MAE 6230 – Computational Fluid Dynamics

Spring 2020

Homework # 7

Due date: Tuesday May 19th, 2020 on Slack or by email.

Concepts: Second-order unsteady Navier-Stokes solver on semi-complex geometry including scalar transport.

Teamwork: Please turn in a single document per team of two, with your names on it. Provide a few sentences explaining in details the contribution of each team member to the overall homework.

1 Objectives

You will finish up your flow solver in this final homework. The following four main objectives will be pursued:

1. Implementation of an explicit second order Adams–Bashforth (AB) temporal discretization for the convective term.
2. Implementation of an implicit second order Crank–Nicolson (CN) temporal discretization for the viscous term using alternating direction implicit (ADI).
3. Implementation of a new flow configuration (i.e., modifications to the boundary conditions): a wake flow in a channel.
4. Implementation of a scalar transport equation following the same discretization (AB + CN with ADI) using various spatial discretizations.

2 Adjustments to the Navier-Stokes solver

2.1 Spatial discretization – second order finite volumes on a staggered mesh

The spatial discretization was extensively discussed in homework #5, please leave it unchanged in this work.

2.2 Temporal discretization – the fractional step method

We will finish the implementation of the methodology described by Kim and Moin (1985), known as fractional step method. Their method combines an explicit Adams–Bashforth discretization of the convective term with a Crank–Nicolson discretization of the viscous term, and uses a projection idea to handle the pressure coupling. In homework #4, you used an explicit forward scheme in time for simplicity.

Step 1. The momentum equations are advanced without a pressure term. We will use a delta notation to shorten the expressions for the spatial derivatives presented above. Similarly, the effect of staggering is ignored in the equations to simplify the expressions (do not ignore staggering in your code!). Finally, H is used as shorthand for the convective term.

$$\frac{\boldsymbol{u}^\star_{i,j}-\boldsymbol{u}^n_{i,j}}{\Delta t}=\frac{3}{2}H^n_{i,j}-\frac{1}{2}H^{n-1}_{i,j}+\frac{1}{2\,\mathrm{Re}}\left(\frac{\delta^2}{\delta x^2}+\frac{\delta^2}{\delta y^2}\right)\left(\boldsymbol{u}^\star_{i,j}+\boldsymbol{u}^n_{i,j}\right). \tag{1}$$

$H^n_{i,j}$ represents the convective term written from the velocity at time n, while $H^{n-1}_{i,j}$ represents the convective term written from the velocity at time $n-1$. Adams–Bashforth consists of a linear extrapolation of the convective term from $n-1$ and n to $n+\frac{1}{2}$, providing second order accuracy for the temporal integration. The viscous term is based on Crank–Nicolson, which is implicit and second order accurate as well. Combining this equation with an alternating direction implicit (ADI) idea, we can rewrite it as a two-step update that involves only tridiagonal solves:

$$\begin{aligned}\left(1-\frac{\Delta t}{2}\frac{1}{\mathrm{Re}}\frac{\delta^2}{\delta x^2}\right)\Delta\boldsymbol{u}^{\star\star}_{i,j} &= \frac{\Delta t}{2}\left(3H^n_{i,j}-H^{n-1}_{i,j}\right)+\frac{\Delta t}{\mathrm{Re}}\left(\frac{\delta^2}{\delta x^2}+\frac{\delta^2}{\delta y^2}\right)\boldsymbol{u}^n_{i,j}, & (2)\\ \left(1-\frac{\Delta t}{2}\frac{1}{\mathrm{Re}}\frac{\delta^2}{\delta y^2}\right)\Delta\boldsymbol{u}^{\star}_{i,j} &= \Delta\boldsymbol{u}^{\star\star}_{i,j}, & (3)\end{aligned}$$

followed by the velocity update $\boldsymbol{u}^\star_{i,j}=\boldsymbol{u}^n_{i,j}+\Delta\boldsymbol{u}^\star_{i,j}$. Note that tridiagonal Gauss elimination (explicitly presented in class) is ideally suited for equations 2 and 3.

