# MAC for Incompressible Navier-Stokes Flow

A. Donev, Fall 2021

The complete Navier Stokes equation, i.e., the <u>momentum conservation</u> equation, valid for inhomegeneous fluids as well, is:

$$\frac{\text{momentum}}{\text{density}} = \underset{\substack{\uparrow \\ \text{mass density}}}{\rho} \, \vec{u} \leftarrow \text{fluid velocity}$$

$$(\rho \vec{u})_t + \vec{\nabla} \cdot (\rho \vec{u} \vec{u}^T) + \nabla p = \nabla \cdot (\overset{\leftrightarrow}{\sigma}_v) + \rho \vec{g}$$

pressure ($p$); viscous stress tensor ($\overset{\leftrightarrow}{\sigma}_v$); (e.g.) gravity ($\vec{g}$)

This needs to be supplemented with (at least) mass conservation:

$$\rho_t + \nabla\cdot(\rho\vec{u}) = 0$$

(continuity equation)

Equivalent form of momentum eq:

$$\rho\vec{u}_t + \rho\,\vec{u}\cdot\nabla\vec{u} = \nabla\cdot(\vec{\sigma}_v) + \rho\vec{g}$$

Where viscous stress tensor is:

$$\overleftrightarrow{\sigma}_v = \eta\,(\vec{\nabla}\vec{u} + \vec{\nabla}^T\vec{u})$$

$$\sigma_{ij}^{(v)} = \eta(\partial_i u_j + \partial_j u_i)$$

Often one can make two key assumptions:

$\rho = \text{const}$ ← means flow is Low Mach number

$M_a = \dfrac{\|u\|_\infty}{c_{\text{sound}}} \ll 1$

$\eta = \text{const}$

constant viscosity

2

This allows for some key simplifications of the equations:

$$\nabla\cdot(\rho\vec{u}) = \rho\,\nabla\cdot\vec{u} = 0 \Rightarrow \nabla\cdot\vec{u}=0$$

(incompressible flow)

This leads to more simplifications:

$$\nabla\cdot\overleftrightarrow{\sigma} = \eta\,\nabla^2\vec{u} \quad (\eta = \text{const})$$

why: $\partial_j \sigma_{ji} = \eta\,\partial_j^2 u_i + \eta\,\partial_i \underbrace{(\partial_j u_j)}_{\text{zero}}$

$$\begin{cases} \rho\big(\underbrace{\vec{u}_t + \vec{u}\cdot\vec{\nabla}\vec{u}}_{D_t\vec{u}\ \text{(advective derivative)}}\big) + \nabla p = \eta\,\nabla^2\vec{u} \\ \nabla\cdot\vec{u} = 0 \end{cases}$$

Pressure is the Lagrange multiplier for incompressibility

Challenge:

1) $\vec{u}\cdot\vec{\nabla}\vec{u}$ is nonlinear
2) Incompressibility constraint

Formally, NS equations are a differential-algrebraic (DAE) system of equations of index 2.
This is hard even for ODEs!

In periodic / unbounded domains, we can eliminate pressure using a projection operator. Start with Hodge decomposition theorem:

$$\forall \vec{v} = -\vec{\nabla}\varphi + \vec{\nabla}\times\vec{A}$$
$$= -\vec{\nabla}\varphi + \vec{u}$$

s.t. $\vec{\nabla}\cdot\vec{u} = 0$

$$\vec{u} = \overleftrightarrow{\mathcal{P}}\,\vec{v} = \vec{v} + \vec{\nabla}\varphi$$

projection operator

How to compute $\varphi$?

$$\vec{\nabla} \cdot \vec{u} = \nabla \cdot \vec{v} + \nabla^2 \varphi = 0$$

$$\Rightarrow \quad -\nabla^2 \varphi = \nabla \cdot \vec{v} \Leftarrow$$ Poisson equation for pressure

(Solving the incompressible equations is always at least as hard as solving a Poisson equation (elliptic), so generally much more expensive than simple advection-diffusion).

