Exercise 2: Modelling fluid & heat flow in Python

M.Geo.239
8 nov 2019

Elco Luijendijk, eluijen@gwdg.de

What is Python?

- Python is a programming language
- Designed late 1980’s by Guido van Rossum at mathematical research institute in the Netherlands. Later development at NIST, Google
- Designed to be easy and readable, to allow quick development of useful code
- Very widely adopted in various applications, web development, software development
- Probably the most popular programming language in science

Programming languages

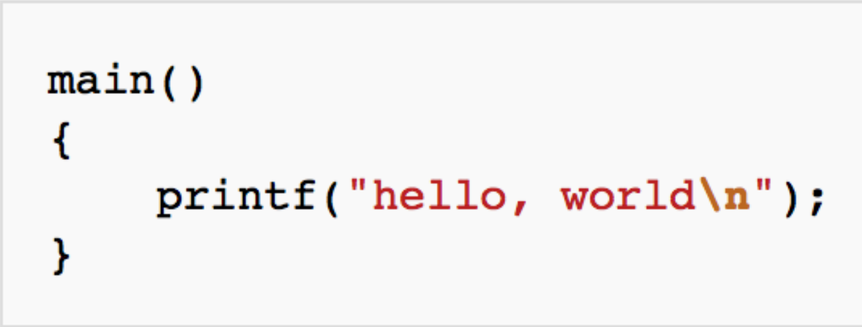
- Low level programming languages: C, C++, Fortran
- Around since the ~1960s-1970s
- Syntax very close to the internal language of computers
- But quite far from human language…
- Programs tend to run very fast, but are relatively difficult to design and maintain
- Used for heavy computational tasks: complex numerical models: coupled surface or subsurface flow, landscape evolution, seismic models, climate models, etc…

- High level programming languages: R, Python, Ruby
- Developed later, 1980s-2000s
- More easily readable syntax, relatively close to the language we, humans, use
- Code runs relatively slow
- But code is much easier to design and to maintain
- Used for lighter every day research tasks and first versions of more complex models

Syntax examples

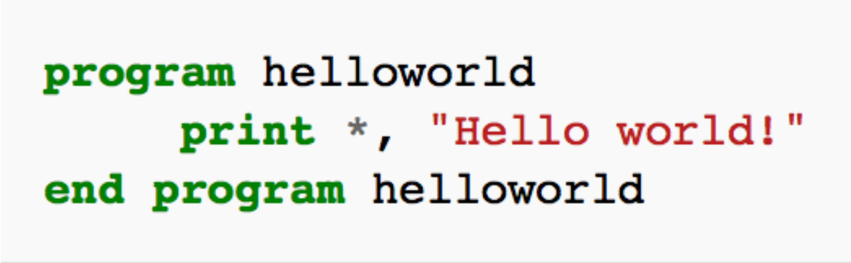
- Hello world example in C++, Fortran and Python:

C



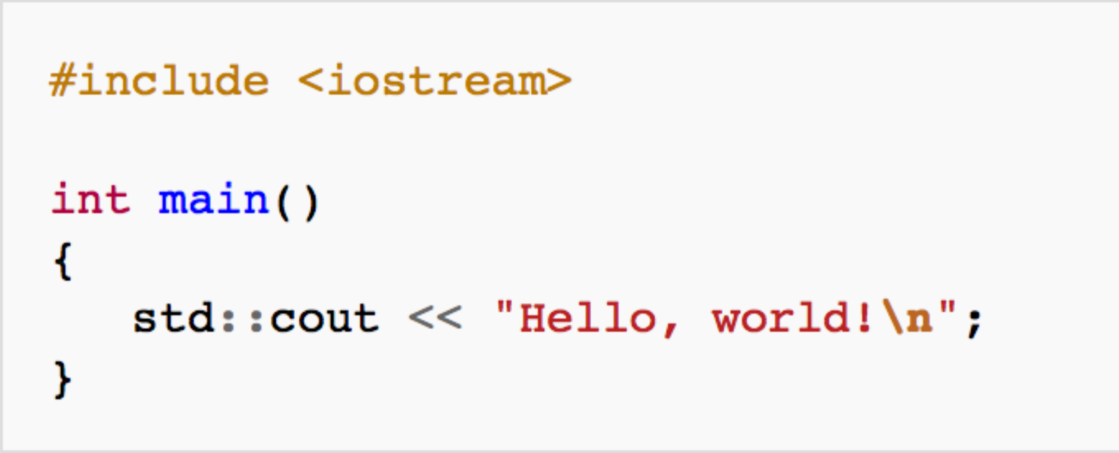
```c
main()
{
    printf("hello, world\n");
}
```

