

Homework 4

Projection Method and Staggered Grid for Incompressible Flows

due February 18 2016, 12:00

The motivation for this exercise is to demonstrate the advantages of a staggered grid compared to a co-located one for incompressible flows.

When solving the incompressible Navier–Stokes equations, other techniques are normally used compared to those for the solution of the compressible Navier–Stokes equations. One common method which ensures that the velocity field is divergence free is the projection (or pressure correction) method. A detailed description of this method will be given later in the course. However, we use a simple version of it for our model equation.

A strongly simplified one-dimensional analogue to the incompressible Navier–Stokes equations, neglecting the viscous and non-linear convection terms, is given by

$$u_t = -p_x , \quad (1)$$

$$u_x = 0 . \quad (2)$$

Let us start with the discretisation of the time derivative using a first-order accurate backward Euler scheme

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -(p_i^{n+1})_x \quad \Rightarrow \quad u_i^{n+1} = u_i^n - \Delta t (p_i^{n+1})_x . \quad (3)$$

Taking the x -derivative of the equation above yields

$$(u_i^{n+1})_x = (u_i^n)_x - \Delta t (p_i^{n+1})_{xx} . \quad (4)$$

Requiring $u_x = 0$ at $t = t^{n+1}$ results in

$$(p_i^{n+1})_{xx} = \frac{1}{\Delta t} (u_i^n)_x . \quad (5)$$

Your task consists of the following

1. discretise the equations (5) and (3) using central differences second order in space for
 - (a) a co-located grid,
 - (b) a staggered grid (see figure 1);
2. write a program that solves the discretised equations derived above (for both grids).

Solve the equations with initial and boundary conditions

$$u(t = 0, x) = 2, \quad p(t, x = 1) = 5 .$$

The spatial domain is $x \in [-1, 1]$. **At each time step, p^{n+1} is computed first from equation (5), and then u^{n+1} is obtained from the equation (3).**

The implementation of the boundary conditions should be done very carefully. Note that for the given problem explicit boundary conditions for the velocity are not required. At the outflow

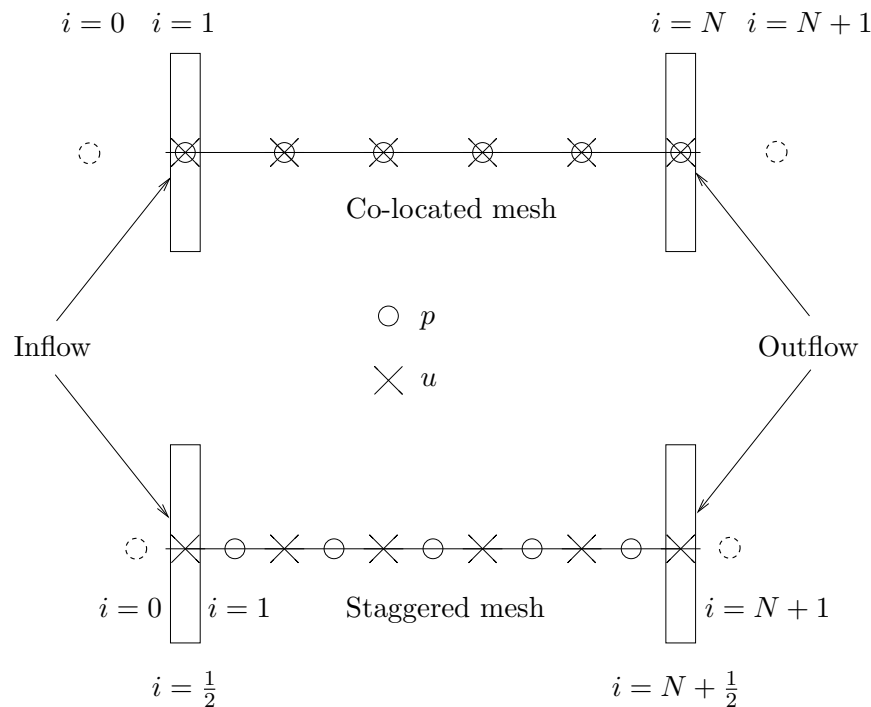


Figure 1: Schematics of a co-located and a staggered grid. Note the numbering of the nodes in both cases and the use of the so-called “ghost points” shown with dashed circles.

Dirichlet boundary conditions for the pressure are given while at the inflow Neumann conditions should be used (first order, *i.e.* $p_1 = p_2$ for the co-located grid and $p_0 = p_1$ for the staggered one). Think how these two conditions should be applied. The values in the “ghost points” p_0 and p_{N+1} can be approximated by linear extrapolation when necessary.

For the report for both co-located and staggered grid you need to

- write the discretised form of the equations (5) and (3)
- write explicitly all equations at the boundaries in the discretised form
- write the equations (5) and (3) in the matrix form, namely

$$\begin{aligned} \underline{\underline{A}} \underline{p}^{n+1} &= \underline{\underline{M}} \underline{u}^n + \underline{b}, \\ \underline{u}^{n+1} &= \underline{u}^n - \underline{\underline{C}} \underline{p}^{n+1} + \underline{d}, \end{aligned}$$

spelling out all the vectors and matrices. Vectors \underline{b} and \underline{d} are needed for correctly including the boundary conditions.

- plot pressure and velocity fields at different time steps. Compare the results for both grids and explain the different behaviour of the solutions.

The idea is to demonstrate the existence of spurious checker-board solutions. Something to keep in mind is that if the chosen initial solution is exact it might happen that you will not observe these spurious solutions. Therefore, perturb your initial condition with some low-amplitude noise ($\sim \mathcal{O}(0.01)$) in the *internal points* of the domain. The MATLAB function `rand` produces random numbers that can be used for that purpose.

- use different $\Delta t/\Delta x$. What effects do you observe? Integrate the equations for long enough such that you can see whether the spurious solutions grow or disappear completely.