FINITE ELEMENT FORMULATIONS AND ADVANCED APPLICATIONS

Rainald Löhner

MS 4C7, Fluids and Materials
School of Computational Sciences and Informatics
George Mason University, Fairfax, VA 22030, USA

ACKNOWLEDGEMENTS (1)

GMU/SCS:

Chi Yang, Juan Cebral, Jarek Tuszyński, Jacob Waltz, Fernando Camelli, Fumiya Togashi, Joaquin Arteaga-Gomez, Rommel Espinoza, Marcelo Castro, Fernando Mut, Sunil Appanaboyima

SAIC, McLean, VA:

Joseph Baum, Hong Luo, Eric Mestreau, Dmitri Sharov, Orlando Soto

General Atomics and ES3, San Diego, CA:

Daniele Pelessone, Charles Charman

NRL:

Ravi Ramamurti, Alex Shostko, Choong Oh

ACKNOWLEDGEMENTS (2)

ESI, Paris:

Jean Roger, Pierre de Kermel, Ming Zhu, Philippe Ravier, Eberhard Haug, Jan Clinkemaier

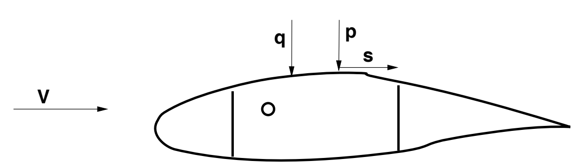
CIMNE, Barcelona:

Eugenio Oñate, Ramon Ribo, Chris Morton, Sergio Idelssohn, Carlos Sacco

Institut für Statik, TU Braunschweig:

Elmar Walhorn, Björn Hübner

THE COMPLETE SYSTEM



THERMAL FIELD (1)

$$\mathbf{LHST} \cdot \Delta\mathbf{T} = \left(\frac{1}{\Delta t}\mathbf{C} + \Theta\mathbf{K}\right) \cdot \Delta\mathbf{T} = \mathbf{f}_t^i + \mathbf{f}_t^e$$

Δt : Timestep
$\mathbf{C}$: Heat Capacitance Matrix
$\mathbf{T}$: Nodal Temperatures
$\mathbf{K}$: Heat Conduction Matrix
$\mathbf{f}$: Internal and External Thermal Loads

THERMAL FIELD (2)

Split Into Degrees of Freedom:

f : On Fluid Surface
t : Remaining Ones

$$\left\{\begin{matrix}\mathbf{LHST}_{ff} & \mathbf{LHST}_{ft} \\ \mathbf{LHST}_{tf} & \mathbf{LHST}_{tt}\end{matrix}\right\} \cdot \begin{pmatrix}\Delta\mathbf{T}_f \\ \Delta\mathbf{T}_t\end{pmatrix} =$$

$$\begin{pmatrix}\mathbf{f}_f \\ \mathbf{f}_t\end{pmatrix}^i + \begin{pmatrix}\mathbf{f}_f \\ \mathbf{f}_t\end{pmatrix}^e + \begin{pmatrix}\mathbf{L} \cdot (\mathbf{q}_f + \Theta\Delta\mathbf{q}_f) \\ 0\end{pmatrix}$$

$\mathbf{L}$: Load Matrix
$\mathbf{q}_f$: Heat Loads on Surface

SOLID REGION (1)

$$\mathbf{LHSS} \cdot \Delta\dot{\mathbf{X}} = (\alpha\mathbf{M}_s + \beta\mathbf{D} + \gamma\mathbf{K})\,\Delta\dot{\mathbf{X}} =$$

$$\mathbf{f}_s^i + \mathbf{f}_s^e + \Theta\Delta\mathbf{f}_s^e + \mathbf{Q}(\mathbf{T} + \Theta\Delta\mathbf{T})$$

$\mathbf{X}$: Displacement Vector
$\dot{\mathbf{X}}$: Velocity Vector
$\mathbf{M}_s$: Mass Matrix
$\mathbf{D}$: Damping Matrix
$\mathbf{K}$: Stiffness Matrix
$\mathbf{f}_s^i$: Internal (Stiffness, Damping, Inertia) Forces
$\mathbf{f}_s^e$: External (Gravity, Fluid Surface, ..) Forces
$\mathbf{Q}$: Thermal Stress Matrix
$\mathbf{T}$: Nodal Temperatures