Fortran



```fortran
program helloworld
     print *, "Hello world!"
end program helloworld
```

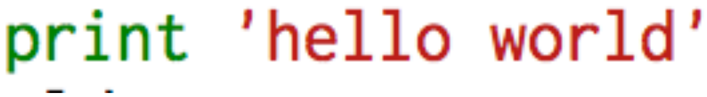
C++



```cpp
#include <iostream>

int main()
{
   std::cout << "Hello, world!\n";
}
```

Python



```python
print 'hello world'
```

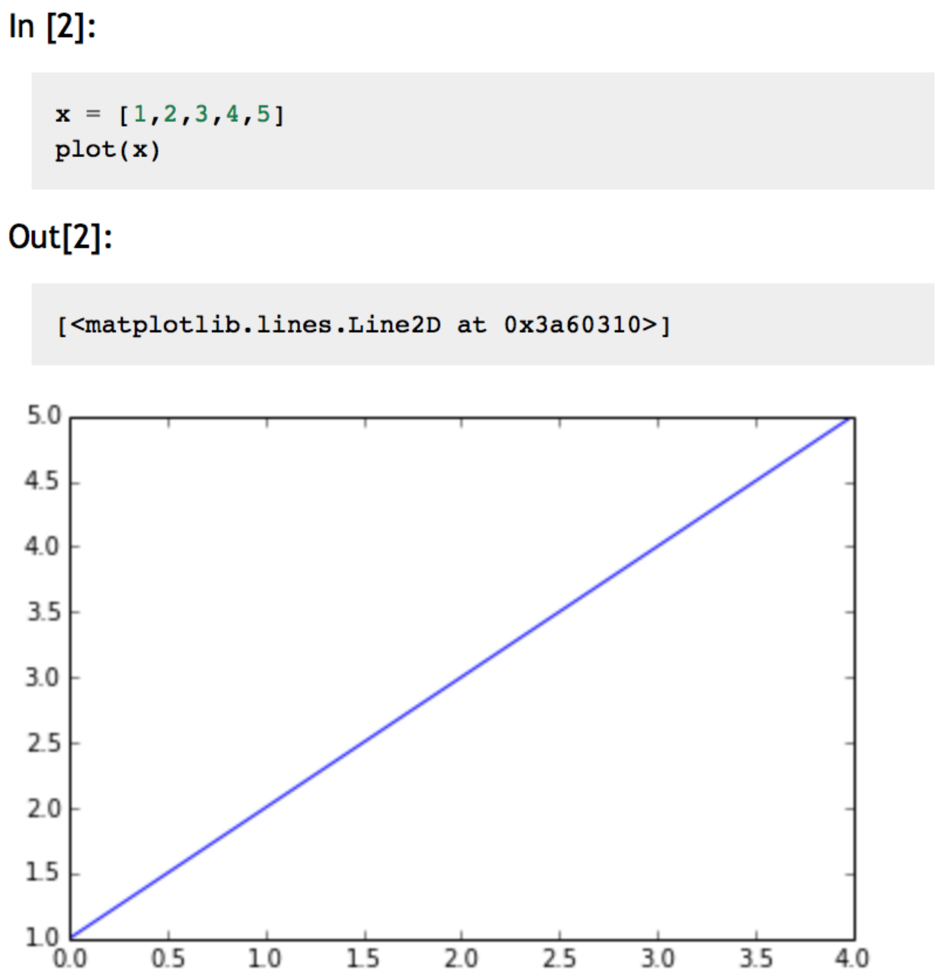
Advantages of Python

- Python is a scripting language: no need to compile code before running it.