In periodic/unbounded domains we can write NS in unconstrained form (c.f. vorticity-stream):

$$u_t = \mathcal{P}\left[-u \cdot \nabla u + \nu \nabla^2 u + f\right]$$

$\nu = \eta/\rho$ gravity/wind/etc.

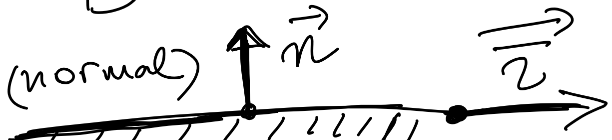
5

The problem is that this does NOT work with boundary conditions!
First methods for NS were based on the projection method of A. Chorin, but I will present a different version that deals correctly with BCs.

Key point to remember:
One cannot seperate pressure & velocity when there are physical boundaries: must solve for both at the same time.
In other words, we will need to solve something like Stokes equation not Poisson equation to get evolution of $v/p$.

# BCs for incompressible NS

At a physical boundary, the following BCs are common.




$\vec{u} = u_n + \vec{u}_\tau$

**Normal component** (pick one)
(also for Euler eqs)

① No penetration (Dirichlet)

$u_n = \vec{u}\cdot\vec{n} = 0$ ← normal vector

② Specified normal stress:

(Neumann) $\vec{n}\cdot\overleftrightarrow{\sigma}\cdot\vec{n} = -p + 2\eta \frac{\partial}{\partial n} u_n$

**Tangential component** (pick one)
(only if $\eta > 0$)

① Specified slip (or no slip)

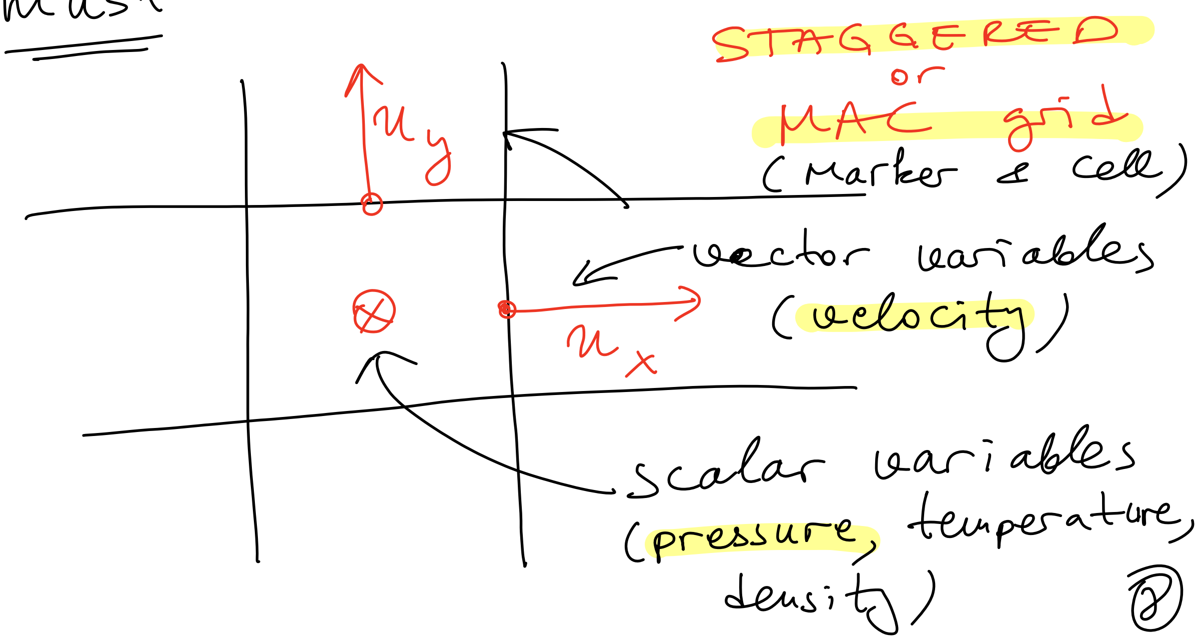
$\vec{u}_\tau = \vec{n} - \vec{u}\cdot\vec{n} = 0$ (Dirichlet)

