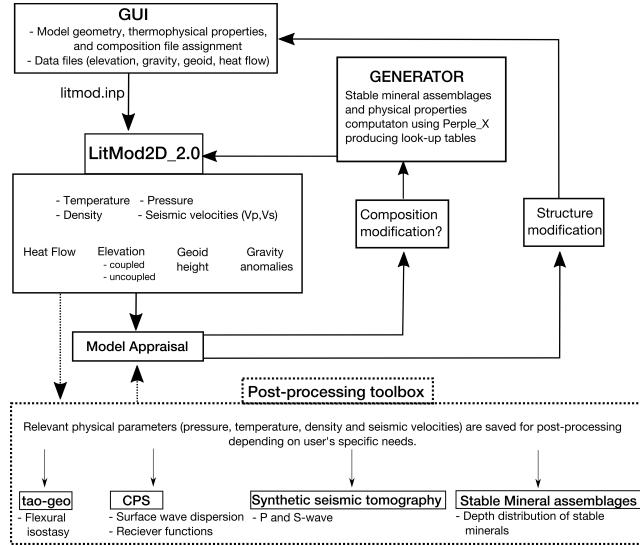


Figure 1: General modelling work-flow in the LitMod2D_2.0

This GUI uses packages from python which do not come pre-installed with stand-alone python installation. This GUI is python2.7 compatible. All packages used here can be installed using "pip" a python package manager which can be easily installed in Windows or Linux distributions (<https://pip.pypa.io/en/stable/installing/>).

I would highly recommend to make a virtual python environment for this GUI and install the required libraries in that environment. To set-up a python environment do following:

```
pip install virtualenv
virtualenv -p /usr/bin/python LitMod2D_2.0
```

Above command will make a LitMod2D_2.0 folder in the current folder with a dedicated python. Now you have to activate it by running following:
source ./LitMod2D_2.0/bin activate

Now stay here and navigate to the GUI folder and run the following command:

```
pip install -r requirements.txt
```

This should have you almost everything needed for the GUI.

Now every time you have to run LitMod2D_2.0 you have activate this en-

The screenshot shows a terminal window with several command-line sessions:

- Session 1 (Top):** Shows the output of the command `docker images`. It lists various Docker images with their tags, IDs, creation dates, and sizes. One image, `ajay6763/litmod2d_2.0 final`, is highlighted.
- Session 2 (Middle):** Shows the command `xhost +` being run to enable X11 access.
- Session 3 (Bottom):** Shows the command `docker run` being used to start a new container. The command specifies the image (`ajay6763/litmod2d_2.0`), removes old containers (-r), attaches to the terminal (-t), and connects to the host's X11 display (-v). It also sets environment variables for the X11 socket and QT_X11_NO_MITSHM.
- Session 4 (Bottom-most):** Shows the user navigating into the container directory (`/home/LitMod2D_2.0_package_dist_users/GUI`) and running the application with `python main.py`.

Annotations with arrows point to specific parts of the terminal output:

- An arrow points to the top session with the text "listing docker images".
- An arrow points to the middle session with the text "running the docker container".
- An arrow points to the bottom-most session with the text "running GUI in inside the container".

Figure 2: Docker images list

vironment as explained above and run the GUI. To test the installation a test model LitMod_test_model is provided. User can load this model in the GUI or test the post-processing steps explained in the next sections.

Once you have everything installed you need to add the LitMod2D to your path.

2.2 Docker way

Docker can be installed from the official website(<https://www.docker.com/products/docker-desktop>).

Once you have docker installed and working properly, you need to download the image of LitMod2D_2.0 from the docker hub. To do this run the following command in a terminal (for Linux or Mac OS) and command prompt (Windows OS):

```
docker pull ajay6763/litmod2d_2.0:final
```

To download it might take a while. Once the downloading finishes check if you have the image downloaded by running following command:

```
docker images
```

The output of the above command should give you the first line in the Fig. 2.

Docker image of the LitMod2D_2.0 behave as a stand-alone distribution (like a virtual machine). Since there is a GUI involved, we need to attach a server (X11 for Linux) to the docker image in order see the graphical components. Now to run the GUI follow the following instructions depending on your operating system:

2.2.1 Linux

To connect to a host server run following command:

```
xhost +
```

Now, you are ready to run the GUI. Follow the instruction in the section 2.2.4.

