Homework 4

Iterative Methods

due February 15, 23:59

Guidelines:

- Use only a **single pdf**. Preferably scan your homework, if you take a photo, make sure it is sharp and bright;
- **Include all plots in the pdf** in the right place;
- Do not present plots without commenting them: always write a (short) description of what the plot tells you and what you can conclude from it;
- You can work in groups up to 3;
- If you work in groups:
  - write the names of all group members at the beginning of the report;
  - you can work and discuss together but each group member is required to submit an individual report written with his/her own words; **copy-paste reports will not be accepted**
- Please use the following naming convention: "**surname_hwX.pdf**" and include all Matlab files as a single **separate** archive: "**surname_hwX.zip**". X is the number of the homework.

An efficient method for the computation of time-dependent incompressible flows is the projection method. In this method, the main effort at each time step is to solve a Poisson equation for the pressure. When discretising the Navier–Stokes equations on a staggered grid one can show that homogeneous Neumann boundary conditions for the pressure should be imposed.

Let us consider a two-dimensional flow. Then we have

$$\begin{aligned} \nabla^2 p = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = f \quad \text{in } \Omega \\ \frac{\partial p}{\partial n} = 0 \quad \text{on } \partial\Omega \end{aligned} \tag{1}$$

Since the boundary condition on all boundaries involves only derivatives of the pressure, to find an unique solution we need to specify the value of the pressure at some point in the domain.

In this homework you are asked to apply an iterative method to solve equation (1) for a domain $\Omega = [0, L_x] \times [0, L_y]$ (set $L_x = L_y = 1$ for this problem). The forcing function f is explicitly given as

$$f(x,y) = \cos(3\pi x)(e^{y^2} + 2y^2 e^{y^2} - e) \ . \tag{2}$$

Equation (1) should be discretised with second-order central differences. Use a grid with $N_x \times N_y$ cells with spacing $h_x = L_x/N_x$ and $h_y = L_y/N_y$, see figure 1. Assemble the numerical solution $\underline{p}$ and the inhomogeneous part $\underline{f}$ as one-dimensional vectors of the form

$$\underline{p} = (p_{1,1}, p_{2,1}, ..., p_{N_x,1}, p_{1,2}, ..., p_{N_x-1,N_y}, p_{N_x,N_y})^T \ , \tag{3}$$

$$\underline{f} = (f_{1,1}, f_{2,1}, ..., f_{N_x,1}, f_{1,2}, ..., f_{N_x-1,N_y}, f_{N_x,N_y})^T \ , \tag{4}$$

where $p_{i,j}$ and $f_{i,j}$ are the values of p and f on the cell centres with the coordinates $x_i = (i - 1/2)h_x$ and $y_j = (j - 1/2)h_y$ (the black dots in figure 1).

Equation (1) is rewritten as a linear system of $N_x \times N_y$ equations of the form $\underline{\underline{L}}\ \underline{p} = \underline{f}$, in which the boundary conditions have yet to be implemented. The homogeneous Neumann condition means that the pressure in cells close to boundary should be equal to those at dummy nodes (gray dots in figure 1), i.e.,

$$p_{1,j} = p_{0,j}, \quad p_{N_x,j} = p_{N_x+1,j}, \quad p_{i,1} = p_{i,0}, \quad p_{i,N_y} = p_{i,N_y+1}\ . \tag{5}$$

This corresponds to a discretised homogeneous Neumann condition of order one.

Two iterative methods will be used in this homework: Gauss-Seidel (GS) and Successive Over-Relaxation (SOR). Both of these methods, when applied to the discretised Poisson equation using central differences, can be written explicitly as

$$p_{i,j}^{(m+1)} = (1-\omega)p_{i,j}^{(m)} + \frac{\omega}{2(1+\beta^2)}\left[p_{i+1,j}^{(m)} + p_{i-1,j}^{(m+1)} + \beta^2(p_{i,j+1}^{(m)} + p_{i,j-1}^{(m+1)}) - h_x^2 f_{i,j}\right]\ , \tag{6}$$

where the Gauss-Seidel method corresponds to the case of $\omega = 1$. $p^{(m)}$ denotes the iterative solution after m number of iterative steps and $\beta = h_x/h_y$.