- You can run Python interactively, type in a command and see the result straight away

- Python is free and open source, and (in my experience) completely bug-free (in contrast to Matlab)

- Very active development of many modules for scientific applications



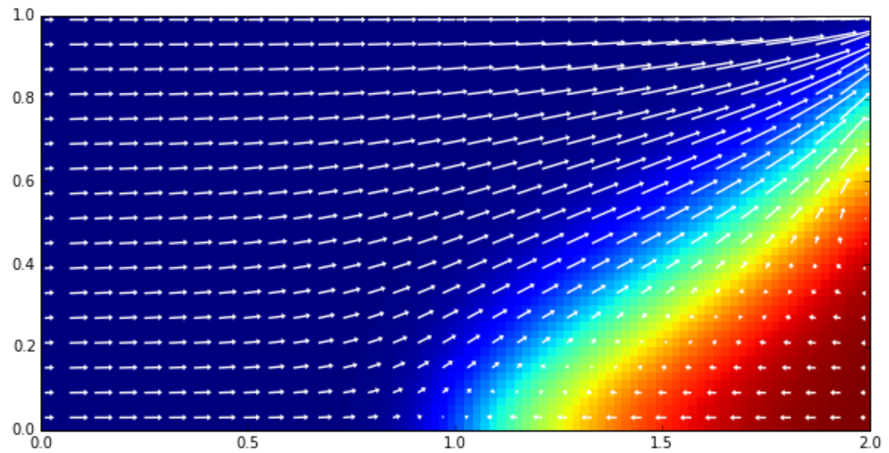
In [2]:

```
x = [1,2,3,4,5]
plot(x)
```

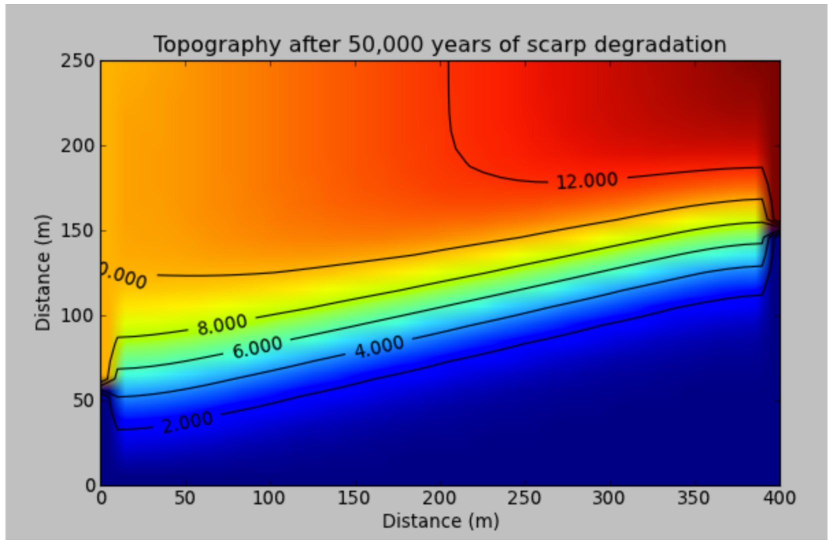
Out[2]:

```
[<matplotlib.lines.Line2D at 0x3a60310>]
```

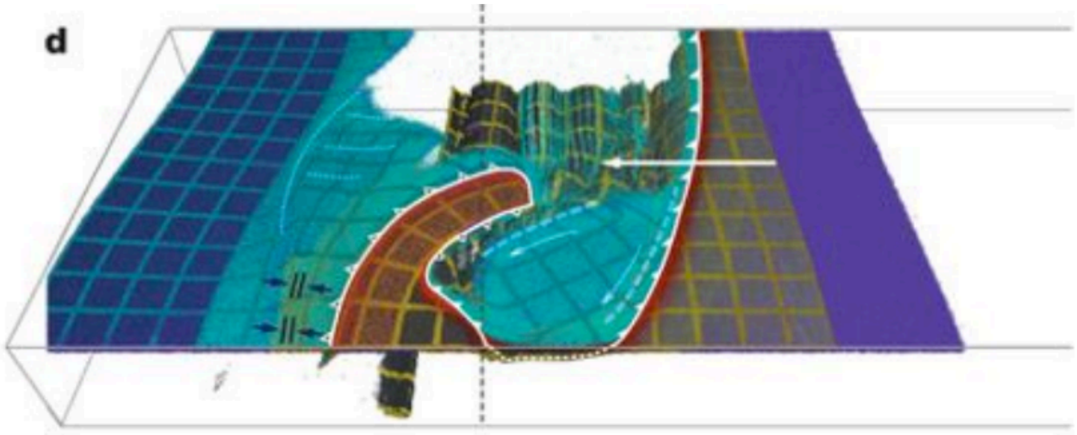
Some examples of Python in earth sciences



