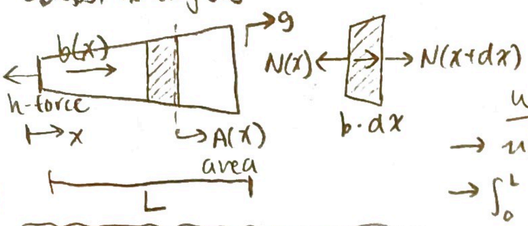
ME C180 Midterm 1 Cheat sheet

mt1

Stress Analysis: Static Bodies



Strong Form: $\boxed{\frac{d}{dx}\left(AE\frac{du}{dx}\right)+b(x)=0}$ $0\le x\le L$

Boundary Conditions: $AE\frac{du}{dx}\Big|_{x=0}=h$ // natural BC

$u(L)=g$ // essential BC

WEAK FORM:

$\rightarrow v(x)\frac{d}{dx}\left(AE\frac{du}{dx}\right)+v(x)b(x)=0$ // multiply both sides by $v(x)$

$\rightarrow \int_0^L$ " " $+\int_0^L$ " " $=0$

$\rightarrow \int_0^L\left(\frac{d}{dx}\left[v(x)AE\frac{du}{dx}\right]-\frac{dv(x)}{dx}AE\frac{du}{dx}\right)dx+\int_0^L v(x)b(x)dx$ // product Rule.

use fundamental Thm. of calculus

natural BCs! $\rightarrow v AE\frac{du}{dx}\Big|_L - vAE\frac{du}{dx}\Big|_0 - \int_0^L \cdots dx = 0$

essential BC

$\boxed{\int_0^L \frac{dv}{dx}AE\frac{du}{dx}\,dx = \left(vAE\frac{du}{dx}\Big|_L\right)-h\,v(0)+\int_0^L v(x)b(x)\,dx}$ (W)

$\boxed{\text{Find } u(x)\in \mathcal{S}=\{f(x)\mid f(L)=g\} \text{ such that } \uparrow \text{ holds true}}$ (WP)

Remarks

- $v(x)$ should be set $=0$ @ essential BC to be considered "admissable"
-

MINIM. FORM (Take heat Cond. Example).

$\int_0^L\left(\frac{dv}{dx}KA\frac{dT}{dx}\right)dx = -v(L)h - vKA\frac{dT}{dx}\Big|_0+\int_0^L v(x)Q(x)dx$. (W)

$\rightarrow$ let $v(0)=0$ (admissibility) $\Rightarrow \underbrace{\int_0^L\left(\frac{dv}{dx}KA\frac{dT}{dx}\right)dx}_{\triangleq\, a(T,v)}=\underbrace{-v(L)h+\int_0^L vQdx}_{\triangleq\, \ell(v)}$

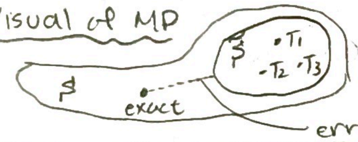
$\boxed{a(T,v)=\ell(v) \text{ for all } v\in\mathcal{V}}$
(new weak-Form intermediate)

$\boxed{I(T)\triangleq\frac{1}{2}a(T,T)-\ell(T) \xrightarrow[\text{CLAIM}]{} \text{Minimizing } I(T) \text{ for all } T\in\mathcal{S} \text{ is equivalent to the (WP)}}$

(MP ⇒ WF) (WF ⇒ MP): Vanberg's Theorem (not taught)

Remarks (useful for proving CLAIM)
- $a(\cdot,\cdot)$ is bilinear! (linear in each elm)
(MP) - $\ell(\cdot)$ is linear!

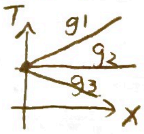
Visual of MP



$I(T_1)=n_1$
$I(T_2)=n_2$ } find min value
$I(T_3)=n_3$
⋮

$\{T_1,T_2,T_3\}\in\mathcal{S}$ where $T(x)=g+cx$

Notice: $T\in\tilde{\mathcal{S}}=\{f(x)\mid f(x)=g+cx\}$ satisfies the essential BC.



Stiffness & force ($\underline{K}$ and $\underline{f}$)

let $\mathcal{S}^N\subset\mathcal{S}$ $\mathcal{S}^N=\{f(x)\mid f(x)=\psi_g(x)+\sum_{j=1}^N a_j\psi_j(x)\}$

let $\mathcal{V}^N\subset\mathcal{V}$ $\mathcal{V}^N=\{f(x)\mid f(x)=\sum_{j=1}^N c_j\psi_j(x)\}$

★ $\psi_g(x)\rightarrow$ known & satisfies EBC
★ $\psi_i(x)\rightarrow$ $=0$ for all i @ EBC.
c_j and a_j are just coefficients.

(Apprx WP): $\boxed{\text{Find } u^N\in\mathcal{S}^N \text{ so } a(u^N,v^N)=\ell(v^N) \text{ for all } v^N\in\mathcal{V}^N}$

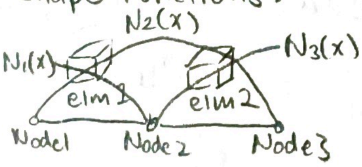
Let $\psi_g=0$ (EBC satisfied) $\psi_i=x^i$ (0 when $x=0$, otherwise forms polynomial)

$\mathcal{S}^N=\{f^N\mid f^N=\sum_{j=1}^N a_jx^j\}$ $\mathcal{V}^N=\{f^N\mid f^N=\sum_{k=1}^N c_kx^k\}$

$\boxed{_{N\times N}K_{ij}=a(\psi_i,\psi_j)}=\frac{ij}{i+j-1}\pi^{i+j-1}\ldots$

★ $\boxed{_{N\times 1}f_i=\ell(\psi_i)}=\pi^i\cdot 1$

Shape Functions:



Designed to Satisfy Kronecker Property.

$N_I(x_J)=\delta_{IJ}\begin{cases}1 & \text{when } I=J\\ 0 & \text{when } I\ne J\end{cases}$

$\tilde{u}=\underline{N}\cdot\underline{u}=\sum u_jN_j(x)=[N_1(x)\ldots N_j(x)]\,\underline{u}$

$\tilde{u}'=\underline{N}'\underline{u}=\sum u_jN_j'(x)=[N_1'(x)\ldots N_j'(x)]\,\underline{u}$

$\tilde{u}=\underline{N}\cdot\underline{u}$ $\underline{N}\triangleq[N_1(x),\ldots,N_j(x)]$

$\tilde{u}'=\underline{N}'\cdot\underline{u}$ $\underline{B}\triangleq[N_1'(x),\ldots N_j'(x)]$

Application to Static Body Ex

$\int_0^L v'AEu'dx=\int_0^L bvdx-hv(0)=0$ (WD)

$\Rightarrow \int_0^L \underline{v}^T\underline{B}^TAE\underline{B}\underline{u}\,dx=\int_0^L\underline{v}^T\underline{N}^Tbdx-h\underline{v}^T\underline{N}^T(0)=0$ // pull $\underline{v}^T$ out

$\Rightarrow \underbrace{\int_0^L \underline{B}^TAE\underline{B}\,dx}\ \underline{u}=\underbrace{\int_0^L\underline{N}^Tbd x-h\underline{N}^T(0)}$

$\Rightarrow \boxed{\underline{K}\quad\underline{u}\quad=\quad\underline{f}}$ final prob.

$\underline{K}\triangleq\begin{bmatrix}\int_0^L B_1AEB_1dx & \cdots & \int_0^L B_1AEB_Ndx\\ \vdots & \ddots & \\ \int_0^L B_NAEB_1dx & \cdots & \int_0^L B_NAEB_Ndx\end{bmatrix}$

Dealing with unknowns in 2 different vectors
Matrix partitioning:

$\begin{bmatrix}\underset{6\times6}{R} & \underset{6\times1}{A_2}\\ \underset{1\times6}{A_2^T} & \underset{1\times1}{A_1}\end{bmatrix}\begin{bmatrix}\underline{u_f}\\ \underline{u_d}\end{bmatrix}=\begin{bmatrix}\underline{f_f}\\ \underline{f_d}\end{bmatrix}$

Local → Global

$k_{ij}^e = \int_{le} B_i^e AEB_j^e \, dx$

$f_i^e = \int_{le} N_i b \, dx - h N_i(0) + f_i N_i(L)$

Linear elem $k^e = 2\times2$ — $e \propto h^2$

Quadratic elem — $e \propto h^3$

$\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} = N \quad k^e = 3\times3$!

h is the size of the element!

