

Computational Geodynamics Projects

C. Thieulot

June 15, 2017

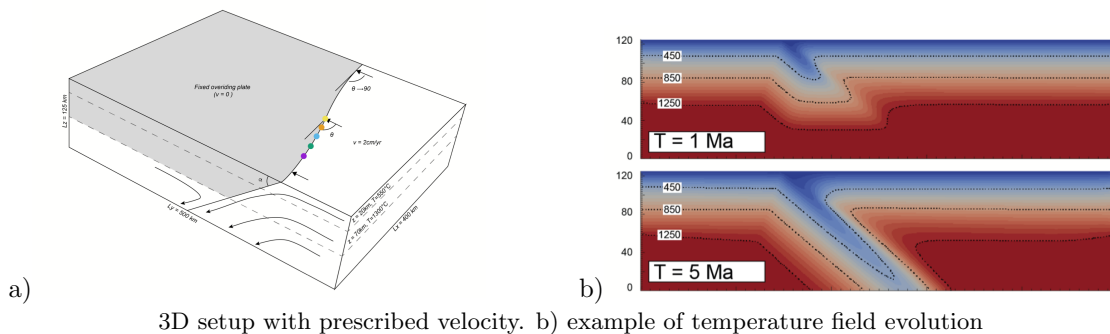
The number of stars next to each project indicates its (estimated) level of difficulty. You can score very high marks with any of the 4 projects: simply choose the one that is in adequation with your motivation and how advanced of a programmer you think you are.

Please let me know by email which project you have chosen.

Corner flow subduction (*)

- In this experiment the velocity is prescribed in geometrically simple subducting and overriding plates, while the velocity in the mantle is computed by means of an analytical solution coined 'corner flow velocity'. Details are to be found in G.K. Batchelor, *An introduction to fluid dynamics*. Cambridge University Press, 1967.
- Write a subroutine which prescribes the velocity in a lithospheric sized domain
- Use this velocity to drive the system in time (choose the appropriate values for the coefficients in the heat transport equation)
- prescribe a constant temperature value at the top, and fix the temperature on both sides, but only in the plates (along lengths l_1 and l_2 . Choose an appropriate plate temperature model.
- Run the model over millions of years with different velocities.
- Measure the depth of the isotherm 800° as a function of time.
- Is steady state ever reached ?

If all goes well, you should be able to recover similar results:



2D Stokes flow and iterative solvers (**)

- Download, install and study the SimpleFEM code from www.cedricthieulot.net/simplefem.html
- The benchmark present in the downloaded 2D version will not be used here. You will therefore model the following setup: the box is a cuboid of size $L_x = 500km$, $L_y = 500km$. Either no-slip or free slip boundary conditions are to be prescribed. It contains a fluid M (mantle) with density $\rho_M = 3300kg/m^3$ and viscosity $\mu_M = 10^{22}Pa.s$ which fills the domain. Additionally, you will implement a sphere (S) of higher density $\rho_S = 3350kg/m^3$ and with a viscosity $\mu_S \in [10^{22-26}]Pa.s$ placed at the center $C(x_c, y_c)$ of the domain. Its radius is $100km$. The functions b_1 and b_2 should not be used anymore and the two commented lines above where b_1 and b_2 are called should be used instead. In order to implement the density and viscosity for the mantle and the sphere, you will create two new functions `my_mu` and `my_rho` which take the coordinates of a point as argument and take the corresponding value at that location. Set the penalty parameter to $\lambda = 10^{28}Pa.s$ and do not change it.

- Document how the maximum (downward) velocity in the domain varies with a) resolution, b) viscosity contrast ($\mu^* = \mu_S/\mu_M$), type of boundary conditions.
- Use the Jacobi Solver that you wrote to solve the system. What do you observe ?
- Implement a Gauss-Seidel iterative solver and test it first on a simple system of which the solution is known (think of one of the FD or FE codes you wrote in class). Use the algorithm presented in https://en.wikipedia.org/wiki/Gauss-Seidel_method
Use it to solve the Stokes system. What do you observe ?
- Same question as above with a Conjugate Gradient solver. Simply use the algorithm presented in https://en.wikipedia.org/wiki/Conjugate_gradient_method
Note: the CG solver is applicable to symmetric matrices.
- bonus: Create a matrix P (further called the preconditioner) which is zero except on its diagonal where $P_{i,i} = 1/\sqrt{A_{i,i}}$. The original linear system to solve $A \cdot V = b$ can be left multiplied by P to obtain:

$$P \cdot A \cdot P \cdot P^{-1} \cdot V = P \cdot b$$

Defining

$$A' = PAP \quad b' = P \cdot A \quad V' = P^{-1} \cdot V$$

one can solve the new new system $A'V' = b'$ for V' and then recover $V = P \cdot V'$. Try all three solvers again on this system, and for those which converge (if any), compare the number of iterations that the iterative solver needs to reach the required tolerance with and without the preconditioner.

3D FEM Stokes solver (***)

- Download, install and study the SimpleFEM code from www.cedricthieulot.net/simplefem.html
- Rewrite the code so that it can now solve the Stokes equation in 3D.
tip: be very careful about the `icon` array. Draw on paper a $5 \times 4 \times 3$ element grid, and label the nodes and elements. Make sure that your code retrieves this numbering for each element.
- The benchmark present in the downloaded 2D version cannot be extended to 3D. You will therefore model the following setup: The box is a cuboid of size $L_x = 500km$, $L_y = 500km$, $L_z = 1000km$. Either no-slip or free slip boundary conditions are to be prescribed. It contains a fluid M (mantle) with density $\rho_M = 3300kg/m^3$ and viscosity $\mu_M = 10^{22}Pa.s$ which fills the domain. Additionally, you will implement a sphere (S) of higher density $\rho_S = 3350kg/m^3$ and with a viscosity $\mu_S \in [10^{22-26}]Pa.s$ placed at the center $C(x_c, y_c, z_c)$ of the domain. Its radius is $100km$. The functions b_1 and b_2 should not be used anymore and the two commented lines above where b_1 and b_2 are called should be used instead. In order to implement the density and viscosity for the mantle and the sphere, you will create two new functions `my_mu` and `my_rho` which take the coordinates of a point as argument and take the corresponding value at that location. Set the penalty parameter to $\lambda = 10^{28}Pa.s$ and do not change it.
- Bonus: Run the same experiment with the 2D version of the code. Comment on the difference of the maximum of the velocity magnitude in the domain between 2D and 3D.

Full 3D thermomechanically coupled FEM code (****)

- Download, install and study the source code Lab1 from www.cedricthieulot.net/mantledynamics.html
- This code solves alternatively the Stokes and the heat transport equation. It is in fact the SimpleFEM code with temperature calculations added to it, and formulated as a function of the Rayleigh number.
- Rewrite/adapt this code to 3D and look at the questions in Instructions Lab 1. Given the time left and the time taken to run the code, try to address these questions.

Iterative solvers for the heat transport equation (***)

For both 2D FDM and FEM pure-diffusion codes, use/implement the following solvers:

- direct solver
- Jacobi solver
- Gauss-Seidel solver (https://en.wikipedia.org/wiki/Gauss-Seidel_method)
- Conjugate Gradient solver (https://en.wikipedia.org/wiki/Conjugate_gradient_method)
- SSOR solver https://en.wikipedia.org/wiki/Successive_over-relaxation)
- GMRES solver (https://en.wikipedia.org/wiki/Generalized_minimal_residual_method) (BONUS)

Please report the number of iterations that each solver requires to achieve a given tolerance. Explore the effect of resolution.