Solving the velocity field for a 2-D flow in the case of constant viscosity with both Navier-Stokes equation and stream function

Yu Geng

Abstract

Two different 2-D incompressible flow problems with constant viscosities were investigated. The first problem was a fluid with constant density driven by the boundary condition. The time evolution of pressure and velocity was solved with the Navier-Stokes equation. Using different experimental materials, the result showed that reducing the viscosity affects the flow patterns and increases the time needed to reach the steady state. The second problem was a buoyancy driven flow in a vertical gravity field for a density structure with two vertical layers. The instantaneous velocity field as the flow starts was solved with the stream function – vorticity formulation. The maximum magnitude of the velocity field increases linearly with gravity and density contrast, but decreases inversely with viscosity. Finite element and finite difference approaches produced consistent solutions with only small differences. The accuracy of the two methods needs to be further investigated. The discrepancy did not reduce at all when the grid number was increased to twice of the default setting for both methods.

Keywords   Navier-Stokes equation; vorticity; stream function

Contents

1 Introduction

1.1 Strategies to solve for velocity field

1.1.1 The Navier-Stokes equation

The deformation of continuous media usually results from the balance of internal and external forces. In order to relate deformation and forces, an equation of motion must be used. This is the so-called *momentum equation* which describes the conservation of momentum for a continuous medium in the gravity field:

$$\nabla \cdot \sigma + \rho \vec{b} = \rho \frac{d\vec{u}}{dt}. \qquad (1.1)$$

The time-dependent Navier-Stokes equation for an incompressible flow[1] is in the form of

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) = \nabla \cdot \sigma + \vec{f} \qquad (1.2)$$

which can be derived by introducing pressure into the momentum equation (1.1), and using the relationship between total stress and deviatoric stress (Gerya, 2009).

In the case that there is little varying in the flow direction[2] i.e.

$$|\vec{u} \cdot \nabla \vec{u}| \approx 0, \qquad (1.3)$$

the Navier-Stokes equation reduces to

$$(\mathbb{D}) \quad \rho \left(\frac{\partial \vec{u}}{\partial t} \right) = \nabla \cdot \sigma + \vec{f} \qquad (1.4)$$

where $\vec{f} = \rho \vec{b}$. Note that the deviatoric stress in Eqn. (1.4) does not include pressure and is different from the Cauchy stress in Eqn. (1.1).

Assume that the flow is Newtonian[3], then

$$\sigma = 2\mu\epsilon - pI \qquad (1.5)$$

where

$$\epsilon = \frac{1}{2}[\nabla \vec{u} + (\nabla \vec{u})^T]. \qquad (1.6)$$

The Navier-Stokes equation describes the conservation of momentum for a fluid in the gravity field (Gerya, 2009). It is always solved together with the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0. \qquad (1.7)$$

If the density change within each time step is neglectable, a time evolution problem can be treated as a series of steady-state problems. The conservation of fluid requires that the net flow into or out of a rectangle is zero:

$$(\mathbb{D}) \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \qquad (1.8)$$

where u and v are the velocity components of a general flow in the xy plane.

This is the form of the *continuity equation* appropriate to an incompressible fluid.

1.1.2 The vorticity formulation

The momentum equation (1.1) and the continuity equation (1.8) are coupled and are difficult to solve directly. We introduce the stream function, defined as the volumetric flow rate between any two points in a steady flow (Turcotte and Schubert, 2014), to solve both equations for 2D incompressible flow with only one scalar

---

[1]From The FEniCS Tutorial
[2]From Geodynamics lecture notes
[3]From Geodynamics lecture notes

1

variable (Ben-Artzi et al., 2005), whose derivatives are given by

$$u = -\frac{\partial \psi}{\partial y}, \ v = \frac{\partial \psi}{\partial x}. \qquad (1.9)$$

With these two functions, the 2D incompressibility condition is automatically satisfied, which can be verified by substituting Eqn. (1.9) into Eqn. (1.8).

Once we have all the essential equations (1.1)∼(1.9) for this study, the 2D Stokes equations can be then modified into

$$\frac{\partial}{\partial y}\left(\frac{\partial \sigma'_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} - \frac{\partial P}{\partial x} + \rho g_x\right) = 0,$$
$$\frac{\partial}{\partial x}\left(\frac{\partial \sigma'_{yy}}{\partial y} + \frac{\partial \sigma_{yx}}{\partial x} - \frac{\partial P}{\partial y} + \rho g_y\right) = 0.$$

Eliminate pressure from the above two equations,

$$\left(\frac{\partial^2 \sigma'_{xx}}{\partial x \partial y} + \frac{\partial^2 \sigma_{xy}}{\partial y^2}\right) - \left(\frac{\partial^2 \sigma'_{yy}}{\partial x \partial y} + \frac{\partial^2 \sigma_{yx}}{\partial x^2}\right)$$
$$= \frac{\partial \rho}{\partial x}g_y - \frac{\partial \rho}{\partial y}g_x. \quad (1.10)$$