Location Matrix.

$$LM = \begin{bmatrix} 1 & 2 & 3 & 4 & 6 & 5 \\ 2 & 3 & 4 & 5 & 7 & 6 \end{bmatrix} \begin{matrix} \to \text{loc. Node 1} \\ \to \text{loc. Node 2.} \end{matrix}$$

} Goes up to however N many nodes per elem.

elem. # 1 2 3 4 5 6

Example: $k_{11}^{e=2} \to K_{22}$

$k_{12}^{e=5} \to K_{67}$

$k_{22}^{e=5} \to K_{77}$

mt1

Quadrature & Isoparametric mappings

Trapezoidal Quadrature : $\int_{-1}^{1} f(x)dx \approx 1\cdot f(-1) + 1\cdot f(1)$ — error $\sim h^3, f''(c)$

Simpson's Rule : $\int_{-1}^{1} f(x)dx \approx \frac{1}{3}f(-1) + \frac{4}{3}f(0) + \frac{1}{3}f(1)$ — error $\sim h^5, f^{4}(c)$

Gauss Quadrature : $\int_{-1}^{1} f(x)dx \approx f(-\frac{1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}})$ — error $\sim h^{2nq+1}$

"parent domain" ⟺ "physical domain"

-1 0 1 → c x_1^e x_2^e → x

x(c)   c(x)

[A] Isoparametric Map

$x(c) = x_1^e N_1^e(c) + x_2^e N_2^e(c)$

physical → Parent

[B] Inverse

$N_i^e(x) = N_i^e(c)\cdot c(x)$

Parent → physical

$f_i^e = \int_{le} b(x) N_i^e(x) dx = \int_{-1}^{1} [b(x)\, x(c) N_i^e(c)] \underbrace{\frac{dx}{dc}}_{\text{Jacobian}} dc$

$k_{ij}^e = \int_{le} \frac{dN_i^e(x)}{dx} AE \frac{dN_j^e(x)}{dx} = \int_{-1}^{1} \Big(\underbrace{\frac{dN_i^e(c)}{dc} AE \frac{dN_j^e(c)}{dc}}_{\text{"Jacobian"}} \Big) \underbrace{\frac{1}{dx/dc}}_{\text{Problem-specific}} dc$

Heat-Conduction 3-D steady-State:

$\frac{d}{dt}\int_R e\, du = \int_R Q\, dv - \int_{\partial R} q\cdot n \, da$

what's inside = what is gen − what leaves

Div Thm

$\int_R \underbrace{\dot{e} - Q - \text{div}(q) du}_{=0!} = 0$

Aside: Divg. Thm. $\iiint \Leftrightarrow \iint$

$\int_R \text{div}(q)\, du = \int_{\partial R} q\cdot n \, da$

$\dot{e} = Q + \text{div}(q) \overset{(S)}{\Rightarrow} q = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \underbrace{\begin{bmatrix} D_{xx} & * & * \\ * & D_{yy} & * \\ * & * & D_{zz} \end{bmatrix}}_{\text{conduction const. tensor}} \begin{bmatrix} \partial T/\partial x \\ \partial T/\partial y \\ \partial T/\partial z \end{bmatrix} \Rightarrow c\frac{\partial T}{\partial t} = K\nabla^2 T + Q$

Types of BC.

1. Prescribed Temp $T(x,t) \to$ ess BC
2. " " Heat flux $h(x,t) \to$ nat BC
3. Convection BC → nat BC
   $q\cdot n = h(T - T_{ref})$

Weak Form Heat Conduction:

Recall: $c\dot{T} = \text{div}(\underline{D}\nabla\underline{T}) + Q$ //gov eq.

BC: on $\partial\Omega_T$ $T = \hat{T}$ (given) Ess BC

on $\partial\Omega_q$ $q^T \hat{n} = h(x)$ (given) Nat BC.

Set up trial space: $S^h = \{T(x) \mid T(x) = \hat{T}(x) \text{ on } \partial\Omega_T\}$

$V^h = \{w(x) \mid w(x) = 0 \text{ on } \partial\Omega_T\}$

$\Rightarrow \underbrace{\int_\Omega \nabla w^T \underline{D} \nabla T \, du}_{} = \underbrace{\int_\Omega wQ\, du - \int_{\partial\Omega q} -h w\, da}_{}$ (WF)

//long and winded proof.

Turn into (WP) $a(w,T) = \ell(w)$

$\underline{K}\underline{T} = \underline{f}$

$\nabla \underline{T}^h = \underline{B}\,\underline{T}$

$\nabla \underline{w}^h = \underline{B}\,\underline{w}$

where $\underline{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \cdots & \frac{\partial N_n}{\partial x} \\ \frac{\partial N_1}{\partial y} & \cdots & \frac{\partial N_n}{\partial y} \\ \frac{\partial N_1}{\partial z} & \cdots & \frac{\partial N_n}{\partial z} \end{bmatrix}_{n\times3}$

"derivative of shape func. matrix"

(WP) $\Rightarrow \underline{w}^T \Big[\underbrace{\int_\Omega \underline{B}^T \underline{D}\, \underline{B}\, du}_{\underline{K}}\, \underline{T} \underbrace{- \int_\Omega \underline{N}^T Q\, du + \int_{\partial\Omega q} \underline{N}^T h\, da}_{\underline{f}} \Big] = 0$

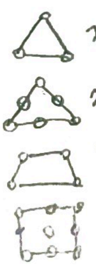
$\boxed{\underline{K}\,\underline{T} = \underline{f}}$

Dimensional Sanity Check.

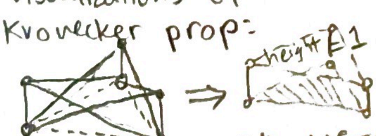
$\underline{B}^T_{n\times3}\; D_{3\times3}\; \underline{B}_{3\times n} = \underline{K}_{n\times n}$

$\underline{N}^T_{n\times1} Q_{1\times1} + \underline{N}^T_{n\times1} h_{1\times1} = \underline{f}_{n\times1}$

Elements for multi-D problems. (NO ISOPARAMETRISM)

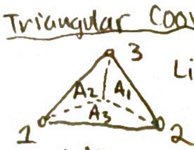


Visualizations of Kronecker prop:



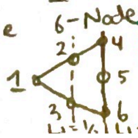
superposition $T_e^h = \underline{N}^e T^e$

Triangular Coordinates



$L_i = \frac{A_i}{2A}$ $N_1^e = L_1$, $N_2^e = L_2$, $N_3^e = L_3$!

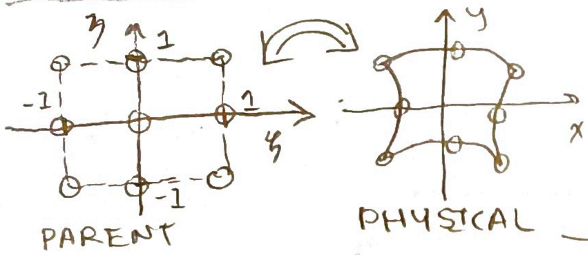
6-Node



$N_1 = \frac{(L_1)(L_1 - 1/2)}{1(1 - 1/2)}$

Clever!

Multi-Dim Isoparametrism



Let $N_i^{iso}(\eta, \xi)$ and $dN_i^{iso}/d(\eta,\xi)$ be EASY one

maps $xy \to \eta\xi$

mt1

$$\boxed{N_i(x,y) \triangleq N_i^{iso}(\xi(x,y), \eta(x,y))}$$

$$\boxed{\underbrace{\underset{\sim}{x}}_{parent} = \sum N_i^{iso}(\xi,\eta)\, \underbrace{\underset{\sim}{x_i}^e}_{physical}}$$

Force vector: $f_i = \int_{\Omega e} N_i(\underset{\sim}{x}) \cdot [\text{Prob. Spec}](\underset{\sim}{x})\, dx$

$$\Rightarrow \int_{-1}^{1}\int_{-1}^{1} N_i^{iso}(\xi,\eta)\cdot[\text{P.S.}]\cdot(\underset{\sim}{x}(\xi,\eta)) \left|\frac{\partial(x,y)}{\partial(\xi,\eta)}\right| d\xi\, d\eta$$

quadrature applied $\Rightarrow \boxed{\sum \underbrace{W_e}_{weight} N_i^{iso}\underbrace{(\bar\xi, \bar\eta)}_{quadr.\ points}[\text{P.S}]\cdot(\underset{\sim}{x}(\bar\xi,\bar\eta))\, \underbrace{J(\bar\xi,\bar\eta)}_{Jacobian}\, d\text{-}d\text{-}}$

Stiffness Matrix K

$\underline{K_e} = \int_{\Omega e} B_e^T[\text{P.S}]B_e\, d\underset{\sim}{x}$ Recall $\underline{B_e} = \begin{bmatrix} \partial N_i^e/\partial x & \cdots & \partial N_n^e/\partial x \\ \partial N_i^e/\partial y & \cdots & \partial N_n^e/\partial y \end{bmatrix}$

$$\boxed{K_e \Rightarrow \int_{-1}^{1}\int_{-1}^{1}\left(\frac{\partial N_A}{\partial \xi} + \frac{\partial N_B}{\partial \xi}\bigg|_{\bar\xi,\bar\eta}\right)[\text{P.S}]\left(\frac{\partial N_A}{\partial \eta} + \frac{\partial N_B}{\partial \eta}\bigg|_{\bar\xi,\bar\eta}\right) J^{-1}\bigg|_{\bar\xi,\bar\eta} d\xi\, d\eta}$$

apply quadr. $\Rightarrow \sum W_e(\text{"}\quad\text{"})[\text{P.S}](\text{"}\quad\text{"})J^{-1}$ don't forg. inverse

Notice: $\begin{bmatrix} \partial N_i/\partial x \\ \partial N_i/\partial y \end{bmatrix} = \underbrace{\begin{bmatrix} \partial\xi/\partial x & \partial\xi/\partial y \\ \partial\eta/\partial x & \partial\eta/\partial y \end{bmatrix}}_{\text{JACOBIAN}} \begin{bmatrix} \partial N_i^{iso}/\partial\xi \\ \partial N_i^{iso}/\partial\eta \end{bmatrix}$

Transient Heat Problems

Recall (s) $\underbrace{\rho c \dot T}_{\text{(B)-term (now nonzero)}} = \underbrace{\text{div}(D\nabla T) + Q}_{\text{(A)-term}}$

From Lab: How to find Jacobian

$$\boxed{\hat J(\varepsilon) = \frac{d\hat x(\varepsilon)}{\partial\varepsilon} = \sum_{i=1}^{p+1} x_i^e \frac{d\hat N_i^e}{d\varepsilon}}$$

(A) $\to \underset{\sim}{w}^T\Big(\boxed{-\int_\Omega B^T \underline{D} \underline{B}\, d\Omega}\ \underset{\sim}{T}$ K

$+\boxed{\int_\Omega \underline{N}^T Q\, d\Omega - \int_{\partial\Omega} \underline{N}^T h\, da}\Big)$ F

(B) $\to \underset{\sim}{w}^T \boxed{\int_\Omega \underline{N}^T \rho c \underline{N}\, d\Omega}\ \dot{\underset{\sim}{T}}$

"mass matrix" M

$$\boxed{\underline{M}\dot{\underset{\sim}{T}}(t) + \underline{K}\underset{\sim}{T}(t) = \underset{\sim}{F}(t)}$$

new term

METHODS for transient

Forward-Euler (explicit): $\underline{M}\left(\frac{T_{n+1} - T_n}{\Delta t}\right) + \underline{K}\underset{\sim}{T}_n = \underset{\sim}{F}_n$ where n is time-step

- used for short-time comp. if $\Delta t \to \infty$ solution will NOT converge

Backward-Euler (implicit): $\underline{M}\left(\frac{T_{n+1} - T_n}{\Delta t}\right) + \underline{K}\underset{\sim}{T}_{n+1} = \underset{\sim}{F}_{n+1}$

- used for long-time comp. No limit on size of Δt.