② Specified traction

$\vec{\tau}\cdot\overleftrightarrow{\sigma}\cdot\vec{n} = \eta\left[\frac{\partial \vec{u}_\tau}{\partial n} + \frac{\partial u_n}{\partial \vec{\tau}}\right]$

# Numerical Method: MAC

What I will describe can be viewed as a finite difference but also (discontinuous Galerkin) finite element method. It is the simplest 2nd order method but only works for regular grid. I will illustrate in 2D but also works in 3D. Domain must be an orthogonal cuboid.




8

Scalars "live" on a standard finite volume / difference grid:

$\nabla\cdot\vec{v}$ or pressure $\rightarrow$ cell centered

but vector fields like velocity have different components "living" on distinct grids

$\vec{\nabla}p$ or velocity $\rightarrow$ face centered

Velocities are on a staggered grid.

Finite difference MAC operators / matrices

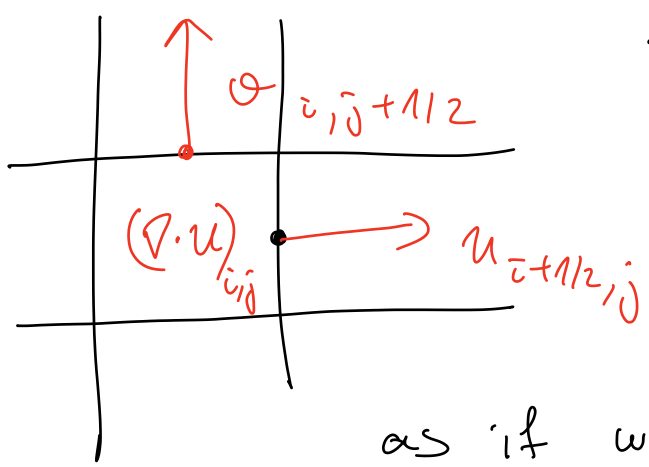
$\vec{u} = (u, v)$

(1) $\nabla\cdot\vec{u} \longleftrightarrow D\vec{u}$

discrete divergence

$$(D\vec{u})_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \approx u_x + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \approx v_y$$

9



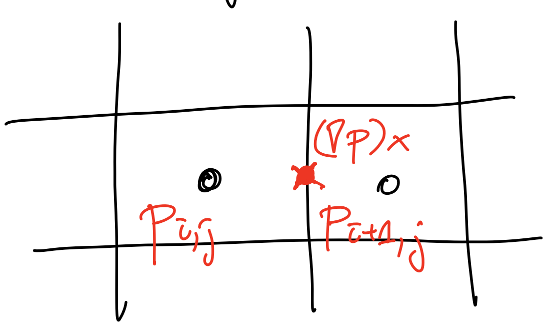

This really is a divergence of a flux through <u>faces</u>, exactly as if we were doing finite volume schemes for the scalar fields

② $\nabla p \longleftrightarrow G p$

Discrete gradient

$$(Gp)^x_{i+1/2,j} = \frac{p_{i+1,j} - p_{i,j}}{\Delta x}$$

$$(Gp)^y_{i,j+1/2} = \frac{p_{i,j+1} - p_{i,j}}{\Delta y}$$




$D = -G^T$ (also $\ast$)

adjoint

$\nabla\cdot = -\nabla^\ast$

③ Scalar / vector Laplacian is standard 5 pt Laplacian, e.g.

$$\left(L_c P\right)_{i,j} = \frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{\Delta x^2} + \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{\Delta y^2}$$

Important: $L_c = DG$
$= -DD^* \preceq 0$

Similarly

$$\left(L^x u\right)_{i+1/2,j} = \frac{u_{i+3/2,j} - 2u_{i+1/2,j} + u_{i-1/2,j}}{\Delta x^2} + \frac{u_{i+1/2,j+1} - 2u_{i+1/2,j} + u_{i+1/2,j-1}}{\Delta y^2}$$

