Emphasis: Algorithms & Implementations

Syl

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

Fortran 08 ← look at this

Notation

Equations: Unsteady Compressible N-S (UCNS)

Conservative form

shorthand notation

Let ϕ be some variable

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

$\phi_{,1} = \frac{d\phi}{dx} = \frac{d\phi}{dx_1}$

$\phi_{,2} = \frac{d\phi}{dy} = \frac{d\phi}{dx_2}$

$\phi_{,12} = \frac{d^2\phi}{dx\,dy} = \frac{\partial^2\phi}{\partial x_2 \partial x_1}$

$\phi_{,33} = \frac{d^2\phi}{dx_3^2}$

$\phi_{,12} = \frac{d^2\phi_1}{dx_1 dx_2}$

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

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

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

u_i where $i = 1, 2$ or 3

Continuity Equation

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

$$\frac{d\rho}{dt} + \vec{\nabla} \rho \underset{\sim}{u} = 0$$

Momentum Equation

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

$$\frac{d}{dt} \rho u + \nabla \rho \underset{\sim}{u}^2 + \frac{dP}{dx_i} = \nabla \tau + \frac{dF}{dx_i}$$

Energy Equation

$$[\rho e_{tot}]_{,t} + \sum_{i=1}^{3} [\rho u_i e_{tot}]_{,i} + \sum_{i=1}^{3} [u_i P]_{,i} = \sum_{i=1}^{3}\sum_{j=1}^{3} [\tau_{ij} u_j]_{,i} - \sum_{i=1}^{3} b_i u_i + r - \sum_{i=1}^{3} q_{i,i}$$

where

ρ = density

P = pressure

e_{tot} = internal energy + KE

$= e + \frac{u_i u_i}{2}$

τ_{ij} = viscous stress Tenson

$\sim \mu S_{ij} = \mu [u_{i,j} + u_{j,i} + \lambda \delta_{ij}$

b_i = body Force

q_i = heat flux vector

$-k T_{,i}$

r = heat supply

Note:

in summations the summing index is repeated

⇒ hence if index repeats, we sum

Contracted / Dummy index

Free Index: mentioned once

01/15/08

Indecial / Einstein Summation Convention

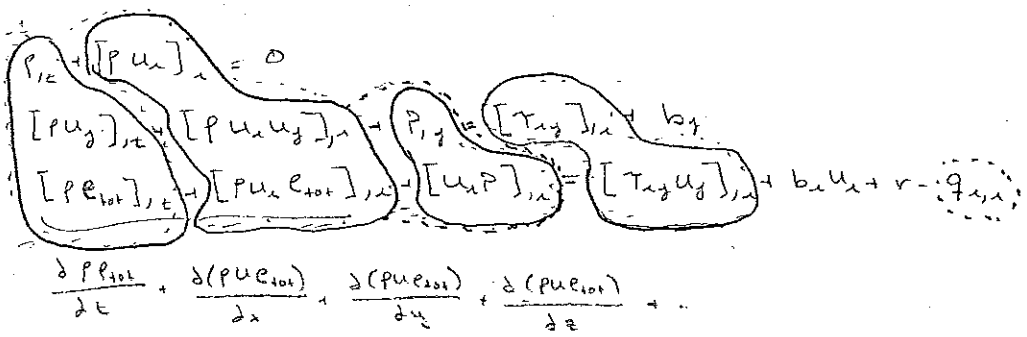
1) indices $(x, y, z, l \ldots)$
   they can appear no more than twice in a term

2) if they appear once ⇒ Free Index
   - they are free to be 1, 2, or 3
   Hence 3 distinct equations for momentum

3) if they appear twice ⇒ Contracted Index
   they must be summed over that index
   3 terms added
   dont need "Σ" they are implied

4) we have error if all terms dont have the same free indices

Ex



Further Notational Abstraction

- collect $\frac{\partial}{\partial t}$ terms $\equiv \underset{\sim}{U}$: $\begin{Bmatrix} \rho \\ \rho u \\ \rho e_{tot} \end{Bmatrix} = \begin{Bmatrix} \rho \\ \rho u_y \\ \rho e_{tot} \end{Bmatrix} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_{tot} \end{Bmatrix}$ (5×1)

- - - collect terms $\frac{\partial}{\partial x_x}$ and/or $\frac{\partial}{\partial x_y}$

Fluxes $\equiv F_x = \begin{Bmatrix} \rho u_x \\ \rho u_x u_y \\ \rho u_x e_{tot} \end{Bmatrix} + \begin{Bmatrix} 0 \\ P\delta_{xy} \\ P u_x \end{Bmatrix} - \begin{Bmatrix} 0 \\ \tau_{xy} \\ \tau_{xy} u_y \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ q_x \end{Bmatrix}$

advective Flux F_x^{adv} + diffusive Flux F_x^{diff}

$P_{,y} = [P\delta_{xy}]_{,x}$

$\delta_{xy} = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$

$P_{xy} = \begin{bmatrix} P & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & P \end{bmatrix} \cdot \nabla$

Note

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

body terms

$$\underset{\sim}{\mathcal{F}} = \begin{Bmatrix} 0 \\ b_y \\ b_y u_y + r \end{Bmatrix}$$

UCNS

$$\underset{\sim}{U}_{,t} + \underset{\sim}{F}_{x,x} = \underset{\sim}{\mathcal{F}}$$

$$\underset{\sim}{U}_{,t} + \underset{\sim}{F}_{x,x}^{adv} + \underset{\sim}{F}_{x,x}^{diff} = \underset{\sim}{\mathcal{F}}$$

Note

$F_x(U)$

$\mathcal{F}(U)$

Unknowns

ρ :	1	6 unknowns
u_x :	3	equation of state
p :	1	1 new equation
e_{tot}-T :	1	

Model : 1D scalar advection - diffusion

$\underline{u} \to \phi$

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

$\phi_{,t} + a\phi_{,x} - K\phi_{,xx} = q$

(a: adv, K: diff)

Simplifying the U.C. N.S.

Euler Equations

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

$\underline{U}_{,t} + \underline{F}^{adv}_{x,x} = \underline{0}\underline{f}$

Note:
in Finite Volume
$F_1 = F$
$F_2 = G$
$F_3 = H$

Incompressible

$\rho = \rho_0$ const

continuity equation

$F_{x,x} = [\rho u_x]_{,x} = \rho_0 u_{x,x}$

$\rho_{,t} = 0$

$\therefore \quad 0 + \rho_0 u_{x,x} = 0$

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

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

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

Note
This suggests that Ic flow produces a divergence free velocity field

energy
$c_p T_{,t} + u_x c_p T_{,x} = \underbrace{\frac{1}{2}\nu S_{xy} S_{xy}}_{\text{viscous dissipation}} + \tilde{K} T_{,xx} + r$

Steady

$u_{i,t} = 0$

$F_{x,x} = \underline{0}\underline{f}$

Isothermal flow

neglects energy equation

Note

compress
$[\rho u_x u_y]_{,y}$

Product rule
$u_x [\rho u_y]_{,y} + \rho u_y u_{x,y}$

if Ic

Ic
$u_y u_{x,y}$

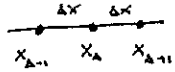
1/18/08

Numerical Methods

1) Finite Difference methods (FDM)

- replace continuous derivatives w/ discrete approx of neighboring pts

example



$$\phi_{,x}|_{x_A} \approx \underbrace{\frac{\phi|_{x_A+\Delta x} - \phi|_{x_A-\Delta x}}{2\Delta x}}_{\text{central diff } o(\Delta x^2)} \approx \underbrace{\frac{\phi|_{x_A} - \phi|_{x_A-\Delta x}}{\Delta x}}_{\text{upwind } o(\Delta x)}$$

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

$\phi_{,xy}$

2) Finite Elements method

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

$$\underset{\sim}{U}_{,t} + \underset{\sim}{F}_{A,A} - \underset{\sim}{\mathcal{P}} = \underset{\sim}{0}$$

- take the dot product of unk vector (W) with our equations

$$W \cdot (\underset{\sim}{U}_{,t} + \underset{\sim}{F}_{A,A} - \underset{\sim}{\mathcal{P}}) = 0 \qquad 1 \text{ equation}$$

- Integrate over entire domain (Ω)

$$\int_\Omega W \cdot (\underset{\sim}{U}_{,t} + \underset{\sim}{F}_{A,A} - \underset{\sim}{\mathcal{P}})\, d\Omega = 0$$

Note
W is a 5×5 vector of weight functions depending on position x
W(x,y,z)

- look for solutions such that, for any W, the above is true
- Integrate by parts (most real cases) (Galerkin Weak Form)

$$\int_\Omega (\underset{\sim}{W} \cdot \underset{\sim}{U}_{,t} - \underset{\sim}{W}_{,A} \cdot \underset{\sim}{F}_A - \underset{\sim}{W} \cdot \underset{\sim}{\mathcal{P}})\, d\Omega + \int_\Gamma W F_A n_A \, d\Gamma = 0$$

Note
Integration by Parts

$$\int_\Omega a b_{,A}\, d\Omega = \int_\Omega [ab]_{,A}\, d\Omega - \int_\Omega a_{,A} b\, d\Omega$$

$$= \int_\Gamma ab\, n_A\, d\Gamma - \int_\Omega a_{,A} b\, d\Omega$$

2D 3D

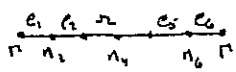
- This is a single scalar equation

1/22/08

- Look for solutions using simple functions (Global Method)

DISCRETIZE INTO FE

- 1-D



- assume some functional variation over each element

$$\underset{\sim}{U}(x) = \sum_{A=1}^{n_n=7} N_A(x)\, \underset{\sim}{U}_A$$

for each A there is a 3×1 matrix (4 in 2D, 5 in 3D)

- Perform an identical expansion on $\underset{\sim}{W}$

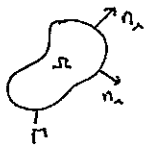
$$\underset{\sim}{W}(x) = \sum_{B=1}^{n_n} N_B(x)\, W_B$$