In homework #5, it was seen that the viscous CFL (aka Fourier number) is the most stringent restriction to the time step size. The CN treatment of the viscous term fully eliminates this restriction, allowing you to take significantly larger time steps (but still constrained by the convective CFL).

Step 2. A geometric projection is used to enforce that the divergence of the velocity should be zero, leading to the pressure Poisson equation,

$$\nabla\cdot\left(\nabla p^{n+1}\right)_{i,j}=\frac{1}{\Delta t}\left.\nabla\cdot\boldsymbol{u}^\star\right|_{i,j}. \tag{4}$$

To solve this equation, make use of the fastest linear solver you have at your disposal from homework #6. For reference, the solver I ended up using that combined both simplicity and performance was a conjugate gradient solver preconditioned by a Gauss-Seidel relaxation.

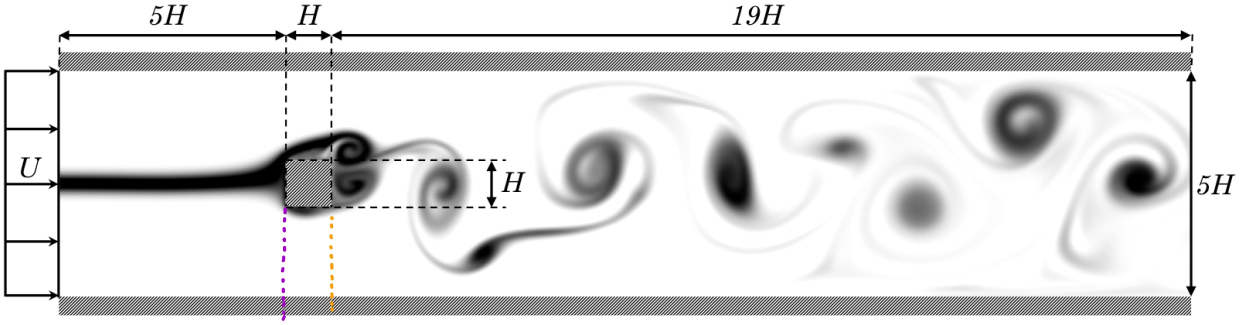
Step 3. Once you have solved for the pressure, the velocity must be corrected using

$$\boldsymbol{u}^{n+1}_{i,j}=\boldsymbol{u}^\star_{i,j}-\Delta t\left.\nabla p^{n+1}\right|_{i,j} \tag{5}$$

With these modifications, your code is exactly mass and momentum conserving, second order in time and space, and not constrained by viscous CFL. As far as incompressible viscous flows are concerned, we are getting very close to a state-of-the-art research code...

3 Unsteady wake flow behind an obstacle

A flow configuration corresponding to a wake behind a square obstacle will be considered.



cell 101... 100 in code cell

Compared to the lid-driven cavity flow, you will need to:

- introduce no-slip conditions along all four sides of the square obstacle, 101... 100
- replace the top sliding wall condition by a static wall,
- provide a Dirichlet boundary for the velocity at the inflow (where $\boldsymbol{u} = (U, 0)^\mathsf{T}$),
- use a Neumann condition on the velocity at the outflow,
- and like for the lid-driven cavity, use Neumann conditions on pressure for all surfaces.

Note on the outflow: you will need to ensure that the RHS of the pressure Poisson equation is exactly zero. To that end, you should adjust the outflow velocity so that its bulk value is equal to U (implying that mass is exactly conserved).

We will target two Reynolds numbers (defined with U and H): 20 and 400, using a mesh characterized by $\Delta x = \Delta y = H/20$.

4 Scalar transport

We will also solve a scalar transport equation. For a scalar ϕ (you can think of it as a very low concentration of dye, for example), we can write

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\boldsymbol{u}\phi) = \nabla \cdot (\mathcal{D} \nabla \phi) \, , \tag{6}$$

where $\mathcal{D}$ is the diffusion coefficient of ϕ. We will use a non-dimensional diffusion coefficient equal to 10^{-4} (i.e., the Schmidt number in the case $\mathrm{Re} = 20$ is $\mathrm{Sc} = 500$, and in the case $\mathrm{Re} = 400$, it is $\mathrm{Sc} = 25$). It is convenient to store the scalar at the cell center, like the pressure. Solve this equation using second order accuracy in time (AB for convection, CN for diffusion using ADI), second order central differencing in space for the diffusion term, and implement first order unwinding, second order central differencing, and third order QUICK for the spatial discretization of the convective term.