With these operators we can write a discrete <u>NS</u> equation

$$\begin{cases} \rho \dfrac{d\vec{u}}{dt} + G p = \eta \begin{pmatrix} L_x u \\ L_y v \end{pmatrix} - \vec{N}(\vec{u}) \\ D\vec{u} = 0 \end{cases}$$

where $\vec{N}(\vec{u}) \Longleftrightarrow \vec{u} \cdot \vec{\nabla} \vec{u}$

(DAE system of index 2)

<u>Things we need</u>:

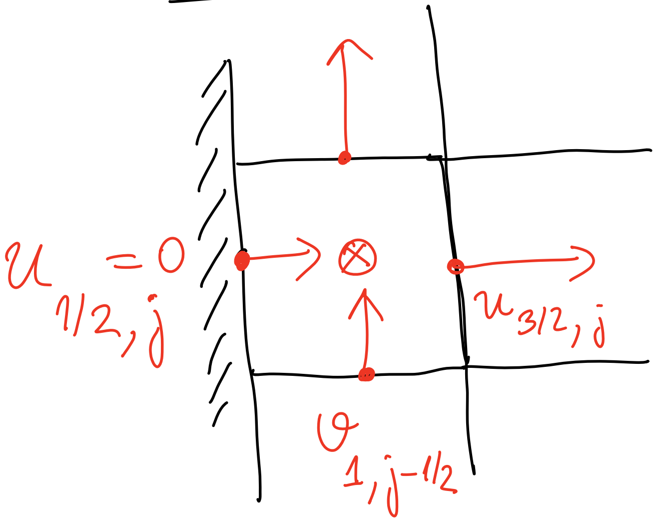
a) Temporal integrator

b) How to discretize $\vec{N}(\vec{u})$

c) Boundary conditions

The "easiest" of these is BCs, <u>especially</u> for Dirichlet BCs on $\vec{u}$ (e.g. no slip / no penetration)

# No-slip BCs



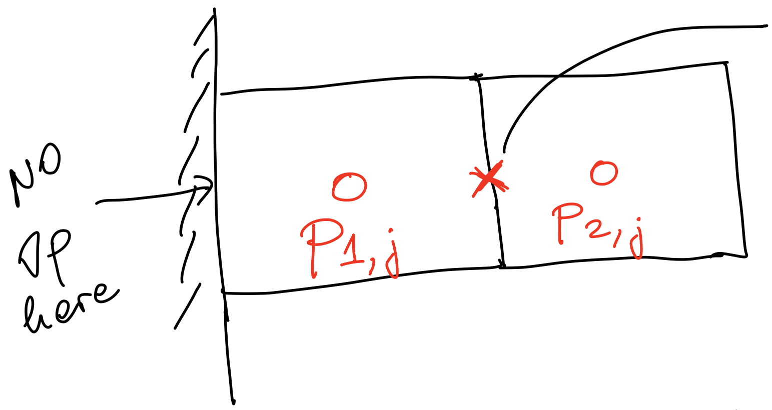

Discrete Divergence is easy to write:

zero (no slip)

$$(D\vec{u})_{1,j} = \frac{u_{3/2,j} - \cancel{u_{1/2,j}}}{\Delta x} + \frac{v_{1,j+1/2} - v_{1,j-1/2}}{\Delta y}$$

Since normal velocity is given (zero for no-slip), the discrete divergence $D$ maps from velocity on interior faces to cell centers.