Convergence requirements

① Consistency → Finite diff eq is generally related to the true model.

True model: $\underset{\sim}{T}_{n+1} = \Psi(\underset{\sim}{T}_n, t_n, t_{n+1}, \underset{\sim}{F}_n, \underset{\sim}{F}_{n+1})$ Ψ is some func.

Finite-Dif.: $\underset{\sim}{T}(t_{n+1}) = \Psi(\underset{\sim}{T}(t_n), t_n, t_{n+1}, \underset{\sim}{F}_n, \underset{\sim}{F}_{n+1}) + e(\Delta t^p)$ e is some error funct. $\boxed{p > 1}$ for consisten.

② Stability: Errors in computation DECAY over time.

Form: $\dot T(t) = \lambda T(t) \Rightarrow \lambda = \lambda_R + i\lambda_I$

$T(t) = e^{\lambda_R t}(\cos(\lambda_I t) + i\sin(\lambda_I t))$ must be < 1 for stability.

Forward-Euler Stability: $\frac{T_{n+1} - T_n}{\Delta t} = \lambda\Delta t\, T_n$

$\Rightarrow T_{n+1} = (1 + \lambda\Delta t)T_n$ $\boxed{T_n = (1+\lambda\Delta t)^n T_0}$

$|(1+\lambda\Delta t)| < 1$

Backward's Euler Stability $\frac{T_{n+1} - T_n}{\Delta t} = \lambda\Delta t\, T_{n+1}$

$\Rightarrow T_{n+1} = \frac{T_n}{1 - \Delta t\lambda} \Rightarrow \boxed{T_n = \left(\frac{1}{1-\Delta t\lambda}\right)^n T_0}$

$\left|\frac{1}{1-\Delta t\lambda}\right| < 1$

Apply to Transient Problem.

$\underline{M}\dot{\underset{\sim}{T}} + \underline{K}\underset{\sim}{T} = \underset{\sim}{0}$ Guess $\underset{\sim}{T} = \underset{\sim}{\Psi}e^{-\lambda t}$

plug in

$(\underbrace{\underline{K} - \lambda\underline{M}}_{\text{generalized eigenvalue}})\underbrace{\underset{\sim}{\Psi}}_{\text{generalized eigenvector}} = \underset{\sim}{0}$

There exist n Ψ_i as e-vecs $\underset{\sim}{\Psi_1} \dots \underset{\sim}{\Psi_n}$

$\underset{\sim}{\Psi_i}^T \underline{M}\, \underset{\sim}{\Psi_j} = 0$

$\underset{\sim}{\Psi_i}^T \underline{M}\, \underset{\sim}{\Psi_j} = 1$

Stability hinges on LARGEST eig-val

$$\boxed{\lambda_{max} \leq \lambda_{max}^e}$$

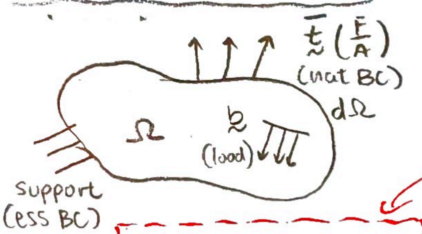
$$\boxed{\dot T^{(j)}(t) + \lambda_j T^{(j)}(t) = 0}$$

λ_i are eig vals of $\underline{M}$

$\underset{\sim}{T}(t) = \sum_{i=1}^{n} T^i(t)\underset{\sim}{\Psi_i}$

since $\Psi_i \dots \Psi_n$ form a basis for $\mathbb{R}^2$

STRESS ANALYSIS



Basis: $\boxed{\text{Cauchy's Law: } \underset{\sim}{t}(\underset{\sim}{x}, \underset{\sim}{n}) = \underset{\sim}{S}(\underset{\sim}{x})\underset{\sim}{n}}$

$\underset{\sim}{S} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \vdots \\ \sigma_{yx} & \sigma_{yy} & \\ \vdots & \cdots & \sigma_{zz} \end{bmatrix}$ stress tensor.

mt2

Force Bal: $\int_\Omega \underset{\sim}{b}\, dv = \int_{\partial\Omega} \underset{\sim}{t}\, dv$

apply Cauchy's law for $\underset{\sim}{t}$ & divergence thm.

$\int_\Omega \left[b + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + \ldots + \frac{\partial \sigma_{zz}}{\partial z} \right] dv = 0$

$\boxed{\tilde{\nabla}^T \underset{\sim}{\sigma} + \underset{\sim}{b} = 0}$

symmetric!

Definitions, symmetry redundancy removed

$\underset{6\times1}{\underset{\sim}{\sigma}} \triangleq \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix}$ $\underset{3\times6}{\tilde{\nabla}^T} \triangleq \begin{bmatrix} \partial/\partial x & 0 & 0 & \partial/\partial y & 0 & \partial/\partial z \\ 0 & \partial/\partial y & 0 & \partial/\partial x & \partial/\partial z & 0 \\ 0 & 0 & \partial/\partial z & 0 & \partial/\partial y & \partial/\partial z \end{bmatrix}$ $\underset{6\times1}{\underset{\sim}{\epsilon}} \triangleq \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yz} \end{bmatrix}$ $\underset{\sim}{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ $\underset{\sim}{D} \triangleq \begin{bmatrix} E & & \\ & E & \\ & & E \\ & & & G \\ & & & & G \\ & & & & & G \end{bmatrix} = \underset{\sim}{D}^T$

$\Rightarrow \tilde{\nabla}^T(\underset{\sim}{D}\underset{\sim}{\epsilon}) + \underset{\sim}{b} = 0$

$\Rightarrow \tilde{\nabla}^T(\underset{\sim}{D}(\tilde{\nabla}\underset{\sim}{u})) + \underset{\sim}{b} = 0$ **(S)**

$(S) \rightarrow (W)$ $\underbrace{\int_{\partial\Omega t} \underset{\sim}{u}^T \bar{\underset{\sim}{t}}\, dA + \int_\Omega \underset{\sim}{u}^T \underset{\sim}{b}\, dv}_{\ell(v)} = \underbrace{\int_\Omega (\tilde{\nabla}\underset{\sim}{u})^T \underset{\sim}{D} \tilde{\nabla}\underset{\sim}{u}\, dv}_{a(u,v)}$ **(W)**

Turn into Linalg Again

$\underset{\sim}{u} = \underset{\sim}{N}\underset{\sim}{d} \Rightarrow \tilde{\nabla}\underset{\sim}{u} = \tilde{\nabla}\underset{\sim}{N}\underset{\sim}{d} = \underset{\sim}{B}\underset{\sim}{d}$

$\underset{\sim}{d} \triangleq \begin{bmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \\ \vdots \end{bmatrix}$ $\underset{\sim}{N} \triangleq \begin{bmatrix} N_1 & 0 & N_2 & 0 & \cdots \\ 0 & N_1 & 0 & N_2 & \end{bmatrix}$ $\underset{3\cdot\text{dof}}{\underset{\sim}{B}} = \tilde{\nabla} \underset{\text{dof}\cdot(\text{ndof})}{\underset{\sim}{N}}$

NODE 1 ⇒ cols 1&2, NODE 2 ⇒ cols 3&4

$= \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \cdots \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \end{bmatrix}$

$(W) \xrightarrow{\text{becomes}} \cancel{\underset{\sim}{u}^T_{\text{mod}}} \left[\underbrace{\int_\Omega \underset{\sim}{B}^T \underset{\sim}{D} \underset{\sim}{B}\, dv}_{\underset{\sim}{K}} \underbrace{\underset{\sim}{d}}_{\underset{\sim}{d}} - \underbrace{\int_\Omega \underset{\sim}{N}^T \underset{\sim}{b}\, dv - \int_{\partial\Omega t} \underset{\sim}{N}^T \bar{\underset{\sim}{t}}\, da}_{\underset{\sim}{F}} \right] = 0$

$\boxed{\underset{\sim}{K}\underset{\sim}{d} = \underset{\sim}{F}}$

$\boxed{\underset{\sim}{k}^e = \int_{\Omega e} \underset{\sim}{B}_e^T \underset{\sim}{D} \underset{\sim}{B}_e\, dv}$

$\boxed{\underset{\sim}{f}^e = \int_{\Omega e} \underset{\sim}{N}_e^T \underset{\sim}{b}\, dv + \int_{\partial\Omega e} \underset{\sim}{N}_e^T \bar{\underset{\sim}{t}}\, dA}$

DYNAMIC PROBLEMS

Static → Dynamic : $\tilde{\nabla}^T\underset{\sim}{\sigma} + \underset{\sim}{b} = 0 \Longrightarrow \tilde{\nabla}^T\underset{\sim}{\sigma} + \underset{\sim}{b} = \rho\ddot{\underset{\sim}{u}}$ new term!