2.2.2 Mac OS

To connect to X11 server in Mac OS we to need install xquartz and can be done by running the following command in terminal:

```
brew cask install xquartz
```

Now, we need to allow connections from network clients and can be done by running following command in the terminal and setting the preferences as shown in Fig. 3:

Now we will get the IP address using following command and it to the xhost server:

```
$ ip=$(ifconfig en0 | grep inet | awk '$1=="inet" {print \$2}')
$ xhost +$ip
```

Now, you are ready to run the GUI. Follow the instruction in the section 2.2.4.

2.2.3 Windows 10

Similar to the Mac OS, in windows we need to install a Xming server in and can be downloaded from here <https://sourceforge.net/projects/xming/>.

Now, launch XLaunch from the start menu and select the option Start no client.

Now open powershell/command-promt. In the powershell we will set the DISPLAY variable with the IP address. IP address can be found using ipconfig command. Run the following command in the powershell where my_IP_address is your IP address:

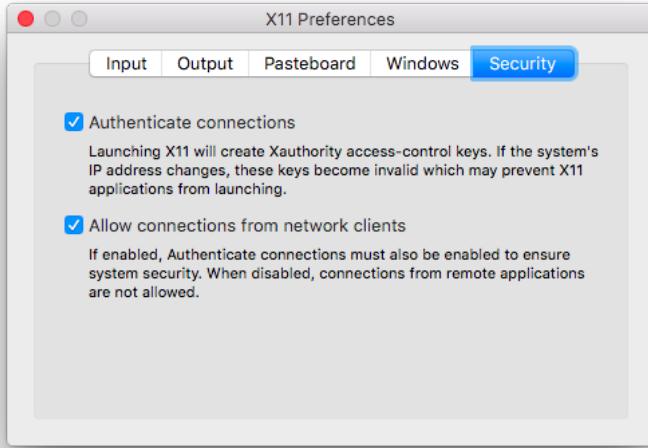


Figure 3: Xquartz preferences for Mac OS

```
set-variable -name DISPLAY <my_IP_address>:0.0
```

Now, you are ready to run the GUI. Follow the instruction in the section 2.2.4.

2.2.4 Running GUI from the image

Docker images are run as containers which is an instance of a docker image. But before running a container we need to share data between the host machine (your computer) and the container so that you can save your work and use it outside of the container. This is done in the command below where host_directory is the absolute path of the folder you want to share with the container and container_directory (e.g., /home/work) is the absolute path inside the container where host folder will be shared. This allows to share and reflect changes in this directory on your host computer and inside the container.

Now, to run the images you will have to run the following command (For Linux and Mac OS):

```
docker run -rm -ti -v host_directory:container_directory -net=host -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix -env QT_X11_NO_MITSHM=1 image_id
```

For, Windows run the following command:

```
docker run -ti -rm -v host_directory:container_directory -e DISPLAY=$DISPLAY  
image_id
```

image_id should be replaced with the value of LitMod2D_2.0 image id in the Fig. 2.

After a successful execution of the docker run command your prompt in the terminal will change to as shown in Fig. 2. You can see the full distribution and directory structure inside the container as shown in section above.

Now, you are ready to run the LitMod2D_2.0 GUI. To check if everything is working fine run python main.py in the GUI folder and load the model from LitMod_test_model by browsing through the GUI.

3 How to make and work with models?

Before making model user should have following in the project folder:

- Surface observables (e.g., elevation, gravity, geoid height) in files. Each file separate for these observables should have three columns, first column being distance along the profile sampled according to the resolution you want (e.g., 2 km , 5 km), second column being the value of the observable, and the third being the error in the observable.
- Material files as two digit code name and contains pressure and temperature dependent thermo-physical properties, which are generated using the Generator module. To generate these files follow the README provided in the GENERATOR_Linux folder.

To start the LitMod2D_2.0 go into GUI folder and run main.py (in Linux type " python main.py" in Windows you can double click on the main.py file), running which Fig. 4 will appear. Here you have three options. Build model option is to build a model from scratch. Load Model is to load a previously build model and the last option is about help.

*Note: In dialogue boxes, put your mouse in a field and additional information will appear.

3.1 Build Model

Before building a model user should have a clear idea and sketch of the model user wants to build. User should know nodes along which different bodies will be connected. A model is build from top-to-bottom and left to right and every time user wants to exit and wants to save the model, user should close the model by clicking the close Model button on top right. After clicking close model click on the save option which will open a dialogue box about some info about the model.