The discrete gradient $G = -D^*$ therefore maps from cell centers to interior faces.




$$(Gp)_{3/2,j} = \frac{p_{2,j} - p_{1,j}}{\Delta x}$$

We do not need ghost cells to define either D or G – this is the real beauty of the staggered grid and why it is capturing the physics of incompressible flow.

$$L_c = D\,G$$

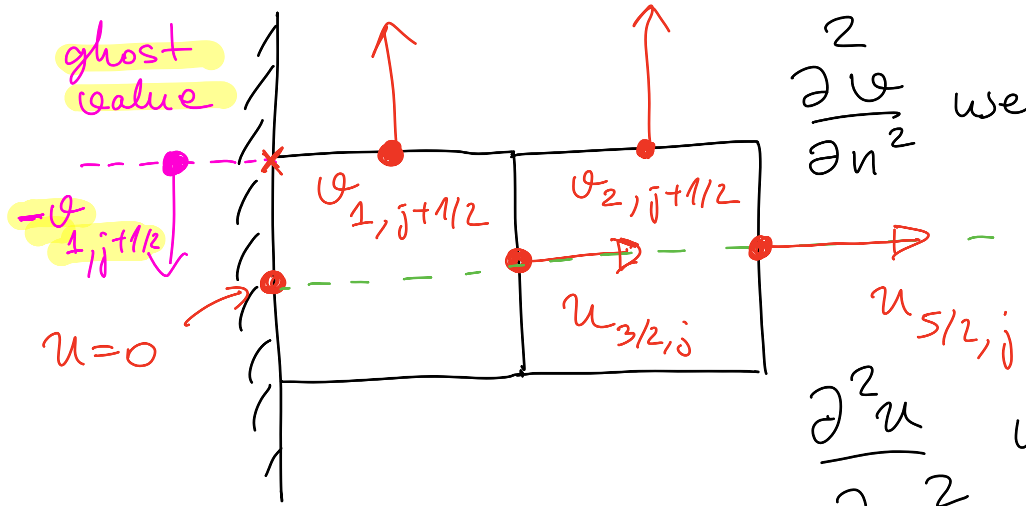
looks like a discrete Laplacian with Neumann BCs for pressure

$$\frac{\partial p}{\partial n} = 0 \quad \text{at impermeable boundary}$$

But this is misleading! Viscosity couples pressure and velocity at boundaries. The MAC grid shows that no BCs are needed for the pressure at no-slip walls.

For the discrete viscous Laplacian, we can reuse the things we did in 1D for advection-diffusion.

The idea is that perpendicular velocity is on a finite-difference (non-staggered) grid, but parallel velocity is on a finite-volume (staggered) grid.




$\frac{\partial^2 v}{\partial n^2}$ uses ghost cells

$\frac{\partial^2 u}{\partial n^2}$ uses BCs directly

This is done in the same way as if the $x$ direction were the only dimension.

Corners are not that hard but they can lead to singularities (driven lid cavity test)

Stress-based BCs can also be handled but they are less natural than no slip – see article by Boyce Griffith linked on course webpage

# Advection

First the "easy" case. Often the velocity advects other quantities like concentration or temperature, assumed here to be <u>scalar</u> (even if multi-component) fields:

$$\partial_t \underbrace{c + u \cdot \nabla c}_{D_t c} = \mu \nabla^2 c$$

$u \cdot \nabla c \equiv \nabla \cdot (u c)$ since $\nabla \cdot u = 0$

This is exactly the finite volume setup we studied in 1D for advection-diffusion equations:

$u$ is a face-centered velocity and $c$ is a cell-centered conserved quantity

One can generalize 2nd order Godunov methods to 2D/3D (good final project)

But what about $u \cdot \nabla u$ ?

One <u>cannot</u> do the <u>extrapolation to midpoint in time</u> for this term due to the staggered grid (it has been done for cell-centered velocities but this has other issues like BCs).

So we will do <u>MOL</u>, with second order in space <u>finite-difference</u> discretization.