- Substitute into Weak Form

$$\int_\Omega \sum_{B=1}^{n_n} N_B \underset{\sim}{W}_B \cdot \sum_{A=1}^{n_n} N_A \underset{\sim}{U}_{A,t} - \sum_{B=1}^{n_n} N_{B,A} W_B \cdot \underset{\sim}{F}_A \left(\sum_{A=1}^{n_n} N_A \underset{\sim}{U}_A\right) - \sum_{B=1}^{n_n} N_B \underset{\sim}{W}_B \, \mathcal{P}\left(\sum_{A=1}^{n_n} N_A \underset{\sim}{U}_A\right) d\Omega$$

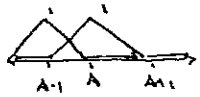
$$+ \int_\Gamma \sum_{B=1}^{n_n} N_B \underset{\sim}{W}_B \cdot \underset{\sim}{F}_A \left(\sum_{A=1}^{n_n} N_A \underset{\sim}{U}_A\right) n_A \, d\Gamma = 0$$

Nomenclature 2D



n_A is normal to Γ a function of x,y,z

N_A is shape function



n_n is the # of nodes

- every term has summation of W_B,

$$\sum_{B=1}^{n_n} W_B \cdot \left\{ \int_\Omega N_B \sum_{A=1}^{n_n} N_A \underset{\sim}{U}_{A,t} - N_{B,A} F_A \left(\sum_{A=1}^{n_n} N_A U_A\right) - N_B \mathcal{P}(\dots)\, d\Omega + \int_\Gamma N_B F_A(\dots) n_A \, d\Gamma \right\} = 0$$

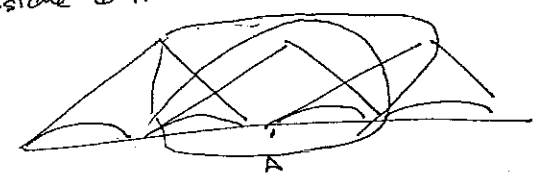
G_B

- $0 = \sum_{B=1}^{n_n} W_B \cdot \underset{\sim}{G}_B$ therefore G_B = zero for every $B = 1 \dots n_n$

Arbitrary W_B's $(Nsd+2)\, n_n$ equations

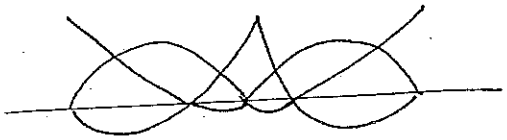
$(Nsd+2)\, n_n$ unknowns

Aside & Heirarchical 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

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

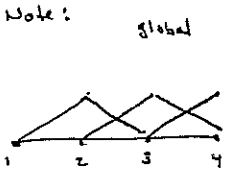
- on each element e we can form G_b^e

$$G_b^e = \int_{\Omega^e} N_b \left[\sum_{a=1}^{nen=2} N_a \underset{\sim}{U}_{a,t}^e - \underset{\sim}{f}\left(\sum_{a=1}^{3} N_a \underset{\sim}{U} \right) \right] + N_{b,i} F_i(\dots) d\Omega^e \quad \text{en} \equiv \#\text{ element nodes}$$

$$+ \int_{\Gamma^e} N_b F_i(\dots) n_i \, d\Gamma^e$$

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

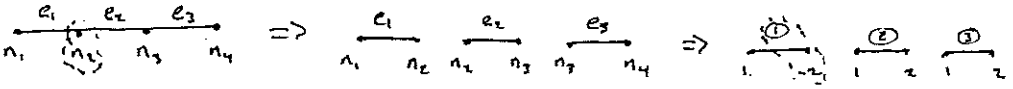
Note:



- 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 element's integrals requires a table to communicate between local & global descriptions } connectivity array

ien (el#, local node#) = global node #

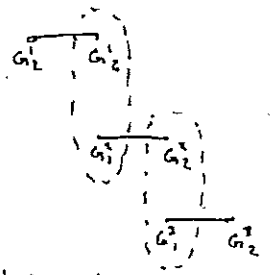
① ② n_2

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

- For our example: local views of residual G_1

First element Ω_1

$$e_1 = \begin{cases} \underset{\sim}{G}_1^1 = \dots \\ \underset{\sim}{G}_2^1 = \dots \end{cases} \qquad e_2 = \begin{cases} G_1^2 = \dots \\ G_2^2 = \dots \end{cases} \qquad e_3 = \begin{cases} G_1^3 = \dots \\ G_2^3 = \dots \end{cases}$$



- Sum contributions element-by-element

$$\underset{\sim}{G}_B = \mathop{\mathbb{A}}_{e=1}^{nel} \underset{\sim}{G}_b^e$$

after element

① $\underset{\sim}{G}_1 = \underset{\sim}{G}_1^1$ $\quad \underset{\sim}{G}_2 = \underset{\sim}{G}_2^1$ $\quad \underset{\sim}{G}_3 = 0$ $\quad \underset{\sim}{G}_4 = 0$

② $\underset{\sim}{G}_1 =$ $\quad \underset{\sim}{G}_2 = \underset{\sim}{G}_2^1 + G_1^2$ $\quad \underset{\sim}{G}_3 = G_2^2$ $\quad \underset{\sim}{G}_4 = 0$

③ $\underset{\sim}{G}_1 = \underset{\sim}{G}_1^1$ $\quad \underset{\sim}{G}_2 = \underset{\sim}{G}_2^1 + G_1^2$ $\quad \underset{\sim}{G}_3 = G_2^2 + G_1^3$ $\quad \underset{\sim}{G}_4 = G_2^3$

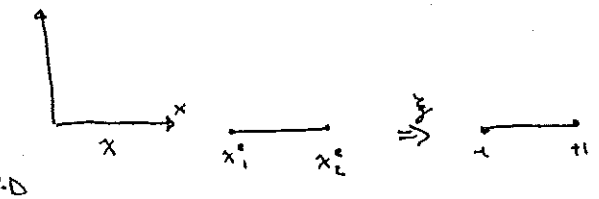
Code:

```
G_B = 0
Do e = 1, nel
  Do b = 1, nen
    G_ien(e,b) = G_ien(e,b) + G_b^e
  enddo
enddo
```

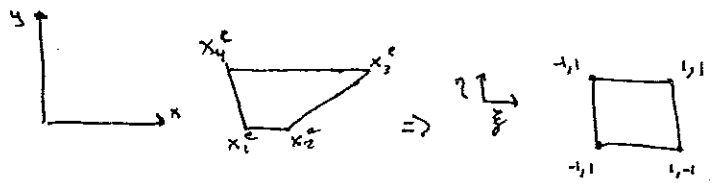
Finite Element Method (cont)

- Mapping between physical and parent domains

1-D



2-D



$X(\xi) \leftrightarrow \xi(x)$

$\Omega^e \leftrightarrow \square \quad \square$

$d\Omega^e \leftrightarrow d\square$

$N_{b,x}(\xi) = \frac{\partial N_b}{\partial x_x} = \frac{\partial N_b}{\partial \xi_j}\frac{\partial \xi_j}{\partial x_x}$

$= N_{b,\xi_j}\,\xi_{j,x}$

$\int_{\Omega^e} f(x)\,d\Omega^e = \int_D f(x(\xi))\,D(\xi)\,d\square$

determinant of the Jacobian

$D(\xi) = \det(x,\xi)$

$= \det\left(\frac{\partial x_i}{\partial \xi_j}\right)$

- now every element can be written in a common way

$$G_b^e = \int_{\square}\left[N_b(\xi)\left\{\sum_{a=1}^{n_{en}} N_a(\xi)\dot{U}_{a,t}^e - \mathcal{F}\left(\sum_{a=1}^{n_{en}} N_a U_a^e\right)\right\} - N_{b,\xi_j}(\xi)\,\xi_{j,x}\,F_x\left(\sum_{a=1}^{n_{en}} N_a U_a^e\right)\right] D(\xi)\,d\square$$

$$+ \int_{\square_\Gamma} N_b F_x\left(\sum_{a=1}^{n_{en}} N_a U_a^e\right) n_x D_\Gamma(\xi)\,d\square_\Gamma$$

= change in every element

Note for Navier-Stokes

$F_{i,x}(U_k U_{i,j})$

$$U_{i,x} = \sum_{a=1}^{n_{en}} N_{a,\xi}(\xi)\,\xi_{j,x}\,U_{a,i}^e$$

$$= \sum_{a=1}^{n_{en}}\left\{\frac{\partial N_a}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial N_a}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial N_a}{\partial \zeta}\frac{\partial \zeta}{\partial x}\right\} U_a^e$$

Note:
Galerkin method (above) is the simplest method but it is unstable for advection dominated flows.
Thus we add a term to stabalize

- approximate the integration with quadrature

$i = 1,2,3$

$$G_b^e = \int g_b^e(\xi)\,d\square + \int \ldots d\square_\Gamma$$

$$\simeq \sum_{k=1}^{nint} \underbrace{g_b^e(\xi^k)}_{\substack{\text{integrand}\\ \text{eval @}\\ k^{th}\text{ quad pt of }k^{th}}} \underbrace{W_k}_{\substack{\text{quad}\\ \text{weight}}}$$

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

$G_B = \mathbb{A} G_b^e$

$G_B(U, U_t) = 0$ when U, U_t are "correct"

- Stabilization of Galerkin

Weak form = Galerkin + $\sum_{e=1}^{n_{el}} \int_{\Omega^e} \hat{\mathcal{L}}^T \underbrace{\hat{W}}_{N_b^e \text{ if } W \text{ is factored out (elemental)}} \cdot \underbrace{\tau(\mathcal{L}U - \mathcal{F})}_{\text{residual}}\,d\Omega_e$

differential operators abstract notation of strong form PDE

$\mathcal{L}U = U_{,t} + F_{x,x} = U_{,t} + F_{x,x}^{adv}(U) + F_{x,x}^{diff}(U)$

$\mathcal{L}U - \mathcal{F} = 0$

regardless of what $\hat{\mathcal{L}} N_b^e \tau$ is, it multiplies a "residual", if U is soln we add zeros