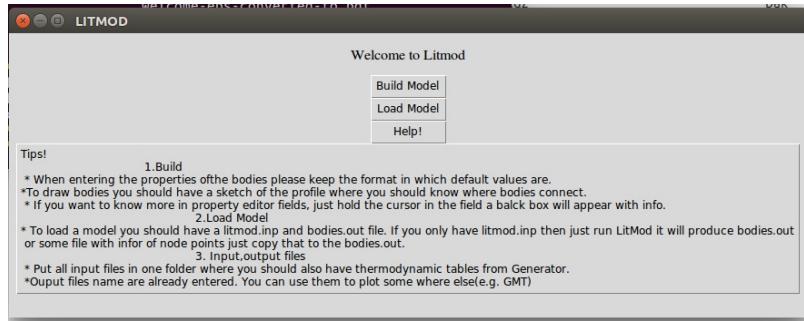


Figure 4: Welcome Page

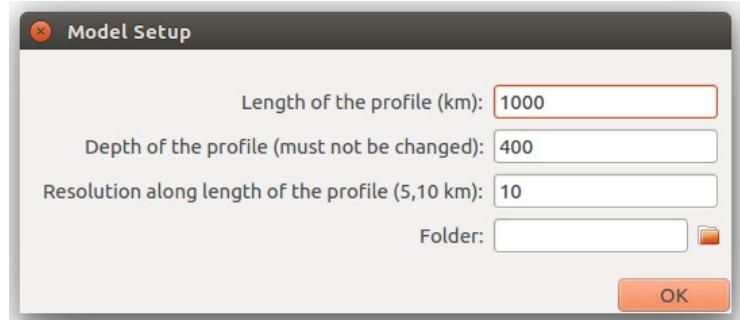


Figure 5: Model Setup

After user hits Build Model option a dialogue box appears (Fig. 5) asking for information about the model and another dialogue box asking for digitized file where you already have nodes of the bodies (e.g. digitized sketch, Moho depths, LAB depths; two column, X (distance along profile, km) Y(depth, -ve, km)). This digitized file will be plotted in background and you can click on the plotted points.

**Note: Bodies are added from left to right

- **Start of the profile (km):** This is the left most start point of profile(it can be negative too. In that case all the observable files should have same limits).
- **End of the profile(km) :** This the right most end point of profile.
- **Depth of the profile:** This the depth of the profile in km. It must be 400km, so it must not be changed.
- **Resolution along the profile:** resolution of the profile
- **Folder :** Here user selects the folder in which observable files are put and it becomes the working folder for LitMod2D_2.0. All the outputs files are

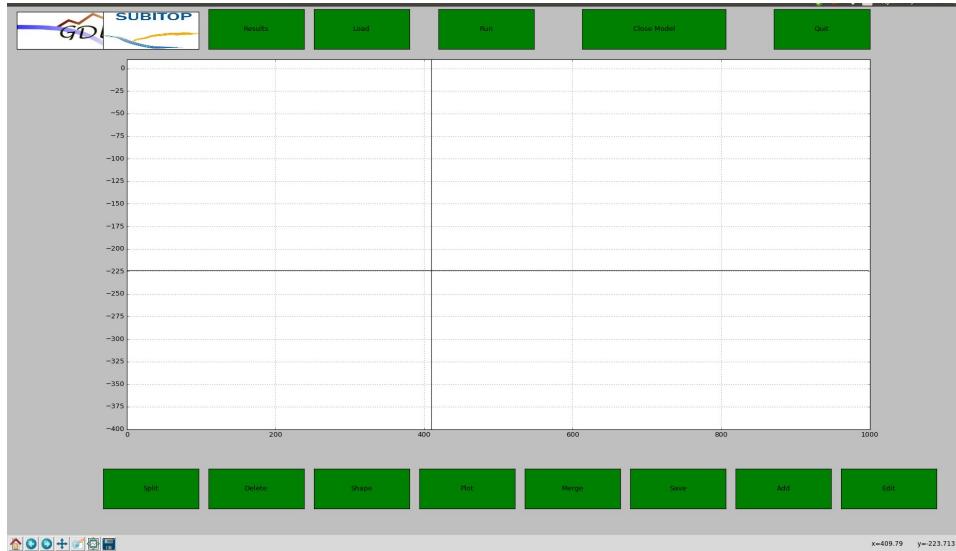


Figure 6: Build Model Window

stored in this folder.