This is not great for high Reynolds number flows, but really is that I know nothing of. So lets do <u>FD MOL</u> 2nd order:

$$\vec{u} \cdot \nabla \vec{u} = \begin{bmatrix} (u^2)_x + (uv)_y \\ \hline (uv)_x + (v^2)_y \end{bmatrix} \begin{array}{l} \leftarrow u \\ \\ \leftarrow v \end{array}$$

Idea: Place $u^2$ and $v^2$ on <u>cell centers</u> and $uv$ on grid <u>corners</u> (In 3D, there are both corners/nodes & edges)


$$\left[(u^2)_x\right]_{i+1/2,j} = \frac{(u^2)_{i+1,j} - (u^2)_{i,j}}{\Delta x}$$

$$\left[(uv)_y\right]_{i+1/2,j} = \frac{(uv)^{(y)}_{i+1/2,j+1/2} - (uv)^{(y)}_{i+1/2,j-1/2}}{\Delta y}$$

To improve stability, we upwind to define $u^2$, $v^2$, and $uv$:

$$\left[u^2\right]_{i,j} = \overline{u}_{i,j} \cdot \underbrace{u_{i-1/2,j}}_{\text{upwind}}$$

if $\overline{u}_{i,j} \geq 0$

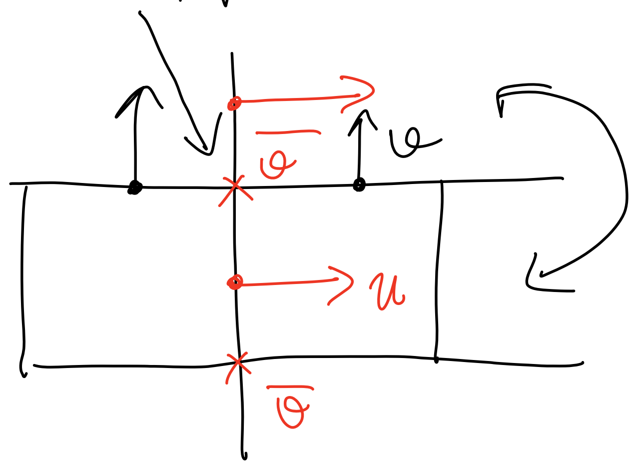
$$\overline{u}_{i,j} = \frac{u_{i+1/2,j} + u_{i-1/2,j}}{2}$$

The notion here is to upwind in the direction of the derivative:

$$(uv)^y_{i+1/2,\,j+1/2} = \overline{v}_{i+1/2,\,j+1/2} \cdot \underbrace{u_{i+1/2,\,j}}_{\text{upwind}}$$

if $\overline{v}_{i+1/2,\,j+1/2} \geq 0$

$$\overline{v}_{i+1/2,\,j+1/2} = \frac{v_{i+1,\,j+1/2} + v_{i,\,j+1/2}}{2}$$




choose upwind value

For low Re numbers one may prefer centered discretizations, which are non dissipative but are dispersive.

One can combine in code:

$$\left(u^2\right)_{i,j} = \underbrace{\frac{u_{i+1/2,j} + u_{i-1/2,j}}{2}}_{\bar{u}_{i,j}} \left( \frac{1 \mp \gamma}{2} u_{i+1/2,j} + \frac{1 \pm \gamma}{2} u_{i-1/2,j} \right)$$

$$0 \leq \gamma \leq 1$$

Choose + sign if $\bar{u}_{i,j} \geq 0$

$\gamma = 0$ for centered

$\gamma = 1$ for upwinding

Adaptive $\gamma$ would be a "poor person's" limiter.

There are better ways but none are very natural on the staggered grid.

# Time Stepping

The last piece is how to integrate in time. The standard/old approach has been to use projection methods, but I suggest a better alternative that is equally efficient (I learned from Boyce Griffith).