SOLID REGION (2)

Split Into Degrees of Freedom:

f: On Fluid Surface
s: Remaining Ones

$$\left\{\begin{matrix}\mathbf{LHSS}_{ff} & \mathbf{LHSS}_{fs} \\ \mathbf{LHSS}_{sf} & \mathbf{LHSS}_{ss}\end{matrix}\right\} \cdot \begin{pmatrix}\Delta\dot{\mathbf{X}}_f \\ \Delta\dot{\mathbf{X}}_s\end{pmatrix} =$$

$$\begin{pmatrix}\mathbf{f}_f \\ \mathbf{f}_s\end{pmatrix}^i + \begin{pmatrix}\mathbf{f}_f \\ \mathbf{f}_s\end{pmatrix}^e + \begin{pmatrix}\mathbf{L} \cdot (\mathbf{s}_f + \Theta\Delta\mathbf{s}_f) \\ 0\end{pmatrix} +$$

$$\begin{pmatrix}\mathbf{Q}_f(\mathbf{T} + \Theta\Delta\mathbf{T}) \\ \mathbf{Q}_s(\mathbf{T} + \Theta\Delta\mathbf{T})\end{pmatrix}$$

$\mathbf{L}$: Load Matrix
$\mathbf{s}_f$: Fluid Stresses on Surface

FLUID REGION (1)

$$\mathbf{LHSF} \cdot \Delta\mathbf{U} = \left(\frac{1}{\Delta t}\mathbf{M}_f + \theta_f \mathbf{J}\right)\Delta\mathbf{U} = \mathbf{f}^i + \mathbf{f}^e$$

$\mathbf{U}$: Vector of Unknowns
$\mathbf{M}_f$: Mass Matrix
$\mathbf{J}$: Jacobian of Discretized Fluxes
$\mathbf{f}$: Internal and External Forces

FLUID REGION (2)

Split Into Degrees of Freedom:
s : On Solid Surface
f : Remaining Ones

$$\begin{Bmatrix} \mathbf{LHSF}_{ss} & \mathbf{LHSF}_{sf} \\ \mathbf{LHSF}_{fs} & \mathbf{LHSF}_{ff} \end{Bmatrix} \cdot \begin{pmatrix} \Delta\mathbf{U}_s \\ \Delta\mathbf{U}_f \end{pmatrix} =$$

$$\begin{pmatrix} \mathbf{f}_s \\ \mathbf{f}_f \end{pmatrix}^i + \begin{pmatrix} \mathbf{f}_s \\ \mathbf{f}_f \end{pmatrix}^e$$

CONTINUITY ACROSS DOMAINS (1)

1. Temperatures: CTD → CSD

$$\mathbf{T}_s = \mathbf{I}_{st}\mathbf{T}_t$$

$\mathbf{I}_{st}$: 3-D Interpolation Matrix

2. Temperatures: CTD → CFD

$$\mathbf{T}_f = \mathbf{I}_{ft}\mathbf{T}_t$$

$\mathbf{I}_{ft}$: Surface to Surface Interpolation Matrix

3. Velocities: CSD → CFD

$$\mathbf{v}_f|_{\Gamma_s} = \mathbf{I}_{fs}\mathbf{v}_s = \mathbf{I}_{fs}\dot{\mathbf{X}}_\mathbf{s}$$

$\mathbf{I}_{ft}$: Surf-Surf Interpolation Matrix

CONTINUITY ACROSS DOMAINS (2)

4. Thermal Loads: CFD → CTD

$$\mathbf{q}_f = \mathbf{G}_{tf}\mathbf{U}_f + \mathbf{G}_{ts}\mathbf{U}_s$$

5. Mechanical Loads: CFD → CSD

$$\mathbf{s}_f = \mathbf{G}_{sf}\mathbf{U}_f + \mathbf{G}_{ss}\mathbf{U}_s$$

ASSEMBLED SYSTEM (1)