The following derivations are tedious and are selectively omitted. By applying the *constitutive relation* and reformulating Eqn. (1.10) using the stream function, we end up with the so-called *vorticity formulation*, assuming that the viscosity is constant:

$$(\mathbb{D}) \quad \frac{\partial^2 \omega}{\partial y^2} + \frac{\partial^2 \omega}{\partial x^2}$$
$$= \frac{1}{\eta}\left(\frac{\partial \rho}{\partial x}g_y - \frac{\partial \rho}{\partial y}g_x\right), \quad (1.11)$$

where vorticity ω is defined as the curl of velocity or the Laplacian of stream function

$$(\mathbb{D}) \quad \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \omega. \qquad (1.12)$$

In the 2-D case, vorticity is constantly perpendicular to the X-Y plane and can be thereby treated as a scalar.

Eqn. (1.11) and Eqn. (1.12) can be solved subsequently such that the solution of the first equation will become the R.H.S. of the second equation.

1.2 The goal of the study

In this project, we are going to study two different 2-D incompressible flow problems with constant viscosities. The first problem is a fluid with constant density driven by the boundary condition. The time evolution of the velocity field and density distribution will be solved with the coupled Navier-Stoles equation and continuity equation. The second problem is to program a numerical model for buoyancy driven flow in a vertical gravity field for a density structure with two vertical layers. The stream function – vorticity formulation will be solved with both finite element and finite difference approaches to investigate the instantaneous velocity field as the flow starts. The pros and cons of the finite element and finite difference approaches will be compared and discussed in terms of stability, computation time, etc.

2 Methodology

2.1 The Navier-Stokes equation

2.1.1 Weak form reformulation

The Navier-Stokes equation (1.4) and the continuity equation (1.8) can be reformulated into weak forms with a modified

version of the classical splitting method, the so-called incremental pressure correction scheme (IPCS) introduced by Goda (1979). The procedures can be summarized as below:

Firstly, we compute a tentative velocity $\vec{u}^*$ based on the pressure from the previous time step. The variational problem for this first step is

$$\int_\Omega \rho \frac{u^* - u^n}{\Delta t} v d\Omega + \int_\Omega \rho u^n \cdot \nabla u^n v d\Omega$$
$$+ \int_\Omega \sigma(u^{n+1/2}, p^n) \epsilon(v) d\Omega + \int_{\partial\Omega} p^n \hat{n} v ds$$
$$- \mu \int_{\partial\Omega} \nabla u^{n+1/2} \cdot \hat{n} v ds = \int_\Omega f^{n+1} v d\Omega. \tag{2.1}$$

Secondly, we use the computed tentative velocity to compute the new pressure p^n:

$$\int_\Omega \nabla p^{n+1} \nabla q d\Omega = \int_\Omega \nabla p^n \nabla q d\Omega$$
$$- \Delta t^{-1} \int_\Omega \nabla \cdot u^* q d\Omega. \tag{2.2}$$

Finally, we compute the corrected velocity u^{n+1} from Eqn. (2.2). Multiplying this equation by a test function v, we obtain

$$\int_\Omega u^{n+1} v d\Omega = \int_\Omega u^* v d\Omega$$
$$- \Delta t \int_\Omega \nabla(p^{n+1} - p^n) v d\Omega. \tag{2.3}$$

2.1.2 Implementation in FEniCS

At each time level, the solution algorithm consists of the following steps:

1. Compute the intermediate velocity u^* from Eqn. (2.1).

2. Solve the Poisson equation for the pressure p^{n+1} resulting in the variational problem (2.2).

3. Update the velocity from (2.3).

In this manner, we may thus solve the incompressible Navier - Stokes equations efficiently by solving a sequence of three linear variational problems in each time step.

2.2 The vorticity formulation

2.2.1 The finite element approach

Multiply Eqn. (1.11) with a test function and integrate over the study domain,

$$\int_\Omega v \nabla^2 \omega d\Omega = \int_\Omega f v d\Omega. \tag{2.4}$$

The second-order spatial derivative of u can be transformed to a first-derivative of u and v by applying the technique of integration by parts. The formula reads

$$\int_\Omega v \nabla^2 \omega d\Omega =$$
$$- \int_\Omega \nabla v \nabla \omega d\Omega + \int_{\partial\Omega} v \frac{\partial \omega}{\partial n} ds, \tag{2.5}$$

where $\partial\omega/\partial n = \nabla\omega \cdot n$ is the derivative of ω in the outward normal direction n on the boundary.

Taking advantage of the property that

$$v = 0 \text{ on } \partial\Omega, \tag{2.6}$$

the variational form becomes

$$(\mathbb{V}) \quad -\int_\Omega \nabla v \nabla \omega d\Omega = \int_\Omega f v d\Omega. \tag{2.7}$$

In the discretized form, the trial solution and the test function could be given

as the sum of basis functions (Gander and Wanner, 2012):

$$\omega = \sum_{i=0}^{N-1} w_i \varphi_i(x,y),$$

$$v = \sum_{j=0}^{N-1} b_j \varphi_j(x,y).$$

where N is the total number of nodes over the 2-D domain.

