ALE FORMULATION (1)

<u>Given</u>: Mesh Velocity Field $\mathbf{w} = (w^x, w^y, w^z)$

<u>Then</u>: Euler eqns.

$$\left\{ \begin{array}{l} \rho \\ \rho u^x \\ \rho u^y \\ \rho u^z \\ \rho e \end{array} \right\}_{,t} + \left\{ \begin{array}{l} (u^i - w^i)\rho \\ (u^i - w^i)\rho u^x + p \\ (u^i - w^i)\rho u^y \\ (u^i - w^i)\rho u^z \\ (u^i - w^i)\rho e + u^i p \end{array} \right\}_{,i} = - \left\{ \begin{array}{l} \rho \\ \rho u^x \\ \rho u^y \\ \rho u^z \\ \rho e \end{array} \right\} \nabla \cdot \mathbf{w}$$

<u>Options</u>:

a) Integrate As Is:

$$\mathbf{M}_c|^n \, \Delta\mathbf{u} = \mathbf{r} + \mathbf{s}$$

b) Use ‘Finite Volume’ Form:

$$\frac{d}{dt} \int_{\Omega(t)} \mathbf{u} d\Omega + \int_{\Omega(t)} \nabla \cdot \mathbf{F} d\Omega = 0$$

$\Rightarrow$

$$\mathbf{M}_c|^{n+1} \, \mathbf{u}^{n+1} - \mathbf{M}_c|^n \, \mathbf{u}^n = \mathbf{r}$$

$\Rightarrow$

$$\mathbf{M}_c|^{n+1} \, \Delta\mathbf{u} = \mathbf{r} - \left(\mathbf{M}_c|^{n+1} - \mathbf{M}_c|^n \right) \cdot \mathbf{u}^n$$

ALE FORMULATION (2)

Boundary Conditions:

- Navier-Stokes:

$$\Delta\rho\mathbf{v} = \Delta\rho\mathbf{w}$$

- Euler:

$$\Delta\rho\mathbf{v}^* = \Delta\left[\rho\left(\mathbf{w} + \alpha\mathbf{t} + \beta\mathbf{n}\right)\right]$$

$$\Delta\rho\mathbf{v}^{n+1} = \Delta\left[\rho\left(\mathbf{w} + \alpha\mathbf{t}\right)\right]$$

  - Given $\mathbf{t}$:

$$\Delta\rho\mathbf{v}^{n+1} = \Delta\rho\mathbf{w} + \left[\left(\Delta\rho\mathbf{v}^* - \Delta\rho\mathbf{w}\right)\cdot\mathbf{t}\right]\cdot\mathbf{t}$$

  - Given $\mathbf{n}$:

$$\Delta\rho\mathbf{v}^{n+1} = \Delta\rho\mathbf{v}^* - \left[\left(\Delta\rho\mathbf{v}^* - \Delta\rho\mathbf{w}\right)\cdot\mathbf{n}\right]\cdot\mathbf{n}$$

GEOMETRIC CONSERVATION LAW (1)

Desired: Motion of Mesh Should Not Affect Solution:

$$\mathbf{u} = const.$$

Assume: $\rho = 1$; $\Rightarrow$ Continuity

$$\mathbf{M}_c\big|^{n+1} - \mathbf{M}_c\big|^{n} + \int_{\Omega^n} N^i \nabla \cdot (\mathbf{v} - \mathbf{w}) d\Omega = 0$$

- So-Called Geometric Conservation Law
- Determines $\mathbf{M}^{n+1}$
- Required for Severe/Rapid Mesh Movement

ALE MESH VELOCITIES (1)

Given:

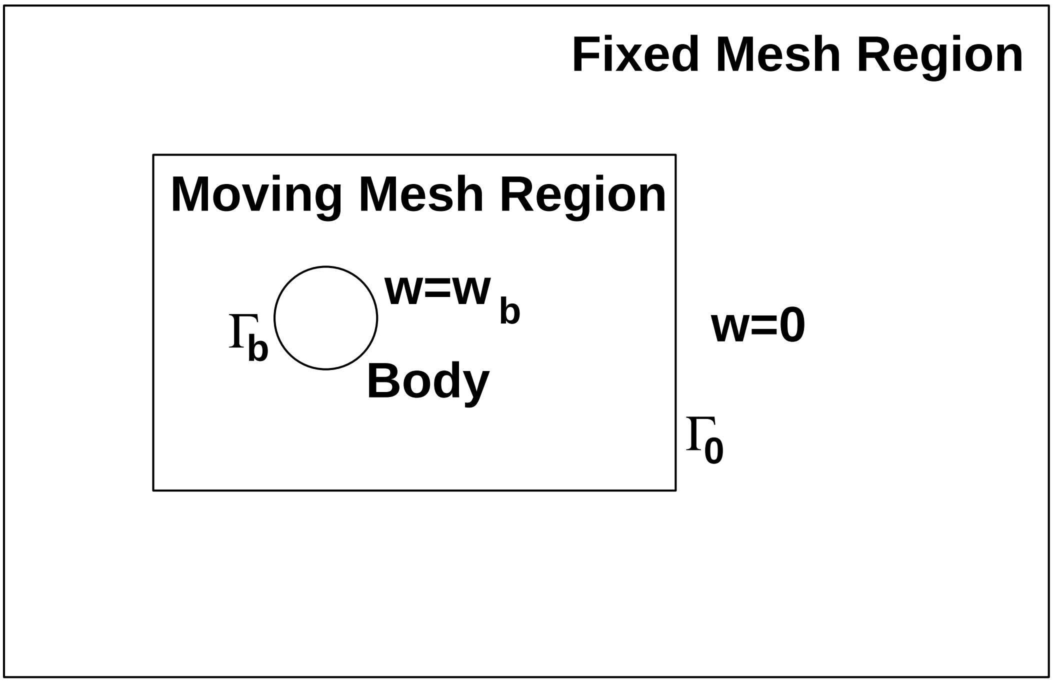
Moving Surfaces:

$$\mathbf{w}|_{\Gamma_0} = \mathbf{w}_0$$

At Some Distance From Moving Surfaces:

$$\mathbf{w}|_{\Gamma_1} = 0$$

Find: $\mathbf{w}$ That Minimizes Grid Distortion



ALE MESH VELOCITIES (2)

Options:

- Analytic User-Prescribed Function
- Smoothing of Coordinates
- Smoothing of Velocities

MESH VELOCITIES FROM ANALYTIC FUNCTIONS

Given: Distance From Moving Surface δ

$$\mathbf{w} = \mathbf{w}(\mathbf{x}|_\Gamma) \cdot f(\delta)$$

$\delta = 0$: $\Rightarrow f(\delta) = 1$

$\delta = \delta_1 \Rightarrow f(\delta) = 0$

Pros/Cons:

- User-Prescribed
- Fast (Algebraic) Once δ Is Computed
- δ: Problematic for:
  - Multibody Configurations
  - Parallel Machines
- Not General for Multibody Configurations

SMOOTHING OF COORDINATES

Idea: Replace Edges by Spring System and Relax

$$\mathbf{x}^{n+1}\big|_\Gamma = \mathbf{x}^n|_\Gamma + \Delta t \cdot \mathbf{w}|_\Gamma$$

Spring Force:

$$\mathbf{f}_i = \sum_{j=1}^{ns_i} c(|\mathbf{x}_j - \mathbf{x}_i|)(\mathbf{x}_j - \mathbf{x}_i)$$

Solve by Relaxation/PCG $\Rightarrow \mathbf{x}^{n+1} \Rightarrow$

$$\mathbf{w} = \frac{1}{\Delta t} \cdot \left(\mathbf{x}^{n+1} - \mathbf{x}^n\right)$$

Pros/Cons:

- General
- Ficticious Velocity if Initial Mesh Not In Equilibrium
  - Special $c(|\mathbf{x}_j - \mathbf{x}_i|)$ Functions
- Movement in x May Imply Movement in y, z
- No Guarantee of Non-Negative Elements

SMOOTHING OF VELOCITIES (1)

Idea: Laplacian of Mesh Velocities

$$\nabla \cdot \mathbf{k} \cdot \nabla \mathbf{w} = 0$$

FEM + Relaxation/PCG $\Rightarrow$

$$C^{ii} \Delta \mathbf{w}^i = -\Delta t \cdot K^{ij} \left(\mathbf{w}^i - \mathbf{w}^j \right)$$

$$C^{ii} = \sum_{i \neq j} |K^{ij}|$$

Optimal Δt-Sequence:

$$\Delta t^i = \frac{1}{1 + cos[\frac{\pi(i-1)}{n_v}]} \quad , i = 1, n$$

Pros/Cons:

