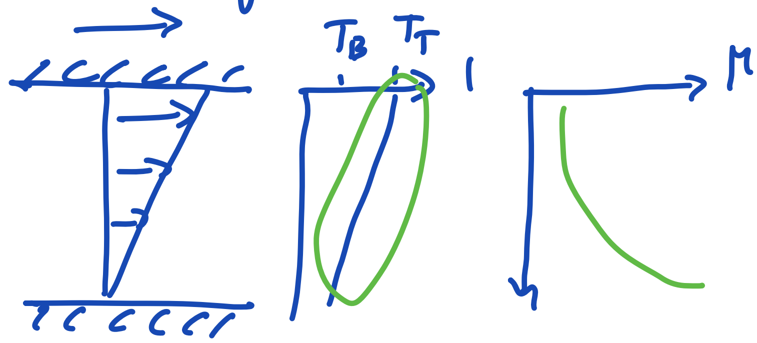
Lecture 25: Discrete Stokes with variable viscosity

Logistics: - HW9 due Th

- may have to cancel class Thursday

Last time: Couette flow with variable viscosity



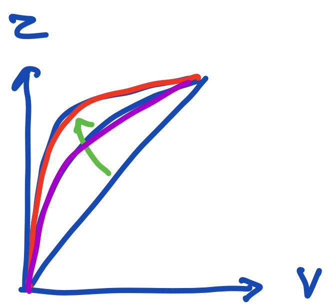
• Arrhenius dependence

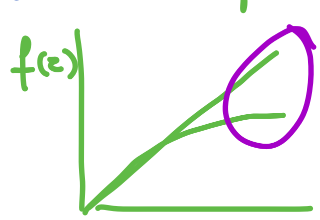
$$\mu = \mu_0 \exp\left(\frac{E_A}{RT}\right)$$

• Reduces to 1D problem

$$-\frac{\partial}{\partial z}\left[\mu(T(z))\frac{\partial v_x}{\partial z}\right] + \frac{\partial \pi}{\partial z} = 0$$

0 assumption





Today: - Numerical discretization with variable viscosity

Stokes with variable viscosity

Governing equs:

lin. mom.: $-\nabla\cdot\left[\mu(T)\left(\nabla\underline{v} + \nabla^T\underline{v}\right)\right] + \nabla\pi = 0$

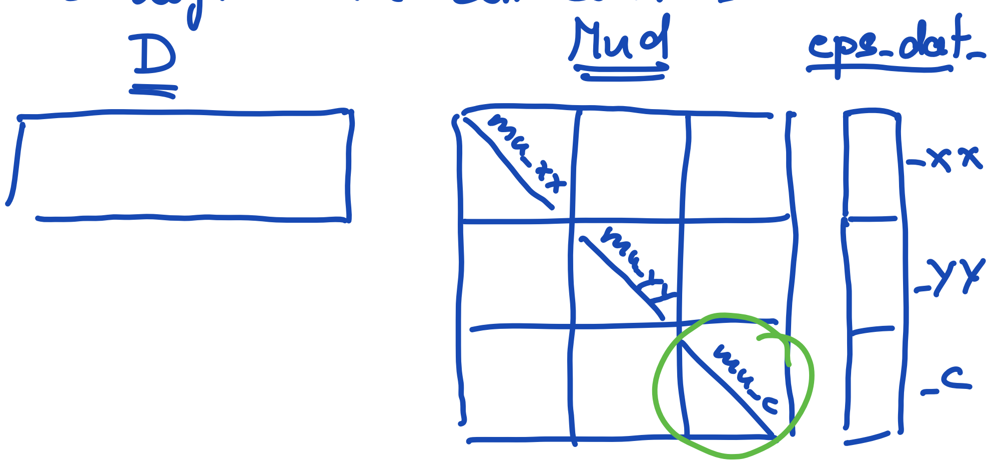
mass: $\nabla\cdot\underline{v} = 0$

Discrete system

$$-\underline{\underline{D}} * \underline{\underline{Mud}} * \underbrace{\underline{\underline{Edot}} * \underline{v}}_{epsdat} + \underline{\underline{Gp}} * \underline{p} = 0$$

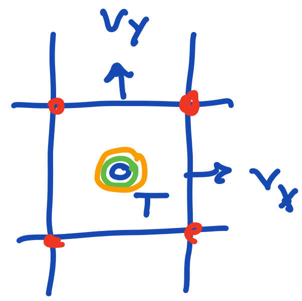
$$\underline{\underline{Dp}} * \underline{v} = 0$$

$\underline{\underline{Mud}}$ is diagonal matrix similar to $\underline{\underline{Kd}}$ that contains the appropriate average of $\mu(T)$, since T is defined in cell centers



$\underline{\underline{Mud}}$ has three diagonal blocks containind the vectors

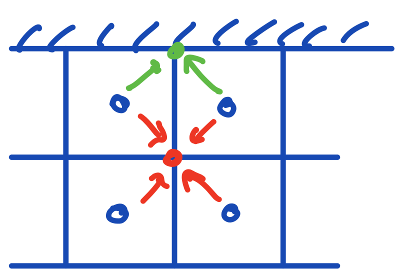
- mu_xx cell center
- mu_yy
- mu_c cell corners



mu_xx and mu_yy no averaging is necessary

mu_c is averaged from cell center to cell corners.

