

Emphasis: Algo-ithms & Implementations

Syl

www.scorec.rpi.edu/kgansen/CFD-06

Fordran

08

← look at this

Notation

Equations: Unsteady Compressible N-S (UCNS)

Conservative form

shorthand notation

Let ϕ be some variable

$$\phi_{,t} = \frac{d\phi}{dt}$$

$$\phi_{,11} = \frac{d^2\phi}{dx^2} = \frac{d^2\phi}{dx_1^2}$$

$$\phi_{,12} = \frac{d^2\phi}{dx_1 dx_2} = \frac{d^2\phi}{dx_2 dx_1}$$

$$\phi_{,112} = \frac{d^3\phi}{dx_1^2 dx_2}$$

$$\phi_{,112} = \frac{d^3\phi}{dx_1 dx_2^2}$$

$$\phi_{,112} = \frac{d^3\phi}{dx_1 dx_2 dx_1}$$

$$\vec{x} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

$$\vec{u} = \underline{u} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

We will often want to refer to specific component in a general way

u_i where $i = 1, 2 \text{ or } 3$

Continuity Equation

$$\rho_{,t} + \sum_{\lambda=1}^3 [\rho u_\lambda]_{,\lambda} = 0$$

$$\frac{d\rho}{dt} + \nabla \cdot \rho \underline{u} = 0$$

Momentum Equation

$$[\rho u_j]_{,t} + \sum_{\lambda=1}^3 [\rho u_\lambda u_j]_{,\lambda} + P_{,j} = \sum_{\lambda=1}^3 [\tau_{\lambda j}]_{,\lambda} + b_j$$

$$\frac{d}{dt} \rho u + \nabla \cdot \rho \underline{u} \underline{u} + \frac{\partial P}{\partial x_i} = \nabla \cdot \underline{\tau} + \underline{b}$$

Energy Equation

$$[\rho e_{tot}]_{,t} + \sum_{\lambda=1}^3 [\rho u_\lambda e_{tot}]_{,\lambda} + \sum_{\lambda=1}^3 [u_\lambda P]_{,\lambda} = \sum_{\lambda=1}^3 \sum_{\beta=1}^3 [\tau_{\lambda\beta} u_\beta]_{,\lambda} - \sum_{\lambda=1}^3 b_\lambda u_\lambda + r - \sum_{\lambda=1}^3 q_{\lambda,\lambda}$$

Note:

in summations the summing index is repeated

→ hence if index repeats, we sum

Contracted/Dummy index

Free Index: mentioned once

where

ρ = density

P = pressure

e_{tot} = internal energy + KE

$$= e + \frac{u_\lambda u_\lambda}{2}$$

τ_{ij} = viscous stress tensor

$$\sim \mu S_{ij} = \mu [u_{i,j} + u_{j,i} + \frac{2}{3} \delta_{ij} \nabla \cdot \underline{u}]$$

b_j = body force

q_λ = heat flux vector

$-k T_{,\lambda}$

r = heat supply

Indexical / Einstein Summation Convention

- 1) indices (x, y, z, t, ...)
they can appear no more than twice in a term
- 2) if they appear once \Rightarrow Free Index
they are free to be 1, 2, or 3
Hence 3 distinct equations for momentum
- 3) if they appear twice \Rightarrow Contracted Index
they must be summed over that index
3 terms added
don't need " Σ " they are implied
- 4) we have error if all terms don't have the same free indices

Ex

$$\begin{aligned}
 & p_{i,z} [P u_x]_{,x} = 0 \\
 & [P u_y]_{,t} + [P u_x u_y]_{,x} + P_{,y} [T_{xy}]_{,x} + b_y u_x + r - q_{x,x} \\
 & [P e_{tot}]_{,t} + [P u_x e_{tot}]_{,x} + [u_x P]_{,x} = [T_{xy} u_y]_{,x} + b_x u_x + r - q_{x,x} \\
 & \frac{\partial P p_{tot}}{\partial t} + \frac{\partial (P u_x e_{tot})}{\partial x} + \frac{\partial (P u_x e_{tot})}{\partial y} + \frac{\partial (P u_x e_{tot})}{\partial z} + \dots
 \end{aligned}$$

Further Notational Abstraction

- collect $\frac{d}{dt}$ terms $\equiv \dot{\underline{U}} = \begin{Bmatrix} P \\ P u \\ P e_{tot} \end{Bmatrix} = \begin{Bmatrix} P \\ P u_y \\ P e_{tot} \end{Bmatrix} = \begin{Bmatrix} P u \\ P v \\ P w \\ P e_{tot} \end{Bmatrix} \quad (\delta \times 1)$
- • collect terms $\frac{d}{dx_x}$ and/or $\frac{d}{dx_y}$
Fluxes $\equiv \underline{F}_x = \begin{Bmatrix} P u_x \\ P u_x u_y \\ P u_x e_{tot} \end{Bmatrix} + \begin{Bmatrix} 0 \\ P \delta_{xy} \\ P u_x \end{Bmatrix} - \begin{Bmatrix} 0 \\ T_{xy} \\ T_{xy} u_y \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ q_x \end{Bmatrix}$

advective Flux \underline{F}_x^{adv} + diffusive Flux \underline{F}_x^{diff}

$$\begin{aligned}
 \underline{T}_{xy} &= [P \delta_{xy}]_{,i} \\
 \delta_{xy} &= \begin{cases} 1 & \text{if } x=y \\ 0 & \text{if } x \neq y \end{cases} \\
 \underline{P}_{xy} &= \begin{bmatrix} P & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & P \end{bmatrix} \cdot \nabla
 \end{aligned}$$

Note

$$\underline{F}_x^{adv} = u_x \underline{U} + P \begin{Bmatrix} \delta_{xy} \\ u_x \end{Bmatrix}$$

body terms

$$\underline{Q} = \begin{Bmatrix} 0 \\ b_y \\ b_y u_y + r \end{Bmatrix}$$

UCNS

$$\underline{U}_{,t} + \underline{F}_{x,x} = \underline{Q}$$

$$\underline{U}_{,t} + \underline{F}_{x,x}^{adv} + \underline{F}_{x,x}^{diff} = \underline{Q}$$

Note
 $\underline{F}_x(\underline{u})$
 $\underline{Q}(\underline{u})$

Unknowns

$$\begin{array}{ll}
 p : 1 & 6 \text{ unknowns} \\
 u_x : 3 & \text{equation of state} \\
 p : 1 & 1 \text{ new equation} \\
 e_{int} - T : 1 &
 \end{array}$$

Model: 1D scalar advection-diffusion

$$\underline{u} \rightarrow \phi$$

$$\phi_{,t} + f(\phi)_{,x} = g$$

$$\phi_{,t} + \underset{\substack{\uparrow \\ \text{adv}}}{a} \phi_{,x} - \underset{\substack{\uparrow \\ \text{diff}}}{K} \phi_{,xx} = g$$

Simplifying the U.C. N.S.

Euler Equations

$$\mu \rightarrow 0 \quad K \rightarrow 0 \quad \therefore F^{\text{diff}} \rightarrow 0$$

$$\underline{U}_{,t} + \underline{F}_{,x} = \frac{\partial \underline{F}}{\partial t}$$

Note:

in Finite Volume

$$F_1 = F$$

$$F_2 = G$$

$$F_3 = H$$

Incompressible

$$p = p_0 \text{ const}$$

continuity equation

$$F_{1,x} = [p u_x]_{,x} = p_0 u_{x,x}$$

$$p_{,t} = 0$$

$$\frac{\partial}{\partial t} 0 + p_0 u_{x,x} = 0$$

$$u_{x,x} = 0 \Leftrightarrow \nabla \cdot \underline{u} = 0$$

$$\tilde{P} = \frac{P}{p}$$

momentum
energy

$$u_{x,t} + u_y u_{x,y} + \tilde{P}_{,x} = \nu u_{x,yy} + \tilde{b}_x$$

energy

$$c_p T_{,t} + u_x c_p T_{,x} = \frac{1}{2} \nu S_{ij} S_{ij} + \tilde{K} T_{,xx} + \tilde{r}$$

viscous
dissipation

Note

This suggests that IC flow produces a divergence free velocity field

Steady

$$u_{,t} = 0$$

$$F_{,x} = \frac{\partial F}{\partial x}$$

Isothermal flow

neglects energy equation

Note

$$\frac{\text{compress}}{[p u_x u_y]_{,y}}$$

Product rule

$$u_x [p u_x]_{,y} + p u_x u_{x,y}$$

if IC

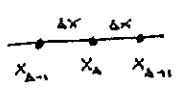
$$\frac{\text{IC}}{u_y u_{x,y}}$$

Numerical Methods

1) Finite Difference methods (FDM)

- replace continuous derivatives w/ discrete approx of neighboring pts

example



$$\phi_{,x}|_{x_n} \approx \frac{\phi|_{x_n+\Delta x} - \phi|_{x_n-\Delta x}}{2\Delta x} \approx \frac{\phi|_{x_n} - \phi|_{x_n-\Delta x}}{\Delta x}$$

central diff $\mathcal{O}(\Delta x^2)$ upwind $\mathcal{O}(\Delta x)$

$$\phi_{,xx}|_{x_n} \approx \frac{\phi|_{x_n+\Delta x} - 2\phi|_{x_n} + \phi|_{x_n-\Delta x}}{\Delta x^2}$$

$\phi_{,xy}$

2) Finite Elements method

- start w/ system in vector form place it in residual form

$$\underline{U}_{,ic} + \underline{F}_{,in} - \underline{F} = 0$$

- take the dot product of unit vector (w) with our equations

$$\underline{W} \cdot (\underline{U}_{,ic} + \underline{F}_{,in} - \underline{F}) = 0 \quad \text{1 equation}$$

- integrate over entire domain (Ω)

$$\int_{\Omega} \underline{W} \cdot (\underline{U}_{,ic} + \underline{F}_{,in} - \underline{F}) d\Omega = 0$$

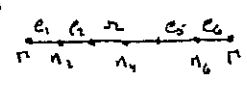
- look for solutions such that, for any \underline{W} , the above is true
- integrate by parts (most real cases) (Galerkin Weak Form)

$$\int_{\Omega} (\underline{W} \cdot \underline{U}_{,ic} - \underline{W}_{,in} \cdot \underline{F}_{,in} - \underline{W} \cdot \underline{F}) d\Omega + \int_{\Gamma} \underline{W} \cdot \underline{n}_n d\Gamma = 0$$

- This is a single scalar equation
- look for solutions using simple functions (Global Method)

DISCRETIZE INTO FE

1-D



- assume some functional variation over each element

$$\underline{U}(x) = \sum_{A=1}^{n_n} N_A(x) \underline{U}_A \quad \text{for each } A \text{ there is a } \begin{matrix} 3 \times 1 \text{ matrix} \\ 4 \\ 5 \\ 20 \\ 30 \end{matrix}$$