$$\mathbf{LHSC}\cdot\begin{pmatrix}\Delta\mathbf{T}_f\\ \Delta\mathbf{T}_t\\ \Delta\dot{\mathbf{X}}_f\\ \Delta\dot{\mathbf{X}}_s\\ \Delta\mathbf{U}_s\\ \Delta\mathbf{U}_f\end{pmatrix} = \begin{pmatrix}\mathbf{f}_f\\ \mathbf{f}_s\\ \mathbf{f}_f\\ \mathbf{f}_t\\ \mathbf{f}_f\\ \mathbf{f}_s\end{pmatrix}^i + \begin{pmatrix}\mathbf{f}_f\\ \mathbf{f}_s\\ \mathbf{f}_f\\ \mathbf{f}_t\\ \mathbf{f}_f\\ \mathbf{f}_s\end{pmatrix}^e + \mathbf{RHSC}\cdot\begin{pmatrix}\mathbf{T}_f\\ \mathbf{T}_t\\ \dot{\mathbf{X}}_\mathbf{f}\\ \dot{\mathbf{X}}_\mathbf{s}\\ \mathbf{U}_s\\ \mathbf{U}_f\end{pmatrix}$$

ASSEMBLED SYSTEM (2)

$$\mathbf{LHSC} =$$

$$\left\{\begin{array}{cccccc}\mathbf{LHST}_{ff} & \mathbf{LHST}_{ft} & 0 & 0 & -\Theta\mathbf{LG}_{ts} & -\Theta\mathbf{LC}\\ \mathbf{LHST}_{tf} & \mathbf{LHST}_{tt} & 0 & 0 & -\Theta\mathbf{G}_{ss} & -\Theta\mathbf{C}\\ -\Theta\mathbf{Q}_f\mathbf{I}_{ft} & -\Theta\mathbf{Q}_f\mathbf{I}_{ft} & \mathbf{LHSS}_{ff} & \mathbf{LHSS}_{sf} & 0 & \\ -\Theta\mathbf{Q}_s\mathbf{I}_{ft} & -\Theta\mathbf{Q}_s\mathbf{I}_{ft} & \mathbf{LHSS}_{fs} & \mathbf{LHSS}_{ss} & 0 & \\ 0 & 0 & 0 & 0 & \mathbf{LHSF}_{ss} & \mathbf{LHSI}\\ 0 & 0 & 0 & 0 & \mathbf{LHSF}_{fs} & \mathbf{LHSF}\end{array}\right.$$

$$\mathbf{RHSC} = \left\{\begin{array}{cccccc}0 & 0 & 0 & 0 & \mathbf{LG}_{ts} & \mathbf{LG}_{tf}\\ 0 & 0 & 0 & 0 & 0 & 0\\ \mathbf{Q}_f\mathbf{I}_{ft} & \mathbf{Q}_f\mathbf{I}_{ft} & 0 & 0 & \mathbf{G}_{ss} & \mathbf{G}_{sf}\\ \mathbf{Q}_s\mathbf{I}_{ft} & \mathbf{Q}_s\mathbf{I}_{ft} & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\end{array}\right\}$$

REQUIREMENTS/PRIORITIES

- General Position/Load Transfer
  - Optimal Methods
  - Optimal Grids
- Modular (Codes/Modules Exchangable)
- General Transient
  - Steady State Possible
- Extendable
  - Multi-Macro-Physics (CEM, CTD, ..)
  - Multi-Micro-Physics (Length/Time)
  - Control
  - Optimization
- Fast Multidisciplinary Problem Definition
- Insightful Visualization

CONSEQUENCES (1)

- General Position/Load Transfer ⇒
  - Arbitrary Grid Transfer
  - Ability to Deal With Different Levels of Abstraction
- Modular ⇒
  - Minimum ‘Discipline Code’ Modification
  - Loose Coupling Approach
  - Load/ Position Transfer Standards/ Protocols
- General Transient ⇒
  - Time-Domain Formulation

CONSEQUENCES (2)

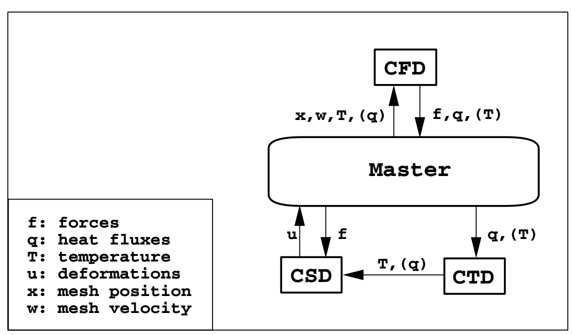
- Extendable ⇒
  - Loose Coupling Approach
- Fast Multidisciplinary Problem Definition ⇒
  - Seamless Integration/Problem Definition for CFD/CSD
  - Fully Automatic Grid Generation for Arbitrary Geometrical Complexity
- Insightful Visualization ⇒
  - CFD and CSD Visualization in Same Package