Following the same procedures, the variational form of Eqn. (1.12) is given as

$$(\mathbb{V}) \quad -\int_\Omega \nabla q \nabla \psi d\Omega = \int_\Omega \omega q d\Omega \quad (2.8)$$

where

$$\psi = \sum_{i=0}^{N-1} \psi_i \varphi_i(x,y),$$

$$q = \sum_{j=0}^{N-1} b_j \varphi_j(x,y).$$

The choice of basis functions depends on the element type. Since we would like to solve the same problem with different approaches and discuss the pros and cons of the two approaches, the same discretization strategy as the finite different approach is suggested to be used for the finite element approach i.e. the element type

should be quadrilateral. The degree of the basis function could be either one or two. Degree one is chosen in this study.

Using the condition that $g_x = 0$ but $g_y \neq 0$, the force function on the R.H.S. of the first variational form (2.7) reduces to

$$f = \frac{g_y}{\eta} \frac{\partial \rho}{\partial x} \quad (2.9)$$

Introduce a central difference representation

$$f \approx \frac{g_y}{\eta} \frac{\rho_{i,j+1} - \rho_{i,j-1}}{2\Delta x}, \quad (2.10)$$

on a regular grid of 21×31 nodes, with $g_y = 10 \ m/s^2$, $\eta = 10^{21} \ Pa \cdot s$, $3200 \ kg/m^3$ and $3300 \ kg/m^3$ for the left and right layers, the force function and the corresponding meshes are visualized in Fig. 2.1.

2.2.2 The finite difference approach

A finite difference representation of the stream function - vorticity formulation can be obtained by using the same strategy as Eq. (3.20), Section 3 of Gerya (2009):

$$\frac{\omega_{i,j-1} - 2\omega_{i,j} + \omega_{i,j+1}}{\Delta x^2} + \frac{\omega_{i-1,j} - 2\omega_{i,j} + \omega_{i+1,j}}{\Delta y^2} = \frac{g_y}{\eta} \frac{\rho_{i,j+1} - \rho_{i,j-1}}{2\Delta x},$$

$$\frac{\psi_{i,j-1} - 2\psi_{i,j} + \psi_{i,j+1}}{\Delta x^2} + \frac{\psi_{i-1,j} - 2\psi_{i,j} + \psi_{i+1,j}}{\Delta y^2} = \omega_{ij}.$$

where the convention that index j corresponds to the x-axis and index i corresponds to the y-axis is adopted.

The global index is calculated based on geometrical indices as

$$k = N_y \times (j-1) + i, \quad (2.11)$$

For both the finite element and the finite difference approaches, after obtaining the solution of the second equation, the stream function will be converted into velocity field based on Eqn. (1.9) using a cen-
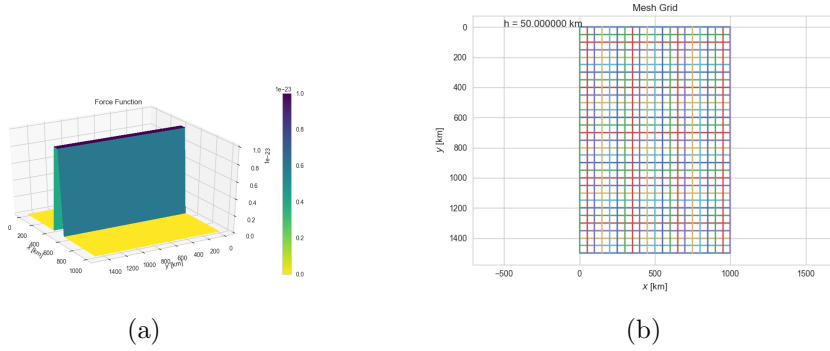
4

(a)   (b)

Fig. 2.1. The discretized force function (a) and the mesh on which it was computed (b). The computation was done on a regular grid of 21×31 nodes, with $g_y = 10 \ m/s^2$, $\eta = 10^{21} \ Pa \cdot s$, 3200 kg/m^3 and 3300 kg/m^3 for the left and right layers.

tral difference approach:

$$v_{x(i,j)} = \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta y},$$

$$v_{y(i,j)} = -\frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta x}.$$

The velocity components should be computed only on the internal nodes of the grid.

added to the top of the study region. The other three boundaries were constraint by no-slip ($v = 0$ m/s) conditions. The density of the fluid is fixed at 3250 kg/m³. The gravity field was ignored. The viscosity is chosen as 10^{21} Pa·s which is the typical value in the Earth's mantle.