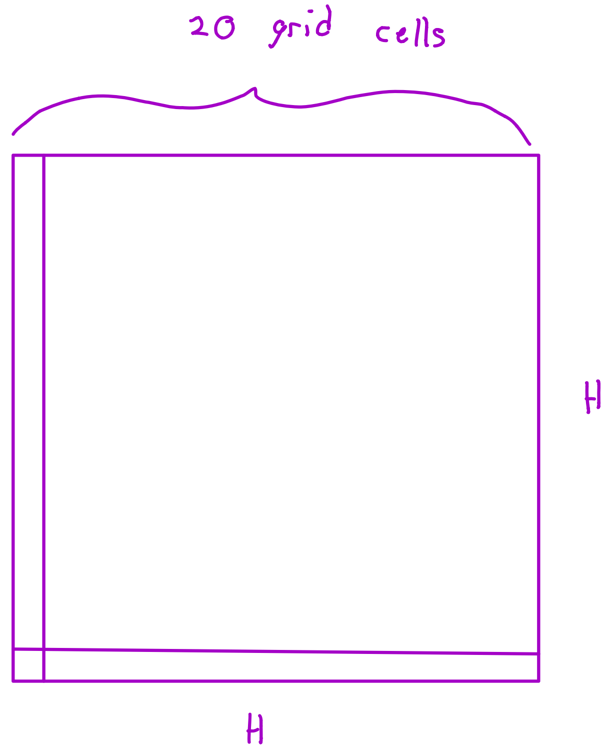
For boundary conditions, use Neumann at the top, bottom, and right boundaries. At the left boundary please use a Dirichlet condition with the profile

$$\phi(y, t) = \exp\left(- \left(\frac{y}{0.2H} \right)^2 \right) , \tag{7}$$

assuming that the domain is centered at $y = 0$.

Please answer the following:
When needed, assume that the depth is unity (i.e., give your answers per unit depth).

1. Plot the drag and lift coefficients of the square (i.e., C_D and C_L) as a function of time from $tH/U = 0$ to $tH/U = 200$, for both Reynolds numbers. Does this match your expectations? Why?
2. Provide a surface plot of ϕ at $tH/U = 200$ for the high Reynolds number case using first order upwinding, second order central differencing, and third order QUICK. Comment briefly on the relative cost and performance of these schemes.
3. Provide a brief critique of your simulations at both Reynolds numbers: are they physically accurate, or are there strong assumptions?
4. What time step size do you use to run at each Reynolds number, and why? Does this match your expectations?
5. Please provide your code.

