About Numerical Modeling and Programming

Excerpts from Taras Gerya's Textbook on Numerical Geodynamic Modeling

Stefan Hergarten

Institut für Geo- und Umweltnaturwissenschaften
Albert-Ludwigs-Universität Freiburg

Excerpts from Taras Gerya's Textbook on Modeling

Geologists in Numerical Modeling

Before starting with numerical modelling we should consider one of the very popular 'myths' among geologists, who often declare (or think) something like:

Numerical modelling is very complicated; it is too difficult for people with a traditional geological background and should be performed by mathematicians.

I used to think like that before I started. I always remember my feeling when I heard for the first time the expression, 'Navier–Stokes equation'. 'Ok, forget it! This is hopeless,' I thought at that time, and that was wrong.

Excerpts from Taras Gerya's Textbook on Modeling

The Seven Golden Rules of Numerical Modeling

Golden Rule 1. Numerical modelling is simple and is based on simple mathematics.

All you need to know is:

- linear algebra,
- derivatives.

Most of the 'complicated' mathematical knowledge is learned in school before we even start to study at university! I often say to my students that all is needed is:

- strong MOTIVATION,
- usual MATHS,
- clear EXPLANATIONS,
- regular EXERCISES.

Motivation is most important, indeed...

Excerpts from Taras Gerya's Textbook on Modeling

The Seven Golden Rules of Numerical Modeling

Golden Rule 2. When numerical modelling looks complicated see Rule 1.

Golden Rule 3. Numerical modelling consists of solving partial differential equations (PDEs).
There are only a few equations to learn. They are generally not complicated, but it is essential to learn and understand them gradually and properly.

Golden Rule 4. Read books on numerical methods several times.
There are many excellent books on numerical methods. Most of these books are, however, written for physicists and engineers and need effort to be 'digested' by people with a traditional geological background.

Excerpts from Taras Gerya's Textbook on Modeling

The Seven Golden Rules of Numerical Modeling

Golden Rule 5. Repeat transformations of equations involved in numerical modelling.
These transformations are generally standard and trivial, but repeating them develops a familiarity with the PDEs (maybe you will even start to like them...), and allows you to understand the structure of the different PDEs.

Golden Rule 6. Visualisation is important!
Without proper visualisation of results, almost nothing can be done with numerical modelling. Modellers often spend more time on visualisation than on computing and programming.

Golden Rule 7. Ask!
This is the most efficient way of learning.

Excerpts from Taras Gerya's Textbook on Modeling

The Nine Programming Rules (Bug Rules)

Bug Rule 1: Think before programming!
Think carefully about the algorithm of your new code and the most efficient way of making modifications to your old code – you will then develop the program faster and more efficiently and will not need too much code re-thinking and re-writing.

Bug Rule 2: Comment!
Making comments in the code is essential to enable the code to be used, debugged and modified correctly. The ratio between comment lines and program lines in a good numerical code is larger then 1:1. Do not be lazy, explain every program line – this will save you a lot of time afterwards!

Bug Rule 3: Programming makes bugs!
We always introduce bugs (i. e. programming errors) while writing a code. We typically introduce at least one bug when we modify one single line and we have to test the modified code until we find the bug!

The Nine Programming Rules (Bug Rules)

Bug Rule 4: Programming means debugging!
Be prepared that only 1 % of the time will be spent on programming and 99 % of your time will be for debugging.

Bug Rule 5: Most difficult bugs are trivial ones!
There are three types of the most common bugs:

- errors in index (90 % of your bugs!), e. g. `y = x(i,j) + z(12)` instead of `y = x(j,i) - z(2)`
- errors in sign, e.g. `y = x + z` instead of `y = x - z` or `y = 1e-19` instead of `y = 1e+19`
- errors in order of magnitude, e.g. `y = 0.0831` instead of `y = 0.00831`.

Don't be surprised that finding these 'trivial' bugs will sometimes be very difficult (we simply don't see them) and will take a lot of time – this is normal.

Excerpts from Taras Gerya's Textbook on Modeling

The Nine Programming Rules (Bug Rules)

Bug Rule 6: If you see something strange – there is a bug!
Be suspicious, do not ignore even small strange things and discrepancies that you see when computing with your code – in 100 % of cases you will find that a bug is the cause. Never try to convince yourself (although this is what we typically tend to do) that a single last digit discrepancy in results with the previous version of the code is due to computer accuracy – it is due to either old or new bugs!

Bug Rule 7: Single bug can ruin 10 000-lines code!
We should really be motivated to carefully debug and test codes. Don't think that one single small error in the code can be ignored – it will spoil results of months of calculations.

The Nine Programming Rules (Bug Rules)

Bug Rule 8: Wrong model looks beautiful and realistic!
Often erroneous models do not look bad or strange and some of them are really beautiful. Therefore, be prepared that of the numerical modelling results you like, some are actually wrong...

Bug Rule 9: Creating a good, correct and nicely working code is possible!
This is what should motivate us to follow the eight previous rules!