3 Results and Discussions

3.1 The Navier-Stokes equation

3.1.1 Default settings

We start with a simple problem as described below: the study region is defined by a rectangular box as $[0, 1000000] \times [0, 1500000]$ and discretized into 20×30 blocks horizontally and vertically. A constant, horizontal velocity of 0.8×10^{-7} m/s flowing to the positive x-direction were

A few selected snapshots from the time evolution of pressure (Fig. 3.1) show that because of the applied velocity boundary condition, the pressure in the upper-right corner gradually increases to about 72 Pa. In contrast, the pressure in the upper-left corner dreceases to -79 Pa. The magnitude of the maximum velocity in the flow is comparable to the velocity boundary condition, 0.9×10^{-7} m/s (Fig. 3.2). The zero velocities occur in the center and the two lower corners of the study region. At these places, the fluid remains steady.
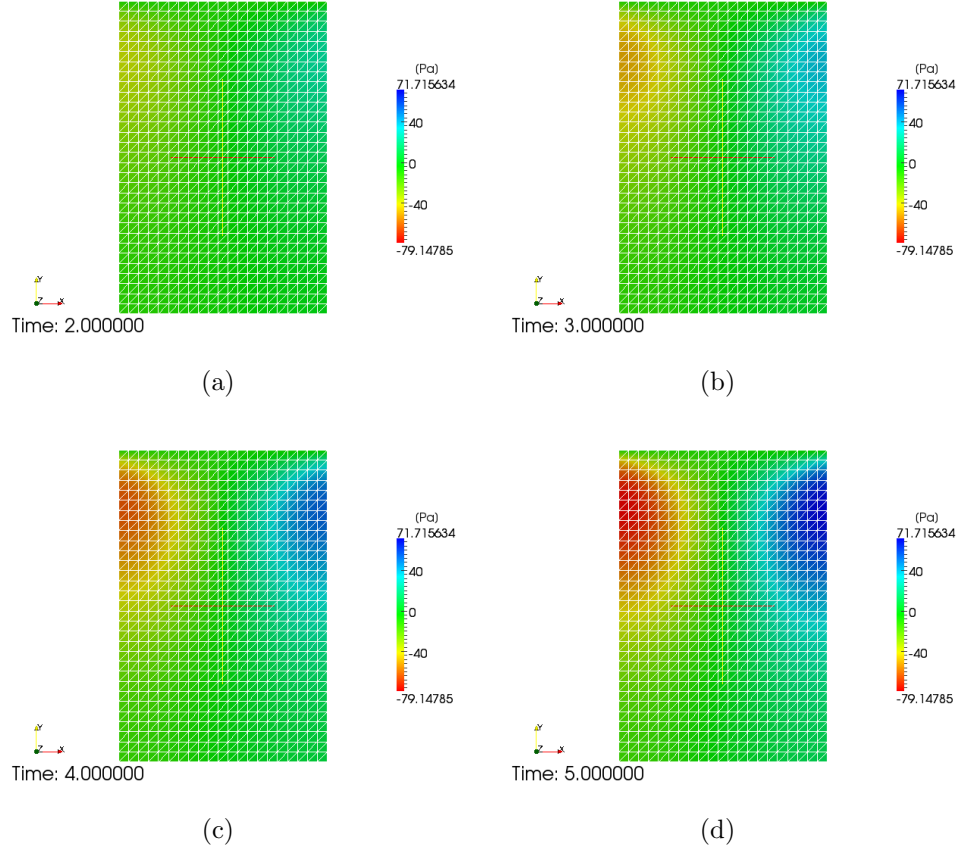
(a)

(b)

(c)

(d)

Fig. 3.1. Selected snapshots from the time evolution animation of pressure at time levels 2.0, 3.0, 4.0 and 5.0 sec: high and low pressures are shown in the upper-right and upper-left corners because of the applied velocity boundary condtion.
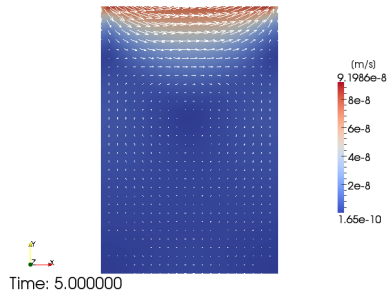

Fig. 3.2. Steady state of the velocity field: the velocity ranges from 0.01 to 9.62 m/s and low values occur in the center and the two lower corners. The viscosity is chosen as 10^{21} Pa·s which is the typical value in the Earth's mantle.

By changing the values of velocity at the open edge at the top, we can examine the behavior of the maximum velocity in the steady-state fluid due to the boundary conditon changes, and plot it as a function of the input (Fig. 3.3). The plot shows that the magnitude changes linearly as the velocity at the open edge increases.
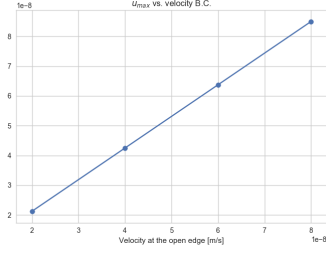
6

Fig. 3.3. Maximum velocity in the steady-state fluid as a function of the boundary condition. The magnitude changes linearly with the velocity at the open edge.