Applying SOR to the discrete Laplace operator, including boundary conditions (5), along $i = 1$ yields

$$\begin{aligned}
p_{1,1}^{(m+1)} &= (1-\omega)p_{1,1}^{(m)} + \frac{\omega}{1+\beta^2}\left[p_{2,1}^{(m)} + \beta^2 p_{1,2}^{(m)} - h_x^2 f_{1,1}\right]\ ,\\
p_{1,j}^{(m+1)} &= (1-\omega)p_{1,j}^{(m)} + \frac{\omega}{2(1+\beta^2)-1}\left[p_{2,j}^{(m)} + \beta^2 p_{1,j+1}^{(m)} + \beta^2 p_{1,j-1}^{(m+1)} - h_x^2 f_{1,j}\right] (j = 2, N_y - 1)\ ,\\
p_{1,N_y}^{(m+1)} &= (1-\omega)p_{1,N_y}^{(m)} + \frac{\omega}{1+\beta^2}\left[p_{2,N_y}^{(m)} + \beta^2 p_{1,N_y-1}^{(m+1)} - h_x^2 f_{1,N_y}\right]\ .
\end{aligned}$$

Note that SOR is applied after the inclusion of the boundary conditions. The equations along $i = N_x$, $j = 1$ and $j = N_y$ should be modified in a similar way. As mentioned before, since the boundary conditions are only on the derivatives of p, the value of the pressure will only be determined up to an additive constant. To fix this constant replace the equation at node $(1, 1)$ with $p_{1,1} = 0$.

Your task is to

a) Investigate if the problem is well posed. Consider in particular the boundary conditions (Compatibility condition!).

b) Implement the GS and the SOR methods. Choose as initial guess for the inner points $\underline{p}^{(0)} = \underline{0}$ and the iterative solution after m iterations is denoted $\underline{p}^{(m)}$. Report the equations at the boundaries, $i = 1$, N_x and $j = 1$, N_y.

c) Choose $N_x = N_y = 20$. Start by using a direct method (Matlab's "\") to compute a reference solution for the problem $\underline{p}_{\text{direct}}$. Present $\underline{p}_{\text{direct}}$ in a suitable plot (contour or 3D surface).
Show the convergence of the iterative methods by plotting the relative residual

$$r^{(m)} = \frac{\|\underline{\underline{L}}\ \underline{p}^{(m)} - \underline{f}\|_2}{\|\underline{f}\|_2} \tag{7}$$

as a function of the number of iterations m in a semi-logarithmic plot. Compare and discuss the two iterative methods. Use different ω for SOR; how does the convergence rate depend on ω? (Include all relevant plots in your report)

d) Repeat part c) for GS only with $N = N_x = N_y = 30, 40, \ldots, 100$. How does the convergence rate change with respect to N?

e) Let's now compare to an efficient iterative method which is commonly used in CFD: the Generalised Minimal Residual (gmres) method[1]. Gmres is implemented in Matlab in the `gmres()` function

[1]`gmres` is an advanced iterative method for the numerical solution of a nonsymmetric system of linear equations. The documentation can be accessed with `doc gmres`.