CODES USED (1)

- Do Not Re-Write CFD/CSD/CTD/... Codes
- Take Codes That Are:
  - Well Proven
  - Benchmarked
  - Debugged
  - Documented
  - Supported
  - (Public Domain)
  - Have a User Base/ Community
- Perform a Loose Coupling ⇒
  - Interpolation
  - Projection
- Provide Intergrated Pre/Post

CODES USED (2)

- CFD
  - FEFLO (Comp/Inco)
- CSD
  - FEEIGEN (Modal)
  - COSMIC-NASTRAN (Linear)
  - DYNA3D (Exp. Nonlinear)
  - GA-DYNA (Exp. Nonlinear)
  - NIKE3D (Imp. Nonlinear)
- CTD
  - COSMIC-NASTRAN (Linear)
  - FEHEAT (Nonlinear)



Loose Coupling

LOOSE COUPLING/STAGGERED SOLUTION

- Solve for CFD with imposed $\mathbf{v}_{sf}$
- Solve for CSD with imposed $\mathbf{s}_{sf}$ and $\mathbf{M}_{fs} \cdot \dot{\mathbf{v}}_{sf}$
- If Error Too Large: Iterate
- Added Mass:
  - Negligible for Solid + Air $[1:10^3 - 1:10^4]$
  - Non-negligible for Solid + Water [1 : 10]

⇒ **MINIMAL CODE RE-WRITE**

LOOSE COUPLING/STAGGERED SOLUTION

1. Navier Stokes → Euler By Setting:

$$\mathbf{v}_{fs} = \mathbf{v}_{fs}^t + \mathbf{v}_{fs}^n$$

Impose:

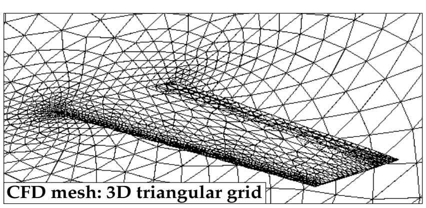
$$\mathbf{v}_{fs}^n = \mathbf{v}_{fs}^n$$

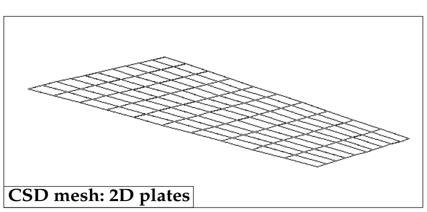
2. Timestepping via Loose Coupling:
   - Explicit CFD and CSD: Negligible Error
   - Implicit CFD or CSD: LHS Jacobians ⇒ Iterate
   - Steady-State: No Error

POSITION/VELOCITY TRANSFER

Why:

- Optimal Grid for Each Discipline ⇒ Different Grid Types/Sizes





Typical Position Transfer Problem

POSITION/VELOCITY TRANSFER ISSUES

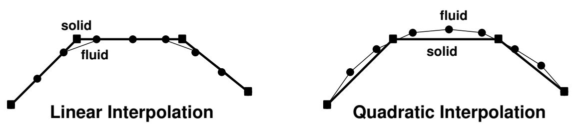
Desired:

- Geometric Fidelity

$$\mathbf{x}_f \approx \mathbf{x}_s \quad ; \quad \mathbf{v}_f \approx \mathbf{v}_s$$

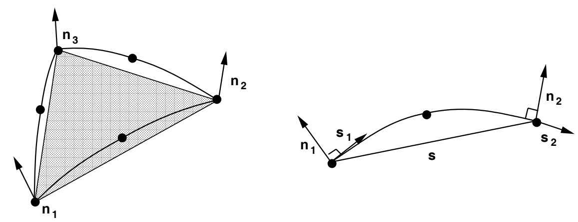
- Speed
- Generality
- Ability to Deal With Lower Dimensionality Abstractions
- Error Indicators

SURFACES NOT SEPARATED (GLUED)