Averaging to cell corners



In interior we average 4 surrounding cell center values

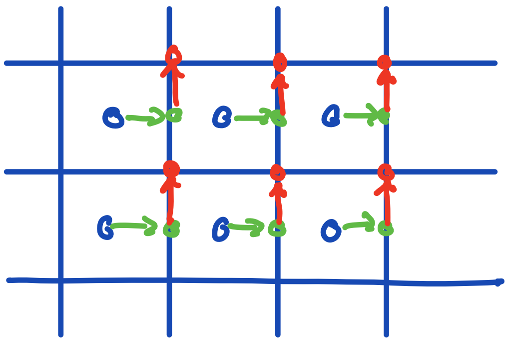
On boundary we average two closest cell center values

need a new averaging matrix M_c (c = corner)

M_c is N_c by N matrix averaging from cell centers to cell corners

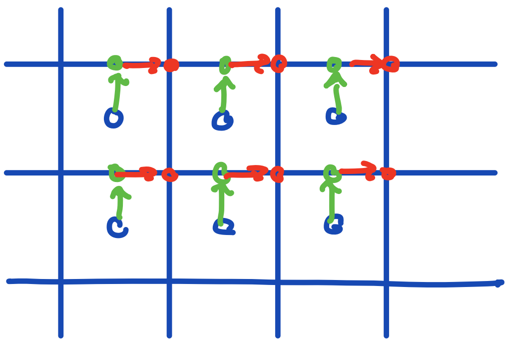
$N_c = (N_x + 1)(N_y + 1)$

This matrix can be built by composing the mean matrices from different grids



1) Centers to x-faces

2) x-faces to corners



1) Centers to y-faces

2) y-faces to corners

Both give the same $\underline{\underline{M_c}}$ matrix

We choose x-faces

In build_stokes_ops.m we have

$\underline{\underline{M_p}}$ mean matrix on primary/pressure grid

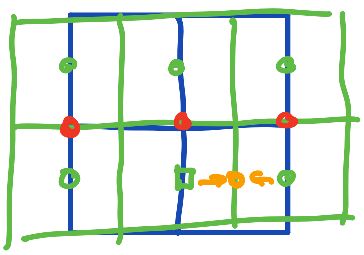
$\underline{\underline{M_x}}$ mean matrix on x-vel. grid

$\underline{\underline{M_p}}\ \underline{T} \rightarrow \underline{T}$ on both x & y faces

$$\underline{\underline{M_p}} = \begin{bmatrix} \underline{\underline{M_{px}}} \\ \underline{\underline{M_{py}}} \end{bmatrix}$$

$$\underline{\underline{M_{px}}} = \underline{\underline{M_p}}[1:\text{Grid.p.Nfx}, :]$$

averages from cell centers to x-faces



We can get corner values from x-face values using

$$\underline{\underline{M_x}} = \begin{bmatrix} \underline{\underline{M_{xx}}} \\ \underline{\underline{M_{xy}}} \end{bmatrix}$$

$\underline{\underline{M_{xy}}}$ averages from x-faces (primary grid) to cell corners of primary grid.

$$\underline{T_c} = \underline{\underline{M_{xy}}}\ \underline{T_x} = \underline{\underline{M_{xy}}}\ \underline{\underline{M_{px}}}\ \underline{T} = \underline{\underline{M_c}}\ \underline{T}$$

$$\boxed{\underline{\underline{M_c}} = \underline{\underline{M_{xy}}}\ \underline{\underline{M_{px}}}}$$

$N_c \cdot N \qquad N_c \cdot Nfx \quad Nfx \cdot N$

$\underline{\underline{Mc}}$ calculates values on corners from values in the center!

Two options:

1) Evaluate first then average

- $\underline{\text{mu\_cen}} = \text{mu}(\underline{T})$ viscosity in cell center
- $\underline{\underline{Mud}} = \text{comp\_mean}(\underline{\text{mu\_cen}}, \underline{\underline{Mc}}, \pm 1, \ldots)$

$\Rightarrow$ similar to how we treat K

2) Average first then evaluate

- $\underline{\underline{Tc}} = \text{comp\_mean}(\underline{T}, \underline{\underline{Mc}}, 1, \ldots)$

  but this blows up if we evaluate $\text{mu}(\underline{\underline{Tc}})$ beause of 0's.

  Instead: $\underline{\underline{Tc}} = \underline{\underline{Mc}}\,\underline{T}$

  this is a vector

- $\underline{\underline{Mud}} = \text{spdiags}(\underline{Tc}, 0, Nc, N_c)$

$\Rightarrow$ apparently second option is best.

Full Mud matrix

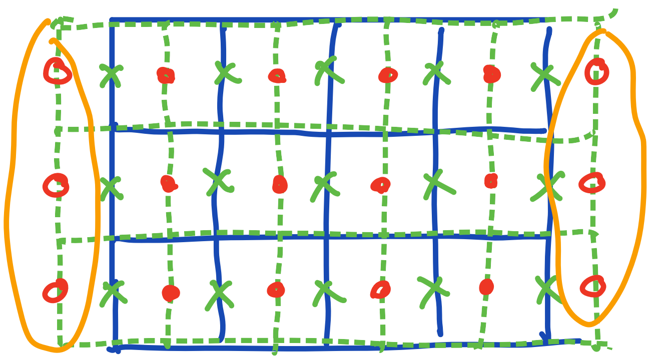
Me is only one of the three blocks in Mud

Ms

$$\begin{bmatrix} I_p & & \\ & I_p & \\ & & M_e \end{bmatrix}$$

But there is a problem, because it turns out eps_dot_xx and eps_dot_yx are not length N.

Consider our standard grid



$N_x = 4 \quad N_y = 3$

eps_dot_xx = Gxx · vx

Additional entries on xmin & xmax bnd of x-velocity grid

$\Rightarrow (N_x+2)\, N_y = 18$

These additional entries are typically eliminated by dirichlet BC's.

Need to come up with an "identity matrix" that copies closest T's into these extra cells.