We want:

$$\begin{cases} \rho \dfrac{u^{n+1} - u^{n}}{\Delta t} + \nabla p^{n+1/2} = -\left[\rho u \cdot \nabla u\right]^{n+1/2} \\ \text{Crank-Nicolson} \longrightarrow + \eta \nabla^2 \left(\dfrac{u^{n} + u^{n+1}}{2}\right) \\ \nabla \cdot u^{n+1} = 0 \end{cases}$$

This is a so-called saddle-point linear system:

$$\left[\begin{array}{c|c} \frac{\rho}{\Delta t} - \frac{\eta}{2} L_v & G \\ \hline -D & \emptyset \end{array}\right] \left[\begin{array}{c} u^{n+1} \\ \hline p^{n+1/2} \end{array}\right] = \left[\begin{array}{c} \left(\frac{\rho}{\Delta t} + \frac{\eta}{2} L_v\right) u^n - \left[\rho u \cdot \nabla u\right]^{n+1/2} \\ \hline \emptyset \end{array}\right]$$

saddle-point (indefinite) system

For Stokes flow, $Re \to 0$, use L-stable backward Euler:

$$\left[\begin{array}{c|c} -\eta L_v & G \\ \hline -D & \emptyset \end{array}\right] \left[\begin{array}{c} u^{n+1} \\ \hline p^{n+1} \end{array}\right] = \left[\begin{array}{c} rhs \\ \hline \emptyset \end{array}\right]$$

How do we solve this saddle-point linear system?

(Harder than solving a Poisson equation for pressure but similar)

Use GMRES with a smart preconditioner (see webpage)

(MULTIGRID method, more on this once we cover FEM)

One can prove (mostly in FEM literature) that the # of GMRES iterations is roughly independent of grid size if the preconditioner is chosen smartly.

For high Re numbers one can approximate the GMRES solution to second-order accuracy using a

Projection Method

① Time lag pressure and solve for $u^{n+1}$ only:

$$\begin{cases} \rho \dfrac{u^{n+1,*} - u^n}{\Delta t} + G p^{n-1/2} = \eta L_\upsilon\left(\dfrac{u^n + u^{n+1,*}}{2}\right) - \left[\rho u \cdot \nabla u\right]^{n+1/2} \\ u^{n+1,*}_{\partial\Omega} = 0 \quad \text{(no slip)} \end{cases}$$

This is just advection-diffusion for velocity

(2) Solve Poisson equation for pressure:

$$L_c \varphi^{n+1} = (DG)\varphi^{n+1} = \frac{D u^{n+1,*}}{\Delta t}$$

Recall: No BCs required on staggered grid

(3) Project velocity onto space of discretely divergence-free fields:

$$u^{n+1} = u^{n+1,*} - \Delta t \, G \varphi^{n+1}$$

(4) Correct pressure (key!)

$$p^{n+1/2} = p^{n-1/2} + \varphi^{n+1} - \frac{\eta \Delta t}{2\rho}\left(L_c \varphi^{n+1}\right)$$

Notes:

1) For periodic domain, this method is equivalent to the full GMRES solve approach

2) For non-periodic domains the projection method is a great preconditioner for the GMRES solver (use inexact sub-solvers), see paper by Boyce Griffith.

Only missing piece is how to get 2nd order (midpoint) estimate for $[u \cdot \nabla u]^{n+1/2}$.

Let's show one approach on a relevant problem.

<u>Boussinesq</u> approximation:

$$\begin{cases} \rho \partial_t u + \nabla \Pi + \rho u \cdot \nabla u = \\ \eta \nabla^2 v + \vec{g} c \\ \partial_t c + u \cdot \nabla c = \mu \nabla^2 c \end{cases}$$

gravity / buoyancy

concentration / temperature / salinity

We have two options for the advected scalar $c$:

1) Discretize $u \cdot \nabla c$ at a specific point in time, as in MOL schemes (e.g. something like third order upwind biased in 2d)

2) Solve the advection-diffusion equation using a second-order high-resolution Godunov method like Fromm + limiters + diffusion as you did in homework

Notation:

$$\nabla \cdot \mathbf{F}^{(adv)} = \text{SpaceTime}\left(c^n, u, s, \Delta t\right)$$

computes advective fluxes over one timestep of duration $\Delta t$:

$$\partial_t c + \underset{\substack{\uparrow \\ \text{kept fixed} \\ \text{in time}}}{u} \cdot \nabla c = \underset{\substack{\uparrow \\ \text{source}}}{s}$$

$$\underbrace{\left[u \cdot \nabla c\right]^{n+1/2}}_{\text{accurate to } O(\Delta t)} \equiv \nabla \cdot F^{(adv)}$$

Recall that this works even if $D = 0$ and is only limited in stability by advective CFL.