Tip: For each of your model you can make folder where you put all observables files, material files from Generator. Later you can load the model by browsing into this folder.

After the user hits OK button build model window will appear (Fig. 6). Here the user has different option.

3.1.1 Add Body

Bodies are added from top to bottom, each body is drawn left to the right. To add a body press the add button on the window (Fig. 6) which will open a dialogue box asking for information about the body (Fig. 7).

**Please note that format in which default values appear should be maintained.

Fields description

- **Name** : Name of the body. Just for your reference
- **Body number** : Index of the body starting from the top.
- **Material** : Type of material of the body
- **Body type**: if you are adding a body which is new, this option should be normal. If you are splitting a body change this option to split. If you are

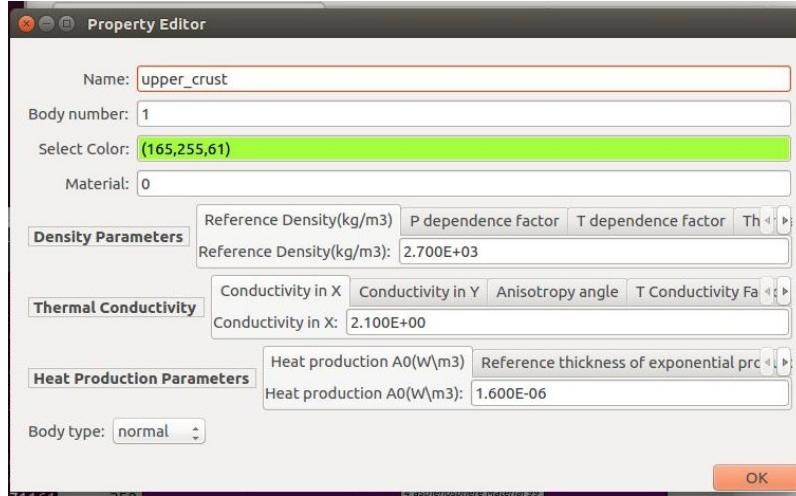


Figure 7: Body property editor

adding an anomaly, change this option to type of anomaly you are adding (thermal or seismic or from a file).

Once the user is done with adding properties user should hit OK and control will be back to the plotting area.

- **to add point:** Middle mouse click (Scroller)
- **to delete current point:** double left click.
- **to close body:** right click

3.1.2 Delete body

The user can delete the last body entered by clicking on the delete button. Let us say a user has drawn three bodies. To delete second body first user has to delete the third body. So essentially you can only delete the last entered body. If you want to delete a body in-between, then you can use the merge function(see following sections).

3.1.3 Change shape of the bodies

To move,delete or add node points of the drawn bodies user can click on shape button. After clicking this button a separate plot will appear where all bodies will be drawn as points (In case you have anomalies in the model other than from a external file then second window also appears where you can edit anomalous bodies too). Now user can move these point by left click drag, to delete a point move the cursor on the point and then press 'd' key on the keyboard (sometimes there are more than one points so keep pressing 'd' until the point

is gone). To add a point go to the point and press 'i' key on the keyboard. After you are done changes can be saved with right click while the cursor is on the edit plot window. Once you have saved changes close the window and come back to the main window and hit plot button which will update the changes. This function will only work after you have closed your model.

**Note: when you see the plot of the bodies (for anomalies) in separate window , you might see some lines connecting different bodies, just ignore them. To have clear idea keep main window where you added bodies in front of you along with this separate window

3.1.4 Edit properties

To edit properties of a body already drawn click on edit button and a dialogue box will appear asking for the index of the body you want to edit.

3.1.5 Split body

This function allows the user to split a body into two (does not work for anomalous bodies). Let us say if you have a crustal body along a body and you want to have different density on the right side than the left side then you would use this option. To split this crustal body click on split body option and enter the asked information about the body and add point near the already existing node in the body and add points along the axis you want to split. The end point should be added close to an already existing point in the body. Once you are done then press right click. The follow the Shape change option instructions, adjust the nodes and refresh back in the main window.