This means that this method is weighted residual method

Finite Element Method (cont)

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

$$\mathcal{L}U = U_{,t} + F^{adv}_{x,x} + F^{dif}_{x,x}$$

$$F^{adv}_{x,x}(U) = \frac{\partial F_x}{\partial U}\frac{\partial U}{\partial x_x} = \underline{\underline{A}}_x \underline{U}_{,x} \qquad \underline{\underline{A}}_x(\underline{U})$$

Quasi-Linearization of Advective Flux

$$F^{dif}_{x,x}(U) = -\left[K_{xy} U_{,y}\right]_{,x}$$

Note: this supposes $F^{diff}(U) \equiv -\underline{\underline{K}}_{xy} U_{,y}$ matrix multiplication

Building $\underline{\underline{K}}_{xy}$ from our definition of F^{diff}_x

$F_{x,x}$: 3×3 $\Rightarrow$ 5×5 matrix

K_{11} = 5×5 matrix

Likewise $\underline{\underline{A}}_x$ represents 3 5×5 matrices

$$\underline{\underline{A}}_x \equiv \frac{\partial F_x}{\partial U}$$

In this form

$$\mathcal{L}U = U_{,t} + \underline{\underline{A}}_x \underline{U}_{,x} - \left[\underline{\underline{K}}_{xy} \underline{U}_{,y}\right]_{,x}$$

$$\mathcal{L}U \equiv \underbrace{\left[\frac{d}{dt} + A_x \frac{\partial}{\partial x_x} - \underline{d}\left[K_{xy}\frac{\partial}{\partial x_y}\right]}_{\frac{\partial}{\partial x_x}}\right] U}_{\mathcal{L}\ \text{linear operator}}$$

  - Definition of $\hat{\mathcal{L}}$
    - Galerkin least squares (GLS)
      - $\hat{\mathcal{L}} = \mathcal{L}$
    - Streamline Upwind Petrov-Galerkin (SUPG) (often used) → Note: this forms a square of the advection term which is positive definite and is stable
      - This is like diffusion just along the streamline of the flow

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

    - Douglas-Wang (Variational Multiscale Method)

$$\hat{\mathcal{L}} = -\mathcal{L}^* = \frac{\partial}{\partial t} + A_x \frac{\partial}{\partial x_x} + \underbrace{\frac{\partial}{\partial x_x}\left[K_{xy}\frac{\partial}{\partial x_y}\right]}_{2^{nd}\ \text{deriv terms}}$$

  - Definition of $\underline{\underline{\tau}}$ (stabilization Matrix) (time scale 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

Finite Element Method (cont)

Stability of Galerkin (cont)

Tracing the stabilization term through the element

Equation Formation

$$\hat{G}^e_b = G^e_b + \int_\square \overset{e}{\delta} N_b(\xi)\, \tau \left(\overset{e}{\mathcal{L}} \overset{h}{U} - \overset{e}{q} \overset{h}{f} \right) D(\xi)\, d\square$$

For SUPG

$$\overset{e}{\delta} N_b = A_k \left(\sum_{c=1}^{n_{en}} N_c(\xi)\, U_c \right) N_{b,j}\, \xi_{j,k}$$

and

$$\overset{e}{\mathcal{L}} U = \sum_{a=1}^{n_{en}} N_a U_{a,i} + A_j \left(\sum_{c=1}^{n_{en}} N_c(\xi) U_c \right) \sum_{a=1}^{n_{en}} N_{a,\xi_m}\, \xi_{m,j}\, U_a - \underline{\text{Second deriv terms}}$$

Review the entire Process

When forming equations, we loop over elements

loop over n_{el}

- ↳ integrate to get $\hat{G}^e_b(\xi^q)$
- loop over quadrature points in each element (n_{qp})
  - ↳ eval $\hat{f}(\xi^q)$
  - loop over shape functions @ each Qpoint in each element (n_{en})

Feed into assembly operator

↳ converts into global equation evaluations

$G_B = A G^e_b$

Alternative Stabilizers

Artificial diffusion Method

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

$$\overset{*e}{G}_b = G^e_b + \int_\square \left[N_{b,\xi_i}\, \xi_{i,x}\, K_{xy}\, N_{a,\xi_m}\, \xi_{m,y} \right] D(\xi)\, d\square$$

K is a diffusivity Matrix

e.g. $\begin{bmatrix} 0 & & \\ & \nu_{ad} + \nu_{ad} & 0 \\ & 0 & \nu_{ad} + \nu_{ad} \end{bmatrix}$ $\quad \nu_{ad} = \frac{\alpha\, u h}{2}$ on a structured grid this produces upwind stencil

Finite Volume Method

Over simplification:

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

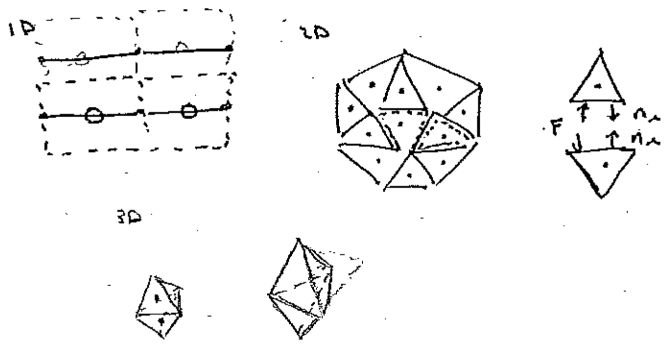
$$\int_{\Omega} (U_{,t} - \mathcal{F}) \, d\Omega = \int_{\Gamma} F_n n_n \, d\Gamma$$

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 (ICC)

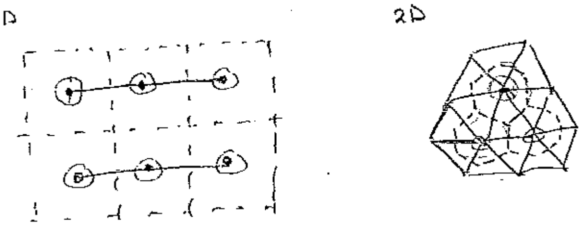
- the center of the C.V. is the sol'n point
- this method developes equations for each cell center by evaluating the Flux F at the boundary of the cell
- the normals have opposite signs therefore Fluxes across boundary leaves one element to the other



2/01/08

- 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 $(U_{,t} \cdot \mathcal{F})$, 1 pt sample times the volum ok up to 2^{nd} order accurate
  - we get $G_{B'} = 0$ where B''s the cell center index (# cells)

Vertex Centered Form



- Down side
  - move Faces to do flux through
  - up front work to comput volumes
- Upside
  - fewer non-linear equations to solve
- Data Structure (IVERT)

  IVERT(edge#, 2) $\Rightarrow$ (nodes on vertices) ends of edges

Finite Element a Finite Volume

$\hat{G}_B$ $\quad$ $G_{B'}$

$G_B(\underset{\sim}{U}, \underset{\sim}{U}_{,t})$

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

$$\hat{G}_B = \int_{\square} N_b \{ A_0 Y_{,t} - \mathcal{F} \} - N_{b,x} F_n(Y) \, D \, d\square + \int_{\square} \hat{\sum} N_b \underset{\approx}{\tau} \{ \mathcal{L} Y - \mathcal{F} \} D \, d\square$$
$$+ \int_{\square_\Gamma} N_b F_n n_n D_\Gamma \, d\square_\Gamma$$

(change of variables UNDER the integral sign)

$$U(\xi) = \sum_{c=1}^{n_{en}} N_a(\xi) U_a \Rightarrow Y = \sum_{a=1}^{n_{en}} N_a(\xi) Y_a^e \quad \text{— nodal values}$$
$$Y_{,t} = N_a(\xi) Y^e_{a,t}$$
$$Y_{,x} = N_{a,\xi_x}(\xi) J_{\xi x} Y_a^e$$

Ref: Haule a Hughes: Comp meth in applied Mech a Eng

ODE can be expressed w/ other vectors consider arbitrary independent variable vector $\underset{\sim}{Y}$.

We can write

$$\underset{\sim}{Y} = \underset{\sim}{Y}(\underset{\sim}{U}) \quad \text{and} \quad \underset{\sim}{U} = \underset{\sim}{U}(\underset{\sim}{Y})$$

replace

$$U_{,t} = U_{,Y} Y_{,t} = \underset{5\times5}{\underset{\approx}{A_0}} Y_{,t}$$

$$F^{adv}_{n} = F_{n,Y} Y_{,x} = \underset{5\times5\times3}{\underset{\approx}{A_n}} Y_{,x}$$

$$\mathcal{F}(U(Y)) = \mathcal{F}(Y)$$

$$F^{diff}_{n} = -[K_{ng} Y_{,g}]_{,n}$$

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

$$\mathcal{L}U = \mathcal{L}Y = \left[A_0 \frac{\partial}{\partial t} + A_n \frac{\partial}{\partial x_n} + \frac{\partial}{\partial x_n}(K_{ng} \frac{\partial}{\partial x_g})\right] Y$$

1^{st} & 3^{rd} component of U vector

$$U_2 = \frac{\rho u_i}{\rho}$$

if $Y_3 = \frac{U_3}{U_1} = v = u_2$

consistency check if $Y = U$
$A_0 \Rightarrow I$ $\quad A_n + K_{ng}$ also go back

$$U = \begin{cases} \rho \\ \rho u \end{cases}$$

$$\rho \cdot A_n \frac{\partial}{\partial x_n}$$

Examples

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

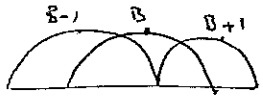
Favor

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

Thought Experiment

Five Vectors in $\hat{G}_B$ become 1

- for each node B we have 1 nonlinear ODE
- if 1-D, we couple to B+1, B, B-1
- for a particular B



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

a-f are coefficients which can depend on Y

Problem:

① n_{nodes} equations $2 * n_{nodes}$ unknowns

② nonlinearity

What happens from 1-D → 3-D

- More a-f coefficients

What happens from scalar to vectors

Y_i becomes vector

a-f become matrix

$$\hat{G}_B = \hat{G}_B(\tilde{Y}_{,t}, \tilde{Y}) = 0$$

$\tilde{\;}$ represents a collection of all the "nodal" values

$\tilde{Y} \equiv (5*n)\times 1$ vector $= \{Y_A\}_{A=1,n}$

$\hat{G}_B = 0$ when we have the correct Y, $Y_{,t}$ vectors

need to convert Non linear ODE to nonlinear Algebraic Eqn

use a predictor multicorrector algorithm to relate $Y_{,t}$ - Y

we can establish nodal level ($Y_{A\,\&\,B}$ to Y_A)

Simplist Form (Explicit method)

$$G_B = \overbrace{G_B^t(Y_{,t}, Y)}^{\text{temporal}} + \overbrace{G_B^s}^{\text{Steady}} = 0$$

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

$$\underset{\approx}{M}(\tilde{Y})\,\tilde{Y}_{,t} + \underset{\approx}{K}(\tilde{Y})\,\tilde{Y} = 0$$

$$\underset{\approx}{M}(\tilde{Y})\,\tilde{Y}_{,t} = -\underset{\approx}{K}(\tilde{Y})\,\tilde{Y}$$

Explicit Forward Euler

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

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

when $Y \neq U$ to get explicit

$$M(Y^n) \overset{\text{lumped}}{=} M^L(Y^n) \Rightarrow \quad Y_A^{n+1} = Y^n + \Delta t \hat{\bar{A}}_o^{-1}\left\{G_A^s(\tilde{Y}^n)\right\}$$

$\bar{A}_o$ is locally averaged

Stability constraints on time step

Usually prohibitively small (bad for viscous flows)

Implicit Generalized Alpha Method

Predictor multi corrector

↑ nonlinear method

Introduce (a) Iteration counter

the ath iterate in the process of moving from time step $n \to n+1$

$$\underset{\sim}{Y}_A^{n+1(a)} = \hat{\underset{\sim}{Y}} + \Delta t\, \hat{\underset{\sim}{Y}}_{,t} + \Delta t\, \underset{\sim}{\gamma}\, (\underset{\sim}{Y}_{A,t}^{n+1(a)} - \hat{\underset{\sim}{Y}}_{A,t}) \qquad \text{OD1}$$

$$\tilde{G}_A(\underset{\sim}{Y}_{,t}^{n+\alpha_m (a)}, \underset{\sim}{Y}^{n+\alpha_f (a)}) = 0 \qquad \text{OD2}$$

Interpolations

$$\text{OD3} \quad \underset{\sim}{Y}_{A,t}^{n+\alpha_m (a)} = \tilde{\underset{\sim}{Y}}_{A,t} + \alpha_m (\underset{\sim}{Y}_{A,t}^{n+1(a)} - \tilde{\underset{\sim}{Y}}_{A,t})$$

$$\text{OD4} \quad \underset{\sim}{Y}_{A}^{n+\alpha_f (a)} = \underset{\sim}{Y}_A^n + \alpha_f (\underset{\sim}{Y}_A^{n+1(a)} - \underset{\sim}{Y}_A^n)$$

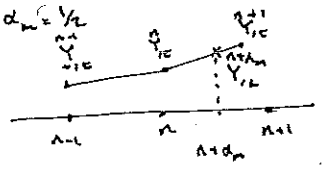
all 4 ODs are for all A

and all 5 eqn/var in each vector

Remarks:

① in OD2 $Y_{,t}$ is evaluated @ $t = t_{n+\alpha_m} = t_n + \alpha_m \Delta t$

let $\alpha_m = 1/2$

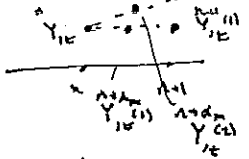


if $0 < \alpha_m < 1$ interpolation

if else extrapolation

② α_m does have to equal α_f

③ both $\underset{\sim}{Y}_{,t}^{n+\alpha_m (a)}$ and $\underset{\sim}{Y}^{n+\alpha_f (a)}$ are interpolative of the current iteration $\underset{\sim}{Y}_{,t}^{n+1(a)}$ and will change with (a) until $\tilde{G}_B = 0$



- The 3 parameters α_m α_f γ
  - to get 2nd order accuracy

    we give up 2 of the 3 free parameters

    express the 3 in terms of 1 new parameter

    ρ_∞ is the amplification factor in large time step limit

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

$$\alpha_f = \frac{1}{1 + \rho_\infty}$$

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

		α_m	α_f	γ
$\rho_\infty = 1$	trapezoidal rule / midpoint rule	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$\rho_\infty = 0$	Gears method	$3/2$	1	1

any inbetween are possible

0.9 only a little temporal damping

There are times we want steady solutions

1) alternative : $\dot{Y}_{i,t} = 0$

- solve Non linear Algebraic equations $G_B(Y)$ for all B Finds local minima easily
- not often used

2) integrate with large timesteps until $\dot{Y}_{i,t} = 0$

3) Expect system to guess + improve systems
(evolutionary algorithms)
(genetic algorithms)

If steady state use Backward Euler :
this is the most dissipitive → largest timestep in the face of non-linearity
Fastest Root to steady state

$\alpha_m = 1$

$\alpha_f = 1$ dont worry about

$\gamma = 1$ 2nd order accuracy

Still have system of equations to solve
Not explicit

$$\hat{G}_{iB}(\tilde{\dot{Y}}_{i,t}, \tilde{Y}) = 0 \quad \text{for all } B$$

ODE

How do we make this happen?

Predictors

assume we are given a $\overset{n}{\underset{\sim}{Y}}$ and $\overset{n}{\underset{\sim}{\dot{Y}}}_{i,t}$
That satisfies ODE 1 + ODE 2

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

Set $\nu = 1$ (iteration count) at previous state
predicts $\overset{n+1}{\underset{\sim}{Y}}^{(\nu)} = \overset{n}{\underset{\sim}{Y}}$

Note : this is the best predictor

Invert ODE 1

$$\overset{n+1}{\underset{\sim}{\dot{Y}}}^{(\nu)} = \frac{\overset{n+1}{\underset{\sim}{Y}}^{(\nu)} - \overset{n}{\underset{\sim}{Y}}^{(\nu)}}{\Delta t \gamma} + \left(1 - \frac{1}{\gamma}\right) \overset{n}{\underset{\sim}{\dot{Y}}}^{(\nu)}_{i,t}$$

Evaluate $\overset{n+\alpha_m}{\underset{\sim}{Y}}^{(\nu)}_{i,t}$ and $\overset{n+\alpha_f}{\underset{\sim}{\dot{Y}}}^{(\nu)}_{i,t}$ using ODE 3 + ODE 4

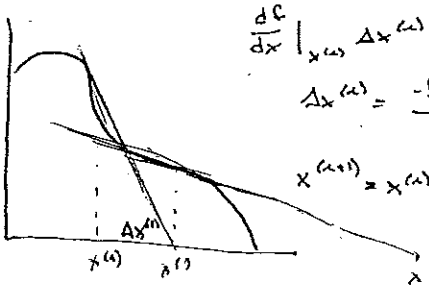
Evaluate $G_B\left(\overset{n+\alpha_m}{\underset{\sim}{Y}}^{(\nu)}_{i,t}, \overset{n+\alpha_f}{\underset{\sim}{\dot{Y}}}^{(\nu)}_{i,t}\right) \neq 0$

Use Newtons method as a corrector

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

$f(x) = 0$

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



$$\left.\frac{df}{dx}\right|_{x^{(\nu)}} \Delta x^{(\nu)} = -f(x^{(\nu)})$$

$$\Delta x^{(\nu)} = \frac{-f(x^{(\nu)})}{\left.\frac{df}{dx}\right|_{x^{(\nu)}}}$$

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

Newton's Method our example

Make one variable "independent" one set by ODE

$\underset{\sim}{Y}^{n+\alpha_f}_{(i)}$ = independent

$\underset{\sim}{Y}^{n+\alpha_m}_{(i)}$ = dependent

Newton's Method

$$\underbrace{G_B\left(\underset{k=1}{\overset{n_{np}}{\sum}} \underset{\sim}{Y}^{n+\alpha_m}_{(i)}, \underset{\sim}{Y}^{n+\alpha_f}_{(i)}\right)}_{\text{non linear Residual } (R_B)} + \underbrace{\underset{A=1}{\overset{n_{np}}{\sum}} \frac{G_B}{\underset{\sim}{Y}^{n+\alpha_f}_{A(i)}}}_{\text{tangent matrix } (M_{BA})} \cdot \underbrace{\Delta \underset{\sim}{Y}^{n+\alpha_f}_{A(i)}}_{\text{increment to improve}} = 0$$

solve for this

Note: summed over nodal points

Note: $\underset{\sim}{Y}^{n+\alpha_f}_{A(i+1)} = \underset{\sim}{Y}^{n+\alpha_f}_{A(i)} + \Delta \underset{\sim}{Y}^{n+\alpha_f}_{A(i)}$

solve for this

Plug back into first predictor

Our linear Matrix problem @ Every time step

$$\sum_{A=1}^{n_{np}} \underset{\sim}{M}_{BA} \, \Delta \underset{\sim}{Y}_A = -\underset{\sim}{R}_B$$

5×5 for each (A,B); 5×1 for each A

$$\underset{\sim}{\hat{M}} \, \Delta \underset{\sim}{\hat{Y}} = -\underset{\sim}{\hat{R}}$$

$5n_{np}\times 5n_{np}$

$$\underset{\sim}{\hat{M}} = \{M_{BA}\}$$

$$\Delta \underset{\sim}{\hat{Y}} = \{\Delta \underset{\sim}{Y}^{n+\alpha_f}_{A(i)}\}$$

$$-\underset{\sim}{\hat{R}} = \{R_B\}$$

Solve linearized Algebraic System of equations to get $\Delta \underset{\sim}{\hat{Y}}$

