Finite Element Modelling for Geosciences: Day 1 Tips

Patrick Sanan

ETH Zurich

ETH zürich

Tips as you start to code

- This is a "Hands-on" course. Long experience has shown that the only way to really learn the FEM is to write codes for yourself.
- Making mistakes and resolving them is part of the process.
- This document gives some quick tips to make this process a little less painful!
- Don't forget that the instructors are here for you.

General Programming: Start "top down"

- Set up the framework, without the "guts"
  - stubs
  - TODOs
- Make sure you can always run your script to completion
- Generate a plot/image (even if it's wrong)
- **test often**

General Programming: Work methodically

- Save working states.
- Change one thing at a time.
- Keep your code readable
- **Test** as you go.

A version control system like Git can be very helpful.

General Programming: Expect to be wrong

- You *WILL* make errors.
- Try to ensure that your errors will be obvious.
- know what you want the answer to look like, before running the code
- Work out small, numerical tests on paper and check them against your code
- **Test often**

Finite Element Programming: Remember the Procedure

Complex FEM codes have a similar form to simple ones (Section 3.2).

1. Define problem parameters
2. Define mesh
3. Define time domain
4. Determine numberings
   - element node numbers $\rightarrow$ point number (and type)
   - boundary node numbers
   - point number (and type) to equation number
5. Initialize global matrices
6. Time Loop
   - Element Loop
     - Compute local matrices and vectors
     - Add to global matrices and vectors
   - Apply boundary conditions
   - Solve system

Finite Element Programming: Challenge your code

- Think: how would you demonstrate that "the code works"?
- Have an exact solution to compare against, when possible
- Break symmetries to test:
  - Different numbers of elements in each direction
  - Irregular grids

MATLAB Tips

- The help system is very useful (e.g. `help magic`)
- Start your script with `clear; close all; clc;`
- Use the debugger