$\tilde{\nabla}^T\underset{\sim}{\sigma} + \underset{\sim}{b} = \rho\ddot{\underset{\sim}{u}}$ **(S)** $\rightarrow \int_\Omega (\tilde{\nabla}\underset{\sim}{w})^T D (\tilde{\nabla}\underset{\sim}{u})\, dv + \int_\Omega \underset{\sim}{w}^T \rho \ddot{\underset{\sim}{u}}\, dv = \int_\Omega \underset{\sim}{w}^T \underset{\sim}{b}\, dv + \int_{\partial\Omega t} \underset{\sim}{w}^T \bar{\underset{\sim}{t}}\, da$ $[N/m^3]$ **(W)**

(W) ⇒ apply $\underset{\sim}{u} = \underset{\sim}{N}\underset{\sim}{d}$ & $\underset{\sim}{w} = \underset{\sim}{N}\underset{\sim}{w}$ $\Rightarrow \int_\Omega \underset{\sim}{B}^T \underset{\sim}{D} \underset{\sim}{B}\, dv\, \underset{\sim}{d} + \int_\Omega \rho \underset{\sim}{N}^T \underset{\sim}{N}\, dv\, \ddot{\underset{\sim}{d}} = \int_\Omega \underset{\sim}{N}^T \underset{\sim}{b}\, dv + \int_{\partial\Omega t} \underset{\sim}{N}^T \bar{\underset{\sim}{t}}\, dA$

$\boxed{\underset{\sim}{K}\underset{\sim}{d} + \underset{\sim}{M}\ddot{\underset{\sim}{d}} = \underset{\sim}{F}}$

Real systems have Damping

$\underset{\sim}{K}\underset{\sim}{d} + \underset{\sim}{M}\ddot{\underset{\sim}{d}} + \underset{\sim}{C}\dot{\underset{\sim}{d}} = \underset{\sim}{F}$

Rayleigh Damping : $\underset{\sim}{C} = (a\underset{\sim}{M} + b\underset{\sim}{K})$
where a & b are coeffs.

Raw Taylor expansions:

$\underset{\sim}{d}_{n+1} = \underset{\sim}{d}_n + \underset{\sim}{v}_n \Delta t + [(1-2\gamma)\underset{\sim}{a}_n + 2\gamma\, \underset{\sim}{a}_{n+1}]$

$\underset{\sim}{v}_{n+1} = \underset{\sim}{v}_n + \Delta t\, [(1-2\beta)\underset{\sim}{a}_n + 2\beta\, \underset{\sim}{a}_{n+1}]$

parameters

Solving via time-stepping method:

NEWARK's method → most common!!

Interpret as $\underset{\sim}{K}\underset{\sim}{d} + \underset{\sim}{M}\underset{\sim}{a} + \underset{\sim}{C}\underset{\sim}{v} = \underset{\sim}{F}$

central-differences | $\gamma = \frac{1}{2}$ $\beta = 0$

① $\underset{\sim}{d}_{n+1} = \underset{\sim}{d}_n + \underset{\sim}{v}_n \Delta t + \frac{\Delta t^2}{2}[\frac{1}{2}\underset{\sim}{a}_{n+1} + \frac{1}{2}\underset{\sim}{a}_n]$
//solvable w/ init values $\underset{\sim}{d}_n, \underset{\sim}{v}_n, \underset{\sim}{a}_n$

② $\underset{\sim}{a}_{n+1} = \underset{\sim}{M}^{-1}(\underset{\sim}{F} - \underset{\sim}{K}\underset{\sim}{d}_{n+1})$
// solvable w/ ①'s solution

③ $\underset{\sim}{v}_{n+1} = \underset{\sim}{v}_n + \Delta t\, [\underset{\sim}{a}_{n+1}]$
//solvable w/ ②'s soln.

Rule of Thumb for centr. diff: $\Delta t \leq h_{min}/c$

Trapezoidal Rule | $\gamma = \frac{1}{2}$ $\beta = \frac{1}{4}$

cannot use this method for determing soln. at once
must iterate & update

MODAL ANALYSIS

forces net to 0 for stability

For modes to exist $\underset{\sim}{M}\ddot{\underset{\sim}{d}} + \underset{\sim}{K}\underset{\sim}{d} = \underset{\sim}{0}$

guess $\underset{\sim}{d} = \underset{\sim}{\psi} e^{i\omega t} \rightarrow \ddot{d} = \omega^2 \underset{\sim}{\psi} e^{i\omega t}$

$\xRightarrow[\text{back}]{\text{plug}} (-\omega^2 \underset{\sim}{M} + \underset{\sim}{K})\underset{\sim}{\psi}\, e^{i\omega t} = \underset{\sim}{0}$

for nontrivial soln, $\det(-\omega^2 \underset{\sim}{M} + \underset{\sim}{K}) = 0$

what comes away:

$\Rightarrow \boxed{\omega_1^2, \omega_2^2, \ldots \omega_n^2}$ ← eigenvalue frequencies

$\Rightarrow \boxed{\underset{\sim}{\psi}_1, \underset{\sim}{\psi}_2, \ldots \underset{\sim}{\psi}_n}$ ← eigenvector modes (shapes)

Properties of ψ_i from $\underset{\sim}{M}, \underset{\sim}{K}$ being symmetric

① $\underset{\sim}{\psi}_i^T \underset{\sim}{M} \underset{\sim}{\psi}_i = \delta_{ij}$ (Kronecker prop)

② $\underset{\sim}{\psi}_i^T \underset{\sim}{K} \underset{\sim}{\psi}_i = \omega_i^2 \delta_{ij}$

③ ω_i^2 is Real $\in \mathbb{R}$

④ $\underset{\sim}{\psi}_1 \ldots \underset{\sim}{\psi}_n$ form a basis for $\mathbb{R}^n$!

thus $\underset{\sim}{d}(t) = \sum_{j=1}^{n} d_j \underset{\sim}{\psi}_j$
//linear combination!

APPLY $\underset{\sim}{d} = \sum d_i \psi_i$ to Rayleigh damped system. $\underset{\sim}{M}\ddot{\underset{\sim}{d}} + \underset{\sim}{C}\dot{\underset{\sim}{d}} + \underset{\sim}{K}\underset{\sim}{d} = \underset{\sim}{F}$

$\Rightarrow \underset{\sim}{M}\ddot{\underset{\sim}{d}} + (a\underset{\sim}{M} + b\underset{\sim}{K})\dot{\underset{\sim}{d}} + \underset{\sim}{K}\underset{\sim}{d} = \underset{\sim}{F}$

$\Rightarrow \underset{\sim}{M}\sum \ddot{d}_j \psi_j + (a\underset{\sim}{M} + b\underset{\sim}{K})\sum \dot{d}_j \ldots$
$+ \underset{\sim}{K}\sum d_j \psi_j = \underset{\sim}{F}$

$\xRightarrow[\text{properties}]{\text{use}} \ddot{d}_k + (a + b\omega_k^2)\dot{d}_k + \omega_k^2 d_k = F_k$ $k = 1 \ldots n$

$\boxed{\text{let } \xi_k = \left(\frac{a}{2\omega_k} + \frac{b\omega_k}{2}\right)}$

$\boxed{\omega_{dk} = \omega_k \sqrt{1 - \xi_k^2}}$

SOLN: $d_k(t) = d_0 e^{-\xi_k \omega_k t}[\cos(\omega_{dk} t) + i\sin(\ldots$

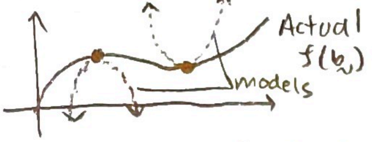
FEA & Optimization cont.

HOW TO FIND $\nabla_b f(\underline{b})$?

① **Response surface models.**

construct parabolic model @ $f(\underline{b})$

$f'(\underline{b}) = f(\underline{b}_0) + \nabla f(\underline{b}_0)(\underline{b}-\underline{b}_0) + \frac{1}{2}(b-b_0)^2 \underline{H}(b_0)$



② **Numerical Derivatives**

$\frac{\partial f_i}{\partial b_i} \approx \frac{f(b_i+\epsilon) - f(b_i)}{\epsilon}$

problem is finding sweet spot for ϵ, OR $\frac{\partial f_i}{\partial b_i} = \frac{1}{\epsilon} \mathrm{Im}[f(b_i + i\epsilon)]$

③ **Analytical Methods** (most common).

Large # of des params → ADJOINT METHOD

TOPOLOGY OPTIMIZATION

mt2

Example of composite Design Optimization

A

Polymer → E_P, ν

Filler → E_F, ν

constraint: $V_{Filler} = 10\%$ total V

$\Delta A = \underline{u}(x_y) \cdot \underline{e}_y = \underline{d} \cdot \underline{e}_y$

LET b_i be a design var. $i=1\ldots$ numels.

$\boxed{E_i = b_i E_F + (1-b_i) E_P}$ rule of comp. mixtures

$\sum b_i \frac{V_i}{V} \le 0.1 \Rightarrow \boxed{\underline{b} \cdot \frac{\underline{V}}{V} \le 0.1}$ constraint

For COMPOSITE: $0 \le b_i \le 1$

For Topology optm. $b_i = 1$ OR 0!

// use adjoint method.

$\underline{K}(\underline{b})\, \underline{d} = \underline{F}$

$\underline{K} = \int \underline{B}^T \underline{D}\, \underline{B}\, dv = \int \underline{B}^T (b_i \underline{D}_F + (1-b_i)\underline{D}_P)\, \underline{B}\, dv$

FLUID CLASS PROBLEMS

mass balance: start: $\frac{\partial}{\partial t}\int_V \rho\, dv = -\int_{\partial V} \rho \underline{v} \cdot \hat{n}\, dA$

using divg. thm. (Practice final) // global statement for mass conservation in a control vol.