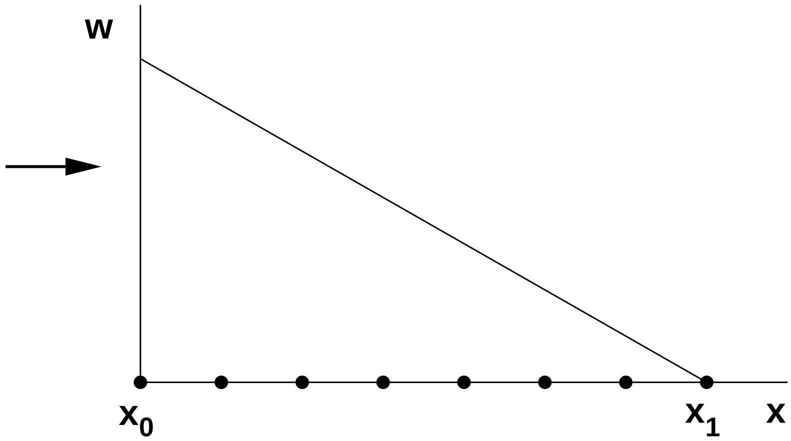
- General
- No Ficticious Velocity if Initial Mesh Not In Equilibrium
- Movement in $x \Rightarrow$ No Movement in y, z
- No Guarantee of Non-Negative Elements
- Element-Based Codes

$$\nabla^2 \mathbf{w} \approx - \left(\mathbf{M}_l - \mathbf{M}_c \right) \cdot \mathbf{w}$$

- Equivalent to $k = h^2$

SMOOTHING OF VELOCITIES (2)

1-D Case:



$$\frac{\partial^2 w}{\partial x^2} = 0 \quad , \quad w(0) = w_0 \quad , \quad w(x_1) = 0$$

$\Rightarrow$

$$w(x) = w_0 \left(1 - \frac{x}{x_1} \right)$$

$\Rightarrow$

$$\frac{\partial w}{\partial x} = g_v$$

SMOOTHING OF VELOCITIES (3)

$\forall$ 2 Elements: Change In Size δh During a Timestep:

$$\delta h = (w_2 - w_1) \cdot \Delta t = \Delta w \cdot \Delta t$$

$\Rightarrow$ Size-Ratio of $\forall$ 2 Elements i, j:

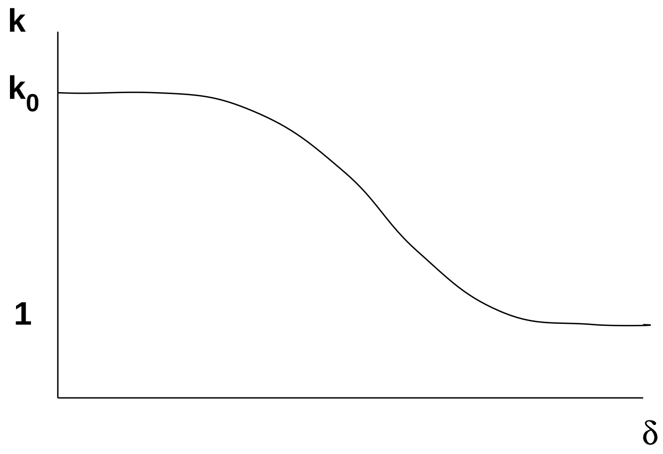
$$\left.\frac{h_i}{h_j}\right|^{n+1} = \frac{h_i + \Delta w_i \cdot \Delta t}{h_j + \Delta w_j \cdot \Delta t} = \frac{h_i + g_v \cdot h_i \cdot \Delta t}{h_j + g_v \cdot h_j \cdot \Delta t} = \left.\frac{h_i}{h_j}\right|^{n}$$

$\Rightarrow$ **All Elements Deformed in the Same Way**

SMOOTHING OF VELOCITIES (4)

Desired:

- Close to Body:
  - Small Elements
  - $k \gg 1 \Rightarrow |\nabla \mathbf{w}| \to 0$
- Away from Body:
  - Larger Elements
  - $k \to 1 \Rightarrow |\nabla \mathbf{w}|$ Laplacian