- perform an identical expansion on \underline{W}

$$\underline{W}(x) = \sum_{B=1}^{n_n} N_B(x) \underline{W}_B$$

- substitute into Weak Form

$$\int_{\Omega} \sum_{B=1}^{n_n} N_B \underline{W}_B \cdot \sum_{A=1}^{n_n} N_A \underline{U}_A - \sum_{B=1}^{n_n} N_{B,in} \underline{W}_B \cdot \underline{F}_{,in} - \sum_{B=1}^{n_n} N_B \underline{W}_B \cdot \underline{F} - \sum_{B=1}^{n_n} N_B \underline{W}_B \cdot \underline{F} (\sum_{A=1}^{n_n} N_A \underline{U}_A) d\Omega + \int_{\Gamma} \sum_{B=1}^{n_n} N_B \underline{W}_B \cdot \underline{F} (\sum_{A=1}^{n_n} N_A \underline{U}_A) \underline{n}_n d\Gamma = 0$$

- every term has summation of \underline{W}_B ,

$$\sum_{B=1}^{n_n} \underline{W}_B \cdot \left\{ \int_{\Omega} N_B \sum_{A=1}^{n_n} N_A \underline{U}_A - N_{B,in} \underline{F}_{,in} - N_B \underline{F} (\dots) d\Omega + \int_{\Gamma} N_B \underline{F} (\dots) \underline{n}_n d\Gamma \right\} = 0$$

$$0 = \sum_{B=1}^{n_n} \underline{W}_B \cdot \underline{G}_B \quad \text{therefore } \underline{G}_B = \text{zero for every } \underline{G}_B \quad B=1 \dots n_n$$

Arbitrary \underline{W}_B 's $(Nsd+2)n_n$ equations
 $(Nsd+2)n_n$ unknowns

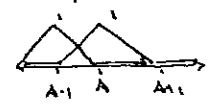
Note
 \underline{W} is a 5x5 vector of weight functions depending on position $x = W(x, y, z)$

Note
 Integration by Parts
 $\int_a^b b_{,x} dx = \int_a^b [ab]_{,x} dx - \int_a^b a_{,x} b dx$
 $= \int_a^b b_{,x} dx - \int_a^b a_{,x} b dx$

Nomenclature 2D



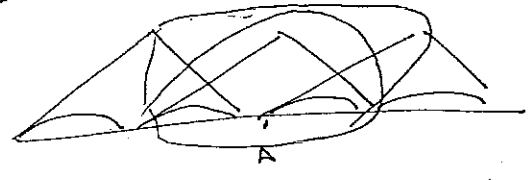
\underline{n}_n is normal to Γ a function of x, y, z
 N_A is shape function



n_n is the # of nodes

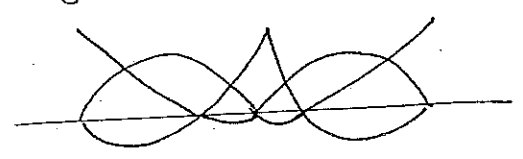
1/22/08

Aside @ Hierarchical Basis Function



Function A interacts with

Lagrange interpolant basis



Finite Element Method (Cont.)

- Element Method
- breaks the integral over Ω into the summation of integrals over each element

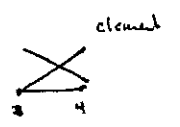
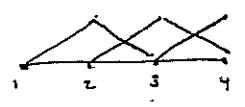
$$\int_{\Omega} \dots d\Omega = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \dots d\Omega_e$$

- on each element e we can form G_B^e

$$G_B^e = \int_{\Omega_e} \sum_{a=1}^{n_{doF}} N_a U_a^e - \int_{\Omega_e} \left(\sum_{a=1}^{n_{doF}} N_a U_a^e \right) \cdot \mathbf{N}_b \cdot \mathbf{F}_b(\dots) d\Omega_e + \int_{\Gamma_e} N_b \mathbf{F}_b(\dots) n_b d\Gamma_e$$

this term applies if element boundary is part of the real boundary

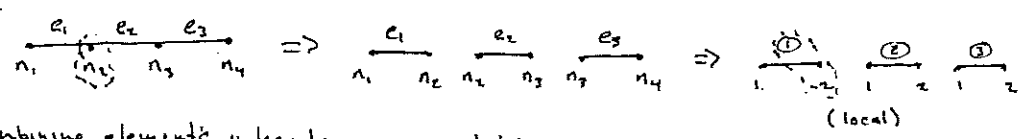
Note: global



- we want each element to look the same
- the solution inside an element depends on the nodes of that element

1/25/08

- subdomains



- recombining elements' integrals requires a table to communicate between local = global descriptions } connectivity array
- $ien(e_i\#, local\ node\ \#) = global\ node\ \#$
 (1) (2) (n1)

local node \ el#	1	2
1	1	2
2	2	3
3	3	4

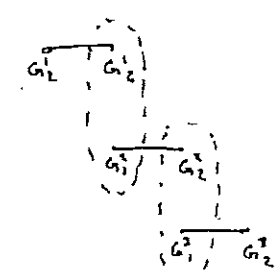
- For our example: local views of residual G

First element Ω_1

$$e_1 = \begin{cases} G_1^1 = \dots \\ G_2^1 = \dots \end{cases}$$

$$e_2 = \begin{cases} G_1^2 = \dots \\ G_2^2 = \dots \end{cases}$$

$$e_3 = \begin{cases} G_1^3 = \dots \\ G_2^3 = \dots \end{cases}$$



- Sum contributions element-by-element

$$G_B = \sum_{e=1}^{n_{el}} A_{nb}^e G_B^e$$

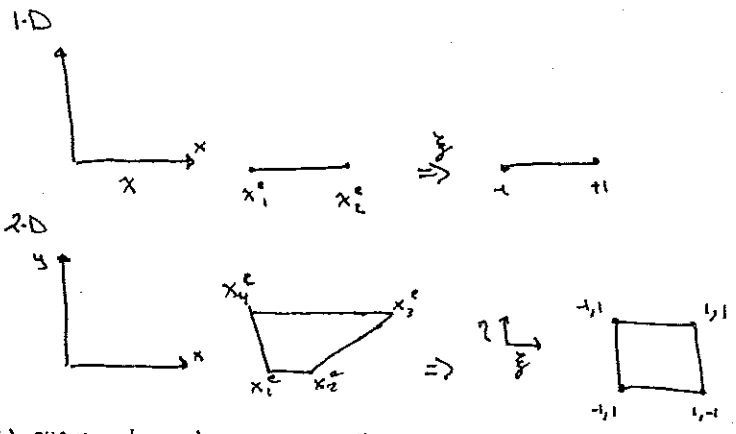
- after element
- ① $G_1 = G_1^1$ $G_2 = G_2^1$ $G_3 = 0$ $G_4 = 0$
 - ② $G_1 = G_1^1 + G_1^2$ $G_2 = G_2^1 + G_2^2$ $G_3 = G_3^2$ $G_4 = 0$
 - ③ $G_1 = G_1^1 + G_1^2 + G_1^3$ $G_2 = G_2^1 + G_2^2 + G_2^3$ $G_3 = G_3^2 + G_3^3$ $G_4 = G_4^3$

```

Code:
G_B = 0
Do e = 1, nel
  Do b = 1, nen
    G_B(ence, b) = G_B(ence, b) + G_B^e
  enddo
enddo
    
```

Finite Element Method (cont)

- Mapping between physical and parent domains



$$\begin{aligned}
 x(\xi) &\leftrightarrow \xi(x) \\
 \Omega^e &\leftrightarrow \bar{\Omega} \square \\
 d\Omega^e &\leftrightarrow d\Omega \\
 N_{b,i}(\xi) &= \frac{\partial N_b}{\partial x_i} = \frac{\partial N_b}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_i} \\
 &= N_{b,j} \xi_{j,i} \\
 \int_{\Omega^e} f(x) d\Omega^e &= \int_{\bar{\Omega}} f(x(\xi)) D(\xi) d\Omega \\
 \text{determinant of the Jacobian} & D(\xi) = \det(x, y) \\
 &= \det \left(\frac{\partial x_i}{\partial \xi_j} \right)
 \end{aligned}$$

- now every element can be written in a common way

$$\begin{aligned}
 G_b^e &= \int_{\Omega} \left[N_b(\xi) \left\{ \sum_{a=1}^{n_{eq}} N_a(\xi) U_{a,t}^e - \mathcal{F} \left(\sum_{a=1}^{n_{eq}} N_a(\xi) U_a^e \right) \right\} - N_{b,j} \xi_{j,i} \mathcal{F}_{j,i} \left(\sum_{a=1}^{n_{eq}} N_a(\xi) U_a^e \right) \right] D(\xi) d\Omega \\
 &+ \int_{\Omega} N_b F_{i,j} \left(\sum_{a=1}^{n_{eq}} N_a U_a^e \right) n_i D(\xi) d\Omega \quad \text{= change in every element}
 \end{aligned}$$

Note for Navier-Stokes

$$\begin{aligned}
 F_{i,j}(U_x, U_y) \\
 U_{i,t} &= \sum_{a=1}^{n_{eq}} N_{a,i}(\xi) \dot{U}_a^e \\
 &= \sum_{a=1}^{n_{eq}} \left\{ \frac{\partial N_a}{\partial \xi} \frac{\partial \xi}{\partial x_i} + \frac{\partial N_a}{\partial \eta} \frac{\partial \eta}{\partial x_i} - \frac{\partial N_a}{\partial \xi} \frac{\partial \xi}{\partial x_i} \right\} U_a^e
 \end{aligned}$$

Note:
Galerkin method above is the simplest method but it is unstable for advection dominated flow.
Thus we add a term to stabilize

- approximate the integration with quadrature

$$\begin{aligned}
 G_b^e &= \int_{\Omega} g_b^e(\xi) d\Omega + \int \dots d\Omega \\
 &\approx \sum_{k=1}^{n_{int}} \underbrace{\frac{g_b^e(\xi_k) W_k}{n_{int}}}_{\substack{\text{integral quad} \\ \text{eval @ } \xi_k \\ \text{K}^{\text{th}} \text{ quad of } k^{\text{th}}}}
 \end{aligned}$$

- The method basically "samples" over the complex function at as many quadrature points as is required to integrate G_b^e

$$\begin{aligned}
 G_b &= \sum G_b^e \\
 G_b(U, U_c) &= 0 \text{ when } U, U_c \text{ are "correct"}
 \end{aligned}$$