$\Rightarrow \boxed{\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \underline{v}) = 0}$ // everything here is a funct of space + time

FORM 1

$= \boxed{\frac{\partial \rho}{\partial t} + \nabla\rho \cdot \underline{v} + \rho\, \mathrm{div}(\underline{v}) = 0}$ FORM 2

$= \boxed{\frac{D\rho}{Dt} + \rho\, \mathrm{div}(\underline{v}) = 0}$ FORM 3 // material derivative defined

MOMENTUM BAL Start: $\rho \frac{D}{Dt}(\underline{v}) = \mathrm{div}(\underline{S}) + \underline{b}$

// $\underline{S}$ is stress tensor: see STRESS ANALYSIS.

$\Rightarrow \rho \frac{D}{Dt}(\underline{v}) = \tilde{\nabla}^T \underline{\tau} - \nabla P + \underline{b}$ FORM 1

$= \boxed{\rho \frac{\partial \underline{v}}{\partial t} + \rho(\nabla \underline{v}) \cdot \underline{v} = \tilde{\nabla}^T \underline{\tau} - \nabla P + \underline{b}}$ FORM 2

// see STRESS ANALYSIS for $\tilde{\nabla}^T$ & $\underline{\tau}$

unsteady term. — convective term — shear gradient — pressure gradient — body forces

For reference

$\underline{v} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$ $\nabla \underline{v} \triangleq \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \cdot & \cdot \\ \frac{\partial w}{\partial x} & \cdot & \cdot \end{bmatrix}$ $\dot{\underline{\epsilon}} \triangleq \begin{bmatrix} \partial u/\partial x \\ \partial v/\partial y \\ \partial w/\partial z \\ \partial u/\partial y + \partial v/\partial x \\ \partial v/\partial z + \partial w/\partial y \\ \partial u/\partial z + \partial w/\partial x \end{bmatrix}$ $\underline{D} \triangleq 2\mu \begin{bmatrix} 2/3 & -1/3 & -1/3 & & & \\ & 2/3 & -1/3 & & & \\ & & 2/3 & & & \\ & & & 1/2 & & \\ & & & & 1/2 & \\ & & & & & 1/2 \end{bmatrix}$

HOW DO WE FIND $\underline{\tau}$?

$\underline{\tau} = \underline{D}\,\dot{\underline{\epsilon}} = \underline{D}(\tilde{\nabla}\underline{v})$ // $\dot{\underline{\epsilon}}_{6\times1} = \tilde{\nabla}_{6\times3}\, \underline{v}_{3\times1}$ "strain rates"

One last necessary relation: $P = \rho \frac{RT}{m}$

PUT ALL TOGETHER

① $\frac{D\rho}{Dt} + \rho\, \mathrm{div}(\underline{v}) = 0$ // mass bal

② $\rho \frac{D\underline{v}}{Dt} = \tilde{\nabla}^T \underline{\tau} - \nabla P + \underline{b}$ // momentum bal

③ & ④ $\underline{\tau} = \underline{D}\,\dot{\underline{\epsilon}} = \underline{D}(\tilde{\nabla}\underline{v})$

⑤ $P = \rho \frac{RT}{m}$

combine with incompressible: $\frac{\partial \rho}{\partial t} = 0$ $\frac{D\rho}{Dt} = 0 \Rightarrow \mathrm{div}(\underline{v}) = 0$

(S) = $\begin{cases} \mathrm{div}(\underline{v}) = 0 \\ \rho_0 \frac{\partial \underline{v}}{\partial t} + \rho_0 \nabla \underline{v} \cdot \underline{v} = -\nabla P + \tilde{\nabla}^T \underline{\tau} + \underline{b} \end{cases}$

(W) = $\begin{cases} \int_\Omega q\, \mathrm{div}(\underline{v})\, dv = 0 \\ \int_\Omega \left(\rho_0 \underline{w}^T \frac{\partial \underline{v}}{\partial t} + \rho_0 \underline{w}^T \nabla \underline{v} \cdot \underline{v} + \tilde{\nabla}\underline{w}\, \underline{\tau} - \mathrm{div}(\underline{w}) P\right) \\ = \int_\Omega \underline{w}^T \underline{b}\, dv + \int_{\partial\Omega_t} \underline{w}^T \underline{t}\, dA \end{cases}$

scalar q / $\underline{w}$ test funct. vector

PRACTICAL ISSUES

① **Steady State**: $\left(\frac{\partial *}{\partial t}\right) = 0$!

Removes $\begin{bmatrix} \underline{M} & \underline{0} \\ \underline{0} & \underline{0} \end{bmatrix} \begin{bmatrix} \dot{\underline{v}} \\ \dot{P} \end{bmatrix} \to 0$ term.

② **Creeping Flow** $\rho_0 \frac{D\underline{v}}{Dt} \approx 0$ Further Removes $\begin{bmatrix} \underline{R} \\ \underline{0} \end{bmatrix} \to 0$ term

③ Turbulent $Re_L = \frac{UL}{\nu}$

& Advection INSTABILITIES!

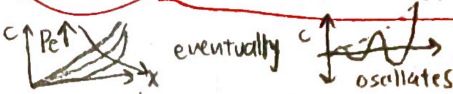
let $c(x) \triangleq$ concentration of something in fluid.

BC: $c(0) = 0$ $c(L) = 1$

Governing Dif eq: $\frac{\partial c}{\partial t} + v \frac{\partial c}{\partial x} = K \frac{\partial^2 c}{\partial x^2}$ K is some parameter of model

Assume steady state $\frac{\partial c}{\partial t} = 0$.

Exact Soln to $v \frac{\partial c}{\partial x} = K \frac{\partial^2 c}{\partial x^2}$ $c(0) = 0$, $c(L) = 1$ $\boxed{c(x) = \frac{1 - e^{Pe\frac{x}{L}}}{1 - e^{Pe}}}$



$\boxed{Pe \triangleq \frac{vL}{K}}$ "Peclet #"

$Pe \ll 1$ diffusion dominated

$Pe \gg 1$ advection dominated

How to prevent oscillation?

① reduce L (length of chanel?)

② **Petrov-Galerkin Scheme**

$M_A = N_A + \alpha W_A$ → $\alpha = 1$ full upwinding

$W_1 = -1/2$ $W_2 = 1/2$

FEA SUBSPACES

$\underline{v} \in S_v = \{ \underline{v} \mid \underline{v} = \bar{\underline{v}} \text{ on } \partial_v\Omega \}$

$P \in S_P = \{ P \mid P = \bar{P} \text{ on } \partial_P\Omega \}$

$\boxed{\underline{v} = \sum_{A=1}^{nv} N_A(\underline{x}) \underline{v}_A}$ → num nodes

$\boxed{P = \sum_{A=1}^{np} \phi_P(\underline{x}) P_A}$

Expanded:

$\underset{3\times1}{\underline{v}} = \underset{3\times3nv}{\begin{bmatrix} N_1 & & & N_2 & & & \cdots \\ & N_1 & & & N_2 & & \\ & & N_1 & & & N_2 & \end{bmatrix}} \underset{3nv\times1}{\begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ \vdots \end{bmatrix}}$

N_A → velocity shape fcts.

Φ → P shape fcts.

$\underset{1\times1}{P} = \underset{1\times np}{[\Phi_1\ \Phi_2\ \ldots]} \underset{np\times1}{\begin{bmatrix} P_1 \\ P_2 \\ \vdots \end{bmatrix}}$

$\underline{K}_P^T \triangleq$ $\underline{K}_P \triangleq$ $\underline{K} \triangleq$ $\underline{F} \triangleq$ $\underline{M} \triangleq$ $\underline{R}(\underline{v}) \triangleq$

Mass (w) → $\underline{q}^T \int_\Omega \underline{\Phi}^T \underline{G}\, dv\, \underline{v} = 0$

Momentum (w) →

$\int_\Omega \underline{N}^T \rho_0 \underline{N}\, dv \frac{d\underline{v}}{dt} + \int_\Omega \underline{B}^T \underline{D}\, \underline{B}\, dv\, \underline{v} - \int_\Omega \underline{G}^T \underline{\Phi}\, dv\, \underline{P} + \ldots$

$\int_\Omega \rho_0 \underline{N}^T (\nabla \underline{N} \underline{v}) \cdot \underline{N}\underline{v}\, dv = \int_\Omega \underline{N}^T \underline{b}\, dv + \int_{\partial\Omega} \underline{N}^T \underline{t}\, dA$

$\begin{bmatrix} \underline{M} & \underline{0} \\ \underline{0} & \underline{0} \end{bmatrix} \begin{bmatrix} \frac{\partial \underline{v}}{\partial t} \\ \frac{\partial P}{\partial t} \end{bmatrix} + \begin{bmatrix} \underline{K} & -\underline{K}_P \\ \underline{K}_P^T & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{v} \\ \underline{P} \end{bmatrix} + \begin{bmatrix} \underline{R}(\underline{v}) \\ 0 \end{bmatrix} = \begin{bmatrix} \underline{F} \\ \underline{0} \end{bmatrix}$ STATE SPACE RLZ mt2

Operator Splitting + Chorin Split

$\dot{y} = \underline{M}\,y \rightarrow \dot{y} = (\underline{A} + \underline{B})y$

Exact Soln: $y = y_0 e^{Mt}$

$= y_0 e^{(A+B)t} = y_0 e^{At} e^{Bt}$

Forward-Euler (explicit) soln:

$y_{n+1} = y_n + (\underline{A} + \underline{B})\Delta t\, y_n$

Nothing Fancy here!

