ESA Sysnova System Studies - Lunar Caves

Documentation Database IGMAS modelling

MatLab Data visualisation app

Author
Frank de Veld

June 17, 2020

Use of database

If one wants to use the database, one must have access to the following files:

- 'Data_visualisation_app.mlapp': an executable file which is the main file for visualisation
- 'Retrieve_coordinates.m': a function for delivering the coordinates to the app described above
- All the result data files either ending on 'X.mat', 'Y.mat' or 'Z.mat'. The total number of data files should be equal to the product of the number of ticks and boxes (but this is a technicality; if things work properly, you already have all data files, or you generate them yourself)

If all these files are downloaded, one can simply run the app 'Data_visualisation_app.mlapp' and adjust the sliders and the check-boxes as desired. As of now, there are three parameters to vary (cavity depth, density difference and grid spacing), and four lava tube models to choose from. Each combination of parameters leads to a contour plot of the g_z gravity signal, with annotations.

Creation of database

If the database still needs to be created, the following files are needed:

- 'Voxel_main.m': the main script for preparation for IGMAS importing
- 'stlread.m': a function needed to import 3D model files in MatLab
- 'Voxel_creation.m: a function which transforms cavity coordinates to a filled voxel model
- 'Write_basic_model_file.m': a function to create an IGMAS .model file
- 'Export_stations_file.m': a function to export an IGMAS .stations file
- 'xml2struct.m': a function to efficiently transform IGMAS result .station files to MatLab structures
- 'Data_export.m': a function to transform IGMAS result .stations files to X, Y and Z coordinates for contour plots

In addition to this, one should have access to the graphical computer system IGMAS for gravity modelling.

First, one should choose the parameter space to model in 'Voxel_main.m' in the lava tube model array, the grid spacing array, the model density array and the cavity depth array. Note that the total number of configurations to be modelled is the product of the lengths of all arrays; this quickly adds up. As all of this needs to be done manually, and can be a laborious task. In this same file, one can change the station resolution. It is advised to use the same resolution throughout the whole modelling process. All of this can be done at lines 35 to 39.

After running this script (taking 1-30 seconds, depending mostly on the grid spacing), one ends up with a .model file, a .vxo file and a .stations file. The names of the files begins with something like:

'Model_name'_GS_'Grid_Spacing'_MD_'Model_Density'_CD_'Cavity_Depth'

All of these are used in IGMAS. Now transfer the files to the computer which has IGMAS installed, and press 'File -> Import -> Model', 'File -> Import -> VoxelCube' and 'File -> Import -> Stations' to import all three files. After this, press 'Tools -> Calculate Anomalies' and check 'gz'. This calculation takes about one to twenty seconds. When this has been completed, press 'File -> Export -> Stations' with the same file name as the imported files, but with the suffix '_results.stations'. Transfer this file to the Results folder.

Then, the script 'Data_export.m' can be used. It is automated to work on all files of one lava tube model at the same time, but can be adapted to only work on a selection on it for verification. Make sure the arrays on lines 12 to 15 are the same as in 'Voxel_main.m', as well as the station resolution.

When this has all been done, the app "Data_visualisation_app.mlapp' can be used as desired.

Due to the structure of file importing, it is advised to download the whole folder 'Database' from GitHub for use.

Improvements for later use

- To look into the 'best' way of determining at what levels contours should be plotted, and on what this should depend
- To also make it possible using which stations the contours should be created, and what kind of patterns work the best, with some kind of user interface. This would likely be a separate applet