**Note: If for some reasons split does not work then you can load model again and try to split it again. If you get an error message after splitting a body, even though you can see that the split body is plotted in the main window, I highly recommend not to proceed with this split body in the model. On the main screen everything will look normal but internally there will be some inconsistency which can affect splitting, changing shapes later when model becomes complicated.

This problem stems from the fact that some geometric shapes exception handling needs to be improved. I will fix in near future. If you are interest in this and want to provide your support to the community in fixing this, I would really appreciate if you contact me. I can walk you through the GUI code and both of us can benefit from this.

3.1.6 Anomalies in the sublithospheric mantle

In this GUI anomalies (Composition,thermal,seismic) are added on top of the completely closed profile. You can edit the shape of these anomalies and properties (type and amount of anomaly) with shape and edit function respectively.

These types of anomalies can be drawn in the profile.

There is another way to enter anomalies where you enter them in a file and you choose the file (For more information about this file refer to the LitMod2D_2.0_usage.pdf, supplied in this folder. Only seismic anomalies can be added in this way.

** Note that if you have selected anomalies in form of a file then all other anomalies (drawn on the profile) are not considered even if you have added them.

3.1.7 Save

When you are done with the profile and you have closed it by clicking on the Close model button, it can be saved by clicking on save button. After you click on the save button a window will appear asking for some more information.

User can choose observable files, where you should have three columns with distance, data value, error. Total length and sampling of these observables should be exactly same as that in your profile. One important thing is to have starting and end point of these observables data as that of start and end you choose to make your profile. After browsing the these file try to keep only the name of the input file try to delete the absolute path. Here making a folder for each of your model, which can be at any location in your computer, helps keeping things in track.

Every time you save a model, a back-up of three files, i) LitMod2D_2.0.inp, ii) bodies_GUI.dat, and iii) bodies_GUI_envelops.out, with a date and time added. You can later rename these set of files and load them again.

3.1.8 Run model

To run a model first you have to save the model by clicking on the save button, but before that, your model should be closed. You should also put the observables file (topography, Bouguer, geoid, free air, heat flux) and composition files (e.g. 80 81 88 99 etc) which you have associated with the bodies in your model, in the same folder.

A successful run of LitMod2D produces following output files:

```
Model_directory
└── topo_out.dat -- isostatic elevation(m) 3 column: 1-distance(km),
    2-coupled, 3-uncoupled
└── bouguer_out.dat -- bouguer anomaly 2 column: 1-distance(km),
    2- bouguer anomaly (mGal)
```

```

└── geoid_out.dat -- geoid height 3 column: 1-distance(km), 2-2D
    geoid (m), 3-1D geoid (m)
└── FA.out.data -- Free-Air anomaly 2 columns: 1-distance(km), 2-free
    air anomaly (mGal)
└── SHF.out.dat -- Surface heat flow 2 columns:: 1-distance(km),
    2-SHF (mW/m2)
└── post_processing_output.dat -- 9 columns in sequence: distance
    (km), depth (km), temperature (oC), pressure (bar), Vp (km/s), Vs(km/s),
    density (km/m3), material code, index in the material file

```

*Note: To run a model you should have LitMod2D_2.0 program executable for windows or Linux based on your system. Executable for Linux is provided with the distribution . Name of these executables should be LitMod2D_2.0_V4_VS_Windows for windows and should be in LitMod2D_2.0_package folder and for Linux it should be in same folder with name LitMod2D_2.0_V4_VS_Linux.

3.1.9 Load Model

This option lets the user load previously build models. To load models user need three files, 1. LitMod2D_2.0.inp, which is a input file to the LitMod2D_2.0 program and 2. bodies_GUI.out , this file contains nodes point of the bodies in the model and color of the body and bodies_GUI_envelops.out. Units of nodes points in bodies_GUI.dat are in kilometres.

This option also allows you to restore changes while you are working. For instance if something goes wrong (e.g. split body , merge body) you can load last saved session and start from there again. You can also track these three files saved in same folder at any time you have saved them, rename them and can load them again.

4 Post-processing toolbox

Post-processing toolbox contains a set of codes/scripts linking with outputs from LitMod2D_2.0 with other software. At this point is it coupled with "Computer programs in Seismology (CPS)" tool (Herrmann, 2013) and "tao-geo" (Garcia-Castellanos et al., 2002) software. It also includes scripts to produce stable phase and mineral assemblages in the profile. Installations of coupled software is explained below.

