PX390, Spring 2018, Assignment 6: dissipative flow

February 2, 2018

You have been tasked with finding flow patterns through a periodic array of obstacles: these obstacles are taken to be uniform in the z direction, so the flow can be solved on the x, y plane. The obstacles are partly permeable to the flow, so instead of completely halting the flow, they provide a frictional force that tends to slow the fluid down. The overall average fluid flow is controlled so that it is constant in time.

The flow may be assumed to be relatively slow and viscous, so the time-stepping stability criterion can be taken to be dominated by the viscosity (the ∇^2 term).

The equations to be solved are the incompressible Navier-Stokes equations for the fluid velocity $\mathbf{u}$, a vector lying in the (x, y) plane; these are solved by time-evolving the (scalar) fluid vorticity $\omega = (\nabla \times \mathbf{u}).\hat{\mathbf{z}}$ as a function of space and time

$$\frac{\partial \omega}{\partial t} + \mathbf{u}.\nabla\omega = \hat{\mathbf{z}}.\nabla \times \mathbf{F} + \nu\nabla^2\omega \tag{1}$$

where $\mathbf{F}(x, y, t)$ is the external force and ν is a (positive) viscosity term.

To account for the obstacles, and a constant external driving force, the external force is

$$\mathbf{F} = \mathbf{G} - K(x, y)\mathbf{u} \tag{2}$$

where $\mathbf{G}$ is time-varying but spatially constant, and $K(x, y)$ represents the frictional force due to the obstacles (specified by the coefficients input file). Note that $\mathbf{G}$ does not lead to a change in the vorticity: this force is varied to maintain the overall mean fluid flow, but does not need to be explicitly calculated.

To find the velocity $\mathbf{u}$ from ω, we use a fluid potential field ϕ. In general, the velocity can then be written $\mathbf{u} = (\nabla\phi) \times \hat{\mathbf{z}} + \mathbf{u}_0$, and in this problem the velocity $\mathbf{u}_0$ is externally controlled (set by the input parameters). To find ϕ, we solve the equation

$$\hat{\mathbf{z}}.(\nabla \times \mathbf{u}) = -\nabla^2\phi = \omega. \tag{3}$$

The velocity does not change if ϕ is changed by a constant, so you can arbitrarily pick a point on the plane to set $\phi = 0$. Note also that a solution in the (x, y) plane only exists if the mean value of ω integrated over one periodic cell is zero:

analytically this mean value is conserved, but you may wish to force the mean of ω to be zero at each timestep.

The boundary conditions in the x and y direction are periodic, with $\mathbf{u}(x,y) = \mathbf{u}(x + IL_x, y + JL_y)$ for any integers I and J (these conditions also hold for ω and ϕ). We take the vorticity at $t = 0$ to be everywhere zero.

Given a vorticity field $\omega(x,y)$, a solution scheme consists of iterating the following steps:

- Find the flow field ϕ given the scalar vorticity ω.
- Find the velocity $\mathbf{u}$
- Use this velocity to find the vorticity field at the next timestep.

1 Specification

The input and output x and y grid have M and N grid points respectively, which will include the end of the domain; these grids will be taken to run from $x_0 = 0$ to $x_{M-1} = L_x - \delta x$, and $y = 0$ to $y_{N-1} = L_y - \delta y$ (the grid spacings are $\delta x = L_x/M$ and $\delta y = L_y/N$). The grid points you use in setting up any matrix problem are up to you. Ideally, for the sake of efficiency, any matrix used should be a narrow banded matrix.

It is acceptable to use methods that are first order in time: you may use an explicit scheme to solve the time-stepping equations.

2 Breaking down the problem

Finding the velocity field associated with a particular vorticity is the main difficulty: first write a code that can do this and check the velocity output using simple vorticity fields.

Note that there is only one nonlinearity in the problem: the convection term. For small velocities, analytical solutions can be found to this periodic problem.

3 Input

The input parameters will be read from a file called 'input.txt'.

1. L_x: right x boundary of domain.
2. L_y: right y boundary of domain.
3. M: number of x grid points.
4. N: number of y grid points.
5. t_f: time to run simulation over.

6. t_d: simulation diagnostic timestep.

7. ν: viscosity.

8. u_{0x}: x component of average velocity.

9. u_{0y}: y component of average velocity.

I will leave you to determine whether these should be read as integers or floating point numbers based on their meaning (note that the example input file does not necessarily have decimal points on all the values that should be floating point).

It is sufficient to use scanf to read in these parameters and simply test that the reads were successful: I am not specifically forbidding the use of other ways of reading input but I found roughly 50% of students attempting a more complicated input scheme got it wrong.

You may assume $M, N > 1$, $L_x, L_y > 0$, and $\nu, K > 0$.

The function $K(x, y)$ will be given at the $M \times N$ grid points as one column of double precision numbers in a file called 'coefficients.txt'. To find the grid point corresponding to the line in the file, the grid points will be in order (x_0, y_0), (x_1, y_0), (x_2, y_0), ..., $(x_{M-1}, y_0), (x_0, y_1)$ etc.

There are example input files and output file on the assignment page.

4 Output:

The code should the vorticity $\omega(x, y)$ to a file 'output.txt' in single column format for all the grid points (x_i, y_j) (in the same order as the coefficients file), for each time $t = 0, t = t_D, t = 2t_D$, etc. You do not need to output the value at $t = t_f$.