3.1.2   More boundary conditions

Some experiments can be done by using more complicated boundary conditions: an additional velocity boundary condition can be added at the bottom of the study region, with the same magnitude and direction. The flow pattern may differs for different fluids because of their different Reynold's numbers. In this project, behaviors of the matters with significantly different viscosities were studied, as shown in Fig. 3.4. The viscosity of granite is in the scale of 10^{19}. In order to bridge the gap, three imaginary materials with viscosities 2.3×10^{13}, 2.3×10^{11}, and 2.3×10^8 Pa·s were created.

Density and viscosity are critical parameters determining the flow patterns, which can be easily seen from the definition of Reynold's number:

$$Re = \frac{\rho \bar{u} L}{\mu} \qquad (3.1)$$

where μ is the *dynamic viscosity* in Pa·s. To simply the calculation, the average velocity $\bar{u}$ was estimated by multiplying the maximum velocity with a factor of 0.5.

As expected, the addition of the B.C. at the bottom, along with the existing B.C. at the top created two vortexes: the one at the top flows clockwisely and the one at the bottom flows counterclockwisely. However, the pattern cannot be produced when changing the properties of the fluid. A first experiment shows that when reducing the viscosity, the pattern gradually disappeared (Fig. 3.5). As the viscosity decreases from the 10^{19} level to the 10^{13} level, there is some slight changes in the flow pattern: the centers of the vortexes became closer to the top the the bottom. As the viscosity dropped to about 10^8 Pa·s and lower, the pattern disappeared.

However, another possible reason for the above observation is that the time selected for the animations were not long enough. To exclude this factor, a second test was done by using lengths of 5.0 sec, 50 sec, 500 sec and 5000 sec, respectively (Fig. 3.6). The materials being used and all the other parameters remained the same. In the revised test, the flow patterns showed up for all cases after a period of time. However, the reduction of viscosity still exerted some effects on the looks of the patterns.

3.2   The vorticity formulation

3.2.1   Solutions

With the vorticity and stream function fixed at zeros on the boundaries, the experiment started from 21×31 nodes, $g_x = 0, g_y = 10 \ m/s^2$ for the vertical gravity field, $3200 \ kg/m^3$ and $3300 \ kg/m^3$ for the left and right layers. The model size is $1000 \times 1500 \ km$ and a constant viscosity of $\eta = 10^{21} \ Pa \cdot s$ was used. The finite el-

7

```
full_list = {'air': (1.2, 17.4e-6),
             'honey': (1.40e3, 10.0),
             'methanol': (0.792e3, 5.44e-4),
             'glycerol': (1.261e3, 1.2),
             'mercury': (13.534e3, 1.526e-3),
             'granite': (2.70e3, 3e19),
             'dummy_1': (1.153e3, 2.3e13),
             'dummy_2': (1.153e3, 2.3e11),
             'dummy_3': (1.153e3, 2.3e8)}
# structure: {'matter name': (density [kg/m^3], viscosity [Pa.s])}
```

Fig. 3.4. A section of the source code in the project, showing the materials being studied. The densities and viscosities are given in the units of kg/m³ and Pa·s, respectively. Three imaginary materials with viscosities 2.3×10^{13}, 2.3×10^{11}, and 2.3×10^{8} Pa·s were created to bridge the gap between granite and the other common fluids.
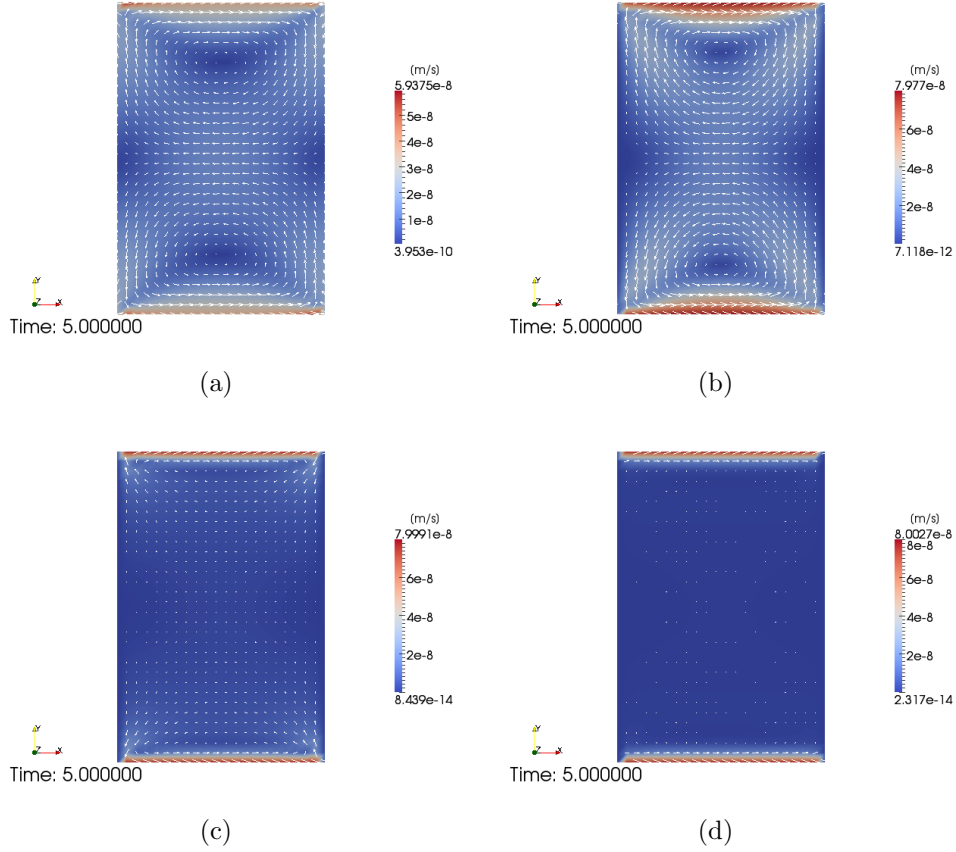