Stabilization of Galerkin

Weak form = Galerkin + $\sum_{e=1}^{n_{el}} \int_{\Omega^e} \hat{\mathcal{L}}^T W \mathcal{L}(U) d\Omega^e$
 differential operators abstract notation of strong form PDE

$$\begin{aligned}
 \mathcal{L}U &\equiv U_{,t} + F_{i,j} = U_{,t} + F_{i,j}^{adv}(U) + F_{i,j}^{diff}(U) \\
 \hat{\mathcal{L}}U &= \mathcal{L}U - \mathcal{F} = 0
 \end{aligned}$$

regardless of what $\hat{\mathcal{L}} N_b^e \mathcal{L}$ is, it multiplies a "residual", if U is sol'n we add zeros
 This means that this method is weighted residual method

N_b^e is W is factored out (elemental)
 $\mathcal{L}(U) = \text{residual}$

Finite Element Method (cont)

- Stabilization of Galerkin (cont)
- $\hat{\mathcal{L}}$ to suggest "linear" operator

$$\mathcal{L}U = U_{,t} + F_{1,1} + F_{1,2}$$

$$F_{1,1}^{adv}(U) = \frac{\partial F_{1,1}}{\partial U} \frac{\partial U}{\partial X_1}$$

↗ $\hat{\mathcal{L}} = \underline{A}_1 U_{,1} + \underline{A}_2(U)$
 Quasi-Linearization of Advective Flux

$$F_{1,1}^{diff}(U) = -[K_{12} U_{,2}]_{,1}$$

Note: this suggests
 $F^{diff}(U) \equiv -K_{12} U_{,2}$
 matrix multiplication

Building K_{12} from our definition of F^{diff}

$$F_{1,1} = 3 \times 3 \Rightarrow 5 \times 5 \text{ matrix}$$

$$K_{1,1} = 5 \times 5 \text{ matrix}$$

Likewise \underline{A}_i represents 3 5×5 matrices

$$\underline{A}_1 \equiv \frac{\partial F_{1,1}}{\partial U}$$

In this form

$$\mathcal{L}U = U_{,t} + \underline{A}_1 U_{,1} - [K_{12} U_{,2}]_{,1}$$

$$\hat{\mathcal{L}}U \equiv \left[\frac{\partial}{\partial t} + \underline{A}_1 \frac{\partial}{\partial X_1} - \frac{\partial [K_{12} \frac{\partial}{\partial X_2}]}{\partial X_1} \right] U$$

↓
linear operator

- Definition of $\hat{\mathcal{L}}$

- Galerkin least squares (GLS)

$$\hat{\mathcal{L}} = \mathcal{L}$$

- Streamline Upwind Petrov-Galerkin (SUPG) (often used)

$$\hat{\mathcal{L}} = A_1 \frac{\partial}{\partial x}$$

- Douglas-Wang (variational multiscale method)

$$\hat{\mathcal{L}} = -\mathcal{L}^* = \frac{\partial}{\partial t} + A_1 \frac{\partial}{\partial X_1} + \frac{\partial}{\partial X_2} [K_{12} \frac{\partial}{\partial X_2}]$$

2nd deriv terms

- Definition of $\underline{\mathcal{T}}$ (stabilization matrix)

beyond the scope of this class

It is possible to use analysis to estimate very well the functional behavior of each entity of the matrix to get stability = H.O. accuracy

→ Note: this forms a square of the advection term which is positive definite and is stable.
 • This is like diffusion just along the streamlines of the flow

Finite Element Method (cont)

Stability of Galerkin (cont)

Tracing the stabilization term through the element

Equation Formation

$$\hat{G}_b^e = G_b^e + \int_{\Omega} \sum_{\alpha=1}^{n_{eq}} N_{\alpha}(\beta) \tau(\beta) (U - \varphi) D(\beta) d\Omega$$

For SUPG

$$\hat{M}_b = A_{\alpha} \left(\sum_{c=1}^{n_{eq}} N_c(\beta) U_c \right) N_{b,j} \bar{J}_{j,m}$$

$$\int U = \sum_{a=1}^{n_{eq}} N_a U_{a,\tau} + A_{\beta} \left(\sum_{c=1}^{n_{eq}} N_c(\beta) U_c \right) \sum_{a=1}^{n_{eq}} N_{a,j} \bar{J}_{m,j} U_a - \text{Second deriv terms}$$

Review the entire process

when forming equations, we loop over elements

- loop over Net
 - ↳ integrate to get $\hat{G}_b^e(\beta^q)$ (N_{e1})
 - ↳ loop over quadrature points in each element
 - ↳ eval $\hat{g}(\beta^q)$ (N_{q?})
 - ↳ loop over shape functions @ each Qpoint in each element (N_{en})

Feed into assembly operator

↳ converts into global equation evaluations

$$G_B = A G_b^e$$

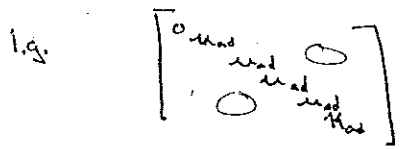
Alternative Stabilizers

Artificial diffusion Method

- not concerned with consistency
- get stability by adding diffusion
- the idea is to mimic physical diffusion

$$\hat{G}_b^e = G_b^e + \int_{\Omega} \left[N_{b,j_1} \bar{J}_{j_1} K_{j_1 j_2} N_{b,j_2} \bar{J}_{j_2} \right] D(\beta) d\Omega$$

K is a diffusivity matrix



$$max = \frac{\alpha U h}{2}$$

on a structured grid this produces upwind stencil

Finite Volume Method

Over simplification:

FVM can be thought of as FEM where $W = \text{constant}$ ($W_{,n} = 0$)

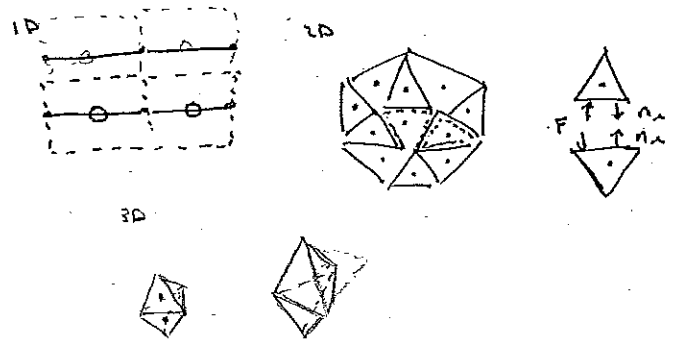
$$\int_{\Omega} (u_{,t} - q) d\Omega = \int_{\Gamma} F_n d\Gamma$$

Ω : is volume of element cell
 Γ : is the faces that close Ω

For this method work is applied to all volumes created by vertices.

Cell Centered Form (CC)

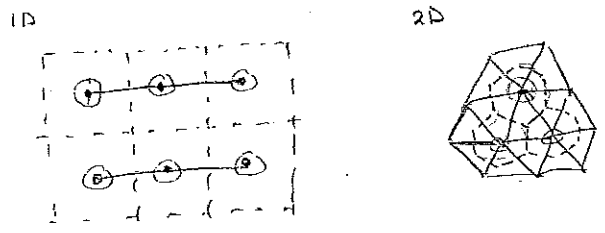
- the center of the C.V. is the sol'n point
- this method develops equations for each cell center by evaluating the Flux F at the boundary of the cell
- the normals have opposite signs therefore Fluxes across boundary leaves one element to the other
- In 3D we loop over the faces of the mesh and compute F_n and combine with n_n then send the result to cell center
- a second loop integrates over the cells ($u_{,t} - F$), 1 pt sample times the volume ok up to 2nd order accurate
- we get $G_B = 0$ where B is the cell center index (# cells)



2/6/08

Vertex Centered Form

- Down side
 - move faces to do flux through
 - up F-out work to compute volumes
- Up side
 - fewer non-linear equations to solve
- Data Structure (IVERT)
 - IVERT(edge^{id}, 2) \Rightarrow ^{nodes on} returns ends of edges



Finite Element a Finite Volume

$$G_B = G_{B'}'$$

- Both depend on the spatial distribution of conserved vector u and its time deriv. $u_{,t}$
- Regardless of choice of method this is a non-linear system of ODEs

$$G_B = \int_{\Omega} N_b \{ A_n Y_n - q \} - N_{b,n} F_n(Y) D d\Omega + \int_{\Gamma} N_b \Gamma_n \{ \{ Y, F \} \} D d\Gamma + \int_{\Omega} N_b F_n n_n D d\Omega$$

(change of variables UNDER the integral sign)

$$u(\xi) = \sum_{a=1}^{n_n} h_a(\xi) u_a \Rightarrow Y = \sum_{a=1}^{n_n} h_a(\xi) Y_a^e - \text{nodal values}$$

$$Y_{,t} = N_{b,t}(\xi) Y_{,t}^e$$

$$Y_{,n} = N_{b,n}(\xi) Y_{,n}^e$$

R&P: Hough a Hough: comp meth in applied Mech a Eng

ODE can be expressed w/ other vectors
consider arbitrary independent variable vector Y

We can write $Y = Y(\underline{u})$ and $\underline{u} = \underline{u}(Y)$

replace $u_{,t} = u_{,t} Y_{,t} = \frac{\partial}{\partial t} Y_{,t}$ (1st & 3rd component of u vector)

$$F_{,n} = F_{,n}(Y) = A_{,n} Y_{,n}$$

$$F_{,n}^{eff} = -[K_{ng} Y_{,n}]_{,n}$$

$$F_n = -K_{ng} Y_n$$

$$\frac{\partial u}{\partial t} = \frac{\partial Y}{\partial t} = \left[A_0 \frac{\partial}{\partial t} + \Delta_0 \frac{\partial}{\partial x} + \frac{\partial}{\partial x} (K_{ng} \frac{\partial}{\partial x}) \right] Y_{,t}$$

consistency check if $Y = u$
 $A_0 \Rightarrow I A_1 + K_{ng}$ also go back

$$f = A_n \frac{\partial}{\partial x}$$

$$u = \begin{cases} \delta \\ \delta u \end{cases}$$

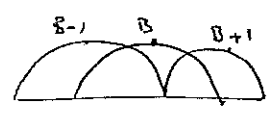
Examples

$$Y = \begin{Bmatrix} u \\ v \\ w \\ T \end{Bmatrix} = \begin{Bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{Bmatrix} \text{ or } \begin{Bmatrix} P \\ W \\ V \\ W \\ T \end{Bmatrix}$$

BC - can set Pressure (1 atm)
density becomes more constant
as we approach IC Flow

Thought Experiment

- Five Vectors in \hat{G}_B known
- for each node B we have 1 nonlinear ODE
- if 1-D, we couple to B+1, B, B-1
- for a particular B



$$\hat{G}_B(Y) a Y_{B,C} + b Y_{B-1,C} + c Y_{B+1,C} + d Y_B + e Y_{B-1} + f Y_{B+1} = 0$$

a-f are coefficients which can depend on Y

Problem:

- 1 nodes equations 2 = n nodes unknowns
- nonlinearity
- what happens from 1-D to 3-D
- more a-f coefficients
- what happens from scalar to vectors
- Y_i becomes vector
- a-f become matrix

$$\hat{G}_B = \hat{G}_B(\underline{Y}_{1,C}, \underline{Y}) = 0$$

\underline{Y} represents a collection of all the "nodal" values.

$$\underline{Y} \equiv (5 \times n) \times 1 \text{ vector} = \begin{Bmatrix} Y_A \end{Bmatrix}_{A=1}^n$$

$\hat{G}_B = 0$ when we have the correct $Y, Y_{1,C}$ vectors
 need to convert Non linear ODE to nonlinear Algebraic Eqn
 use a predictor multicorrector algorithm to relate $Y_{1,C} = Y$
 we can establish nodal level ($Y_{A,C}$ to Y_A)

Simplified Form (Explicit Method)

$$G_B = \overset{\text{temporal}}{G_B^t}(Y_{1,C}, Y) + \overset{\text{steady}}{G_B^s} = 0$$

$$\underline{M}(\underline{Y}) \underline{Y}_{1,C} + \underline{K}(\underline{Y}) \underline{Y} = 0$$

$$\underline{M}(\underline{Y}) \underline{Y}_{1,C} = -\underline{K}(\underline{Y}) \underline{Y}$$

we can factor out the $Y_{1,C}$ vector b/c it is linear in $Y_{1,C}$

Explicit Forward Euler

$$Y_{1,C} = \frac{Y^{n+1} - Y^n}{\Delta t} + \text{all else evaluated at } \underline{Y}^n$$

$$M(\underline{Y}^n) \frac{Y^{n+1} - Y^n}{\Delta t} = -K(\underline{Y}^n) \underline{Y}^n = -G_B^s(\underline{Y}^n)$$

when $Y \neq u$ to get explicit

$$M(\underline{Y}^n) \approx M^t(\underline{Y}^n) \Rightarrow Y_{\Delta}^{n+1} = Y^n + \Delta t \bar{A}_0^{-1} \{G_B^s(\underline{Y}^n)\}$$

\bar{A}_0 is locally averaged

Stability constraints on time step

usually prohibitively small (bad for viscous flows)

Implicit Generalized Alpha Method

Predictor multi correction
 ↑
 nonlinear method

Introduce iteration counter

the i th iteration in the process of moving from time step $n \rightarrow n+1$

$$\tilde{Y}_A^{n+1(i)} = \tilde{Y}_A^n + \Delta t \tilde{Y}_A^n + \Delta t \tilde{g}_A^{n+1(i)}(\tilde{Y}_A^{n+1(i)} - \tilde{Y}_A^n) \quad \text{OD1}$$

$$\tilde{G}_A^{n+1(i)}(\tilde{Y}_A^{n+1(i)}, \tilde{Y}_A^n) = 0 \quad \text{OD2}$$

Interpolations

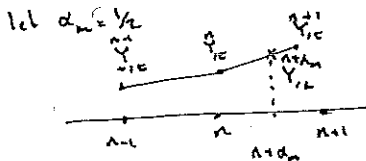
$$\text{OD3 } \tilde{Y}_A^{n+dm} = \tilde{Y}_A^n + \alpha_m (\tilde{Y}_A^{n+1} - \tilde{Y}_A^n)$$

$$\text{OD4 } \tilde{Y}_A^{n+d\alpha} = \tilde{Y}_A^n + \alpha_s (\tilde{Y}_A^{n+1} - \tilde{Y}_A^n)$$

all 4 ODEs are for all A
 and all 5 eqn/var in each vector

Remarks:

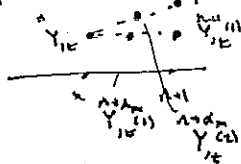
① in OD2 \tilde{Y}_A is evaluated @ $t = t_{n+dm} = t_n + \alpha_m \Delta t$



if $0 < \alpha_m < 1$ interpolation
 if else extrapolation

② α_m does have to equal α_s

③ both \tilde{Y}_A^{n+dm} and $\tilde{Y}_A^{n+d\alpha}$ are interpolations of the current iteration \tilde{Y}_A and will change with i until $\tilde{G}_A = 0$



• The 3 parameters α_m , α_s , γ

to get 2nd order accuracy

we give up 2 of the 3 free parameters

express the 3 in terms of 1 new parameter

that is the amplification factor in large time step limit

P_{∞}

$$\alpha_m = \frac{1}{2} \left(\frac{3 - P_{\infty}}{1 + P_{\infty}} \right)$$

$$\alpha_s = \frac{1}{1 + P_{\infty}}$$

$$\gamma = \frac{1}{2} + \alpha_m - \alpha_s$$

		α_m	α_s	γ
$P_{\infty} = 1$	trapezoidal rule midpoint rule	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$P_{\infty} = 0$	Gears method	$\frac{3}{2}$	1	1

any inbetween are possible

0.9 only a little temporal damping

There are times we want steady solutions

- 1) alternative: $Y_{,t} = 0$
 - solve Non linear Algebraic equation $G_B(Y)$ for all B. Finds local minima easily
 - not often used
- 2) integrate with large timesteps until $Y_{,t} = 0$
- 3) Expert system to guess + improve systems (evolutionary algorithms) (genetic algorithms)

If steady state use Backward Euler:

this is the most dissipative \rightarrow largest timestep in the case of non-linearity
Fastest route to steady state

$\alpha_m = 1$
 $\alpha_f = 1$ don't worry about
 $\delta = 1$ 2nd order accuracy

Still have system of equations to solve
Not explicit

$G_B(\tilde{Y}_{,t}, \tilde{Y}) = 0$ for all B OD2

How do we make this happen?

Predictor

assume we are given a $\tilde{Y}_{,t}^n$ and \tilde{Y}^n
 That satisfies OD1 + OD2
 Set $n=1$ (iteration count) at previous state
 predicts $\tilde{Y}^{n+1} = \tilde{Y}^n$
 Invert OD1

$$\tilde{Y}^{n+1} = \frac{\tilde{Y}^{n+1} - \tilde{Y}^n}{\Delta t} + \left(1 - \frac{1}{\delta}\right) \tilde{Y}^n$$

 Evaluate $\tilde{Y}_{,t}^{n+1}$ and \tilde{Y}^{n+1} using OD3 + OD4
 Evaluate $G_B(\tilde{Y}_{,t}^{n+1}, \tilde{Y}^{n+1}) \neq 0$

Note: these are either the IC or come from the previous converged state

Note: this is the best predictor

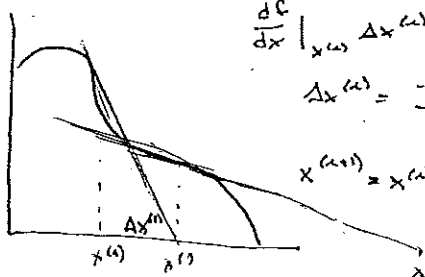
Use Newton's method as a corrector

$f(x) = 0$
 $f(x^{(n)}) + \frac{df}{dx} \Delta x^{(n)} = 0$

$\frac{df}{dx} \Big|_{x^{(n)}} \Delta x^{(n)} = -f(x^{(n)})$

$\Delta x^{(n)} = \frac{-f(x^{(n)})}{\frac{df}{dx} \Big|_{x^{(n)}}}$

$x^{(n+1)} = x^{(n)} + \Delta x^{(n)}$



Note: since guess was not perfect G_B is found $\neq 0$

Newton's Method our example

Make one variable "independent" one set by ODE

Y_A^{n+dp} = independent

Y_B^{n+dp} = dependent

Newton's Method

$$G_B \left(\begin{matrix} n+dp \\ Y_A \\ Y_B \end{matrix} \right)$$

non linear Residual (R_B)

$$\sum_{A=1}^{n+dp} \frac{G_B}{\Delta Y_A}$$

tangent matrix (M_{BA})

$$\Delta Y_A$$

increment to improve

solve for this

Note:

summed over nodal points

$$Y_A^{n+dp} = Y_A^{(n)} + \Delta Y_A^{(n)}$$

solve for this

plug back into first predictor

Qu. Linear Matrix problem @ Every time step

$$\sum_{A=1}^{n+dp} M_{BA} \Delta Y_A = -R_B$$

SxS for each element $(A \rightarrow B)$

$$M \Delta \underline{Y} = -\underline{R}$$

$S_{np} \times S_{np}$

$$\underline{M} = \{ M_{BA} \}$$

$$\underline{\Delta Y} = \{ \Delta Y_A^{n+dp} \}$$

$$\underline{-R} = \{ R_B \}$$

Solve linearized Algebraic System of equations

to get $\Delta \underline{Y}$

If we keep iterating $n = 1, 2, 3, \dots$

eventually $\Delta Y_A^{n+dp} \Rightarrow 0$ for each A

as $R_B \Rightarrow 0$ for each B

We need Y^{n+1} to go to next step

so we write ODE twice

$$\begin{aligned} Y_A^{n+dp} &= Y_A^n + \Delta Y_A^n \\ - Y_A^{n+dp} &= Y_A^n + \Delta Y_A^n \end{aligned}$$

$$\Delta Y_A^{n+dp} = \frac{dY_A}{dt} \Delta t$$

We can then say

$$Y_A^{n+1} = Y_A^n + \Delta Y_A^n$$

This allows us to continue to iterate w/ ODE

Going Deeper into Forming \bar{M}

Review: G_B in FEM

$$G_B = \int_{\Omega} \left\{ N_B \left[A_0 \overset{n+dn}{Y}_{1,c} - \overset{n+dn}{F} \right] - \underbrace{N_{B,12} \overset{n+df}{F} (Y_{1,c})}_{\text{very non linear}} \right\} d\Omega + \sum_{c=1}^{n_{el}} \int_{\Omega} \hat{N}_B^T \left\{ A_0 \overset{n+dn}{Y}_{1,c} + \overset{n+df}{A} \overset{n+df}{Y}_{1,c} - \overset{n+df}{F} \right\} d\Omega + \int_{\Gamma} N_B \overset{n+df}{F} n_x d\Gamma$$

Do not use integration by parts on convective term

$$\overset{4LHS}{G_B} = \int_{\Omega} \left\{ N_B \left(\overset{n+dn}{A_0} \overset{n+dn}{Y}_{1,c} - \overset{n+dn}{F} + \overset{n+df}{A} \overset{n+df}{Y}_{1,c} \right) + N_{B,12} \overset{n+df}{K_{12}} \overset{n+df}{Y}_{1,c} \right\} d\Omega + \dots + \int_{\Gamma} N_B \overset{n+df}{F} n_x d\Gamma$$