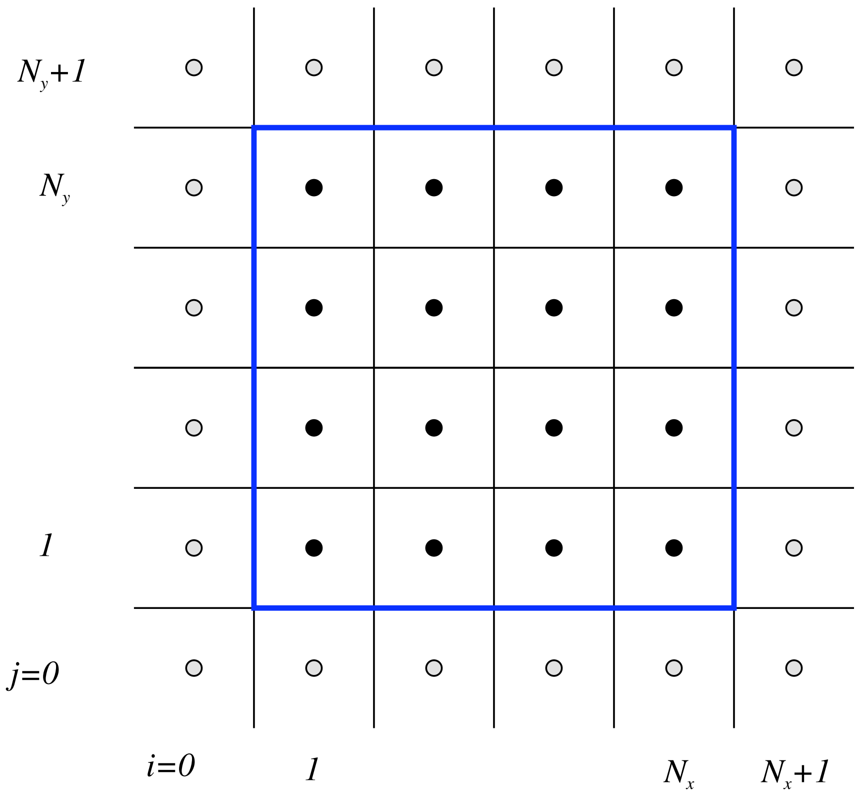

Figure 1: Sketch of the grid. Black circles show the $(N_x \times N_y)$ the internal nodes in which the pressure is evaluated; gray circles represent the ghost nodes situated outside of the computational domain.

```
[p, flag, endrelres, iter, reshist] = gmres(A, f, [], tol);
```

alternatively in python in `scipy.sparse.linalg.gmres`:

```
p, exitCode = scipy.sparse.linalg.gmres(A, f, tol=tol, maxiter=maxit)
```

Solve the problem with $N = 50$. Set the tolerance of all 3 methods to 10^{-6}; set the maximum number of iterations to 10 000 for GS and SOR. Use $\omega = 1.9$ for SOR. Plot and compare the relative residual $r^{(m)}$ as a function of the number of iterations m for all three methods in a semi-logarithmic plot. What do you observe?

A code template can be found on the course homepage.

1 Some suggestions...

1. Before starting, you need to spend the right amount of time to understand how the template works.
2. For the parameters left in the template, p_{direct} plotted as it is in *figure(2)* (line ≈ 32) is reported in Figure 2. If is not the same or reasonably similar, something is wrong.

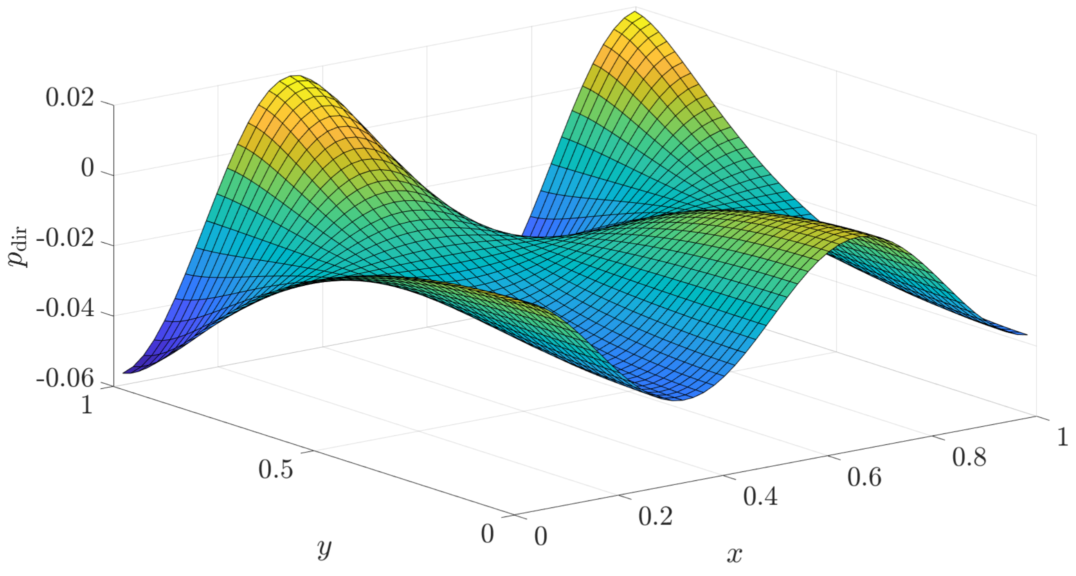

Figure 2: Variable p_{direct}, plotted in *figure(2)* (line ≈ 32 of the template).