FloPy: Python interface to the widely used groundwater model Modflow
https://github.com/modflowpy/flopy

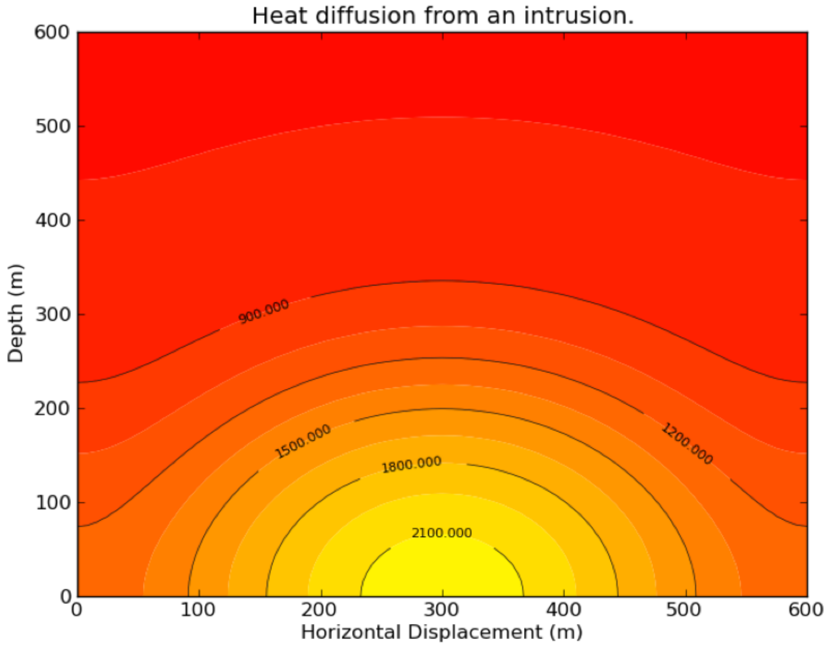


Landlab: landscape evolution
https://landlab.github.io/#/

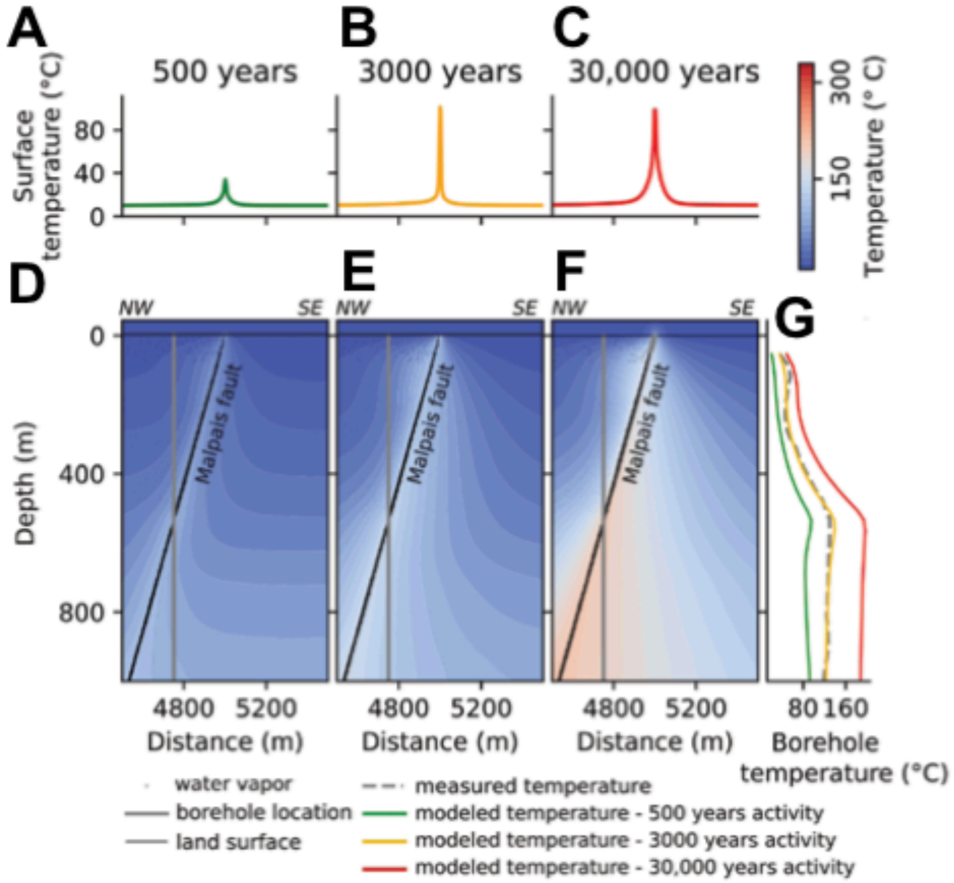


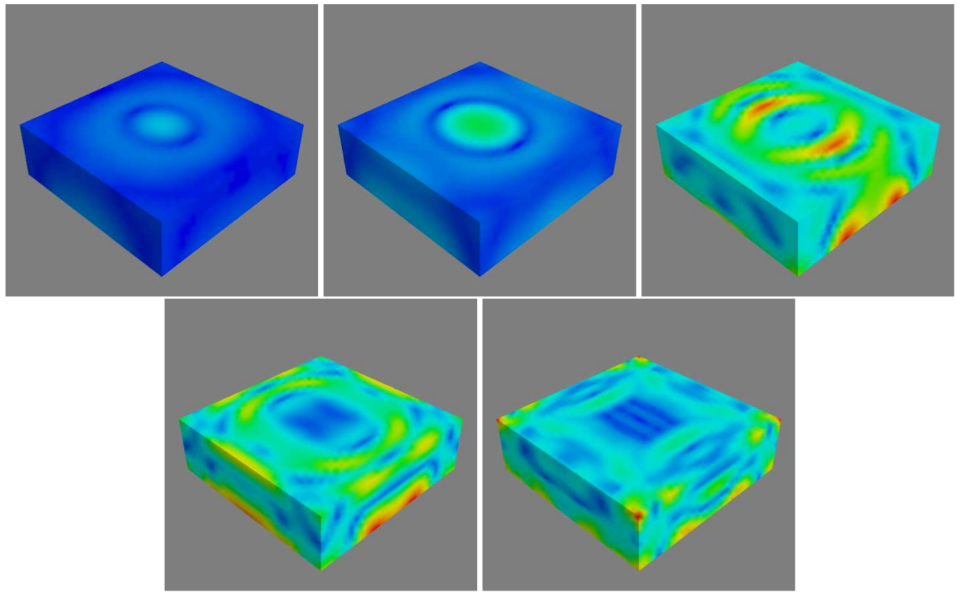
Underworld: Geodynamics
https://www.underworldcode.org

Python & the generic numerical model escript



Heat conduction





Temperatures in a hydrothermal system (Louis et al. 2019)

Propagation of seismic waves

Python in this course

- Remaining exercise in this course will use Python
  - Exercise 2: Steady-state and transient fluid and heat flow using the same finite difference solution as in exercise 1
  - Exercise 3: 2D models of fluid and heat flow using the external numerical module escript (https://launchpad.net/escript-finley)