Fig. 3.5. Final states of the velocity field using different densities and viscosities: (a) granite; (b) imaginary material one; (c) imaginary material two; (d) imaginary material three. The velocity added at the bottom has the same magnitude and direction as the B.C. at the top.

ement and finite difference approaches basically produced the same result, as shown in Fig. 3.7. The magnitude of velocities range from 0.0 to 0.4×10^{-7} km/s in the
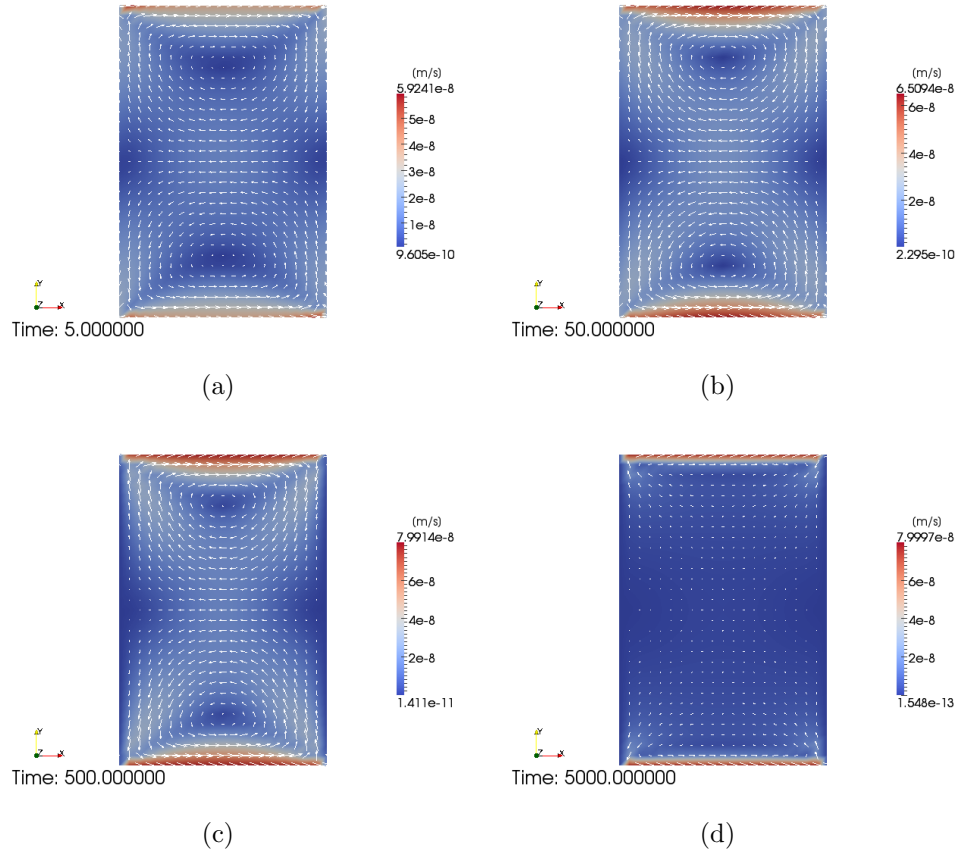
Fig. 3.6. The same experiment as in Fig. 3.5 repeated with larger time lengths. The time used for the four animations were respectively 5.0 sec, 50 sec, 500 sec and 5000 sec. The revised experiment showed that the reduction of viscosity did not stop the transfer of the velocity B.C. to the inner part of the fluid, but the flow pattern has changed due to the reduction of viscosity.

clockwise flow. Zero velocities basically occurred at the center and the four corners of the study region.