Both have * or **

THE SPLIT:

The setup for Forw-Euler $\Rightarrow$	$\dot{y}^*_n = A y^*_n$	$\dot{y}^{**}_n = \underline{B} y^{**}_n$
The CRITICAL SUBSTITUTION afterwards	$y^*_n = y_n$	$y^{**}_n = y^*_{n+1} = y_n$

☆ SET ☆ $y^{**}_n = y_n$!! ☆

The Split Completely Cascaded:

$y^*_{n+1} = A\Delta t\, y^*_n + y^*_n = A\Delta t\, y_n + y_n$

F-E algebra; substitute $y^*_n = y_n$

$y^{**}_{n+1} = B y^{**}_n \Delta t + y^{**}_n = B y^*_{n+1} \Delta t + y^*_{n+1} = (B\Delta t + 1) y^*_{n+1}$

F-E algebra; sub $y^{**}_n = y^*_{n+1}$; Pull y^*_{n+1}

$= (B\Delta t + 1)(A\Delta t + 1) y_n = y_{n+1}$

... sub $y^*_{n+1} = $ ♀; Use ☆ relation

The Chorin Split as applied to Fluids:

NS eqs $\rightarrow \operatorname{div}(\underset{\sim}{u}) = 0$ //incomp.

$\rightarrow \frac{\partial u}{\partial t} = -\nabla \underset{\sim}{u} \cdot \underset{\sim}{u} + \tilde{\nabla}^T \tau(\underset{\sim}{u}) - \nabla P + \underset{\sim}{b}$

☆ SPLIT! $\rightarrow \frac{\partial u^*}{\partial t} = -\nabla \underset{\sim}{u}^* \cdot \underset{\sim}{u}^* + \tilde{\nabla}^T \tau(\underset{\sim}{u}^*) + \underset{\sim}{b}$

$\rightarrow \frac{\partial u^{**}}{\partial t} = -\nabla P$ //isolates P effects to one side of split

APPLY FORW-EULER to split | $u^{**}_n = u_n$

$\boxed{1}$ $\frac{\underset{\sim}{u}^*_{n+1} - \underset{\sim}{u}_n}{\Delta t} = -(\nabla \underset{\sim}{u}_n)\cdot \underset{\sim}{u}_n + \nu \operatorname{div}(\tilde{\nabla}^T \underset{\sim}{u}_n)$

$\boxed{2}$ $\frac{\underset{\sim}{u}^{**}_{n+1} - \underset{\sim}{u}^*_{n+1}}{\Delta t} = -\nabla P_{n+1}$

take div of both sides & set $\operatorname{div}(u^{**}_{n+1}) = \operatorname{div}(u_{n+1}) \rightarrow 0$

$\frac{-\operatorname{div}(u^*_{n+1})}{\Delta t} = -\nabla P_{n+1}$

$\boxed{3}$ $\underset{\sim}{u}^{**}_{n+1} = -\nabla P_{n+1}\Delta t + \underset{\sim}{u}^*_{n+1} = \underset{\sim}{u}_{n+1}$ ☺

work flow: $\boxed{1} \xrightarrow[u^*_{n+1}]{\text{solve for}} \boxed{2} \xrightarrow[\nabla P_{n+1}]{\text{solve for}} \boxed{3} \xrightarrow[u^{**}_{n+1} = u_{n+1}]{\text{solve for}}$ ☺

Optimization

idea: $\min_{\underset{\sim}{b} \in \Omega} f(\underset{\sim}{b})$

2 flavors $\rightarrow$ Parameterized $\rightarrow$ scalar, vector, tensor

$\rightarrow$ Topology $\rightarrow$ change the domain

$\underset{\sim}{b} \triangleq$ "parameter"

$\Omega \triangleq$ "Subspace of available $\underset{\sim}{b}$s"

(this is where constraints play)

General Optimization Form

$\min_{\underset{\sim}{b} \in \Omega} f(\underset{\sim}{b})$

$\Omega = \{\underset{\sim}{b} \mid g(\underset{\sim}{b}) \leq 0\}$

$\frac{\partial f}{\partial \underset{\sim}{b}} = 0$ //local extreme; $\frac{\partial^2 f}{\partial \underset{\sim}{b}^2} \geq 0$ //concave up; calculus review

$\frac{\partial f}{\partial \underset{\sim}{b}} = \nabla_b f \triangleq \begin{bmatrix} \frac{\partial f}{\partial b_1} \\ \vdots \\ \frac{\partial f}{\partial b_n} \end{bmatrix} = 0$ "gradient wrt b"

$\frac{\partial^2 f}{\partial \underset{\sim}{b}^2} = \nabla_b^2 f \triangleq \begin{bmatrix} \frac{\partial^2 f}{\partial b_1^2} & \cdots & \frac{\partial^2 f}{\partial b_1 \partial b_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial b_n \partial b_1} & \cdots & \frac{\partial^2 f}{\partial b_n^2} \end{bmatrix} \geq 0$

$= H(f)$ "Hessian"

EQUALITY CONSTRAINT

minimize $A = lw$ subject to $2(l+w) = P$ //fixed, known per.

① Translate $2(l+w) = P \rightarrow 2(l+w) - P = 0 = g(l,w)$

② setup. $\mathcal{L}(l,w,\lambda) \triangleq A - \lambda g$

$= lw - \lambda[2(l+w) - P]$

① $\frac{\partial \mathcal{L}}{\partial l} = 0$

② $\frac{\partial \mathcal{L}}{\partial w} = 0$

③ $\frac{\partial \mathcal{L}}{\partial \lambda} = g(l,w) = 0$!

// System of 3 eq, 3 unknowns (l, w, λ)

INEQUALITY CONSTRAINT

$\mathcal{L}(\underset{\sim}{x}, \lambda) = f(\underset{\sim}{x}) + \lambda g(\underset{\sim}{x})$

conditions: ① $\frac{\partial \mathcal{L}}{\partial x} = 0$, ② $\frac{\partial \mathcal{L}}{\partial \lambda} = g(\underset{\sim}{x}) \leq 0$, ③ $\lambda \geq 0$, ④ $\lambda \frac{\partial \mathcal{L}}{\partial \lambda} = 0$, $\lambda g(x) = 0$

KKT conditions

Write system out.

① Assume g is inactive $(g < 0, \lambda = 0)$

$\rightarrow$ check if solns violate g.

② Assume g is active $(g = 0, \lambda \neq 0)$

Mini max problem

$\min f(x)$ under $h(x) = 0$, $g(x) \leq 0$

$\min_x \left[\max_{\nu, \lambda \geq 0} \mathcal{L}(x, \nu, \lambda) \right]$

where $\mathcal{L}(x, \nu, \lambda) \triangleq f(x) + \nu h(x) + \lambda g(x)$

conditions: $\frac{\partial \mathcal{L}}{\partial x} = 0$, $\frac{\partial \mathcal{L}}{\partial \nu} = 0$, $\frac{\partial \mathcal{L}}{\partial \lambda} \leq 0$, $\lambda \geq 0$, $\frac{\partial \mathcal{L}}{\partial \lambda}\lambda = 0$

//combination of EQUALITY & INEQ conditions

FEA & Optimization

⓪ Zeroth-order search methods (Nelder-Mead) triangle-flipping method.

- guaranteed to find min ✓
- slow ✗

① First-order method: seed $\underset{\sim}{b}$, $\rightarrow$ find search direction (typically steepest gradient or zig zag) $\rightarrow$

$\rightarrow$ minimize f on search direction s, find loc $\underset{\sim}{b}^*$ of min

Repeat.

② 2nd-order Newton's Method. start $\underset{\sim}{b}^i$ where $\nabla_b f(\underset{\sim}{b}^i) \neq 0$.

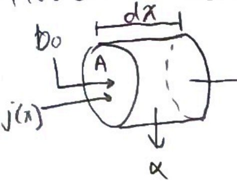
Try to find $\nabla_b f(\underset{\sim}{b}^i + \Delta \underset{\sim}{b}) = 0$

Taylor exp. $\rightarrow \nabla_b f(\underset{\sim}{b}^i) + \nabla_b \nabla_b f(\underset{\sim}{b}^i) \cdot \Delta b = 0$

$\Rightarrow \boxed{\Delta \underset{\sim}{b} = -H^{-1}(f(\underset{\sim}{b}^i)) * \nabla_b f(\underset{\sim}{b}^i)}$ //newton step size in direct steep

HW Review

HW 1.1 derivation of strong-form eqs.



$$\Rightarrow \frac{d}{dx}\left(A D(x) \frac{d\gamma(x)}{dx}\right) + b_0(x) = A\,\gamma(x)\alpha(x)$$

// at end, there is a cap: $j(L)=0 \Rightarrow -D(x)\frac{d\gamma}{dx}=0$ NATURAL (Neumann) BC

// at base, there is a given concentration: $\gamma(0)=\gamma_0$ ESSENTIAL (Dirichlet) BC

HW

HW 1.3

WF: $\int_0^1 v'u' + vu\, dx = 2v(1)$

$\Rightarrow \int_0^1 (vu')' - vu'' + vu\, dx = 2v(1)$

$\Rightarrow \int_0^1 v(u-u'')dx + v(1)u'(1) - \cancel{v(0)u'(0)} = 2v(1)$

0

$\Rightarrow \int_0^1 v(u-u'')dx = 2v(1) - v(1)u'(1)$

$\Rightarrow \underbrace{\int_0^1 v(u-u'')dx}_{0} - v(1)\underbrace{(2-u'(1))}_{0} = 0$

SF: $\boxed{u-u''=0 \qquad u'(1)=2}$

Proving linearity: $a(c_1u_1+c_2u_2, u) = c_1a_1(u_1,u)+c_2a_2(u_2, \ldots$

Proving bilinearity: (add-on)

Show $a(u,u) = a(u,u)$ // squared

HW 1.2 Deriving weak form from strong → see cheatsheet.