Jupyter notebooks

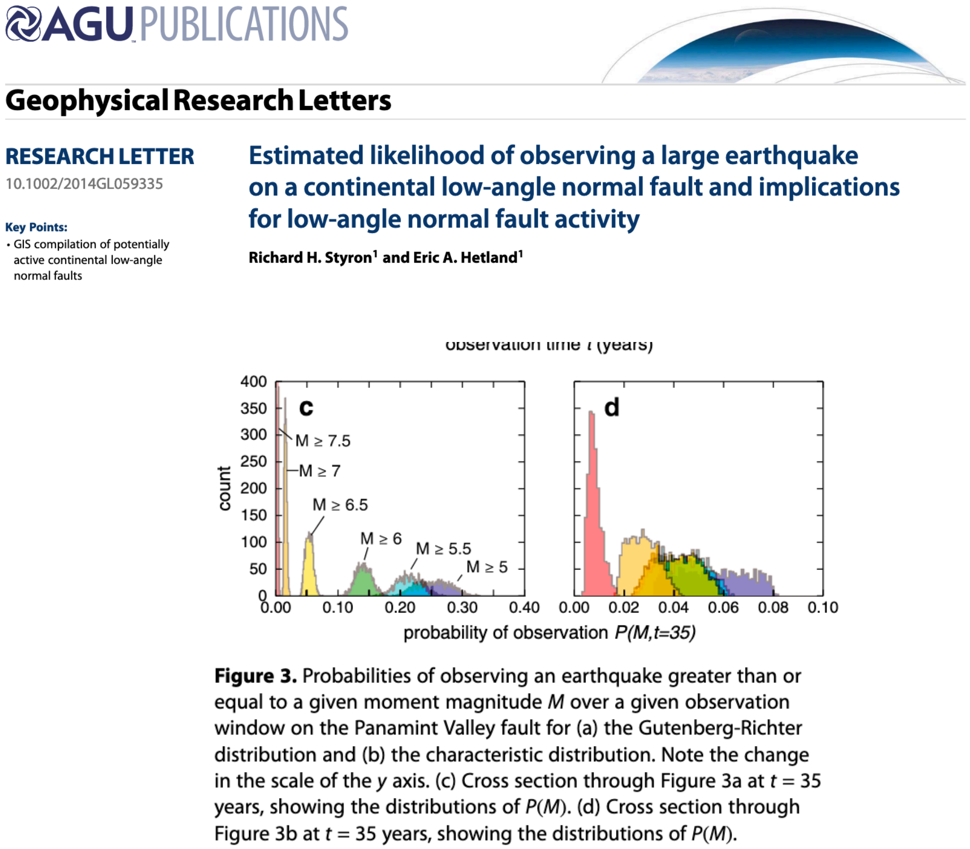
- New this year: We will use Jupyter notebooks. Notebooks contain a mix of text, code and figures and will probably become the new standard for sharing data & analysis

- Some scientific papers have started to supply notebooks as supplementary information. Advantage: you can reproduce the analysis and redo the figures yourself -> truly reproducible science

- See for instance this Nature article: “Why Jupyter is data scientists’ computational notebook of choice”, https://www.nature.com/articles/d41586-018-07196-1

Jupyter notebooks

- See for instance this example this paper:
- https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014GL059335
- and associated notebook:
- http://nbviewer.jupyter.org/github/cossatot/lanf_earthquake_likelihood/blob/master/notebooks/lanf_manuscript_notebook.ipynb



AGU PUBLICATIONS

Geophysical Research Letters

RESEARCH LETTER

10.1002/2014GL059335

Key Points:

- GIS compilation of potentially active continental low-angle normal faults

Estimated likelihood of observing a large earthquake on a continental low-angle normal fault and implications for low-angle normal fault activity

Richard H. Styron[1] and Eric A. Hetland[1]

Figure 3. Probabilities of observing an earthquake greater than or equal to a given moment magnitude *M* over a given observation window on the Panamint Valley fault for (a) the Gutenberg-Richter distribution and (b) the characteristic distribution. Note the change in the scale of the *y* axis. (c) Cross section through Figure 3a at *t* = 35 years, showing the distributions of *P*(*M*). (d) Cross section through Figure 3b at *t* = 35 years, showing the distributions of *P*(*M*).

Jupyter notebooks

- We will first go through a python tutorial notebook: A Crash Course in Python for Scientists

- Notebook can be found here: http://nbviewer.jupyter.org/gist/rpmuller/5920182

- Work through sections I Python Overview and II Numpy and Scipy, up to the harmonic oscillator part

- The notebook is located on stud.ip and can be run using the software package Canopy that is installed on your computer