If user is using the docker image then all of post-processing tools (e.g., CPS, tao-geo) are already installed and ready to be used. In case user is using the github then she/he might have to installed these tools.

LitMod2D_2.0 produces post_processing_output.dat which can be used to plot the results (see RUN model section). This file is read in the post-processing tool-box.

Before using the post-processing tools, used need to add the path where litmod resides in the computer. To do that user have to run the following command in the terminal:

```
export LitModHOME = path_to_litmod_home (e.g., /home/LitMod2D_2.0_package_users)
```

In case of the docker image path to the litmod home is always /home/LitMod2D_2.0_package_dist_users

All the post-processing is done inside the model directory and you must run all the scripts (explained in the following sections) from there.

4.1 Passive Seismological data

Forward prediction of surface wave dispersion curves and receiver functions can be calculated from the seismic velocities distribution with depth at each node along the profile. This done by feeding in seismic velocities to CPS.

CPS can be easily downloaded and install from <http://www.eas.slu.edu/eqc/eqccps.html> and needs to in your path.

To calculate RF and dispersion curves, we first need to calculate seismic velocities in the crust in the case when no material files are used for them. This is done automatically by running a python script, `crustal_velocities_full.py` using empirical relations and input Vp/Vs ratio, inside the `run_dispersion_RF.sh`. To run this script inside of your model folder type following:

```
$LitModHOME/Post_processing/run_dispersion_RF.sh
```

User has to input some information about the model in the terminal. The result of this script is two directories in the model folder (with example content in them):

```
Model_directory
  └── Receiver_Functions
      └── 350km_vel.dat -- 350km-location of the node; Format of the
          file is CPS velocity file
      └── 350km_vel.dat_2.5.eqr.xyz -- RF at 350km with Gaussian width
          of 2.5 (input parameter); Two column file: 1-time (s), 2-amplitude(scale
          to 1)
  └── Surface_Wave_Dispersion_Curves
      └── 350km_vel.dat -- 350km-location of the node; Format of the
          file is CPS velocity file
```

```
|   └─ 350km_R_C_SURF96.out -- R (Rayleigh-input), C (Phase velocity-input)
      dispersion at 350km; Eight column file with CPS dispersion
      file format, Period (s) in 6th and Phase velocity (in the example)
      in 7th column
```

If user want to change the empirical relation between density and seismic velocities, she/he should change it in this python script. This python script also produces post_processing_output_crust_mantle.dat which includes the seismic velocities in the crust. Internally, this script makes velocity input file at each distance node along the profile to be read in the CPS.

4.2 Flexural Isostasy

Flexural isostasy is incorporated via “tao-geo” and can be downloaded and installed from <https://github.com/danigeos/tao-geo> and should be added in your path.

To include flexure in elevation calculation user should run Pressure_Flexure.sh.job in the model directory. To run this script inside of your model folder type following:

```
$LitModHOME/Post_processing/flexure_tao/Pressure_Flexure.sh.job
```

User has to input some information about the model in the terminal. Te (effective elastic thickness) can be changes inside the Pressure_Flexure.sh.job script. The result of this script is two files in the model directory:

```
Model_directory
└─ Deflexiotopotao_TeA.dat -- Three column file: 1-distance(km),
  2-deflection, 3-elevation (m); Te= 10km; Can be changed inside
  the script
└─ Deflexiotopotao_TeB.dat -- Three column file: 1-distance(km),
  2-deflection, 3-elevation (m); Te= 20km; Can be changed inside
  the script
```

4.3 Stable phase and mineral assemblages

To do this user should have full property tables from the GENERATOR module. Opting for option ”1” shown in 8 produces both full property material file along with simple material file with physical properties only. Full property material file contains information on stable phase and mineral assemblages (weight percentage, volume percentage) along with the physical properties (density, seismic velocities). Simple material files, with material code ’90,99..’ are read in the LitMod2D_2.0 whereas full property material files named as ’99_FULL etc.’ are used to produce stable phase and mineral assemblages. Full property material files should be in the model directory.

```

5
Your option is      5
Will this be a sublithospheric mantle?
YES = 1           NO = 0
0
Do you have H2O and K2O in your system?
YES = 1           NO = 0
0
Now enter the bulk composition (in wt%) for the oxides
SiO2 Al2O3 FeO MgO CaO Na2O (in this order,
                                         one per line)
45
4
8
34
3
0.2
*****
Note that these values do not add up to 100%
The absolute error (in wt%) is = 5.799999999999972
If you do not stop here, I will correct to 100%
*****
Name of the body (mantle/crust)
rrrr
Do you need the FULL (prop. + system) table?
For "Yes" type 1 ; For "No" type 0

```

Figure 8: Generator console showing option to generate full property table.