For both methods, different spacing and parameters were tried to investigate the accuracy and stability for comparison. The maximum magnitudes of the velocity field were plotted as functions of input parmeters (Fig. 3.8). The experiment showed that the maximum magnitude of the velocity field increased linearly as the gravity and the density contrast between the two layers increased, but showed an inverse relationship with the viscosity of the fluid. The finite element and finite difference solutions showed a tiny difference. Different from the pipe flow problem for which an analytic solution can be found, without an exact solution to the buoyancy driven flow, one cannot say anything about the accuracy of the two methods in this study. Having the suspect that the grid size could have exerted an effect on the accuracy of the solutions, the grid number

9

was increased from 21×31 to 41×31 for both methods. Completely no reduction in the discrepancy was observed.

3.2.2 Performance test

The FeniCS project implicitly uses just-in-time (JIT) compilation to solve finite element problems. The finite element codes for this project run very fastly. Nevertheless, the finite difference code for the stream function approach was written in Python and is not as efficient as the finite element codes. Therefore, we came up with the idea of accelerating the code using Numba.

A section of codes was selected to do the performance test (Fig. 3.9). The code calls stream_2d() to repeatedly solve the 2D stream function problem with different input parameters. The performance test showed that without using Numba, the code took 58.416270 sec to execute; while after accelerating the subroutine stream_2d(), it took 51.385716 sec to execute.

The first iteration using Numba actually took much longer time to run than ordinary Python, whereas as the execution continued, the later iterations took less time and the acceleration accumulated, which made the total time shorter.

4 Conclusions

- In the first problem, several different velocities were used at the boundary to drive a 2-D incompressible flow, the relationship between the maximum magnitude in the velocity field

and the boundary condition is linear. After an additional velocity boundary condition with the same magnitude and direction was added, two vortexes in the flow were observed.

- Several materials, including imaginary matters, with different viscosities were used to try the experiment. Within the same time period (5 sec), the materials with viscosities 10^{11} Pa·s and 10^8 Pa·s did not show the expected flow patterns. After the animations were extended to 500 sec and 5000 sec, the flow patterns showed up but are still different from those with viscosities 10^{19} Pa·s and 10^{13} Pa·s.

- In the second problem, different input parameters were tried to solve the same buoyancy driven flow velocity field. The results turned out to be that the maximum magnitude in the velocity field is linearly related to the gravity and density constrast, and inversely related to the viscosity.

- Acceleration of Numba is significant when there are sufficient number of iterations in the code to compensate the time lose during the first iteration, otherwise, the time spent on the first iteration makes Numba slower than ordinary Python.

- The solutions given by the finite element and finite difference approaches are consistent. There are some slight differences between the solutions of the two methods, but an exact solution is needed for comparison in order to figure out which

method is more accurate. No reduction in the discrepancy was observed as the grid number was increased from 21×31 to 41×31 for both methods.

References

Ben-Artzi, M., Croisille, J.-P., Fishelov, D., and Trachtenberg, S. (2005). A pure-compact scheme for the stream-function formulation of navier–stokes equations. *Journal of Computational Physics*, 205(2):640–664.

Gander, M. J. and Wanner, G. (2012). From euler, ritz, and galerkin to modern computing. *Siam Review*, 54(4):627–666.

Gerya, T. (2009). *Introduction to numerical geodynamic modelling*. Cambridge University Press.

Goda, K. (1979). A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows. *Journal of Computational Physics*, 30(1):76–95.