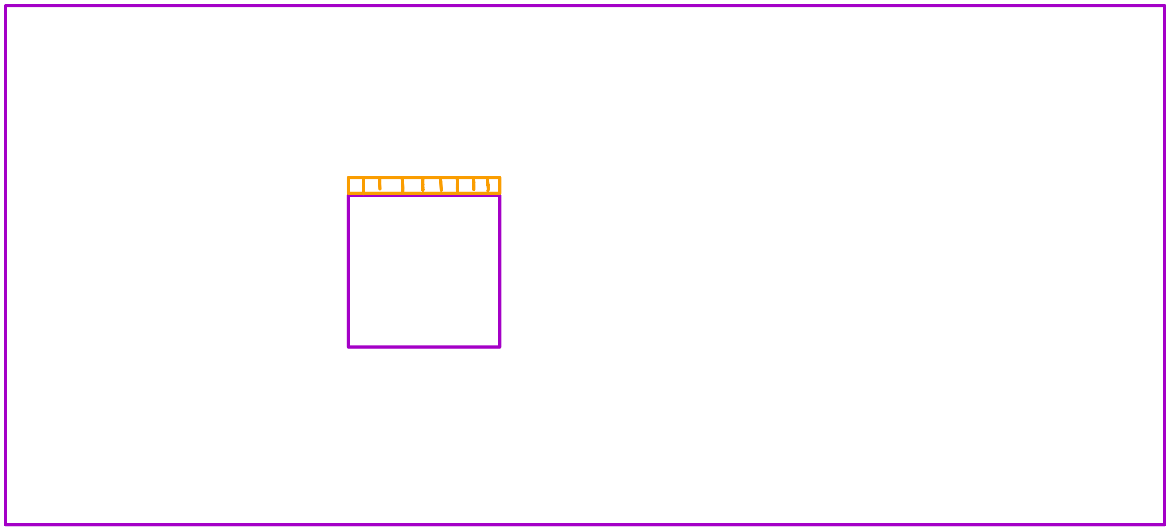


$\Delta x = \Delta y = \frac{H}{20}$

domain: $(5H + H + 19H) \times 5h$

along x-direction: $25 \cdot 20$ grid cells $=$ 500 grid cells

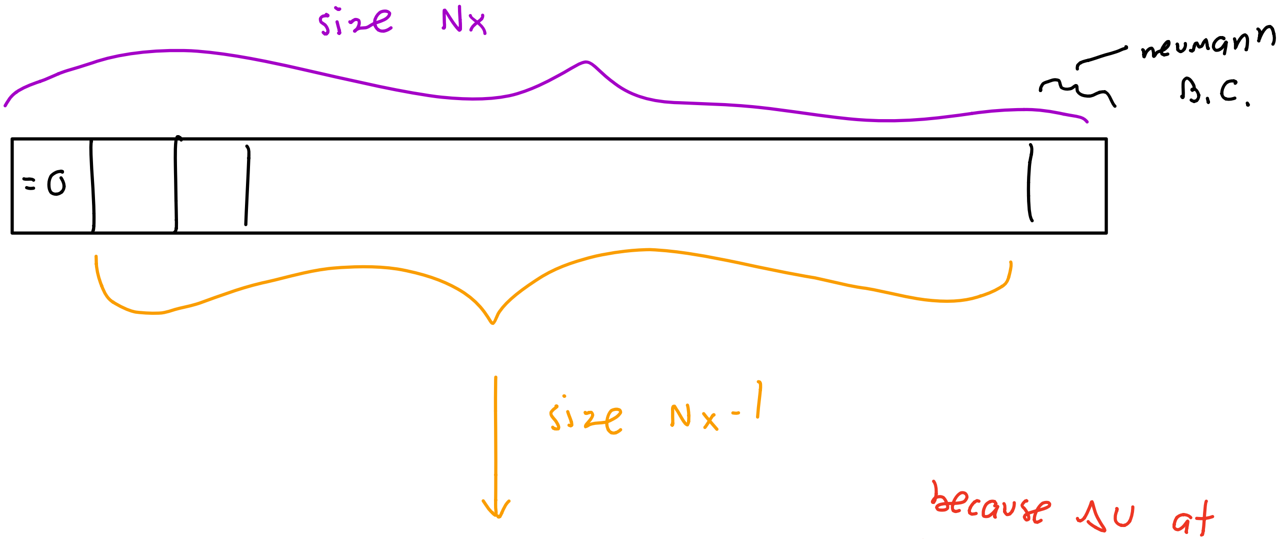
along y-direction: $5 \cdot 20$ grid cells $=$ 100 grid cells



$$\left(1 - \frac{\Delta t}{2}\frac{1}{\mathrm{Re}}\frac{\delta^2}{\delta x^2}\right) U_{j,i} = U_{j,i} - \frac{\Delta t}{2Re}\left(U_{j,i+1} - 2\cdot U_{j,i} + U_{j,i-1}\right)\cdot\frac{1}{dx^2}$$

$$= \frac{-\Delta t}{2Re}\cdot\frac{1}{dx^2}U_{j,i+1} + \left(1 + \frac{\Delta t}{Re}\cdot\frac{1}{dx^2}\right)U_{j,i} - \frac{\Delta t}{2Re}\cdot\frac{1}{dx^2}U_{j,i-1}$$