If we keep iterating $i = 1, 2, 3 \ldots$

eventually

$\Delta \underset{\sim}{Y}^{n+\alpha_f}_{(i)} \Rightarrow 0$ for each A

as $R_B \Rightarrow 0$ for each B

We need $\underset{\sim}{Y}^{n+1}_{(\text{converged})}$ to go to next step

so we write ODE twice

$$\underset{\sim}{Y}^{n+\alpha_f}_{A} = \underset{\sim}{Y}^{n}_{A} + \alpha_f \left(\underset{\sim}{Y}^{n+\alpha_f}_{A(i+1)} - \underset{\sim}{Y}^{n}_{A}\right)$$

$$- \underset{\sim}{Y}^{n+\alpha_f}_{A(i)} = -\underset{\sim}{Y}^{n}_{A} - \alpha_f \left(\underset{\sim}{Y}^{n+\alpha_f}_{A(i)} - \underset{\sim}{Y}^{n}_{A}\right)$$

$$\Delta \underset{\sim}{Y}^{n+\alpha_f}_{A(i)} = \alpha_f \, \Delta \underset{\sim}{Y}^{n+1}_{A(i)}$$

$$\Delta \underset{\sim}{Y}^{n+1}_{A(i)} = \frac{\Delta \underset{\sim}{Y}^{n+\alpha_f}_{A(i)}}{\alpha_f}$$

We can then say

$$\underset{\sim}{Y}^{n+1}_{A(i+1)} = \underset{\sim}{Y}^{n+1}_{A(i)} + \Delta \underset{\sim}{Y}^{n+1}_{A(i)}$$

This allows us to continue to iterate w/ ODE

Going Deeper into Forming $\tilde{\underset{\sim}{M}}$

Review: G_B in FEM

very non linear

$$G_B = \int_\Omega \left\{ N_B \{ A_0 \overset{n+\alpha_m}{Y}_{,t} - \mathcal{F} \} - N_{B,i} F_i(\overset{n+\alpha_f}{Y}^{(i)}) \right\} d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \hat{\mathcal{L}} N_B \tau \{ A_0 \overset{n+\alpha_m}{Y}_{,t} + \underset{=}{A}_i \overset{n+\alpha_f}{Y}_{,i} - \mathcal{F} \}$$

$$+ \int_\Gamma N_B F_i(\overset{n+\alpha_f}{Y^{(i)}}) n_i \, d\Gamma$$

Do not use integration by parts on convective term

Hold A matrices fixed @ time n

$$\overset{4LHS}{G_B} = \int \left\{ N_B \{ \underset{=}{A_0} \overset{n+\alpha_m}{Y}^{(i)}_{,t} - \mathcal{F} + \underset{=}{A_i} \overset{n+\alpha_f}{Y}^{(i)}_{,i} \} + N_{B,i} K_{ij} Y_{,j} \right\} d\Omega + \ldots$$

(with $F^{adv}_{i,i}$ and F^{diff}_i marked above the terms)

$$+ \int_\Gamma N_B F^{diff}_i n_i \, d\Gamma$$

usually dropped <<1

We are attempting to solve

$$\frac{\partial \overset{4LHS}{G_B}}{\partial \overset{n+\alpha_f}{Y_A}} \Delta \overset{n+\alpha_f}{Y_A} = -G_B$$

$$\underset{=}{A} \, \underset{-}{x} = \underset{-}{b} \qquad \text{solve for } x$$

2/12/08

Counterpart at the Element Level FEM^e_b / Face Level FVM

$$\overset{4LHS}{G^e_b} = \int_\square \Big\{ N_b \{ \underset{=}{A_0} \overset{①}{\sum_{c=1}^{n_{en}}} N_c \overset{n+\alpha_m}{Y}^{(i)e}_{c,t} + \underset{=}{A}_i \overset{②}{\sum_{c=1}^{n_{en}}} N_{c,i} \overset{n+\alpha_f}{Y}^{(i)e}_{c} - \mathcal{F} \} + N_{b,i} \underset{=}{K_{ij}} \overset{③}{\sum_{c=1}^{n_{en}}} N_{c,j} \overset{n+\alpha_f}{Y}^{(i)e}_c$$

$$+ \hat{\mathcal{L}} N_b \underset{=}{\tau} \{ A_0 \overset{④}{\sum_{c=1}^{n_{en}}} N_c \overset{n+\alpha_m}{Y}^{(i)e}_{c,t} + A_i \overset{⑤}{\sum_{c=1}^{n_{en}}} N_{c,i} \overset{n+\alpha_f}{Y}^{(i)e}_c - \mathcal{F} \} \Big\} D \, d\square$$

Terms we differentiate WRT $\overset{n+\alpha_f}{Y}^{(i)e}_a$

②,③,⑤ are already in terms of $\overset{n+\alpha_f}{Y}$

① ④ need work to express $\overset{n+\alpha_m}{Y}$ in terms of $\overset{n+\alpha_f}{Y}$

Start with OD3

$$\overset{n+\alpha_m}{Y}^{(i)e}_{c,t} = \overset{n}{Y}^e_{c,t} + \alpha_m \left(\overset{n+1}{Y}^{(i)e}_{c,t} - \overset{n}{Y}^e_{c,t} \right)$$

Use OD1 inverted to look at delta increment

$$\left(\overset{n+1}{Y}^{(i)e}_{c,t} - \overset{n}{Y}^e_{c,t} \right) = \frac{\overset{n+1}{Y}^{(i)e}_c - \overset{n}{Y}^e_c}{\Delta t \gamma} - \frac{1}{\gamma} \overset{n}{Y}^e_{c,t}$$

Invert OD4 to get new delta increment

$$\overset{n+1}{Y}^{(i)e}_c - \overset{n}{Y}^e_c = \frac{\overset{n+\alpha_f}{Y}^{(i)e}_c - \overset{n}{Y}^e_c}{\alpha_f}$$

Substitute terms

$$\overset{n+\alpha_m}{Y}^{(i)e}_{c,t} = \overset{n}{Y}^e_{c,t} + \alpha_m \left\{ \left\{ \frac{\overset{n+\alpha_f}{Y}^{(i)e}_c - \overset{n}{Y}^e_c}{\alpha_f} \right\} \frac{1}{\Delta t \gamma} - \frac{1}{\gamma} \overset{n}{Y}^e_{c,t} \right\}$$

By the chain Rule, need to evaluate

$$\frac{\partial \overset{n+\alpha_m}{Y_{(i)c}}}{\partial \overset{n+\alpha_f}{Y_{(i)a}}} = \frac{\alpha_m}{\alpha_f \gamma \Delta t} \frac{\partial \overset{n+\alpha_f}{Y_{(i)c}}}{\partial \overset{n+\alpha_f}{Y_{(i)a}}}$$

$$= \frac{\alpha_m}{\alpha_f \gamma \Delta t} \delta_{ca} I_{s\times s}$$

If $c=a$ $\quad \frac{\partial \underline{\xi}}{\partial Y_m} = \underline{I}_s = \delta \underline{\underline{I}}_{s\times s} = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$

Term ①

$$\frac{\partial \left[N_b A_a \sum_{c=1}^{n_{en}} N_c \overset{n+\alpha_m}{Y_{(i)c}} \right]}{\partial \overset{n+\alpha_f}{Y_{(i)a}}} = N_b A_a \underbrace{\sum_{c=1}^{n_{en}} N_a \frac{\alpha_m}{\gamma \alpha_f \Delta t} \delta_{ca}} I_{s\times s}$$

$$= \underbrace{N_b A_a N_a \frac{\alpha_m}{\gamma \alpha_f \Delta t}}$$

$$\frac{\partial G_b}{\partial Y_a} = \int_\Omega (\qquad) D \, d\Omega$$

Freeze A_a, $A_{a,i}$, K_{ij}

By nature of δ_{ca}

Term ②

$$\frac{\partial \left[N_b A_{a,i} \sum_{c=1}^{n_{en}} N_{c,i} \overset{n+\alpha_f}{Y_{(i)c}} \right]}{\partial \overset{n+\alpha_f}{Y_{(i)a}}} = N_b A_{a,i} \sum_{c=1}^{n_{en}} N_c \delta_{ca} I_{s\times s}$$

$$= N_b A_{a,i} N_{a,i}$$

Term ③

$$\frac{\partial \left[N_{b,i} K_{ij} \sum_{c=1}^{n_{en}} N_{c,j} \overset{n+\alpha_f}{Y_{(i)c}} \right]}{\partial \overset{n+\alpha_f}{Y_{(i)a}}} = N_{b,i} \underline{\underline{K}}_{ij} N_{a,j}$$

Term ④ (like 1)

$$= \int_\Omega N_b \tau A_a N_a \frac{\alpha_m}{\gamma \alpha_f \Delta t}$$

in SNPG

Term ⑤ (like 2)

$$= \int_\Omega N_b \tau A_{a,i} N_{a,i}$$

All terms go in

$$\int_\Omega (\qquad) D \, d\Omega$$

added up they make M^e_{ab} matrix

requires quadrature like G^e_b

$$G_B = \sum_{e=1}^{n_{el}} G^e_b \qquad \text{len(el\#, local)} = \text{global}$$

$$M_{AB} = \sum_{e=1}^{n_{el}} M^e_{ab} \qquad \text{twice} \quad \begin{matrix} 1) \; a \to A \\ 2) \; b \to B \end{matrix}$$

$$\tilde{\underline{M}} = \{ M_{AB} \} \qquad R = \{ G_B \}$$

$$\boxed{\tilde{\underline{M}} \, \Delta \tilde{Y} = G_B}$$

Boundary Conditions

Two Types

Dirichlet (Γ_g) aka/or **Neumann (Γ_h)**

Set value of nodes on boundary strong/exact

"essential"

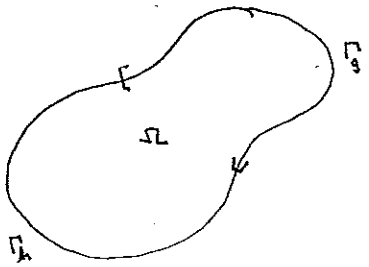
"strong"