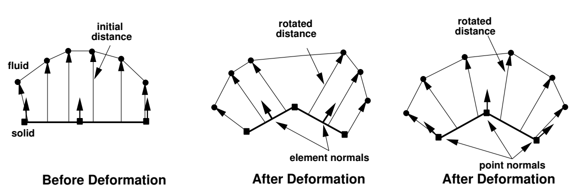
- $\mathbf{x}_1 = \mathbf{x}_2 \Rightarrow \delta = 0$
- $\mathbf{x}, \mathbf{v}$ Interpolated
  - Linear, Quadratic, Local Spline, Least Squares,...
- Typical Cases:
  - Large Deformation CSD, Euler CFD
  - Fine CSD/CFD Grids, Small Deformations

QUADRATIC SURFACE RECOVERY



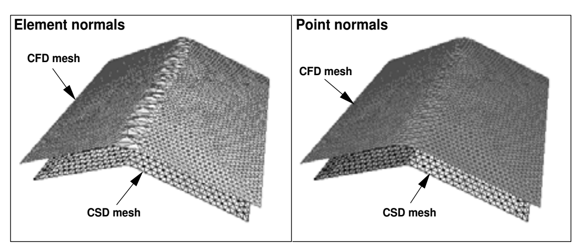
- Recover (Multiple) Normals at Points
  - Limiting Procedures for Normals
- Project Point Normal to Side
- Introduce Mid-Side Points
- Use Quadratic Shape-Functions
  $N^1 = \zeta_1(2\zeta_1 - 1)$, $N^2 = \zeta_2(2\zeta_2 - 1)$
  $N^3 = \zeta_3(2\zeta_3 - 1)$
  $N^4 = 4\zeta_1\zeta_2$, $N^5 = 4\zeta_2\zeta_3$, $N^6 = 4\zeta_1\zeta_3$

SURFACES SEPARATED, TRACKED (1)



- $\mathbf{x}_1 = \mathbf{x}_2 + \delta\mathbf{n}$
- $\mathbf{x}, \mathbf{v}$ Interpolated
  - Linear, Quadratic, Local Spline, Least Squares,...
- $\mathbf{n}$ Updated
  - Face, Normal, Averages, Rotations,...
- Typical Cases:
  - Doubly Loaded Wall
  - Aeroelasticity

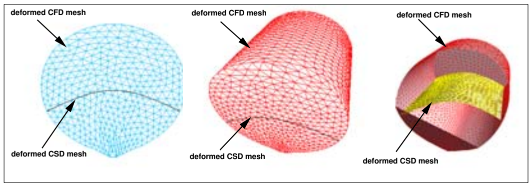
SURFACES SEPARATED, TRACKED (2)



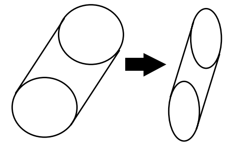
Face vs. Point Normals

$\Rightarrow$ Prefer Face Normals

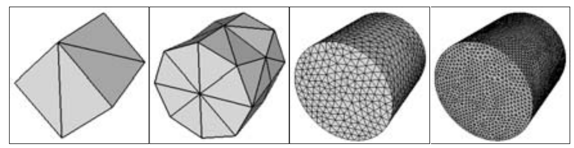
SURFACES SEPARATED, TRACKED (2)



ACCURACY STUDY (1)

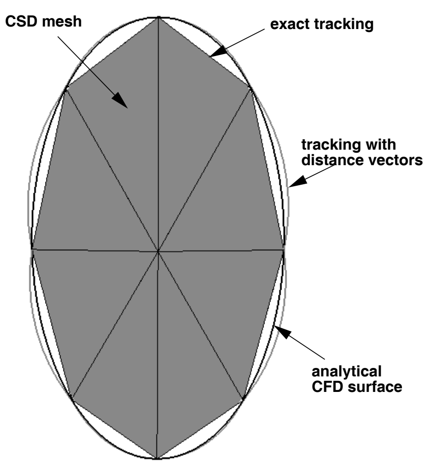


Analytical Deformation Used



Different CSD Grids

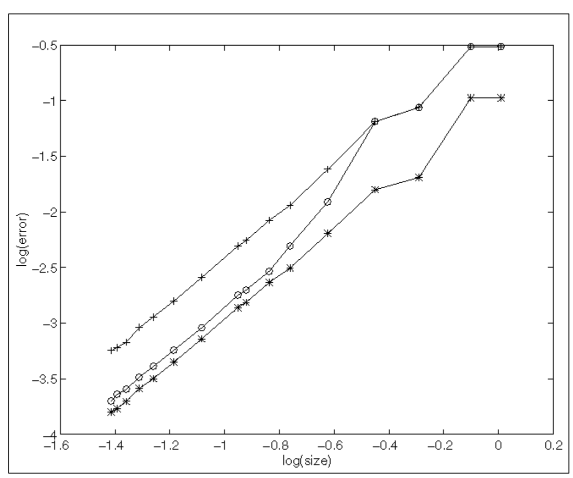
ACCURACY STUDY (2)