Turcotte, D. and Schubert, G. (2014). *Geodynamics*. Cambridge University Press.
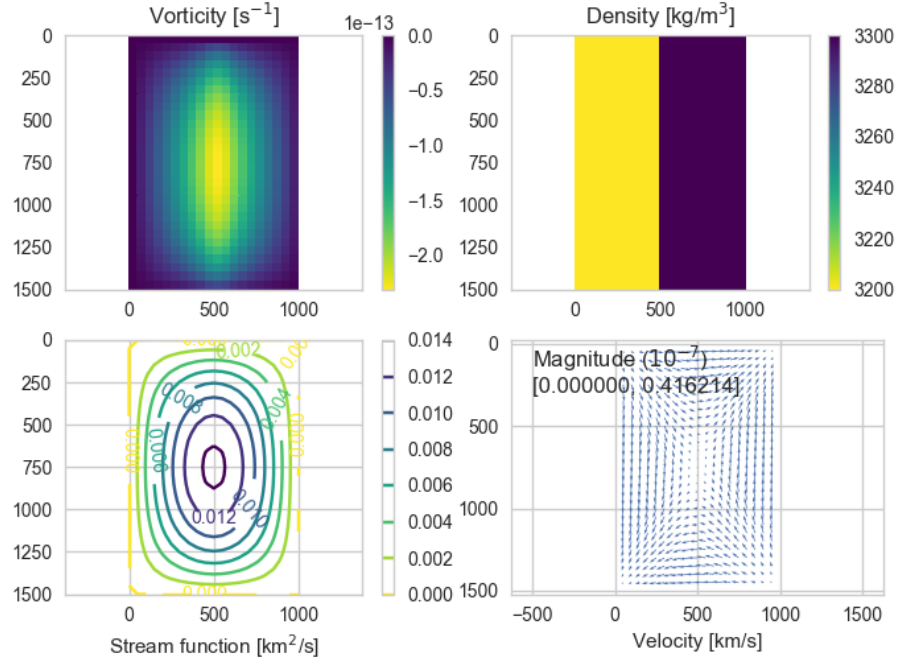
(a)
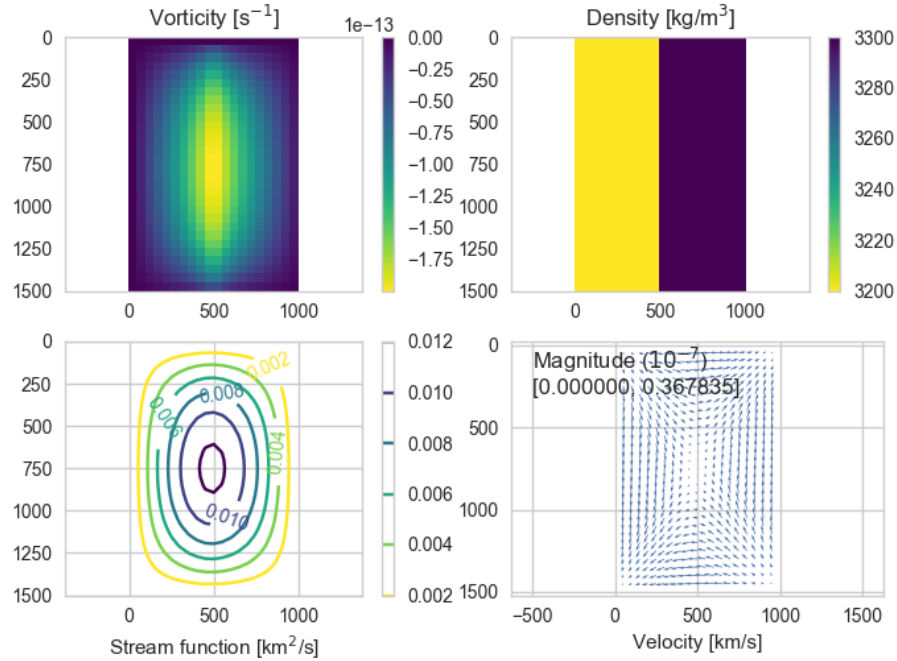
(b)

Fig. 3.7. The finite element (a) and the finite difference (b) solutions to the 2-D buoyancy driven flow problem: velocities range from 0.0 to 0.37×10^{-7} km/s in the resultant clockwise flow; the zero velocities occurred at the center and the four corners of the study region.
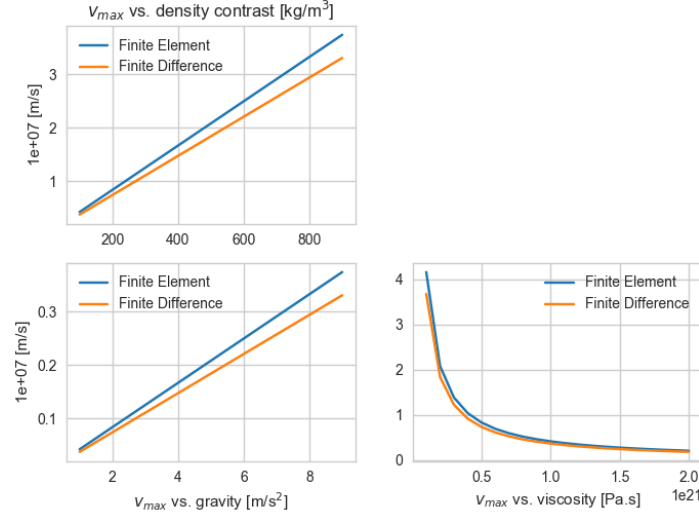
12

Fig. 3.8. Maximum magnitudes of the velocity field as functions of gravity, density contrast between the two layers and viscosity: blue curves, solutions using the finite element approach; orange curves, solutions using the finite difference approach.

```python
# Try with different viscosities.
print("Trying different viscosities...")
logfile = logpath + 'viscosity.log'
fp = open(logfile, 'w')

print("<Performance Test>")
start = time.time()

for eta_i in range(1, 21, 1):
    filename = 'viscosity_%02d' % eta_i
    suptitle = 'Viscosity (10$^{21}$): %f Pa.s' % (eta_i/10.0)
    v_max = stream_2d(xmax, ymax, M, N, eta_i*1e20, gy, \
                      rho_L, rho_R, figpath, filename, suptitle)
    lout = '%f\t%f\n' % (eta_i/10.0, v_max*v_mag)
    fp.write(lout)

end = time.time()
print('<Elapsed Time> %f sec' % (end-start))

fp.close()
print('File saved as: %s\n' % logfile)
```

Fig. 3.9. A section of codes being used to run the performance test. Timing commands were added before and after the loop.

13