By contrast, an RK2 MOL approach is unstable without sufficient diffusion.

# Algorithm: Boussinesq predictor-corrector

1. Solve using GMRES for $u^{n+1,*}$ and $p^{n+1/2,*}$ (predictor):

$$\begin{cases} \rho \dfrac{u^{n+1,*} - u^{n}}{\Delta t} + \nabla p^{n+1/2,*} = -\rho \left[u \cdot \nabla u\right]^{n} \\ \quad + \eta \nabla^2 \left( \dfrac{u^{n} + u^{n+1,*}}{2} \right) + c^{n} \vec{g} \\ \nabla \cdot u^{n+1,*} = 0 \end{cases}$$

and define $u^{n+1/2,*} = \dfrac{u^{n} + u^{n+1,*}}{2}$

2. Update concentration

$$\frac{c^{n+1} - c^{n}}{\Delta t} = \mu \nabla^2 \left( \frac{c^{n} + c^{n+1}}{2} \right) - \begin{cases} \left[u^{n+1/2,*} \cdot \nabla c\right]^{n} & (\text{MOL approach}) \\ \text{SpaceTime}\left(c^{n}, u^{n+1/2,*}, D\nabla^2 c^{n}, \Delta t\right) \end{cases}$$

(3) Solve using GMRES for $u^{n+1}$ and $p^{n+1/2}$ (corrector)

$$\begin{cases} \rho \dfrac{u^{n+1} - u^{n}}{\Delta t} + \nabla p^{n+1/2} = -\rho \left[u \cdot \nabla u\right]^{n+\frac{1}{2},*} \\ \quad + \eta \nabla^2 \left( \dfrac{u^{n} + u^{n+1}}{2} \right) + \left( \dfrac{c^{n} + c^{n+1}}{2} \right) g \\ \nabla \cdot u^{n+1} = 0 \end{cases}$$

This is a <u>one-step</u> RK2 type method but requires <u>two</u> Stokes solves per step, which is the most expensive step by far. We can do cheaper by using Adams–Bashforth for the $u \cdot \nabla u$ term:

# Algorithm Boussinesq AB2

① Predict concentration:

$$\frac{c^{n+1,*}-c^{n}}{\Delta t} = \mu \nabla^{2}\left(\frac{c^{n}+c^{n+1,*}}{2}\right) - \begin{cases} [u^{n}\cdot\nabla c]^{n} \text{ (MOL approach)} \\ \text{SpaceTime}(c^{n}, u^{n}, D\nabla^{2}c^{n}, \Delta t) \end{cases}$$

② Solve using GMRES

$$\begin{cases} \rho\dfrac{u^{n+1}-u^{n}}{\Delta t} + \nabla p^{n+1/2} = \left(\dfrac{c^{n}+c^{n+1,*}}{2}\right) g \\ + \eta \nabla^{2}\left(\dfrac{u^{n}+u^{n+1}}{2}\right) + \\ \left(\underbrace{\dfrac{3}{2}[u\cdot\nabla u]^{n} - \dfrac{1}{2}[u\cdot\nabla u]^{n-1}}_{AB2}\right) \end{cases}$$

(3) Define $u^{n+1/2} = \dfrac{u^n + u^{n+1}}{2}$ and

correct concentration:

$$\frac{c^{n+1} - c^n}{\Delta t} = \mu \nabla^2 \left( \frac{c^n + c^{n+1}}{2} \right) - \begin{cases} \left[ u^{n+1/2} \cdot \nabla c \right]^n & (\text{MOL approach}) \\ \text{SpaceTime}\left( c^n, u^{n+1/2}, D \nabla^2 c^n, \Delta t \right) \end{cases}$$