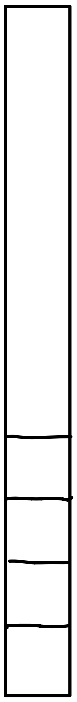
first row ΔU^{**}



because ΔU at boundary is zero

$$\begin{bmatrix} (1+\frac{\Delta t}{Re}) & \frac{-\Delta t}{2Re} & & 0 \\ \frac{-\Delta t}{2Re} & (1+\frac{\Delta t}{Re}) & \frac{-\Delta t}{2Re} & \\ & \frac{-\Delta t}{2Re} & (1+\frac{\Delta t}{Re}) & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \Delta U^{**}_{j,1} \\ \Delta U^{**}_{j,2} \\ \vdots \\ \Delta U^{**}_{j,Nx-2} \end{bmatrix} = \begin{bmatrix} step\_1\_mat\_X_{j,1} \\ step\_1\_mat\_X_{j,2} \\ \vdots \\ step\_1\_mat\_X_{j,Nx-2} \end{bmatrix} \begin{matrix} + \frac{\Delta t}{2Re}\cdot 0 \\ \\ \\ + \frac{\Delta t}{2Re}\cdot 0 \end{matrix}$$

second row



$$\left(1 - \frac{\Delta t}{2} \cdot \frac{1}{Re} \frac{\partial^2}{\partial y^2}\right) U_{j,i}$$

$$= U_{j,i} - \frac{\Delta t}{2Re}\left[\frac{\partial^2}{\partial y^2} U_{j,i}\right]$$

$$= U_{j,i} - \frac{\Delta t}{2Re}\left[\frac{(U_{j+1,i} - U_{j,i}) \cdot \frac{1}{\Delta y} - (U_{j,i} - 0) \cdot \frac{2}{\Delta y}}{\Delta y}\right]$$

$$= \frac{-\Delta t}{2Re\,\Delta y^2} U_{j+1,i} + \left(1 + \frac{\Delta t}{2Re\,\Delta y^2} + \frac{\Delta t}{Re\,\Delta y^2}\right) U_{j,i}$$

$$= \frac{-\Delta t}{2Re\,\Delta y^2} U_{j+1,i} + \left(1 + \frac{3\Delta t}{2Re\,\Delta y^2}\right) U_{j,i}$$

↑

take $j = 0$ or $j =$ right above obstacle boundary

for Neumann B.C. at outlet boundary,

$$\left(1 - \frac{\Delta t}{2Re} \frac{\partial^2}{\partial x^2}\right) U_{j,i} = U_{j,i} - \frac{\Delta t}{2Re}\left(U_{j,i-1} - U_{j,i}\right) \cdot \frac{1}{dx^2}$$

$$= U_{j,i-1}\left(\frac{-\Delta t}{2Re} \cdot \frac{1}{dx^2}\right) + U_{j,i}\left(1 + \frac{\Delta t}{2Re} \cdot \frac{1}{dx^2}\right)$$

$$\frac{\partial\phi}{\partial t} + \overset{\substack{\text{convection}\\ \downarrow}}{\nabla\cdot(\boldsymbol{u}\phi)} = \overset{\substack{\text{diffusion}\\ \downarrow}}{\nabla\cdot(\mathcal{D}\nabla\phi)}\,,$$

using second order accuracy in time (AB for convection, CN for diffusion using ADI), second order central differencing in space for the diffusion term, and implement first order unwinding, second order central differencing, and third order QUICK for the spatial discretization of the convective term.