- test function $v(x)=0$ @ essential BC. "admissable"
- trial functions $u(x) \in \mathcal{S} = \{f(x) \mid f(eBC) = eBC \text{ conditions}\}$

HW 1.4 Minimization Form. *missed this on MT2!!

minimize the functional $I(\Phi) \triangleq \frac{1}{2}a(u,u) - \ell(u)$

method: set $\frac{d}{d\epsilon} I(\Phi+\epsilon v)\Big|_{\epsilon=0} = 0$ where v is an admissable test funct.

intuition: setting $\epsilon=0$ is basically eval $I(\Phi)$, then introduce the tiniest admissible perturbation ϵv and $I(\Phi)$ should not change!

HW 2.1

given minimization form $I(\Phi)$ and trial $\mathcal{S}_n = \{\tilde{\Phi}(x) \mid \Phi_1 \cdot x + \Phi_2 \cdot \frac{1}{3}x^3\}$

Find $\tilde{\Phi}(x)$ that satisfies minimization form.

KEY. set $\frac{\partial I(\tilde{\Phi})}{\partial \Phi_1} = \frac{\partial I(\tilde{\Phi})}{\partial \Phi_2} = 0$

works when $\Phi_{1,2}$ = const! otherwise use $\frac{\partial}{\partial\epsilon} I(\Phi+\epsilon v)\big|_{\epsilon=0} = 0$

HW 2.2 EASY linalg prob

id_f → loc of unknown a $\quad id_d$ → loc of unknown f.

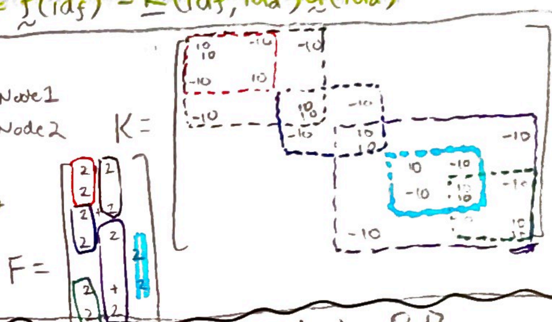
OR: $\underline{K}(id_f, id_f)\,\underline{a}(id_f) = \underline{f}(id_f) - \underline{K}(id_f, id_d)\,\underline{a}(id_d)$

HW 2.3 LM Practice:

LM:	2	7	3	4	1	6	→ loc Node 1
	1	6	4	7	3	5	→ loc Node 2
el	1	2	3	4	5	6	

6 elements, 7 nodes, $\underline{K}_{7\times7}$

$\underline{k}^e = \begin{bmatrix} 10 & -10 \\ -10 & 10 \end{bmatrix}$ $\quad \underset{\sim}{f}^e = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ $\quad \underline{f}_{7\times1}$



HW 3.1 Constructing Lagrange Shape functions & the Kronecker property:

Ex: $N_1^e(x) = \frac{(x-x_2^e)(x-x_3^e)(x-x_4^e)}{(x_1^e-x_2^e)(x_1^e-x_3^e)(x_1^e-x_4^e)}$ ↗ 1 @ x_1^e ↘ 0 else

HW 3.2 Iso parametrism.

^ hat denotes iso parametric realm

Key relations: $N_i(x,y) = \hat{N}_i(\xi(x,y), \eta(x,y))$

$\underset{\text{parent}}{\underset{\sim}{x}} = \sum_{\text{all } i} \hat{N}_i\, \underset{\text{physical}}{\underset{\sim}{x}_i^e}$ // dot product of sorts.

$\hat{J}^e = \sum_{\text{all } i} \frac{\partial \hat{N}_i(\xi)}{\partial \xi}\, \underset{\text{physical}}{x_i^e}$ // dot product of deriv. of parent-dom shape functions

HW 3.3 + Coding Problem on MT 1 ☆☆!!

Assembly of an element m^e matrix of transient heat problem.

#N(gp,:) evaluates N @ quadr. point gp

#dN(gp,:) evals dN @ quadr. point gp

#$\underline{M} = \int_\Omega \underline{N}^T c\, \underline{N}\, d\Omega$ integration term

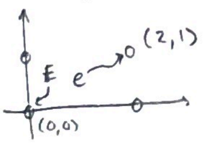
$m^e = \sum_{\text{all gp}} N^{eT} c N^e$ * weight * Jacobian

```
m = zeros(nen, nen)
for gp = 1:nq
  x_parent = x_physical * N(gp,:)'
  J = x_physical * dN(gp,:)'
  M = m + N(gp,:)' * N(gp,:) * c(gp) * J * w(gp)
end.
```

#nen → num nodes / elem

#nq → # of quadrature points.

HW 4.1 2-D Lagrangian



$N_e = \frac{xy}{(2)(1)}$

$N_E = \frac{(x-2)(x-1)}{(-2)(-1)}$ } Examples

$T^e(\vec{x}) = [N_1^e(\vec{x}), N_2^e(\vec{x}), N_3^e(\vec{x}), N_4^e(\vec{x})] \begin{bmatrix} T_1^e \\ T_2^e \\ T_3^e \\ T_4^e \end{bmatrix}$

when focusing on a SINGLE element e

$q = k\nabla T \Rightarrow q^e = kBT^e$

where $B = \begin{bmatrix} \partial N_1^e/\partial x & \cdots & \partial N_4^e/\partial x \\ \partial N_1^e/\partial y & \cdots & \partial N_4^e/\partial y \end{bmatrix}_{2\times4}$

$\underset{\text{const.}}{k}\, \underline{B}_{2\times4}\, T^e_{4\times1} = q^e_{2\times1}$ (heat flux in x & y).

Likewise for anisotropy let $k = D_{2\times2}$

$q^e_{2\times1} = \underline{D}_{2\times2}\, \underline{B}_{2\times4}\, T^e_{4\times1}$ $\quad \underline{D} = \begin{bmatrix} k_{xx} & k_{xy} \\ k_{yx} & k_{yy} \end{bmatrix}$

Tensor!

HW 4.2 Simple Integral evaluation

HW 4.3 Pure ANSYS Fluent Problem

HW 5.1 STRESS ANALYSIS + isoparametrism

$\underset{\sim}{\varepsilon} = \tilde{\nabla}\underset{\sim}{u} \Rightarrow \boxed{\underset{\sim}{\varepsilon}^e = \underline{B}\,\underset{\sim}{d}}$ → $\underset{\sim}{d}$ is displacement vector (given)

we know, $\underset{\sim}{d}$, to find $\underset{\sim}{\varepsilon}^e$, we need

to find $\underline{B} \triangleq \begin{bmatrix} \partial N_1/\partial x & 0 & \partial N_2/\partial x & 0 & \cdots \\ 0 & \partial N_1/\partial y & 0 & \partial N_2/\partial y & \cdots \\ \partial N_1/\partial y & \partial N_1/\partial x & \partial N_2/\partial y & \partial N_2/\partial x & \cdots \end{bmatrix}$ why $\underline{B}$ defined as such? definitions

$\underset{\sim}{d} \triangleq \begin{bmatrix} d_1^e x \\ d_1^e y \\ \vdots \\ d_4^e x \\ d_4^e y \end{bmatrix}$

To get $\frac{\partial N_i}{\partial x}, \frac{\partial N_i}{\partial y}$ from parent-domain $\frac{\partial N_i}{\partial \xi}, \frac{\partial N_i}{\partial \eta}$ ⇒ $\begin{bmatrix} \partial N_i/\partial\xi \\ \partial N_i/\partial\eta \end{bmatrix} = \underbrace{\begin{bmatrix} \partial x/\partial\xi & \partial y/\partial\xi \\ \partial x/\partial\eta & \partial y/\partial\eta \end{bmatrix}}_{J} \begin{bmatrix} \partial N_i/\partial x \\ \partial N_i/\partial y \end{bmatrix}$

Thus ⇒ $\begin{bmatrix} \partial N/\partial x \\ \partial N/\partial y \end{bmatrix} = J^{-1} \begin{bmatrix} \partial N/\partial\xi \\ \partial N/\partial\eta \end{bmatrix}$

Recall: Used to find J

$\frac{\partial x}{\partial \xi} = \sum_{\text{all } i} \frac{\partial \hat{N}_i}{\partial \xi} x_i^e$

$\frac{\partial x}{\partial \eta} = \sum_{\text{all } i} \frac{\partial \hat{N}_i}{\partial \eta} x_i^e$

$\frac{\partial y}{\partial \xi} = \sum_{\text{all } i} \frac{\partial \hat{N}_i}{\partial \xi} y_i^e$

After solving $\underset{\sim}{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{bmatrix}$

$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = \underline{D} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz}=0 \\ \gamma_{xy} \\ \gamma_{yz}=0 \\ \gamma_{xz}=0 \end{bmatrix}$

$\boxed{\underset{\sim}{\sigma} = \underline{D}\,\underset{\sim}{\varepsilon}}$

Cauchy's Law: $\boxed{\underset{\sim}{t} = \underset{\sim}{S}\,\underset{\sim}{n}}$

$\underset{3\times3}{\underset{\sim}{S}}$ is stress tensor

HW 5.2

Edge traction given, find forces $\rightarrow$ more stress analysis equations

HW 5.3

Redefinition of $\tilde{\nabla}$ for 2-D simplified problems

$\vec{u} = \begin{bmatrix} u_r \\ u_z \end{bmatrix}$ // cylindrical coords, NO u_θ!

$$\varepsilon = \begin{bmatrix} \varepsilon_{rr} \\ \varepsilon_{\theta\theta} \\ \varepsilon_{zz} \\ \cancel{\gamma_{r\theta}} \\ \cancel{\gamma_{\theta z}} \\ \gamma_{rz} \end{bmatrix} \begin{matrix} \\ \\ \\ 0 \\ 0 \\ \\ \end{matrix} = \begin{bmatrix} \varepsilon_{rr} \\ \varepsilon_{\theta\theta} \\ \varepsilon_{zz} \\ \gamma_{rz} \end{bmatrix}$$

Find $\underset{4\times 2}{\tilde{\nabla}}$ by coefficient matching

Find $\boxed{B_A = \tilde{\nabla} N_A}$

HW 6.1 → Modal Analysis

what are the different methods of time-stepping?

$$\left.\begin{aligned} d_{n+1} &= f(d_n) \\ x_{n+1} &= g(x_n) \\ a_{n+1} &= h(a_n) \end{aligned}\right\}$$ f, g, h typically follow kinematic eqs.