Set the flux on a boundary to a value. Let 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

if

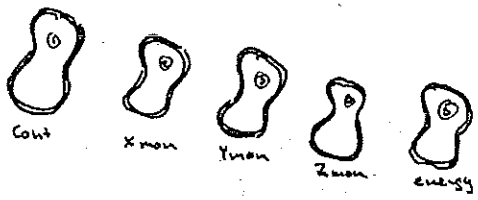
$u\phi_{,x} - \kappa\phi_{,xx}$ convection

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

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

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

Essential BC

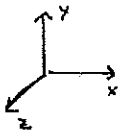
think about which kinds of fields we want for BC for

collect fields into a vector "q"

$$q = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix} = \begin{Bmatrix} P_s^q \\ U_r^q \\ U_s^q \\ U_t^q \\ T_s^q \\ \rho_s^q \end{Bmatrix}$$

Pe-scribed pressure
Velocity component
"
"
Temperature
density

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



$U_r^q = C_1^r U_1 + C_2^r U_2 + C_3^r U_3 = C_n^r U$

rotation components (cosines)

$U_s =$

$U_t =$

where

C_n^r are rotation cosines

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

$|C_1^r| \geq C_{n\neq 1}^r$

$|C_2^s| \geq C_{n\neq 2}^s$

$|C_3^t| \geq C_{n\neq 3}^t$

gets us near xyz system

What we want to accomplish is:
How does the setting of an Eqn model of these BC change our soln variables

With BCs in terms of Y variables

$$q(Y) = \begin{Bmatrix} P_s^q \\ U_r^q \\ U_s^q \\ U_t^q \\ T_s^q \\ \rho_s^q \end{Bmatrix} = \begin{Bmatrix} Y_1 \\ C_1^r Y_{n+1} \\ C_2^s Y_{n+1} \\ C_3^t Y_{n+1} \\ Y_5 \\ \frac{Y_1}{RY_5} \end{Bmatrix}$$

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

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

$P = \rho RT \Rightarrow \rho = \frac{P}{RT} = \frac{Y_1}{RY_5}$

2/15/08

We want to invert these relationships

$$q(q) = \begin{Bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{Bmatrix} = \begin{Bmatrix} P_s^q \\ (U_r^q - C_2^r Y_3 - C_3^r Y_4)/C_1^r \\ \frac{1}{C_2^s}(U_s^q - C_1^s Y_2 - C_3^s Y_4) \\ \frac{1}{C_3^t}(U_t^q - C_1^t Y_2 - C_2^t Y_3) \\ T_s^q \end{Bmatrix}$$

$Y_1 = P_s^q R Y_5$

Prescribed quantity

Links things through the gas laws

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 of p^3 sets Y^3 thus you cannot use 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 of node are zero

In linear problems, this influences says

- if node has solved
  - ↳ Weight function is set to zero
    - $q_s^{\wedge} \rightarrow w^{\wedge} = 0$
    - ↳ this removes that equation to node s from system

Theory:

- Non linear BCs should be handled by constraining the weight function space V to belong to the tangent space of solution space $\mathcal{S}$
- This is accomplished by taking $\underset{\sim}{w}$ to be the variation of $\underset{\sim}{Y}$

$$w_{\substack{r \\ new}} \Leftarrow \frac{\partial \hat{q}_r}{\partial Y_s} w_{\substack{s \\ old}} = \frac{\partial q_r}{\partial Y_1} w_1 + \frac{\partial q_r}{\partial Y_2} w_2 + \frac{\partial q_r}{\partial Y_3} w_3 + \cdots \quad w_5$$

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

example: typical

$i=1$ $p^3_{n=1}$: $w_1 \Leftarrow \frac{\partial p^3}{\partial Y_1} w_1 + \frac{\partial p^3}{\partial Y_2} w_2 + \frac{\partial p^3}{\partial Y_3} w_3 + \frac{\partial p^3}{\partial Y_4} w_4 + \frac{\partial p^3}{\partial Y_5} w_5$

w_1 old

example

$\hat{q}_r = \frac{q_r - \cdots}{c_{ir}}$

$i=2$ $w_2 \Leftarrow \frac{\partial (\hat{q}_2^1)}{\partial Y_1} w_1 + \frac{\partial (\hat{q}_2^1)}{\partial Y_2} w_2 + \frac{\partial (\hat{q}_2^1)}{\partial Y_3} + \frac{\partial (q_2)}{\partial Y_4} + \frac{\partial \cdots}{\partial Y_5}$

$i=3$ $= \frac{-c_1^2}{c_1^2} w_3 - \frac{c_3^2}{c_1^2} w_4$

$w_3 = \frac{-c_1^3}{c_2^3} w_2 - \frac{c_3^3}{c_2^3} w_4$

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

$i=5$ $w_5 = 0$

$w_6 \cdot p^3 R w_5 = \frac{\partial q_6}{\partial Y_5} = \frac{\partial (p^3 R Y_5}{\partial Y^5}$

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

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

Lets look @ the $\underline{\underline{S}}$'s

a given node with pressure BC

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

this makes pressure @ the boundary = 0

$$w^{new} = \begin{bmatrix} 0 & & & \\ & 1 & & 0 \\ & & 1 & \\ 0 & & & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_5 \end{bmatrix}$$

a given density

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

a given velocity (x)

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

ask for a Paper if we don't know more

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

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

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

To account for BC's we must Replace

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

say the nth node has a B.C.

$\underline{\underline{\tilde{S}}}$ = block diagonal $\left[I_{5\times5}\; I_{5\times5} \cdots \overset{n^{th}}{S} \cdots I_{5\times5}\right]$

not how it is coded

$$\tilde{W} \leftarrow \underline{\underline{\tilde{S}}}\,\tilde{W} \qquad \Delta\tilde{Y} \leftarrow \underline{\underline{\tilde{S}}}\,\Delta\tilde{Y}$$

$$[SW]\cdot M[S\Delta Y] = -[SW]\cdot G$$

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

$$W\,M'\Delta Y = W\,G'$$

$$M'\Delta Y = G'$$

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

ierr = input_ffonm(inpfilename); reads in ascii data

Input geom BC IC data

readnblk declares arrays in readarrays

genblk reads data into pointers created
breaks elements into blocks
Reads in ientp (4x8)

genbkb copies to gensav - makes memory data pointers

gread sim to genblk

genint reads in initial condition

Process pre computes and tables all shape functions @ each quad point

Simhex Qpt location of point ξ γ ζ of quad points
Qwt weight of quad pts

---

gen dat

genshp

shpTet uses Qpt and assigns shape functions @ each Qpt

block size 50-60

We seek the $\{\underline{U}^{n+\alpha_f}\}, \{U^{n+\alpha_m}_{i,t}\}$ that will make $\{\underline{R}^M\} = 0$ and $\{\underline{\dot{R}}\} = 0$

$\{P^{n+\alpha_f}\}, \{P^{n+\alpha_m}_{i,t}\}$

Newtons Method

"choose" independent variables

$\{U^{n+1}_{i,t}\}$ and $\{P^{n+1}_{i,t}\}$

Newtons lin mom residual

Mom $$\sum_{B=1}^{n_{sp}} \left\{ \underbrace{\frac{d R^M_A}{\partial U^{n+1}_{i,t}}}_{\textcircled{a}} \Delta U^{n+1}_{B,t} + \underbrace{\frac{d R_A}{\partial P^{n+1}_{i,t}}}_{\textcircled{b}} \Delta P_{i,B} \right\} = -R^M_A$$

Con $$\sum_{B=1} \left\{ \underbrace{\frac{\partial \dot{R}^C_A}{\partial U^{n+1}_{i,t}}}_{\textcircled{c}} \Delta U^{n+1}_{i,t} + \underbrace{\frac{\partial \dot{R}^C_A}{\partial P^{n+1}_{i,t}}}_{\textcircled{d}} \Delta P^B_{i,t} \right\} = -\dot{R}^C_A$$

$$\textcircled{a} \left\{ \underbrace{K^{AB}}_{3\times 3} \Delta U^{n+1}_{i,t} + \underbrace{G^{AB}}_{3\times 1} \Delta P^B_{i,t} \right\} = -R^M_A$$

$$\sum_{B=1}^{n_{sp}} \left\{ \underbrace{D^{AB}}_{1\times 3} \Delta U^{n+1}_{i,t} + \underbrace{C^{AB}}_{1\times 1} \Delta P^{n+1}_{i,t} \right\} = -\dot{R}^C_A$$

LHSK
$$\underbrace{\begin{bmatrix} K & G \\ D & C \end{bmatrix}}_{\text{LHS}\ 4n_{sp}\times 4n_{sp}} \underbrace{\begin{Bmatrix} \Delta U_{i,t} \\ \Delta P_{i,t} \end{Bmatrix}}_{\text{Sol'n}\ 4n_{sp}\times 1} = -\underbrace{\begin{Bmatrix} R^M \\ \dot{R}^C \end{Bmatrix}}_{\text{RHS}\ 4n_{sp}\times 1}$$

In code
K => LHSK
$[D\ C]$ => LHSP

Note
$K^{AB} = A^a K^{ab}$ ← local

@
$$K^{ab}_{ij} = \frac{\partial R^{M,a}_i}{\partial U^{n+1}_{j,t}} = \sum_{c=1}^{n_{en}} \left\{ \frac{\partial R^a_i}{\partial U^{n+\alpha_f,c}_k} \underbrace{\frac{\partial U^{n+\alpha_f,c}_k}{\partial U^{n+1,b}_{j,t}}}_{\text{chain Rule}} + \frac{\partial R^a_i}{\partial U^{n+\alpha_m,c}_k} \frac{\partial U^{n+\alpha_m,c}_k}{\partial U^{n+1,b}_{j,t}} \right\}$$