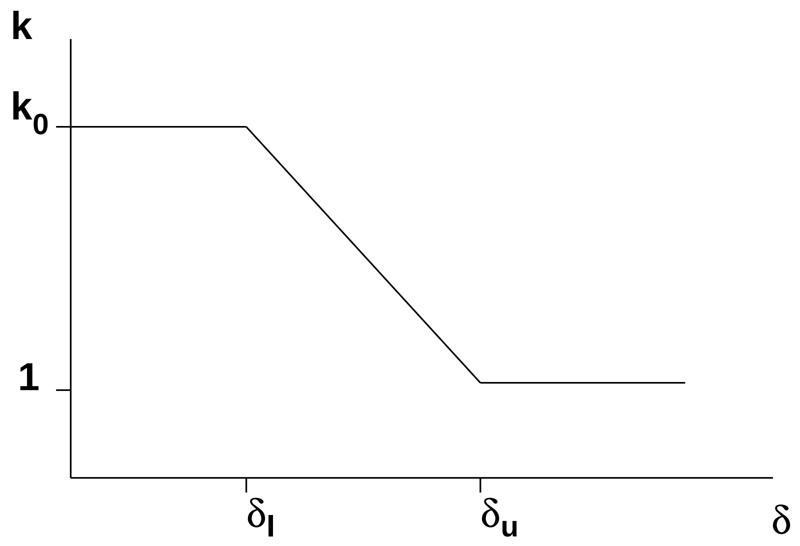
Need:

- Distance From Body Evaluation $\Rightarrow \delta$
- $k = k(\delta)$

SMOOTHING OF VELOCITIES (5)

Then: Use Constant-Linear-Constant $k(\delta)$

$$k = k_0 + (1 - k_0) max\left(0, min\left(1, \frac{\delta - \delta_l}{\delta_u - \delta_l}\right)\right)$$



PDE-BASED DISTANCE EVALUATION: EIKONAL EQN

$$|\nabla \delta| = 1 \quad , \quad \delta_\Gamma = 0$$

Re-write as:

$$\frac{\nabla \delta}{|\nabla \delta|} \cdot \nabla \delta = 1$$

Interpret δ as Steady-State of:

$$\delta_{,t} + \mathbf{v} \cdot \nabla \delta = 1 \quad , \quad \mathbf{v} = \frac{\nabla \delta}{|\nabla \delta|}$$

Remarks:

- Nonlinear Hyperpolic Equation
- Advection Velocity: Unit Gradient of $\delta(\mathbf{x})$
  Eigenvalue = 1
- Solve Using Standard Hyperpolic PDE Solver

PDE-BASED DISTANCE EVALUATION: LAPLACIAN

1D:

$$\delta_{,xx} = -s \quad ; \quad \delta(0) = 0 \quad ; \quad \delta_{,x}\big|_{x_1} = 0$$

$\Rightarrow$

$$\delta = sx\left(x_1 - \frac{x}{2}\right)$$

$\Rightarrow$

$$x_1 = \sqrt{\frac{2\delta_1}{s}} \quad ; \quad \delta_{,x}\big|_0 = \sqrt{2 \cdot s \cdot \delta_1}$$

$\Rightarrow$ For Unit Gradient at $x = 0$:

$$s = 1/2\delta_1 \quad \Rightarrow \quad x_1 = 2\delta_1$$

In General

$$\nabla^2\delta = -s \quad ; \quad \delta\big|_{\Gamma_0} = 0 \quad ; \quad \delta_{,n}\big|_{\Gamma_1} = 0$$

Poisson Condition Imposed As:

$$\delta \leq \delta_1$$

PROJECTIVE PREDICTION OF MESH VELOCITIES

Laplacian/Elasticity Can Be Time-Consuming

$$\mathbf{K} \cdot \mathbf{w} = \mathbf{r}$$

Idea: Start With Good Estimate for $\mathbf{w}$

- $\mathbf{w} = 0$ (Steady State)
- $\mathbf{w}$ From Previous Timesteps

PPMV

Basic Assumption: $\mathbf{K} \approx const.$

Timesteps/Iterations: $n-1, n-2, ..., n-i$:

$$\mathbf{K} \cdot \mathbf{w}^i = \mathbf{r}^i$$

Perform Least Squares Approximation of $\mathbf{r}$ With Basis $\mathbf{r}^i, i = 1, l$:

$$\left(\mathbf{r} - \alpha_i \mathbf{r}^i\right)^2 \rightarrow min$$

$\Rightarrow$

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{s} \quad , \quad A^{ij} = \mathbf{r}^i \cdot \mathbf{r}^j \quad , \quad s^i = \mathbf{r}^i \cdot \mathbf{r}$$

With α_i:

$$\mathbf{w} = \alpha_i \mathbf{w}^i$$

<u>Observations</u>:

- Dramatic Reduction of Nr. of Iterations: `niter` $O(20) \rightarrow O(2)$
- Need only 2-4 Search Vectors $\Rightarrow$ Low Storage