Hold A matrices fixed @ time t

usually dropped << 1

We are attempting to solve

$$\frac{dG_B}{dY_A} \Delta Y_A = -G_B$$

$$\underline{A} \underline{x} = \underline{b} \quad \text{solve for } x$$

2/12/08

Counter part at the Element Level FEM $\overset{c}{b}$
 Full Level FVM

$$\overset{4LHS}{G_b^c} = \int_{\Omega} \left\{ N_b \left[A_0 \sum_{c=1}^{n_{el}} N_c \overset{n+dn}{Y}_{1,c} + \overset{n+df}{A} \sum_{c=1}^{n_{el}} N_c \overset{n+df}{Y}_{1,c} - \overset{n+dn}{F} \right] + N_{b,12} \overset{n+df}{K_{12}} \sum_{c=1}^{n_{el}} N_c \overset{n+df}{Y}_{1,c} \right\} d\Omega + \int_{\Gamma} N_b^T \left[A_0 \sum_{c=1}^{n_{el}} N_c \overset{n+dn}{Y}_{1,c} + \overset{n+df}{A} \sum_{c=1}^{n_{el}} N_c \overset{n+df}{Y}_{1,c} - \overset{n+dn}{F} \right] d\Gamma$$

Terms we differentiate wRT $\overset{n+df}{Y}_{1,c}$

- ①, ③, ⑤ are already in terms of $\overset{n+df}{Y}_{1,c}$
- ② need work to express $\overset{n+dn}{Y}_{1,c}$ in terms of $\overset{n+df}{Y}_{1,c}$

Start with OD3

$$\overset{n+dn}{Y}_{1,c} = \overset{n}{Y}_{1,c} + dn \left(\frac{\overset{n+df}{Y}_{1,c} - \overset{n}{Y}_{1,c}}{\Delta t} \right)$$

Use OD1 inverted to look at delta increment

$$\left(\frac{\overset{n+df}{Y}_{1,c} - \overset{n}{Y}_{1,c}}{\Delta t} \right) = \frac{\overset{n+df}{Y}_{1,c} - \overset{n}{Y}_{1,c}}{\Delta t} = \frac{1}{\delta} \overset{n}{Y}_{1,c}$$

Invert OD4 to get new delta increment

$$\overset{n+df}{Y}_{1,c} - \overset{n}{Y}_{1,c} = \frac{\overset{n+df}{Y}_{1,c} - \overset{n}{Y}_{1,c}}{\delta}$$

Substitute terms

$$\overset{n+dn}{Y}_{1,c} = \overset{n}{Y}_{1,c} + dn \left\{ \frac{\overset{n+df}{Y}_{1,c} - \overset{n}{Y}_{1,c}}{\delta} \right\} = \frac{1}{\delta} \overset{n}{Y}_{1,c}$$

By the chain-rule, need to evaluate

$$\frac{\int_{\text{node } a}^{n+1} \frac{dY_{c,1e}}{dt}}{\int_{\text{node } a}^{n+1} \frac{dY_{c,1e}}{dt}} = \frac{d\alpha_m}{\alpha_f \delta \Delta t} \int_{\text{node } a}^{n+1} \frac{dY_{c,1e}}{dt}$$

$$= \frac{d\alpha_m}{\alpha_f \delta \Delta t} S_{ca} I_{S_{ca}}$$

if $c=a$

$$\frac{dY_c}{dY_a} = I_S = S \frac{I}{S_{SS}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Term ①

$$\frac{\int_{\text{node } a}^{n+1} [N_b A_0 \sum_{c=1}^{n+1} N_c \frac{dY_{c,1e}}{dt}]}{\int_{\text{node } a}^{n+1} \frac{dY_{c,1e}}{dt}} = N_b A_0 \sum_{c=1}^{n+1} N_c \frac{d\alpha_m}{\alpha_f \delta \Delta t} S_{ca} I_{S_{ca}}$$

$$= N_b A_0 N_a \frac{d\alpha_m}{\alpha_f \delta \Delta t}$$

) D d 0

Freeze A_0, A_1, K_{12}
By nature of S_{ca}

$$\frac{dG_b}{dY_a} = \int_0^{\infty} ($$

Term ②

$$\frac{\int_{\text{node } a}^{n+1} [N_b A_1 \sum_{c=1}^{n+1} N_c \frac{dY_{c,1e}}{dt}]}{\int_{\text{node } a}^{n+1} \frac{dY_{c,1e}}{dt}} = N_b A_1 \sum_{c=1}^{n+1} N_c S_{ca} I_{S_{ca}}$$

$$= N_b A_1 N_{a,1}$$

Term ③

$$\frac{\int_{\text{node } a}^{n+1} [N_b K_{12} \sum_{c=1}^{n+1} N_c \frac{dY_{c,1e}}{dt}]}{\int_{\text{node } a}^{n+1} \frac{dY_{c,1e}}{dt}} = N_b K_{12} N_{a,12}$$

Term ④ (like 1)

$$= \int_0^{\infty} N_b I A_0 N_a \frac{d\alpha_m}{\alpha_f \delta \Delta t}$$



Term ⑤ (like 2)

$$= \int_0^{\infty} N_b I A_1 N_{a,1}$$

All terms go in

$$\int_0^{\infty} () D d 0$$

added up they make M_{ab}^e matrix
requires quadrature like G_{12}

$$G_B = \sum_{c=1}^{n+1} A G_{12}^c \quad \text{len(e1, local) = global}$$

$$M_{AB} = \sum_{c=1}^{n+1} M_{ab}^c \quad \text{twice } 1) a \rightarrow A$$

$$2) b \rightarrow B$$

$$\tilde{M} = \{M_{a0}\} \quad R = \{G_{12}\}$$

$$\tilde{M} \Delta \tilde{\varphi} = \underline{G}_B$$

Boundary Conditions

Two Types

Dirichlet (Γ_g) and/or

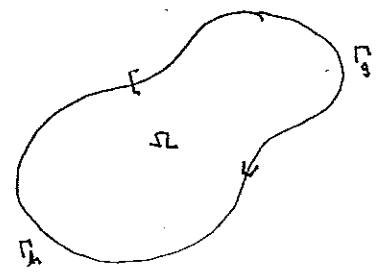
Set value of nodes
on boundary strong/exact

"essential"
"strong"

Neumann (Γ_h)

Set the flux on a boundary
to a value lets nodal
value take on whatever
it needs to to satisfy eqn

"Natural"
often are derivatives
"weak"



$$\Gamma = \overline{\Gamma_g + \Gamma_h}$$

Cover entire domain

Scalar

$$\phi(x_s, y_s) = g(x_s, y_s)$$

For x_s, y_s on Γ_g

is

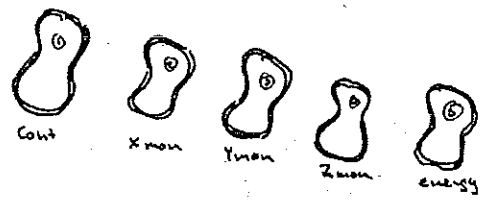
$$u \phi_{,x} - K \phi_{,xx}$$

$$h = -a\phi + K \phi_{,x}$$

convection

The UCNS system is more complicated
we have m equations at each node
and m unknowns

$3D \Rightarrow m=5$ Γ_g Γ_h



if all the boundary
is a flux, fixed
value may not be
controlled. Multiply
by const

$$\frac{\partial x}{\partial t} = \frac{\partial (x+z)}{\partial t}$$

Essential BC

think about what kinds of fields we want for BC for

collect fields into vector "q"

$$q = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix} = \begin{Bmatrix} P_3^* \\ U_r^* \\ U_s^* \\ U_t^* \\ T^* \\ \rho^* \end{Bmatrix}$$

P-scaled pressure
 velocity component
 "
 "
 Temperature
 density

r, s, t is a coordinate system that may/may not be aligned w/ x, y, z



$$U_r^* = C_1^r U_1 + C_2^r U_2 + C_3^r U_3 = C_i^r U_i$$

rotation components (cosines)

$$U_s^* =$$

$$U_t^* =$$

where

C_i^r are rotation cosines

It is possible to choose rot such that there is diagonal dominance

$$|C_i^r| \geq C_{i \neq r}^r \quad \text{gets us near } x, y, z \text{ system}$$

$$|C_i^s| \geq C_{i \neq s}^s$$

$$|C_i^t| \geq C_{i \neq t}^t$$

What we want to accomplish is:
 How does the setting of an (or more) of these BC change our sol'n variables

Write BCs in terms of Y variables

$$q(Y) = \begin{Bmatrix} P_3^* \\ U_r^* \\ U_s^* \\ U_t^* \\ T^* \\ \rho^* \end{Bmatrix} = \begin{Bmatrix} Y_1 \\ C_1^r Y_{r+1} \\ C_2^s Y_{s+1} \\ C_3^t Y_{t+1} \\ Y_5 \\ \frac{Y_6}{R Y_5} \end{Bmatrix}$$

$$Y = \begin{Bmatrix} P \\ U \\ T \end{Bmatrix}$$

$$U_r = C_1^r Y_2 + C_2^r Y_3 + C_3^r Y_4$$

$$P = PRT \Rightarrow P = \frac{P}{RT} = \frac{Y_1}{R Y_5}$$

2/15/08

We want to invert these relationships

$$q(q) = \begin{Bmatrix} P^* \\ U_r^* \\ U_s^* \\ U_t^* \\ T^* \\ \rho^* \end{Bmatrix} = \begin{Bmatrix} (U_r - C_1^r Y_2 - C_2^r Y_3) / C_1^r \\ \frac{1}{C_2^s} (U_s - C_1^s Y_2 - C_3^s Y_4) \\ \frac{1}{C_3^t} (U_t - C_1^t Y_2 - C_2^t Y_3) \\ T^* \\ \frac{P^*}{R Y_5} \end{Bmatrix}$$

P-scaled quantity

Links Hertz through the gas law

In general we may apply any combination of these boundary conditions to any point or section of the boundary

This tells us how to update our variables to be sure that we are always satisfying the BC's

(ITRBC)

Note: setting $p_{3,0}$ sets Y' thus you cannot see both BC

Implicit BCs (large time steps)

In FEM we formally define the solution space to be influenced by Dirichlet BC's.

if we set value @ a node
 ↳ weight functions at node are zero

In linear problems, this influences says

if node has sol'n set
 ↳ weight function is set to zero

$$\int \dot{q}_i^* \rightarrow \dot{c}w^* = 0$$

↳ this removes that equation to nodes from system

Theory:

Non linear BCs should be handled by constraining the weight function space V to belong to the target space of solution space \mathcal{Q}