$\to \alpha_f \gamma \Delta t\, \delta_{cb}\, \delta_{kj}$ $\qquad$ $\to \alpha_m \delta_{cb}\, \delta_{kj}$

$$= \frac{\partial R^{M,a}_i}{\partial U^{n+\alpha_f,b}_j} \alpha_f \gamma \Delta t + \frac{\partial R^{M,a}_i}{\partial U^{n+\alpha_m,b}_{j,t}} \alpha_m$$

$$G^{ab}_i = \frac{\partial R^{M,a}_i}{\partial P^{n+1,b}_{i,t}} = \frac{\partial R^{M,a}_i}{\partial p^{n+\alpha_f,b}} \alpha_f \gamma \Delta t$$

$$D^{ab}_j = \frac{\partial R^{C_{LHS},a}}{\partial U^{n+1,b}_{j,t}} = \frac{\partial R^{C_{LHS},a}}{\partial U^{n+\alpha_f,b}_j} \alpha_f \gamma \Delta t + \frac{\partial R^{C_{LHS},a}}{\partial U^{n+\alpha_m,b}_{j,t}} \alpha_m$$

$$C^{ab} = \frac{\partial R^{C_{LHS},a}}{\partial P^{n+1,b}_{i,t}} = \frac{\partial R^{C,a}}{\partial p^{n+\alpha_f,b}} \alpha_f \gamma \Delta t$$

2/29/08

The Code : Shape Functions

In e3

Shp : N_a
Shgl : N_{a,ξ_i}

xi : ξ

ngauss - # quadrature points

getshp (   )

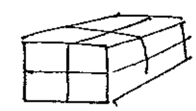
shp & shgl copyied into shape and shdrv

Interpolate the integrand

e3ivar

input

yl = localize nodal values $Y^e_{,a}$
ycl = "
acl : $y_{,t}$ vector
sclr : Value of scalar that transporting
xl = Coord of nodes of elements



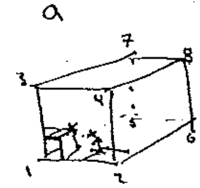
Yl (npro, nshl, nflow)
counter   1-5 unknowns VPT

y(int) interpolates

Pres = Pres + shape · $y^e_a(i, n, i)$ Pres has length npro
u1
u2
u3
T

(e = el, a = 8, unk)

a



8 interior points

get thermo = equation of state
calculates thermo constants

l

solution subscript
l - nodal / local
i - integral

e3 metric

$$\frac{\partial N_a}{\partial x_i} = \frac{\partial N_a}{\partial \xi_a} \frac{\partial \xi}{\partial x_i}$$ } metric

WdetJ = W · determinant of the Jacobian Wg'D

Shg : global grad of shape weight function

e3wmlt.

2/29/08

Plug in finite dimensional space

$$\left\{\begin{matrix} w_i = \sum_{A=1}^{n_{np}} N_i^A w_i^A \\ q = \sum_{A=1}^{n_{np}} N_p^A q^A \end{matrix}\right\} W \qquad \left\{\begin{matrix} u_i = \sum_{C=1}^{n_{np}} N_i^C u_i^C \\ p = \sum_{C=1}^{n_{np}} N_p^C p^C \end{matrix}\right\} Y$$

Plug into weak form (velbar)

$$\sum_{i=1}^{3}\sum_{A=1}^{n_{np}} w_i^A \Big\{ \int_\Omega N_i^A \{ \rho \overset{②}{u_{i,t}} + \overset{③⑬}{\rho \bar{u}_j u_{i,j}} - \overset{④}{f_i} \} + N_{i,j} \{ \overset{⑤}{\tau_{ij}} - \overset{⑥}{p\delta_{ij}} \} d\Omega$$
$$+ \int_\Gamma -N_i^A \{ \overset{⑧}{\tau_{ij}} - \overset{⑨}{p\delta_{ij}} \} d\Gamma$$
$$+ \sum_e^{n_{el}} \int_{\Omega^e} \{ \overset{⑩}{\tau_m u_j N_{i,j}^A \mathcal{L}} + \overset{⑫}{\tau_c N_{i,i}^A u_{j,j}} + \overset{⑭}{\bar{\tau} \mathcal{L}_j N_{i,j}^A \mathcal{L}_k u_{i,k}} \} d\Omega^e \Big\}$$
$$+ \sum_{A=1}^{n_{np}} q^A \Big\{ - \int_\Omega \overset{①⑪}{N_{p,i}^A \bar{u}_i} d\Omega + \int_\Gamma \overset{⑦}{N_p^A u_i n_i} d\Gamma \Big\}$$

rewrite

$$\sum^{3}\sum^{n_{np}} w_i^A \underbrace{\{\cdots\}}_{\overset{m}{R}_i^A} + q \underbrace{\{\cdots\}}_{\overset{c}{R}^A}$$

$$\underset{\sim}{\overset{m}{R}}_{3 n_{np}} \qquad \underset{\sim}{\overset{c}{R}}_{n_{np}}$$

Arbitrary weight functions

$w_i^A \Rightarrow \underset{\sim}{\overset{m}{R}} = 0$ $\quad 3 \times n_{np}$ equations

$q^A \Rightarrow \underset{\sim}{\overset{c}{R}} = 0$ $\quad n_{np}$ equations

> $4 \cdot n_{np}$ equations for $4 n_{np}$ unknowns $\left\{\begin{matrix} u_x \\ u_y \\ u_z \\ p \end{matrix}\right\}$ @ each node

3/4/08

These equations are assembled from element level integrals

Skip time integration

localization

$n + d_m$
$n + d_p$ } indicies
e, i

Note
$\underset{-}{i}$ means no sum

MOM

$$\overset{m}{R}_{\underset{-}{i}}^a = \int_\square \Big[N_{\underset{-}{i}}^a \{ \rho \overset{②}{u_{\underset{-}{i},t}} + \overset{③+⑬}{\rho \bar{u}_k u_{\underset{-}{i},k}} - \overset{④}{f_{\underset{-}{i}}} \} + N_{\underset{-}{i},k}^a \{ \overset{⑤}{\tau_{\underset{-}{i}k}} - \overset{⑥}{p\delta_{\underset{-}{i}k}} \} + N_{\underset{-}{i},k}^a \overset{⑩}{\tau_m u_k \mathcal{L}_{\underset{-}{i}}} + N_{\underset{-}{i},i}^a \overset{⑫}{\tau_c \bar{u}_{k,k}}$$
$$+ N_{\underset{-}{i},l}^a \overset{⑭}{\bar{\tau} \mathcal{L}_l \mathcal{L}_k u_{\underset{-}{i},k}} \Big] D \, d\square$$
$$+ \int_{\square_\Gamma} \Big[N_{\underset{-}{i}}^a \{ \overset{⑧}{\tau_{\underset{-}{i}k}} - \overset{⑨}{p \delta_{\underset{-}{i}k}} \} \Big] D \, d\square_\Gamma$$

CON

$$\overset{c}{R}^a = \int_\square - \overset{①⑪}{N_{p,k}^a \bar{u}_k} D \, d\square + \int_{\square_\Gamma} \overset{⑦}{N_p^a u_k n_k} D_\Gamma \, d\square_\Gamma$$

$$\overset{C4LHS}{R^a} = \int_\square N_p^a u_{k,k} D \, d\square + \int N_{p,k} \overbrace{\frac{\tau_M}{\rho} \mathcal{L}_k}^{\hat{u}} D \, d\square$$

Common
Incompress
compress

Incompressible Formulation

Strong Form (PDE)

$u_{i,i} = 0$

$\mathcal{L}_i = \rho u_{i,t} + \rho u_j u_{i,j} + p_{,i} - \tau_{ij,j} - f_i = 0_i + w_i$

Residual of NS eqn

Multiply by weight function — equivalent to $(w, \mathcal{L}u - f)$

Integrate over domain

Weak form

$$0 = \int_\Omega \{ q u_{i,i} + w_i \mathcal{L}_i \} d\Omega$$

$$= \int \{ q u_{i,i} + w_i \{ \rho u_{i,t} + \rho u_j u_{i,j} + p_{,i} - \tau_{ij,j} - f_i \} d\Omega$$

Integration by parts is on stress and continuity term only

$$0 = \int \{ -q_{,i} \overset{①}{u_i} + w_i \{ \overset{②}{\rho u_{i,t}} + \overset{③}{\rho u_j u_{i,j}} - \overset{④}{f_i} \} + w_{i,j} \{ \overset{⑤}{\tau_{ij}} - \overset{⑥}{p \delta_{ij}} \} d\Omega + \int_\Gamma \{ \overset{⑦}{q u_i n_i} - w_i (\overset{⑧}{\tau_{ij}} - \overset{⑨}{p \delta_{ij}}) n_j \} d\Gamma$$

$$+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \left\{ \left\{ \overset{⑩}{\tau_m u_j w_{i,j}} + \overset{⑪}{\frac{q_{,i}}{\rho}} \right\} \frac{p}{\rho_i} + \overset{⑫}{\tau_c w_{i,i} u_{j,j}} \right\} d\Omega^e$$

stabalized momentum — stabalizes continuity

$$+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \left\{ w_i \rho \overset{⑬}{\hat{u}_j} u_{i,j} + \overset{\text{SUPG Stabalization}}{\overset{⑭}{\bar{\tau}}} \left\{ \frac{\rho \hat{u}_j}{\tau_m} \right\} w_{i,j} \frac{\rho \hat{u}}{\tau_m} u_{i,t} \right\} d\Omega^e$$

conservation restoration

1) $\hat{u}_j := -\tau_m \frac{\mathcal{L}_i}{\rho}$

2) $\bar{u}_j = u_j + \hat{u}_j = u_j - \frac{\tau_m \mathcal{L}_i}{\rho}$

Dimensional considerations

1) terms multiplying q

① $q_{,i} u_i \, d\Omega \quad \frac{1}{L} \frac{L}{T} L^3 \Rightarrow \frac{L^3}{T}$ = [illegible]

⑦ $q u_i n_i \, d\Gamma \quad \frac{L}{T} L^2 \Rightarrow \frac{L^3}{T}$

⑪ $\tau_m \frac{q_{,i}}{\rho} \frac{p}{\rho_i} d\Omega \quad \frac{L^3}{LM} \frac{M}{L^3} \frac{L}{T^2} \frac{L^3}{1} = [\tau_m] \Rightarrow \frac{L^3}{T}$ if $[\tau_m] = T$

note for ⑮ & ⑯: $[q] = \frac{WM}{L^2}$

2) terms multiplying w_i

2-6, 8, 9 $[w_i] = \frac{ML}{T^2}$

⑩ $\tau_m u_j w_{i,j} \mathcal{L}_i \quad T \frac{L}{T} \frac{1}{L} \frac{M}{L^3} \frac{L}{T^2} L^3 = \frac{ML}{T^2}$