$$\frac{\partial \phi}{\partial t} = \frac{\phi_{j,i}^{n+1} - \phi_{j,i}^{n}}{\Delta t}$$

$$\nabla\cdot(D\nabla\phi) = D\nabla^2\phi$$

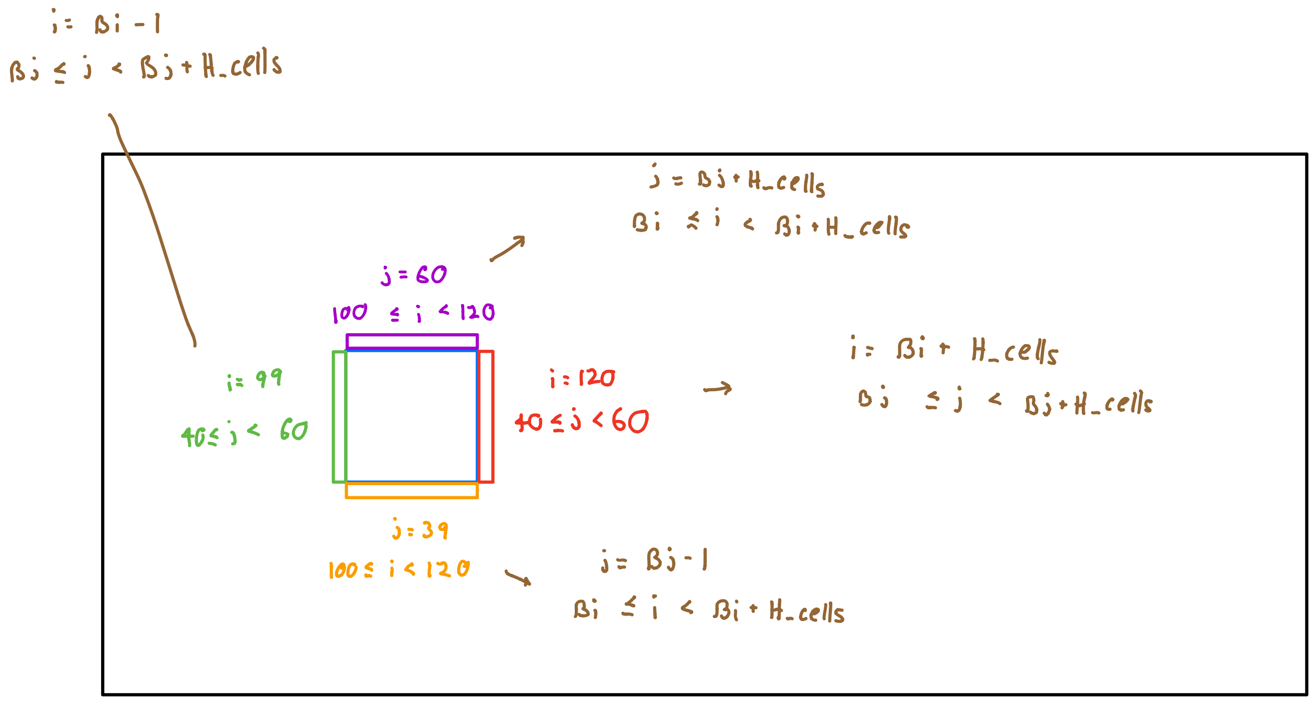
$$= D\left[\frac{\partial^2}{\partial x^2}\phi + \frac{\partial^2}{\partial y^2}\phi\right]$$

$$= D\left[\frac{\phi_{j,i+1} - 2\phi_{j,i} + \phi_{j,i-1}}{\Delta x^2} + \frac{\phi_{j+1,i} - 2\phi_{j,i} + \phi_{j-1,i}}{\Delta y^2}\right]$$

$$\nabla\cdot(u\phi) = \frac{\partial}{\partial x}(u\phi) + \frac{\partial}{\partial y}(v\phi)$$

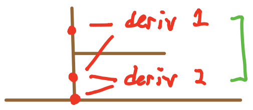
$$\phi_{j,i}^{n+1} = \phi_{j,i}^{n} + \Delta t\,(\text{diffu} - \text{convec})$$

$Bi = 100$
$Bj = 40$
$H\_cells = 20$



Small things that may be an issue

① calculations of $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 v}{\partial x^2}$, $\frac{\partial^2 u}{\partial y^2}$, $\frac{\partial^2 v}{\partial y^2}$ in step 1 near walls may need to be updated.



distance between is dy but maybe take as $\frac{3dy}{4}$?

check for \(2/dy) or \(2/dx)

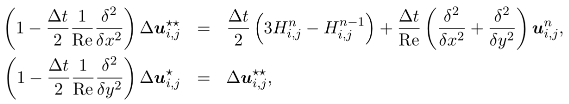
② I take ΔU^{**} and ΔU^{*} to be zero along boundaries. Does this make sense?