Surface Representation

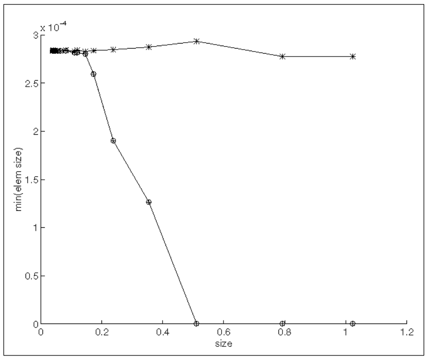
ACCURACY STUDY (3)

Error: $e = \sqrt{\frac{\int_{\Omega_f} (u_i - u_i^a)^2 d\Omega}{(u_{max}^a)^2}}$



Comparison of Tracking Schemes

ACCURACY STUDY (4)



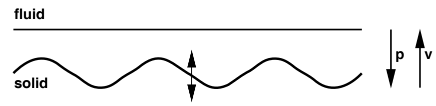
Smallest Element Size

⇒

Separated, Tracked Surfaces Allow for Very Coarse CSD Grids

SURFACES SEPARATED, NOT TRACKED

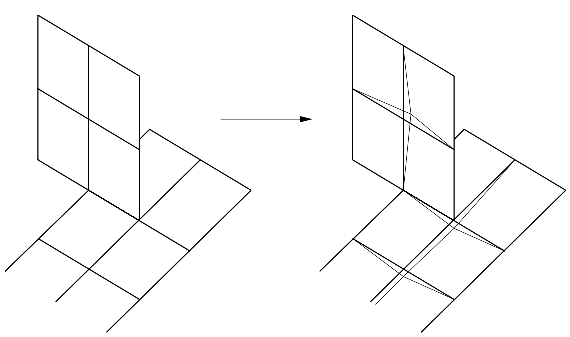
- $\mathbf{x}_1 \neq \mathbf{x}_2$
- Only $\mathbf{v}$ Interpolated
- Typical Cases:
  - Transpiration B.C. for CFD
  - Acoustic Loading



Tracking of Surface Velocities

DOUBLY DEFINED FACES (1)

Why: Doubly Loaded Shells



Unwrapping Doubly Loaded Shells

DOUBLY DEFINED FACES (2)

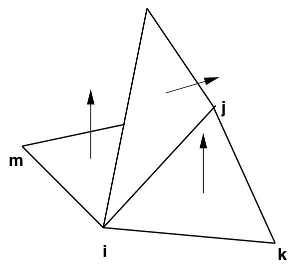
Given: List of Faces From CSD

Approach:

- Find Out the Doubly Defined Faces:
  - Linked List: Faces Surrounding Points
  - For Each Point: Get Faces & Check
- Define New Points and Consistent Surface Normals, Remembering Original CSD Points
- Add Thickness ⇒ Proper Position
- Interpolate

DOUBLY DEFINED FACES (3)

- Build: `fsufa(nedfa,nface)`
- For Multiple Neighbours:
  - Take 'Most Visible Neighbour' From Scalar Product



Most Visible Face

DOUBLY DEFINED FACES (4)

Initialize point array `lpoin(1:npoin)=0`

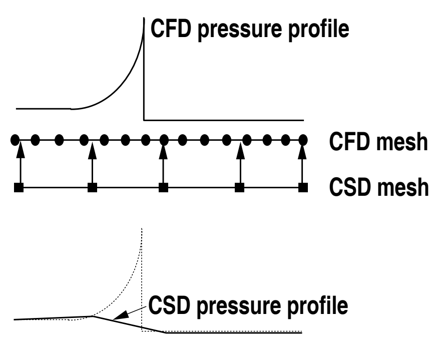
```
do iface=1,nface  ! Loop over the faces
 do inofa=1,nnofa  ! Loop over the face-nodes
 ipoin=bface(inofa,iface)  ! Point number
 if(ipoin.gt.0) then
  if(lpoin(ipoin).eq.0) then
   The point has not yet been surrounded ⇒
   ipold=ipoin
   ipnew=ipoin
  else
   As the point has already been surrounded and
   the point was left unconsidered:
   introduce a new point
   ipold=ipoin
   npoin=npoin+1
   ipnew=npoin
   Transcribe: ipold → ipnew (coord,etc.)
  endif
```