Put the full material files in the model directory (e.g., 99_FULL). Then run "make_mineral_wise_files_full.sh". This produces mineral wise properties along the profile. To run this script do following from the model directory:

```
$LitModHOME/Post_processing/Phase_diagrams/make_mineral_wise_files.sh
```

You will have enter few input parameters when asked in the terminal. The result of the above script is a directory called Phasein the model directory and contents are explained below:

```

Model_directory
└── Phase_diagrams
    └── 99_profile-- this file includes all the minerals physical properties
        and amounts at the P-T prevailing in the sublithospheric mantle
        of the model, because 99 is the code for the sublithospheric
        mantle material file

```

To plot the depth and along profile distribution of stable mineral assemblages user must go to the Phase_diagrams directory and then run the python script there. User can plot a property (e.g. weight%) of stable mineral at a distance point along the profile running 'phase_diagram_1D.py' 9 or depth distribution of individual minerals along the profile using 'phase_diagram_2D.py' 10. To plot, e.g. along profile distribution of properties user should run the following command in the Phase_diagrams directory:

```
python $LitModHOME/Post_processing/Phase_diagrams/phase_diagram_2D.py
```

5 Miscellaneous

User can also use other functionality. They are listed below:

1. Values of coordinates are shown at the bottom right corner of the plot.

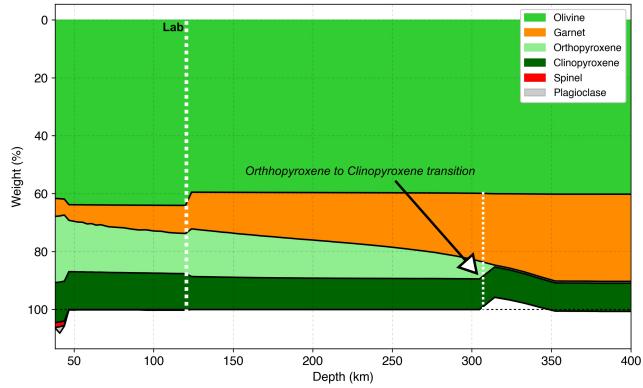


Figure 9: Example of stable mineral wt% distribution at a distance point.

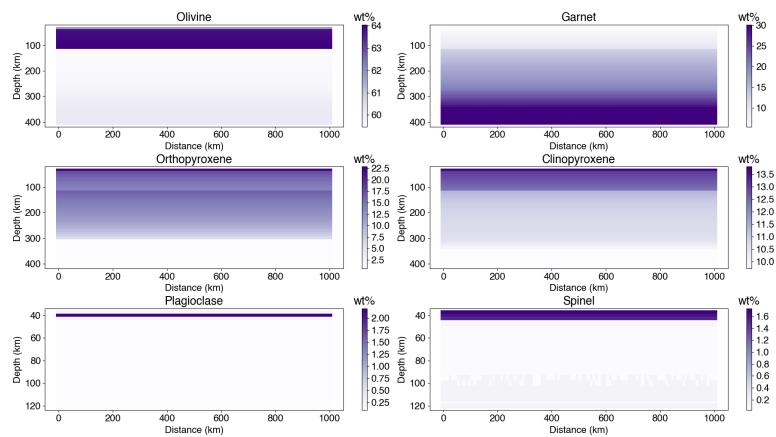


Figure 10: Example of stable mineral wt% distribution along a profile.

It is useful to add points at specific positions.

2. You can zoom in at any area of the profile by clicking at the functions in the left bottom of the window. You can also go to the zoom mode by pressing 'o' on the keyboard. Then you can select area to zoom in with mouse and navigate back and forth with arrow keys on the keyboard. It is useful for small area bodies.
3. To go back out of zoom mode you should press 'o' again.
4. You can drag the profile and to go to this mode you can press 'p' on the keyboard.
5. If you want to go straight to the initial level after zooming at different levels just press 'h' key on the keyboard.