Ny - Bj - H_cells

major things I've changed:

- $+Hx \rightarrow -Hx$
- In n step Gauss-Seidel solves, bounds for code after "compute the norm" and "impose condition that the integral ..."

  Time step was too large



$$\left(1-\frac{\Delta t}{2}\frac{1}{\mathrm{Re}}\frac{\delta^2}{\delta x^2}\right)\Delta \boldsymbol{u}^{\star\star}_{i,j} = \frac{\Delta t}{2}\left(3H^n_{i,j}-H^{n-1}_{i,j}\right)+\frac{\Delta t}{\mathrm{Re}}\left(\frac{\delta^2}{\delta x^2}+\frac{\delta^2}{\delta y^2}\right)\boldsymbol{u}^n_{i,j},$$

$$\left(1-\frac{\Delta t}{2}\frac{1}{\mathrm{Re}}\frac{\delta^2}{\delta y^2}\right)\Delta \boldsymbol{u}^{\star}_{i,j} = \Delta \boldsymbol{u}^{\star\star}_{i,j},$$

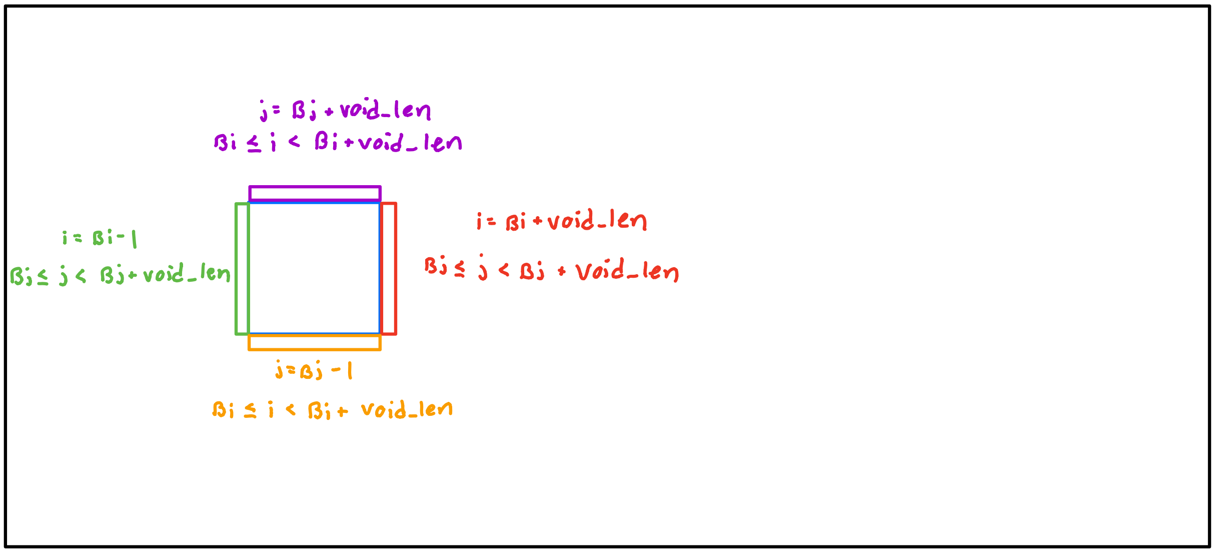
$$\left(1-\frac{\Delta t}{2}\frac{1}{Re}\frac{\partial^2}{\partial x^2}\right)\left(1-\frac{\Delta t}{2}\frac{1}{Re}\frac{\partial^2}{\partial y^2}\right)\Delta U^* = \frac{\Delta t}{2}\left(3H^n-H^{n-1}\right)+\frac{\Delta t}{Re}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)U^n$$

