Compaction Model Documentation

April 11, 2022

Abstract

Brief documentation to accompany the 1D compaction code used for the publication of *Modeling 65 Years of Land Subsidence in California's San Joaquin Valley* For additional questions or errors, please contact M Lees at mlees@stanford.edu or mlees0209@gmail.com.

Contents

1 Installation and execution

The model runs on Python and, optionally, GMT. The code is hosted on Github. To install, you will use the command:

```
git clone https://github.com/mlees0209/compaction_model_StaticPaperVersion
```

To execute the code, you must create a parameter file and use the command:

```
python execute_model.py PARAMFILE.par
```

For examples of how to create a parameter file, see *EXAMPLE_paramfile.par*, where all the options are described.

For the run to be successful, you need Python 3 and the following Python packages to be installed:

- matplotlib
- datetime
- csv
- subprocess
- scipy
- operator
- netcdf4
- seaborn

Additionally, output can be saved as the highly efficient netCDF format if GMT is installed; install instructions can be found at the GMT website. If GMT is not/cannot be installed, there is an option to turn GMT off in "execute_model.py".

Disclaimer: following these steps allowed me to successfully run and execute the script using a conda environment with the above listed packages installed in addition to GMT, on a Mac OS operating system and Ubuntu Linux system. Functioning Conda environments are provided on Github for both operating systems (files with .yml suffix). Please contact me at *mlees@stanford.edu* if you need help.

2 Fundamental Equations and Math

The model solves two governing equations: the evolution of effective stress, and the resulting compaction. In both cases, head is converted to effective stress using $\Delta\sigma' = \Delta\sigma - \rho g \Delta h$, as described in Methods 2 of the paper *Modeling 65 Years of Land Subsidence in California's San Joaquin Valley.*

2.1 Diffusion Equation in Clays

The evolution of effective stress is governed by Equation 1. This equation is a form of the diffusion equation encountered in many physical situations. However, in this context, the specific storage value is a function of stress history, as described in the Methods section of the paper *Modeling 65 Years of Land Subsidence in California's San Joaquin Valley.* The model solves this equation by a finite difference method in all clay layers (interbeds and aquitards). The solver is found in the function called *solve_head_equation_elasticinelastic* defined in *model_functions.py.*

$$\frac{K_v}{S_s}\frac{\partial^2 \sigma'}{\partial z^2} = \frac{\partial \sigma'}{\partial t} \tag{1}$$

In Equation 1, K_v is the vertical hydraulic conductivity, S_s is the specific storage, σ' is effective stress, z is the spatial dimension, and t is time.

2.1.1 Convergence Criteria

The converge of finite difference methods to solve Equation 1 does not seem to have been extensively studied where there is spatially and temporally varying diffusivity. However, one literature source showed the condition in Equation 2 to be a sufficient one for convergence ([1]). If this condition is violated, the model exits with an error; although the condition is sufficient, rather than necessary, we were unable to find a necessary condition for convergence so we use this stricter sufficient condition. The user is not able to use coarser resolution than provided by this criteria at present, although such a flag may be added in future.

$$\frac{\Delta t}{\Delta x^2} < \frac{1}{2max(\frac{K_v}{S_s})} \tag{2}$$

In Equation 2, Δt is the timestep and Δx is the discretisation in the spatial dimension.

2.2 Compaction Equation

The governing equation for the resulting compaction is given in Equation 3. This is computed directly by the model, using the midpoints of the nodes used for solution of Equation 1. The effective stress values at midpoints are calculated by linear interpolation. For a description of the terms in Equation 3, see the text preceding and following Equation M3 in the paper *Modeling 65 Years of Land Subsidence in California's San Joaquin Valley.*

$$b(t_i) = \frac{\Delta z}{\rho g}\Big[S_{skv}\sum_{j=1}^{J-1}\max_{n\le i}(\sigma_j'^n) - \sigma_j'^0) - S_{ske}\sum_{j=1}^{J-1}\max_{n\le i}(\sigma_j'^n) - \sigma_j'^i)\Big] \tag{3}$$

In the coarse-dominated parts of the aquifers (sands and gravels), effective stress history is calculated from head, and the compaction is then calculated using Equation 3 with $S_{skv} = 0$ and an elastic value of S_{ske} for coarse-dominated material. The code uses a single 'node' of thickness $/Deltaz$ equal to the aquifer layer thickness minus the total thickness of clay interbeds within the aquifer.

2.3 Specific Storage

Skeletal specific storage, and therefore specific storage, takes different values depending on stress history if solver mode is 'elastic-inelastic'. This is governed by Equation M2 in the paper *Modeling 65 Years of Land Subsidence in California's San Joaquin Valley.* The model code keeps track of effective stress history at each timestep and sets the skeletal specific storage at the next timestep accordingly.

3 Input Head Timeseries

The ultimate objective is that the model will be able to read in head timeseries files in a flexible variety of fileformats. At present, the file format must be two columns of comma-delimited items, of format:

DATE,MEASUREMENT

DATE should be a string of any format and MEASUREMENT should be a number given to any number of significant figures, in the units of metres. We rely on Python's pandas library to automatically read the dates in any format; if this is not working, see the last paragraph of this subsection for a potential manual solution.

Internally, the model uses Python's Datetime module, and dates are stored as numerically as a number of days since 0000-12-31 using the datetime.date2num method. See https://www.w3schools.com/python/python_datetime.asp and https://matplotlib.org/3.1.1/api/dates_api.html#matplotlib.dates.date2num.

There are some potential issues with future head projections beyond the year 2069, as for certain situations Datetime seems to interpret XX69 as 1969 no matter what XX is. If this occurs, you may need to go into the code and manually specify a date format or parser.

4 Code benchmarking and validation

The code was validated against an analytic solution for a step head drop. Additionally, the code was benchmarked against an independent code for the scenario of a more complicated head time series. Please contact the code author (mlees@stanford.edu) for more information on the benchmarking process.

References

[1] Jordan Lee. "Stability of Finite Difference Schemes on the Diffusion Equation with Discontinuous Coefficients". en. In: (2017), p. 21.