This is accomplished by taking \dot{w} to be the variation of Y

$$w_{new} \leftarrow \frac{\partial \dot{q}_i}{\partial Y_j} w_j = \frac{\partial \dot{q}_i}{\partial Y_1} w_1 + \frac{\partial \dot{q}_i}{\partial Y_2} w_2 + \frac{\partial \dot{q}_i}{\partial Y_3} w_3 + \dots w_5$$

the w 's on the boundary are replaced by this transform

example: trapezoid

$$i=1: \quad \dot{q}_1^* \rightarrow w_1 \leftarrow \frac{\partial \dot{q}_1^*}{\partial Y_1} w_1 + \frac{\partial \dot{q}_1^*}{\partial Y_2} w_2 + \frac{\partial \dot{q}_1^*}{\partial Y_3} w_3 + \frac{\partial \dot{q}_1^*}{\partial Y_4} w_4 + \frac{\partial \dot{q}_1^*}{\partial Y_5} w_5$$

example:

$$\dot{q}_4^* = \frac{q}{c_{1r}} \dots$$

$$i=2: \quad w_2 \leftarrow \frac{\partial (\dot{q}_4^*)}{\partial Y_1} w_1 + \frac{\partial (\dot{q}_4^*)}{\partial Y_2} w_2 + \frac{\partial (\dot{q}_4^*)}{\partial Y_3} w_3 + \frac{\partial (\dot{q}_4^*)}{\partial Y_4} w_4 + \frac{\partial (\dot{q}_4^*)}{\partial Y_5} w_5$$

$$= \frac{-c_{1r}}{c_{1r}} w_3 - \frac{c_{1r}}{c_{1r}} w_4$$

$$i=3: \quad w_3 = \frac{-c_1^s}{c_2^s} w_2 - \frac{c_3^s}{c_2^s} w_4$$

$$i=4: \quad w_4 = \frac{-c_1^t}{c_2^t} w_2 - \frac{c_3^t}{c_2^t} w_3$$

$$i=5: \quad w_5 = 0$$

$$w_6 = p^* R w_5 = \frac{\partial \dot{q}_6}{\partial Y_5} = \frac{\partial (p^* R Y_5)}{\partial Y_5}$$

We just described 6 unique transformations that modify w to account for BC's that ~~we want~~ to act

$$\underline{w}^{new} = \underline{S} \underline{w}^{old}$$

Lets look @ the \underline{S} 's

a given node with pressure BC

$$S_p = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

this makes pressure @ the boundary = 0

$$w^{new} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} w_i \\ \vdots \\ w_b \end{bmatrix}$$

a given density

$$S_p = \begin{bmatrix} 0 & 0 & 0 & 0 & P^4 R \\ & 1 & & & 0 \\ & & 1 & & \\ & 0 & & 1 & \\ & & & & 1 \end{bmatrix}$$

a given velocity (u)

$$S_p = \begin{bmatrix} 1 & & & & \\ & 0 & & & \\ & & 0 & & \\ & 0 & & 0 & \\ & & & & 1 \end{bmatrix}$$

ask for a paper if you know none

These are called a modification to the weight function / space
How does this affect our algorithm

Roll back system to when we had w 's

$$\underline{M} \Delta \underline{Y} = -\underline{G}$$

$$\underline{W} \cdot \underline{M} \Delta \underline{Y} = -\underline{W} \cdot \underline{G}$$

To account for BC's we must replace

$\underline{w} \leftarrow \underline{S} \underline{w}$ in each node and each type of BC

say the i -th node has a B.C.

$$\underline{S} = \text{block diagonal} [I_{1 \times 1}, I_{2 \times 2}, \dots, S^{i \times i}, \dots, I_{n \times n}]$$

not how it is coded

$$\underline{W} \leftarrow \underline{S} \underline{W} \quad \Delta \underline{Y} \leftarrow \underline{S} \Delta \underline{Y}$$

$$[\underline{S} \underline{W}] \cdot \underline{M} [\underline{S} \Delta \underline{Y}] = -[\underline{S} \underline{W}] \cdot \underline{G}$$

$$\underline{W} [\underline{S}^T \underline{M} \underline{S}] \Delta \underline{Y} = \underline{W} [\underline{S}^T \underline{G}]$$

$$\underline{W} \underline{M}' \Delta \underline{Y} = \underline{W} \underline{G}'$$

$$\underline{M}' \Delta \underline{Y} = \underline{G}'$$

BC3 LHS
BC3 RES

- we build the G, M matrix
- Properly apply BC to M, G
- Sol'n are then consistently applied

The Code

Process

Main

Phasta.cc

levr = input_fform(inpfilename); reads in ascii data

input geom BC IC data

readnble declares arrays in readarrays

genblk reads data into point2x created
breaks elements into blocks

reads in ientp (4x8)

genblk copies to gensau - makes data pointer ^{memory}
sim to genblk

qread reads in initial condition

genint

Process

simhex

pre compute and tables all slope functions @ each quad point

Qpt location of point $\frac{x}{y}$ of quad points

Qwt weight of quad points

gen dat

genshp

shpTet

uses Qpt and assigns slope functions @ each Qpt

We scale the $\{U^B\}$, $\{U^{nom}\}$ that will make $\{R^A\} = 0$ and $\{R\} = 0$
 $\{P^{loc}\}$, $\{P_{,c}\}$

Newton's Method

"choose" independent variables

$\{U^B_{,c}\}$ and $\{P_{,c}\}$

Newton's lin mom residual

Mon $\sum_{B=1}^{n_{mp}} \left\{ \frac{\partial R^A}{\partial U^B_{,c}} \Delta U^B_{,c} + \frac{\partial R^A}{\partial P^B_{,c}} \Delta P^B_{,c} \right\} = -R^A$

CON $\sum_{B=1}^{n_{mp}} \left\{ \frac{\partial R^A}{\partial U^B_{,c}} \Delta U^B_{,c} + \frac{\partial R^A}{\partial P^B_{,c}} \Delta P^B_{,c} \right\} = -R^A$

$\left\{ K_{AB} \Delta U^B_{,c} + G_{AB} \Delta P^B_{,c} \right\} = -R^A$

$\left\{ D_{AB} \Delta U^B_{,c} + C_{AB} \Delta P^B_{,c} \right\} = -R^A$

LHSK

$$\begin{bmatrix} K \\ G \\ 0 \\ C \end{bmatrix} \begin{bmatrix} \Delta U_{,c} \\ \Delta P_{,c} \end{bmatrix} = - \begin{bmatrix} R \\ R \end{bmatrix}$$

LHS Sol'n RHS

In code
 $K \Rightarrow$ LHSK
 $\begin{bmatrix} D \\ C \end{bmatrix} \Rightarrow$ LHSK

Note

$K^{AB} = A^a K^{ab}$
 ↑ local

$K_{xy}^{ab} = \frac{\partial R^a}{\partial U^b_{,c}} = \sum_{c=1}^{n_{mp}} \left\{ \frac{\partial R^a}{\partial U^b_{,c}} \frac{\partial U^b_{,c}}{\partial x} + \frac{\partial R^a}{\partial U^b_{,c}} \frac{\partial U^b_{,c}}{\partial y} \right\}$
 chain rule
 $\rightarrow \alpha_f \delta \Delta t$ $\rightarrow \alpha_m \delta c \delta x$

$G_{,c}^{ab} = \frac{\partial R^a}{\partial P^b_{,c}} = \frac{\partial R^a}{\partial p_s} \alpha_f \delta \Delta t$

$D_{,c}^{ab} = \frac{\partial R^a}{\partial U^b_{,c}} = \frac{\partial R^a}{\partial U^b_{,c}} \alpha_f \delta \Delta t + \frac{\partial R^a}{\partial U^b_{,c}} \alpha_m$

$C_{,c}^{ab} = \frac{\partial R^a}{\partial P^b_{,c}} = \frac{\partial R^a}{\partial p} \alpha_f \delta \Delta t$

The Code: Shape Functions

In e3

Shp : Na.
 Shgl : Na, ξ .

$\xi = \xi$

n_{gauss} - # quadrature points

getshp ()

shp & shgl copied into shape and shdu

Interpolate the integrand

~~get~~

e3ivar

input

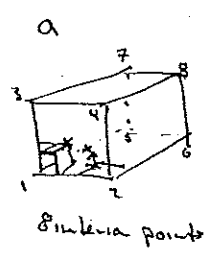
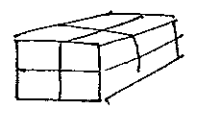
- yl = localize nodal values y, c
- ycl = " " y, c
- act = y, c vector
- scr = value of scalar that transpacting
- xl = coord of nodes of elements

γ (npro, nshl, nflow)
 counter 1-5 unknown VPE

γ (int) interpolates

$$Pres = Pres + shape \cdot \gamma_a(i, n, i)$$
 Pres has length npro

$$\frac{c}{a} = \frac{b}{a-b}$$



get them = equation of state
 calculates thermo constants

solution subscript
 l - nodal/local
 i - integral

e3 metric

$$\frac{\partial Na}{\partial \xi} = \frac{\partial Na}{\partial \xi} \frac{\partial \xi}{\partial x_i}$$

WdetJ = W * determinant of the Jacobian = Wg'D

Shg: global grad of shape function

e3sumit.

Plug in finite dimensional space

$$\begin{aligned} \begin{cases} \omega_n \\ q \end{cases} &= \begin{cases} \sum_{A=1}^{n_{np}} N_n^A \omega_n^A \\ \sum_{A=1}^{n_{np}} N_p^A q^A \end{cases} & \begin{cases} u_n \\ p \end{cases} &= \begin{cases} \sum_{C=1}^{n_{np}} N_n^C u_n^C \\ \sum_{C=1}^{n_{np}} N_p^C p^C \end{cases} \\ W & & Y & \end{aligned}$$

Plug into weak form (velbar)

$$\begin{aligned} \sum_{\alpha=1}^3 \sum_{A=1}^{n_{np}} \omega_n^A \left\{ \int_{\Omega} N_n^A \left\{ p \overset{(1)}{u_{n,1k}} + p \overset{(2)}{u_{n,2k}} - f_{n,1} \right\} + N_{n,1} \left\{ \overset{(3)}{T_{21}} - P \overset{(4)}{\delta_{21}} \right\} d\Omega \right. \\ \left. + \int_{\Gamma} N_p^A \left\{ \overset{(5)}{T_{21}} - P \overset{(6)}{\delta_{21}} \right\} d\Gamma \right. \\ \left. + \sum_e \int_{\Omega_e} \left\{ \overset{(7)}{T_m} u_{2,n} N_n^A, \overset{(8)}{d} + \overset{(9)}{T_o} N_{n,2} u_{2,1} + \overset{(10)}{T_{22}} N_{n,1} \overset{(11)}{d_{ic}} u_{1,k} \right\} d\Omega_e \right\} \\ + \sum_{A=1}^{n_{np}} q^A \left\{ - \int_{\Omega} N_p^A \overset{(12)}{u} d\Omega + \int_{\Gamma} N_p^A u_n n_{\alpha} d\Gamma \right\} \end{aligned}$$

rewrite

$$\begin{aligned} \sum_{\alpha=1}^3 \sum_{A=1}^{n_{np}} \omega_n^A \left\{ \dots \right\} + q \left\{ \dots \right\} \\ \underbrace{\quad}_{\tilde{R}_{3 \times n_{np}}} \quad \underbrace{\quad}_{\tilde{R}_{n_{np}}} \end{aligned}$$

arbitrary weight functions

$$\omega_n^A \Rightarrow \tilde{R}_{\alpha} = 0 \quad 3 \times n_{np} \text{ equations}$$

$$q^A \Rightarrow \tilde{R} = 0 \quad n_{np} \text{ equations}$$

> $\begin{cases} 4 \times n_{np} \text{ equations for } \\ 4 \times n_{np} \text{ unknowns } \end{cases} \begin{Bmatrix} u_x \\ u_y \\ z \\ p \end{Bmatrix}$ @ each node