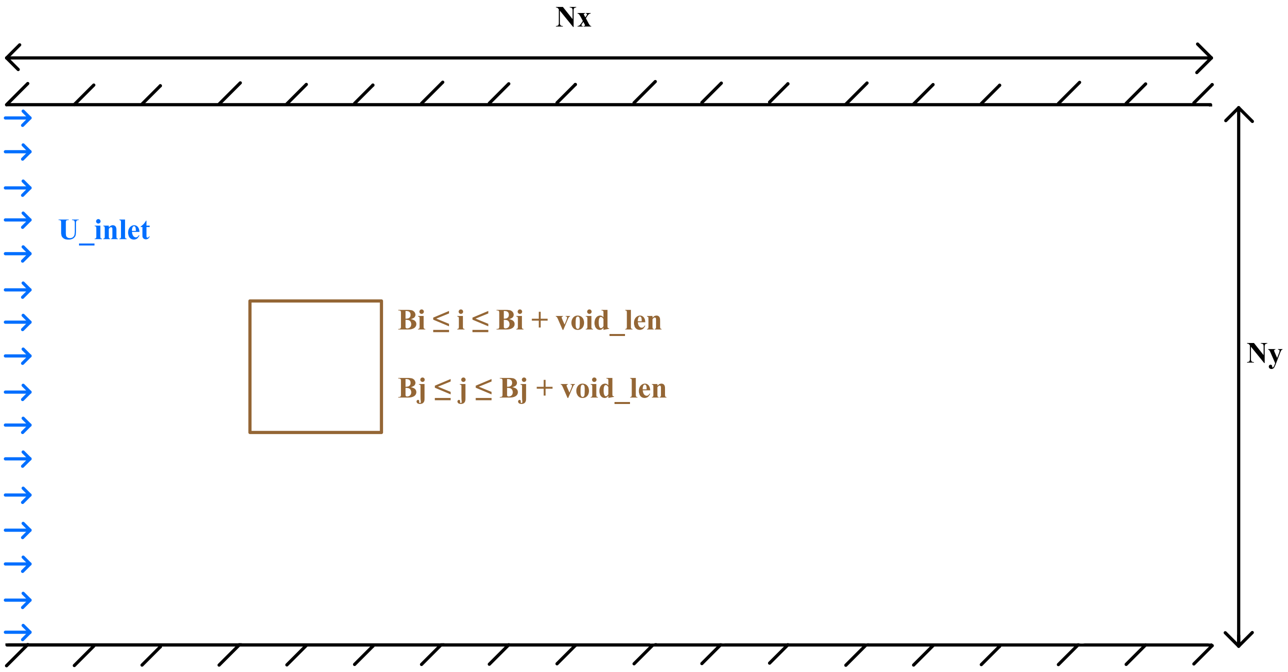
$$\Delta U^* - \frac{\Delta t}{2}\cdot\frac{1}{Re}\frac{\partial^2}{\partial x^2}\Delta U^* - \frac{\Delta t}{2}\cdot\frac{1}{Re}\frac{\partial^2}{\partial y^2}\Delta U^* = \frac{\Delta t}{2}\left(3H^n-H^{n-1}\right)+\frac{\Delta t}{Re}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)U^n$$

$$\frac{1}{\Delta t}\Delta U^* - \frac{1}{2Re}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\Delta U^* = \frac{3}{2}H^n-\frac{1}{2}H^{n-1}+\frac{1}{Re}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)U^n$$

$$\frac{1}{\Delta t}\Delta U^* - \frac{1}{2Re}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\left(U^*-U^n\right) = \frac{3}{2}H^n-\frac{1}{2}H^{n-1}+\frac{1}{Re}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)U^n$$

$$\frac{1}{\Delta t}\Delta U^* = \frac{3}{2}H^n-\frac{1}{2}H^{n-1}+\frac{1}{Re}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\left(\frac{U^*}{2}+\frac{U^n}{2}\right)$$


j= Bj + void_len
Bi ≤ i < Bi+void_len
i= Bi-1
Bj≤ j < Bj+void_len
i= Bi + void_len
Bj ≤ j < Bj + void_len
j=Bj - 1
Bi ≤ i < Bi+ void_len


Nx
U_inlet
Bi ≤ i ≤ Bi + void_len
Bj ≤ j ≤ Bj + void_len
Ny