⑫ $\tau_c w_{i,i} u_{j,j} \quad [\tau_c] \frac{W}{L} \frac{L}{TL} L^3 = [\tau_c] \frac{WL^2}{T} = \frac{WML}{T^2}$ if $[\tau_c] = \frac{M}{LT}$

⑬ $w_i \rho u_{i,j} \hat{u}_j \quad W \frac{M}{L^3} \frac{L}{T} \frac{L}{TL} L^3 = \frac{ML}{T^2}$

⑭ $[\bar{\tau}] \frac{M}{L^3} \frac{L}{T} \frac{1}{T} \frac{W}{L} \frac{M}{L^3} \frac{L}{T} \frac{1}{T} \frac{L}{TL} L^3 = \frac{WML}{T^2}$ if $[\bar{\tau}] = \frac{T^3 L^3}{M}$

3/29/08

In the code

$$\tau_c = \frac{\rho}{8 \tau_m g_{ii}} = \left[\frac{M}{LT}\right] \qquad \tau_m = \frac{1}{\sqrt{\left(\frac{2c_1}{\Delta t}\right)^2 + u_k u_l g_{kl} + c_2 \nu^2 g_{kl} g_{kl}}} = [T]$$

$g_{kl} = \xi_{m,k} \xi_{m,l}$

$$\bar{\tau} = \frac{\tau_m}{\sqrt{\mathcal{L}_m \mathcal{L}_m g_{ii}}} \Rightarrow \left[\frac{T^3 L^3}{M}\right] \qquad \begin{pmatrix} \text{inverse} \\ \text{time scale} \\ \text{time step} \end{pmatrix} + \begin{pmatrix} \text{convection} \\ \Delta t \text{ through el} \end{pmatrix} + \begin{pmatrix} \text{diffusion} \\ \Delta t \text{ across element} \end{pmatrix}$$

In the code, the integral is cooked in a different way (Volker formulation)

Note ⑪

$$\frac{q_{,i} \tau_m \mathcal{L}_i}{\rho} = -q_{,i} \hat{u}_i$$

add to ①

$$-q (u_i + \hat{u}_i) = -q_{,i} \bar{u}_i$$

sim. ⑬ ③

$$w_i \{ \rho u_j u_{i,j} + \rho \hat{u}_j u_{i,j} \} = w_i \{ \rho \bar{u} u_{i,j} \}$$

⑭

$$\rho \hat{u}_j \equiv \mathcal{L}_j$$

$$\tau \mathcal{L}_j w_{i,j} \mathcal{L}_k u_{i,t}$$

Nader Dertiz

Recall what work we need to do

1) Build LHS & RHS of equation system

M^e_{ab} G^e_b

$\Downarrow \underline{A}$ $\Downarrow \underline{A}$

$\underline{M}$ $\underline{G}$ o.R

2) Solve the equation system

sol e3 > sub rou

PHASTA

Main.c → (calls) Phasta.cc

Phasta.cc
- input_fform.cc .... reads the input deck
- Process.f .... rest of program
  - gendat.f .... read digital data / reshapes data for optimal "FEM" talk
    - geombc.dat.1 .... coord connect BC
    - restart.0.1 .... IC data
  - itrdrv.f .... loops over time steps solves N-S
    - solgmr.f .... for current iteration / linearization form, solve Ax= MA4
    - ~~itrbc.f~~ .... satisfy the BC
    - itrBC.f

APP HERE RUN(#)

get/set BC()

→ Solgmr.f
- Solgmrs .... sparse solver
  - elmgmrs .... Form Ax=b
    - loopover blocks .... ability to break element loops into smaller loop; smaller # elements per block
      - asigmr ....
        - local .... $x_A \to x^e_a$, $x \to x_a$ — take global x vector, make local x-vector
        - e3.f .... $\int_\Omega = G^e_b$, M^e_{ab}
          - loop over integration points
            - e3ivar.f .... sample any component in the integrand
            - e3mtrx.f
            - e3conv.f .... calculates $\underline{A}_0$ $\underline{A}_i$; build adv flux vector F_{conv}
            - Γ_r vector built which multiplies N_b or $N_{b,i}$
            - e3source.f
            - e3visc.f .... take body force $\mathcal{F}$ into r
            - e3ls. .... build dif flux F_i, puts into r
            - e3DC .... stabilization SUPG
            - e3wmlt.f .... weight function multiply, complete integran $\Sigma g^e_b W$
            - loops over something
          - end loop quad points

A: as a assemble iteration

$\int N_b \{ \} + N_{b,i} \{ \}$ (Γ_r; 5, 5×3)

$\Gamma_r\{16:20\}$ $\mathcal{F}$

$\Gamma_r\{1:5\}$ $N_{b,1}\{F_1\}$

$\Gamma_r\{6:10\}$ $N_{b,2}\{F_2\}$

$\Gamma_r\{11:15\}$ $N_{b,3}\{F_3\}$

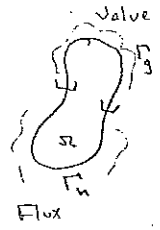
$\int^e_b = N_b \Gamma_r\{16:20\} + N_{b,1} \Gamma_r\{1:5\} + \ldots$

$G^e_b = \sum_{k=1}^{nL} g^e_b(\xi^k) W^k$

Recall: we may set 2 types of BC data

1) essential - fixes a nodes value (u, v, w, T, P, ρ) $\longrightarrow g$
2) Natural - fixes flux through a boundary element $\longrightarrow f_h$

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

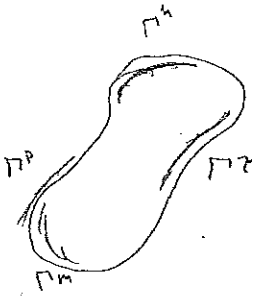
$$\int_{\Box_\Gamma} N_b \left\{ \rho U_n \begin{Bmatrix} \vec{U} \\ 1 \end{Bmatrix} + P \begin{Bmatrix} \vec{0} \\ n_x \\ n_y \\ n_z \end{Bmatrix} - \begin{Bmatrix} \vec{\tau}_n \\ u\tau_{xn} + v\tau_{yn} + w\tau_{zn} \end{Bmatrix} + \begin{Bmatrix} \vec{0} \\ f_n \end{Bmatrix} \right\} D_\Gamma d\Box_\Gamma$$

$$= \underbrace{\int_{\Box_{\Gamma_h^m}} N_b h^m \begin{Bmatrix} \vec{U} \\ 1 \end{Bmatrix} D_\Gamma d\Box_\Gamma}_{\text{mass flux}} + \underbrace{\int_{\Box_{\Gamma - \Gamma_h^m}} N_b \rho U_n \begin{Bmatrix} \vec{U} \\ 1 \end{Bmatrix} D_\Gamma d\Box_\Gamma}_{\text{consistent flux}}$$

$$+ \underbrace{\int_{\Box_{\Gamma_h^p}} N_b h^p \begin{Bmatrix} \vec{0} \\ n \end{Bmatrix} D_\Gamma d\Box + \int_{\Box_{\Gamma - \Gamma_h^p}} N_b P \begin{Bmatrix} \\ \end{Bmatrix} D_\Gamma d\Box_\Gamma}_{\text{Pressure flux}}$$

$$+ \int_{\Box_{\Gamma_h^\tau}} -N_b \begin{Bmatrix} \vec{0} \\ h_{xx} \\ h_{xy} \\ h_{xz} \end{Bmatrix} D_\Gamma d\Box_\Gamma + \int_{\Box_{\Gamma - \Gamma}} N_b \begin{Bmatrix} \vec{0} \\ \vec{\tau} \cdot \end{Bmatrix} D_\Gamma d\Box_\Gamma$$

$$+ \int N_b \begin{Bmatrix} \vec{0} \\ h \end{Bmatrix} D_\Gamma d\Box_\Gamma + \int N_b \begin{Bmatrix} \vec{0} \\ q \end{Bmatrix} D_\Gamma d\Box_\Gamma$$



"Floating Flux BC"
This process computes the flux through the entire boundary using prescribed data in place of consistent flux data, where provided

h values in code → BCFB integer that determines when to use cons, V per.

IBCFB
In compres code CBvar.f calculates coef h by interpolating BCFB computes γ, γ_m needed to γ_n in e36.f → IBCFB array set which to use cons, V h

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

$$\underline{\underline{M}} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \underline{\underline{M}}^e_{ab} \qquad G = \mathop{\mathbf{A}}_{e=1}^{n_{el}} G^e_a$$

A adds all elemental contributions to form the global nodes

Natural BC

Local level

$$\hat{G}^e_b = \int_\Omega [N_b \{ \underline{\underline{A}}_0 \underline{Y}_{,x} \cdots \} + \underbrace{\int_{\Omega_n} N_b \underline{F}_c \cdot \underline{n}_c D_n d\Omega_n}_{\text{only if touches Real BC}}$$

We need to understand this BC

Code creates a list of elements (faces) that need to be evaluated

$$\int N_b (\underbrace{\underline{F}^{adv}_c + \underline{F}^{diff}_c}) \cdot \underline{n}_c D_n d\Omega_n$$

further

$$\int N_b \left(\underbrace{\{\underline{u}\}\underline{u}^T + P\{\delta_{mn}\}}_{\{\tau_{in} u_n\}} - \{\tau_{in}\} + \{q_n\} \right) D_n d\Omega_n$$

n = normal

This suggests several "Natural" fluxes BC's

1) Mass Flux: ρu_n
   if we prescribe it we set $\rho u_n = h^m$ on Γ^m_h

2) Natural Pressure (weak)

3) Normal viscous flux (traction) $P = h^P$ on Γ^P_h

4) Heat Flux $\tau_{jn} = h^v_j$ on Γ^v_h

$-q_n = h^h$ on Γ^h_h

Note: convention states that $h^{\wedge}$ is a given value

Note could be a function of space