3/4/08

These equations are assembled from element level integrals

Skip time integration

localization

$\left. \begin{matrix} n + d_n \\ n + d_p \\ e, n \end{matrix} \right\}$ indices

Note $\hat{\quad}$ means no sum

MOM

$$\tilde{R}_{\alpha}^A \neq 0 = \int_{\Omega} \left[N_n^A \left\{ p \overset{(1)}{u_{\alpha,1k}} + p \overset{(2)}{u_{\alpha,2k}} - f_{\alpha,1} \right\} + N_{n,1}^A \left\{ \overset{(3)}{T_{21}} - P \overset{(4)}{\delta_{21}} \right\} + N_{n,2}^A \overset{(5)}{T_{22}} \overset{(6)}{d_k} u_{\alpha,1k} + N_{n,1}^A \overset{(7)}{T_c} u_{\alpha,k} \right] D d\Omega$$

CON

$$\tilde{R}^A = \int_{\Omega} N_p^A \overset{(8)}{u_k} D d\Omega + \int_{\Gamma} N_p^A u_k n_k D_r d\Gamma$$

CHHS

$$\tilde{R}^A = \int_{\Omega} N_p^A u_{k,k} D d\Omega + \int_{\Gamma} N_p^A \frac{\hat{T}_{m,p}}{p} \hat{T}_k D d\Gamma$$

Incompressible Formulation

Common
 - Incompressible
 - compressible

Strong Form (PDE)

$$u_{i,i} = 0$$

$$\rho_{,i} = \rho_{i,i} + \rho_{ij} u_{i,j} + \rho_{,ii} - \tau_{ij,j} - f_i = 0_{,i} + w_i$$

Residual of NS BOM

Multiply by weight function equivalent to $(w, \delta\psi - F)$

Integrate over domain

Weak form

$$0 = \int_{\Omega} \{ q u_{i,i} + w_i \rho_{,i} \} d\Omega$$

$$= \int_{\Omega} \{ q u_{i,i} + w_i \{ \rho_{i,i} + \rho_{ij} u_{i,j} + \rho_{,ii} - \tau_{ij,j} - f_i \} \} d\Omega$$

Integration by parts is on stress and continuity term only

$$0 = \int_{\Omega} \{ \underbrace{q}_{(1)} u_{i,i} + w_i \{ \underbrace{\rho_{i,i}}_{(2)} + \underbrace{\rho_{ij} u_{i,j}}_{(3)} - \underbrace{f_i}_{(4)} \} + w_{i,j} \{ \underbrace{\tau_{ij}}_{(5)} - \underbrace{\rho_{,ij}}_{(6)} \} \} d\Omega + \int_{\Gamma} \{ \underbrace{q u_i n_i}_{(7)} - w_i \{ \underbrace{\tau_{ij}}_{(8)} - \underbrace{\rho_{,ij}}_{(9)} \} n_j \} d\Gamma$$

$$+ \sum_{e \in \Gamma} \int_{\Omega^e} \{ \underbrace{\tau_m}_{(10)} u_{i,j} w_{i,j} + \frac{\underbrace{q_{,ii}}_{(11)}}{\rho} \} d\Omega + \tau_c w_{i,i} \underbrace{u_{i,j}}_{(12)} \} d\Omega$$

(10) stabilized momentum (11) stabilizes continuity

$$+ \sum_{e \in \Gamma} \int_{\Omega^e} \{ w_i \underbrace{\rho_{ij} u_{i,j}}_{(13)} + \underbrace{\tau}_{(14)} \left\{ \frac{\rho_{,ii}}{\tau_m} \right\} w_{i,j} \frac{\rho_{,ii}}{\tau_m} u_{i,i} \} d\Omega$$

(14) conservation restoration

- 1) $\hat{u}_3 = -\tau_m \frac{q_{,ii}}{\rho}$
- 2) $\bar{u}_3 = u_3 + \hat{u}_3 = u_3 - \frac{\tau_m d}{\rho}$

Dimensional considerations

1) terms multiplying q

$$\textcircled{1} \int_{\Omega} q u_{i,i} d\Omega \quad \frac{1}{L} \frac{1}{T} L^3 \Rightarrow \frac{L^2}{T} = [q]$$

$$\textcircled{2} \int_{\Gamma} q u_i n_i d\Gamma \quad \frac{1}{T} L^2 \Rightarrow \frac{L^2}{T}$$

$$\textcircled{3} \int_{\Omega} \tau_m \frac{q_{,ii}}{\rho} d\Omega \quad \frac{L^3}{L M} \left[\frac{M}{L^3} \right] \left[\frac{L^2}{T^2} \right] \Rightarrow \frac{L^2}{T^2} \text{ if } [\tau_m] = T$$

note for $\textcircled{10} = \textcircled{11}$

$$[q] = \frac{W M}{L^2}$$

2) terms multiplying w_i

$$\tau_c, \delta, \gamma \quad [w_i] = \frac{M L}{T^2}$$

$$\textcircled{12} \int_{\Omega} \tau_c w_{i,i} u_{i,j} d\Omega \quad T \frac{1}{T} \frac{1}{L} \frac{M}{L} \frac{L}{T} L^3 = \frac{M L}{T^2}$$

$$\textcircled{13} \int_{\Omega} \tau_c w_{i,j} u_{i,j} d\Omega \quad [\tau_c] \frac{W}{L} \frac{L}{T L} L^3 = [\tau_c] \frac{W L^2}{T^2} = \frac{W M L}{T^2} \text{ if } [\tau_c] = \frac{M}{L T}$$

$$\textcircled{14} \int_{\Omega} w_i \rho_{ij} u_{i,j} d\Omega \quad \frac{W M}{L} \frac{L}{T} \frac{L}{T L} L^3 = \frac{M L}{T^2}$$

$$[\tau] \frac{M}{L^3} \frac{L}{T} \frac{L}{T} \frac{1}{L} \frac{W M}{L} \frac{L}{T} \frac{L}{T L} L^3 = \frac{W M L}{T^2} \text{ if } [\tau] = \frac{T^2 L^3}{M}$$

In the code

$$\tau_c = \frac{\rho}{8 \tau_m g_{ii}} = \left[\frac{M}{L T^2} \right] \tau_m = \sqrt{\left(\frac{2 \tau_c}{\delta t} \right)^2 + u_k u_k g_{kk} + C_2 v^2 g_{kk} g_{kk}} = [T]$$

$$g_{ii} = \sum_k g_{m,k} g_{m,k}$$

$$\tau = \frac{\tau_m}{\sqrt{g_{ii} g_{ii} g_{ii}}} \Rightarrow \left[\frac{T^2 L^3}{M} \right] \left(\text{inverses time scale} \right) + \left(\text{convection} \right) \cdot \left(\text{diffusion} \right)$$

(time step) (dt through el) (dt across element)

In the code, the integral is coded in a different way (velban formulation)

Note $\textcircled{11}$ $\frac{q_{,ii} \tau_m \rho_{,ii}}{\rho} = -g_{ii} \hat{u}_i$ $\textcircled{14}$

add to $\textcircled{1}$ $-g_{ii} (u_i + \hat{u}_i) = -g_{ii} \bar{u}_i$

sim. $\textcircled{13}$ $\textcircled{3}$ $w_i \{ \rho u_{i,j} u_{i,j} + \rho \hat{u}_i u_{i,j} \}$
 $= w_i \{ \rho \bar{u}_i u_{i,j} \}$

$\rho \hat{u}_i \equiv \delta_i$
 $\tau \delta_i w_{i,j} \rho_{,ii} u_{i,j}$

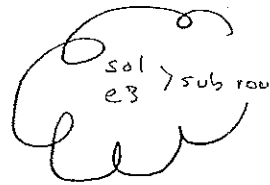
2/21/08

Recall what work we need to do

1) Build LHS & RHS of equation system

$$\begin{matrix} M_{k_0}^c & G_b^e \\ \Downarrow \triangle & \Downarrow \triangle \\ M & G \cdot P \\ = & = \end{matrix}$$

2) solve the equation system



PHASTA

main.c ^{calls} → Phasta.cc

Phasta.cc

input_fform.cc

Process.f

gendat.f

geombe.dat.1

restant.o.1

Iterdrv.f

solgmrf

Herbc.f

ArBC.f

Solgmrf

Solgmrs

elmgms

loopover-blocks

dsigmr

local

e3.f

loop over integration points

e3ivar.f

e3mtrx.f

e3comul.f

r2

e3aou.a.f

e3visc.f

e3ls.

e3os

e3wmlt.f

loops over somethings

end loop quad points

reads the input deck

rest of program

read digital data / restape data for optimal FEM lck

coord connect BC

IC data

loops over time steps solves N-S

for current iteration / linearization solve Ax = Mb

satisfy the BC

sparse solver

Form Ax=b

ability to break element loops into smaller loop smaller # elements per block

$x_b \rightarrow x_b^t$ $x \rightarrow x_t$ take global x-vec -> local x-vec

$$\int_{\Omega} = G_b^e + P_{k_0}^e$$

samples any comping in the integrand

calculates \underline{A}_0 \underline{A}_n

build flux vector \underline{F}_n

vector built which multiplies N_b or $N_{b,n}$

takes body force \underline{F} into \underline{c}

build dir flux \underline{F}_n puts into \underline{r}

stabilization \underline{SOP}

weight function multiply complete integrand $\underline{F}_n W$

APES HZONS RUN (??)

get/set

A: as a assembly iteration

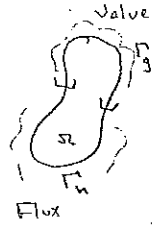
$$\int N_b \{ \} + N_{b,n} \{ \}$$

$r_2 \{16:20\}$ $r_2 \{1:5\}$ $N_b \{F_1\}$
 $r_2 \{6:10\}$ $r_2 \{6:10\}$ $N_{b,n} \{F_1\}$
 $r_2 \{11:15\}$ $N_{b,n} \{F_1\}$

$$G_b^e = \sum_{n=1}^{nL} \int_{\Omega} (F^*) W^n$$

Recall: we may set 2 types of BC data

- 1) essential - fixes a nodes value (u, v, w, T, P, p) $\rightarrow q$
 - 2) Natural - fixes flux through a boundary element $\rightarrow \tilde{r}$
- 3D Face
2D Edge