CENTRAL DIFF. METHODS.

$$\underbrace{\begin{aligned} d_{n+1} &= d_n + \Delta t\, v_n + \frac{\Delta t^2}{2} a_n \\ a_{n+1} &= M^{-1}(F_{n+1} - K d_{n+1}) \\ v_{n+1} &= v_n + \Delta t \left(\frac{a_n + a_{n+1}}{2}\right) \end{aligned}}$$

to use the stepping method, must find K, M, F

HW 8 Derivation of Fluid Mechanics FEA eqs.

assume no body forces

$$\rho \frac{Dv}{Dt} = -\nabla P + \tilde{\nabla}^T \tau \xrightarrow[\text{apart}]{\text{break}} \rho\frac{\partial v}{\partial t} + \rho(\nabla v)\cdot v = \tilde{\nabla}^T \tau - \nabla P + \cancel{b}^{\,0}$$

mat. derivative

$$\tilde{\nabla}^T \tau = \tilde{\nabla}^T (D \varepsilon) = \tilde{\nabla}^T (D(\tilde{\nabla} v))$$

Recall:

$$\tilde{\nabla} \triangleq \begin{bmatrix} \partial/\partial x & & & \partial/\partial y & & \partial/\partial z \\ & \partial/\partial y & & \partial/\partial x & \partial/\partial z & \\ & & \partial/\partial z & & \partial/\partial y & \partial/\partial x \end{bmatrix} \qquad D \triangleq 2\mu \begin{bmatrix} 2/3 & -1/3 & -1/3 & & & \\ -1/3 & 2/3 & -1/3 & & & \\ -1/3 & -1/3 & 2/3 & & & \\ & & & 1/2 & & \\ & & & & 1/2 & \\ & & & & & 1/2 \end{bmatrix}$$

$$\tau \triangleq \begin{bmatrix} \tau_{xx} \\ \tau_{yy} \\ \tau_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{bmatrix}$$

$\tilde{\nabla}^T \tau = \nabla^2 v$ // comes out to standard NS nonlinear term.

Derivation of weak form:

$g \triangleq -\frac{\partial v}{\partial t} - (\nabla v)\cdot v + \nu(\nabla^2 v)$ $\qquad 0 = g - \nabla P$ // div of both sides.

$\operatorname{div}(0) = \operatorname{div}(g - \nabla P) \Rightarrow 0 = q \operatorname{div}(g - \nabla P)$ mult by test func q

CRITICAL STEP:

$\operatorname{div}(q(g-\nabla P)) = q \operatorname{div}(g - \nabla P) + \nabla q \cdot (g - \nabla P)$ // div product rule

$$\Rightarrow \int_\Omega q \operatorname{div}(g-\nabla P)\, d\Omega = \int_\Omega \nabla q (g - \nabla P) - \operatorname{div}(q(g-\nabla P))\, d\Omega = 0\ !$$

$$= \int_\Omega \nabla q (g - \nabla P)\, d\Omega - \underbrace{\int_{\partial\Omega} q\, (g - \nabla P)\, \hat{n}\cdot dA}_{0\,!}$$ Apply div theorem

simplify $\Longrightarrow$ $\boxed{\int_\Omega q \operatorname{div}(g - \nabla P)\, d\Omega = \int_\Omega \nabla q (g - \nabla P)\, d\Omega}$

```
# Material properties
nu = 3.77e-6 # m^2/s
```

everything above ↑ import numpy as np
from dolfin import *

HW

```
# Assumes 57 data files at 1 sec time steps in a sub-folder called Data
first_filenum         = 1
last_filenum          = 57
mesh_filename_root    = './Data/lv_mesh.'
velocity_filename_root = './Data/lv_vel.'
dt                    = 1. / 57.    # Assuming T = 1s

# Load mesh and velocity from the last tstep to allow estimation of v_dot at
# first tstep
mesh_filename = mesh_filename_root + str(last_filenum) + '.h5'
mesh       Mesh()
mesh_in = HDF5File(MPI.comm_world, mesh_filename, 'r')
mesh_in.read(mesh, 'mesh', False)
mesh_in.close()

# creates a vector-valued finite element function space
# vector valued because velocity is a vector
V = VectorFunctionSpace(mesh, "P", 1) # FILL IN HERE
velocity_filename = velocity_filename_root + str(last_filenum) + '.h5'
velocity = Function(V)
vel_in   = HDF5File(MPI.comm_world, velocity_filename, 'r')
vel_in.read(velocity, 'velocity')
vel_in.close()

# Set up output files (result will go in a folder called Results)
fout = File('Results/pressure.pvd')

for i in range(first_filenum, last_filenum+1):
    # Load new mesh
    mesh_filename = mesh_filename_root + str(i) + '.h5'
    mesh          = Mesh()
    mesh_in       = HDF5File(MPI.comm_world, mesh_filename, 'r')
    mesh_in.read(mesh, 'mesh', False)
    mesh_in.close()

    # Define new function spaces Q, V, and T for scalars, vectors, and tensors,
    # respectively
    # FILL IN CODE HERE
    Q = FunctionSpace(mesh, "P", 1)
    V = VectorFunctionSpace(mesh, "P", 1)
    T = TensorFunctionSpace(mesh, "P", 1)

    # Project old velocity field to new mesh. Need to allow extrapolation since
    # domain is changing.
    velocity.set_allow_extrapolation(True)
    velocity_prev = project(velocity, V)

    # Load new velocity field
    velocity_filename = velocity_filename_root + str(i) + '.h5'
    velocity          = Function(V)
    vel_in            = HDF5File(MPI.comm_world, velocity_filename, 'r')
    vel_in.read(velocity, 'velocity')
    vel_in.close()

    # Project gradient of velocity to CG1 space so that div(grad_velocity) can
    # be used to approximate laplacian of velocity
    grad_velocity = project(grad(velocity), T)

    # Define test, trial, and solution functions called p_test, p_trial, and p,
    # respectively
    # FILL IN CODE HERE
    p_test = TestFunction(Q)
    p_trial = TrialFunction(Q)
    p = Function(Q)

    # Define Dirichlet BC to pin the node with the largest x coordinate
    coords      = mesh.coordinates()
    x_dirichlet = -1e8
    y_dirichlet = -1e8
    for j in range(coords.shape[0]):
        if coords[j, 0] > x_dirichlet:
            x_dirichlet = coords[j, 0]
            y_dirichlet = coords[j, 1]

    def dirichlet_node(x, on_boundary):
        return near(x[0], x_dirichlet) and near(x[1], y_dirichlet)

    # This resets the pressure reference to zero every snap shot
    bc = DirichletBC(Q, Constant(0.), dirichlet_node, method='pointwise')

    # Define g vector
    # FILL IN CODE HERE
    g = nu * div(grad_velocity) - dot(grad_velocity, velocity) - (velocity -
velocity_prev)/dt

    # Set up a( , ) and l( ) forms and solve, placing solution in p
    # FILL IN CODE HERE
    a = inner(grad(p_test), grad(p_trial)) * dx
    l = inner(grad(p_test), g) * dx
    solve(a == l, p, bc)
```

→ notice a is a func of both p_trial & p_test
→ l is only a functional p_test

★ critical

Integrand = 0 since eq must hold for all control volum domains Ω

Practice Final

Q1.

ID matrix for multiple DOF:

$$\begin{array}{c|c} 1 & \\ 2 & \end{array}\left[\ \begin{matrix} + & + & + & + & + \end{matrix}\ \right]$$

1 2 3 4 ... n ← GLOBAL NODE loc.

↑ DOF

Workflow:

$k^e_{ab} \xrightarrow{LM}$ (Glob Node a, DOF_a) $\xrightarrow{ID}$ Row in K_{global} → K

$k^e_{ab} \xrightarrow{LM}$ (Glob Node b, DOF_b) $\xrightarrow{ID}$ Col in K_{global} → K

Ex $k^{e=4}_{15}$ → look @ elem. 4 in LM.

→ 1 is assoc. w/loc node 1, DOF 1 $\xrightarrow{\text{extract}}$ LM_{14}

→ 5 is assoc. w/loc node 3, DOF 1 $\xrightarrow{\text{extr.}}$ LM_{34} } Global Nodes

ID → glob node LM_{14}, DOF 1 $\xrightarrow{\text{extract}}$ Row in K

→ glob node LM_{34}, DOF 1 $\xrightarrow{\text{extract}}$ Col in K

> place $k^{e=4}_{15}$ into K(Row, Col)

Q2

Pattern matching to find (W) again

Q3

Pattern matching to find form-fitting gradient operator $\underline{\tilde{\nabla}}$

Q4

Assemble K using k^e, LM, and drop rows/cols where BC = 0

Q5

Operator Dif Eq given Forward Euler

$\frac{x_{n+1} - x_n}{\Delta t} = A$

$\Rightarrow x_{n+1} = A\,\Delta t$ ...

THE SPLIT

$$\left.\begin{aligned} q_{1\,n+1} &= A\,\Delta t\, q_{1n} + q_{1n} \\ q_{1n} &= x_n \rightarrow \text{init cond!} \end{aligned}\right\} (1)$$

$$\left.\begin{aligned} q_{2\,n+1} &= A_2\,\Delta t\, q_{2n} + q_{2n} \\ q_{2n} &= q_{1\,n+1} \end{aligned}\right\} (2)$$

$x_{n+1} = q_{2\,n+1}$

↳ what we look for

Q7

optimization on constraint (see cheatsheets)