For Systems we can't always set just one or the other so we need to break Γ_h into 2 parts

- 1) Part where we set flux
- 2) Part where we compute "consistent flux" yet to be defined

Boundary Integral

$$\int_{\Omega_n} N_b \left\{ \underbrace{p u_n \tilde{u}}_{\text{mass flux}} + p \left\{ \begin{matrix} \tilde{u}_n \\ \tilde{v}_n \\ \tilde{w}_n \end{matrix} \right\} - \underbrace{\left\{ \begin{matrix} \tilde{u}_n \\ \tilde{v}_n \\ \tilde{w}_n \end{matrix} \right\}}_{\text{consistent flux}} + \left\{ \begin{matrix} \tilde{p}_n \\ \tilde{r}_n \end{matrix} \right\} \right\} D_n d\Omega_n$$

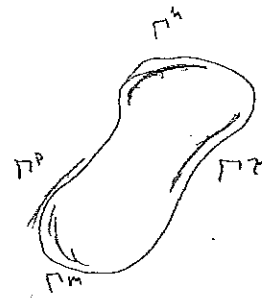
$$= \int_{\Omega_n^m} N_b h^m \tilde{u} D_n d\Omega_n + \int_{\Omega_n^h} N_b p u_n \tilde{u} D_n d\Omega_n$$

Pressure flux

$$+ \int_{\Omega_n^p} N_b h^p \left\{ \begin{matrix} \tilde{u}_n \\ \tilde{v}_n \\ \tilde{w}_n \end{matrix} \right\} D_n d\Omega_n + \int_{\Omega_n^r} N_b p \left\{ \begin{matrix} \tilde{p}_n \\ \tilde{r}_n \end{matrix} \right\} D_n d\Omega_n$$

$$+ \int_{\Omega_n^x} -N_b \left\{ \begin{matrix} \tilde{u}_n \\ \tilde{v}_n \\ \tilde{w}_n \end{matrix} \right\} D_n d\Omega_n + \int_{\Omega_n^z} N_b \left\{ \begin{matrix} \tilde{p}_n \\ \tilde{r}_n \end{matrix} \right\} D_n d\Omega_n$$

$$+ \int_{\Omega_n^k} N_b \left\{ \begin{matrix} \tilde{p}_n \\ \tilde{r}_n \end{matrix} \right\} D_n d\Omega_n + \int_{\Omega_n^q} N_b \left\{ \begin{matrix} \tilde{p}_n \\ \tilde{r}_n \end{matrix} \right\} D_n d\Omega_n$$



"Floating Flux BC"

This process computes the flux through the entire boundary using prescribed data in place of consistent flux data, where provided.

h vales in code \rightarrow BCs integer that determines when to use cons, v per. IBES In compress code e3bvar, l calculates each h by interpolating BCs compute γ, γ_n needed for $\gamma_n q_n$ e3b.p \rightarrow IBES array set with to use convh

In practice we can apply this condition either at the element level or globally

$$\bar{M} = \sum_{e=1}^{n_{el}} \mathbf{A}^e \mathbf{M}_{ns}^e$$

$$\mathbf{G} = \sum_{e=1}^{n_{el}} \mathbf{A}^e \mathbf{G}_a^e$$

\mathbf{A} adds all element contributions to form the global nodes

Natural BC

Local level

$$\bar{G}_b^e = \int_{\Omega} \mathbf{N}_b \left\{ \mathbf{A}_b \cdot \mathbf{Y}_{1,c} \dots \right\} + \int_{\Omega} \mathbf{N}_b \mathbf{F}_2 \mathbf{n}_x \mathbf{D}_r d\Omega_r$$

only if touches Real BC

We need to understand this BC code creates a list of elements (faces) that need to be evaluated

$$\int \mathbf{N}_b \left(\mathbf{F}_{adv} + \mathbf{F}_{diff} \right) \mathbf{n}_x \mathbf{D}_r d\Omega_r$$

further

$$\int \mathbf{N}_b \left(\rho \mathbf{u}_x \mathbf{u}_x + p \left\{ \frac{\rho}{\rho_0} \right\} = \left\{ \frac{\rho}{\rho_0} \right\} + \left\{ \frac{q}{T} \right\} \right) \mathbf{D}_r d\Omega_r$$

$\mathbf{n} = \text{normal}$

This suggests several "Natural" fluxes BC's

- 1) Mass Flux: $\int \rho \mathbf{u}_n$
if we prescribe we set $\rho \mathbf{u}_n = h^m$ on Γ_h^m
- 2) Natural Pressure (weak)
 $P = h^p$ on Γ_h^p
- 3) Normal viscous flux (traction)
 $\tau_{3n} = h^v$ on Γ_h^v
- 4) Heat Flux
 $-q_n = h^h$ on Γ_h^h

Note: convention states that h^* is a given value

Note: could be a function of space

Comments:

[1] This getting $G = -D^T$ (or vice versa) is tedious but it gives some advantages

Gresho's Book ← mathematical

[2] Saves space and work ΔU ΔP
in $e3xxx.f$ x60c [D:C] → stored in LHSP

all element
[2x2]
computations
take place
in these
files

2/20/09

[3] Later we will see that we use iterative solvers
⇒ matrix vector products

$$\vec{r} = A\vec{p}$$

$$\begin{bmatrix} K & G \\ D & C \end{bmatrix} \begin{bmatrix} \Delta U \\ \Delta P \end{bmatrix}$$

↳ this shape

How do we do that if we did not store G ?
→ Hardware the AP product to use $-D^T$

$$\underline{K} \underline{\Delta U} + \underline{G} \underline{\Delta P} = \underline{K} \underline{\Delta U} - \underline{D}^T \underline{\Delta P}$$

[4] Note the N_i^a , the i is just there to help us place the term in the 3×3 K_{ij}^{ab} matrix or D_j (1×3 matrix)

We use the same shape function regardless of the value of i . Furthermore, N_i^a is also the same for stabilize methods like we discussed before.

There are FEM methods, mixed methods, where the polynomial order of velocity is higher than pressure.

boxed {}

[5] It is fairly common when studying incompressible flow to study an auxiliary equation that is derived from the already discretized system,

$$\begin{bmatrix} \underline{\underline{K}} & \underline{\underline{G}} \\ \underline{\underline{G}}^T & \underline{\underline{C}} \end{bmatrix} \begin{bmatrix} \underline{\underline{\Delta U}} \\ \underline{\underline{\Delta P}} \end{bmatrix} = \begin{bmatrix} -\underline{\underline{R}} \\ -\underline{\underline{PC}} \end{bmatrix} \quad \text{Fully coupled momentum-continuity system} \\ \text{FCMC}$$

Extract momentum equation out

$$\underline{\underline{K}} \underline{\underline{\Delta U}} + \underline{\underline{G}} \underline{\underline{\Delta P}} = -\underline{\underline{R}}$$

imagine we know $\underline{\underline{K}}^{-1}$

$$\underline{\underline{\Delta U}} = \underline{\underline{K}}^{-1} [-\underline{\underline{R}} - \underline{\underline{G}} \underline{\underline{\Delta P}}] \rightarrow \text{plug } \underline{\underline{\Delta U}} \text{ into continuity eq.}$$

Second cont

$$-\underline{\underline{G}}^T \underline{\underline{\Delta U}} + \underline{\underline{C}} \underline{\underline{\Delta P}} = -\underline{\underline{R}}$$

$$-\underline{\underline{G}}^T \underline{\underline{K}}^{-1} [-\underline{\underline{R}} - \underline{\underline{G}} \underline{\underline{\Delta P}}] + \underline{\underline{C}} \underline{\underline{\Delta P}} = -\underline{\underline{R}} \quad \leftarrow \text{rearrange}$$

$$\underbrace{\left[\underline{\underline{G}}^T \underline{\underline{K}}^{-1} \underline{\underline{G}} + \underline{\underline{C}} \right]}_{\text{Discrete pressure Poisson equation}} \underline{\underline{\Delta P}} = -\underline{\underline{R}} - \underline{\underline{G}}^T \underline{\underline{K}}^{-1} \underline{\underline{R}}$$

Discrete pressure Poisson equation $(\nabla^2 P) = \nabla \cdot \text{mom}$

we don't work w/ this.

DPPE

Instead, we work with the DPPE

The incompressible code (PHASTA) calls a third party library that solves DPPE as a preconditioner to FCMC.

Why not just solve FCMC?

Eigenvalue spread between continuity equation (ΔP) and momentum equation (ΔU) so large that FCMC would have to be solved to very tight tolerances.

The "solution" solve DPPE (worse tolerance) then solve FCMC to loose tolerance but "constrained" to not violate DPPE solution

	1	2	3
1		2	3
2	4		5
3	7	8	

$$[6] \quad K_{ij}^{ab} = X k e b a \left(\overset{e}{\text{lines}}, \overset{ij}{1:9}, \overset{a}{\text{inshgl}}, \overset{b}{\text{inshgl}} \right)$$

nshgl = number of shape functions in the element
 i.e. tet, linear = 4

Assemble this fill sparse

$$xkete \text{ Assemble} \Rightarrow \text{lhs } k(1:nmb\text{-tot}, 9)$$

-i the number of non-zeros in the matrix

$$XGOC \Rightarrow \text{lhs } P(1:nmb\text{-tot}, 4)$$

[7] Above we mentioned \underline{k}^{-1} . We can't really get this but instead we use $\underline{\underline{k}}^{-1} \approx \underline{\underline{k}}^{-1}$ typically $\underline{\underline{k}} = \underline{\underline{k}}^l$ lumped diagonal of the real \underline{k} .
 Trivial to invert.

Note: \exists something sort of in between compressible and incompressible flow.

\Rightarrow pseudo-compressible flow
 only really sensible for acoustics

The idea is to put $\frac{\partial p}{\partial t}$ back into continuity equation

$$\frac{\partial p}{\partial t} = \frac{\partial p}{\partial p} \frac{\partial p}{\partial t}$$

$$\alpha \frac{\partial p}{\partial p} \frac{\partial p}{\partial t} + p u_{ii} = 0$$

$$\frac{\alpha}{p} \frac{\partial p}{\partial t} + u_{ii} = 0$$

Extra time term in continuity equation.

Probably push $\alpha f_p = \alpha E u$

$$\alpha m_p = \alpha m u$$

$$\gamma_p = \gamma_u$$

This technique is also pursued as a relaxation technique.

$$\frac{\partial p}{\partial t} \rightarrow \frac{\partial p}{\partial \tau} \quad \leftarrow \text{pseudo time}$$