DOUBLY DEFINED FACES (5)

```
  Surround the point with faces obtained from
  fsufa that have point ipold in common
  Modify bface, setting entry of ipold to -ipnew
  lpoin(ipoin)=-1  ! Mark point as surrounded
 endif
 enddo
enddo
```

Restore `bface` to positive values.

INTERPOLATION FROM CFD TO CSD



- Known: $p_f = N_f^i p_{if}$
- Unknown: $N_f^i(\mathbf{x}_s)$
- ⇒ Interpolation From CFD to Known CSD Locations
- Simple
- Every CSD Point/Face Has a Load Value
- **Non Conservative**

CONSERVATIVE LOAD TRANSFER (1)

Desired:

$$p_s(\mathbf{x}) \approx p_f(\mathbf{x})$$

$$\mathbf{f} = \int p_s \mathbf{n} d\Gamma = \int p_f \mathbf{n} d\Gamma$$

Weighted Residual Statement:

$$p_s = N_s^i p_{is} \quad , \quad p_f = N_f^j p_{jf}$$

Or:

$$\int N_s^i N_s^j d\Gamma p_{js} = \int N_s^i N_f^j d\Gamma p_{jf}$$

$$\mathbf{M}_c \mathbf{p}_s = \mathbf{r} = \mathbf{L} \mathbf{p}_f$$

CONSERVATIVE LOAD TRANSFER (2)

WRM Conservative, As $\sum_i N_s^i(\mathbf{x}) = 1$:

$$\int p_s d\Gamma = \int N_s^j d\Gamma p_{js} = \int \sum_i N_s^i N_s^j d\Gamma p_{js}$$

$$= \sum_i \int N_s^i N_s^j d\Gamma p_{js} = \sum_i \int N_s^i N_f^j d\Gamma p_{jf}$$

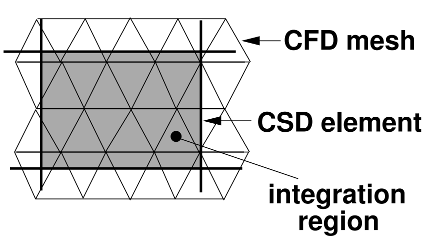
$$= \int \sum_i N_s^i N_f^j d\Gamma p_{jf} = \int N_f^j d\Gamma p_{jf} = \int p_f d\Gamma$$

CONSERVATIVE LOAD TRANSFER (3)

Needed:

$$\mathbf{L} \mathbf{p}_f = \int N_s^i N_f^j d\Gamma p_{jf}$$

Problem: Grids Are Not Nested

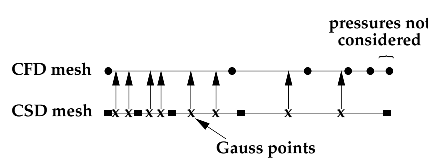


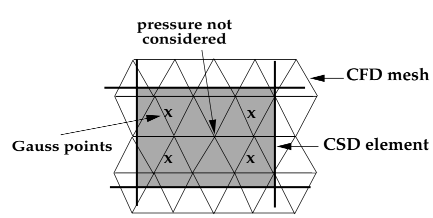
⇒ **Use Gaussian Quadrature**

OPTION 1: LOOP OVER CSD FACES (1)

$$r^i = \int N_s^i N_f^j d\Gamma p_{jf}$$

$$= \sum_s A_s \sum_{qp} W_{qp} N_s^i(\mathbf{x}_{qp}) N_f^j(\mathbf{x}_{qp}) p_{jf}$$





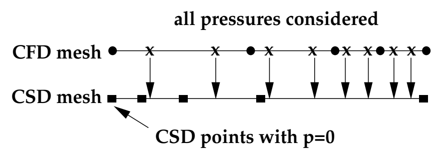
OPTION 1: LOOP OVER CSD FACES (2)

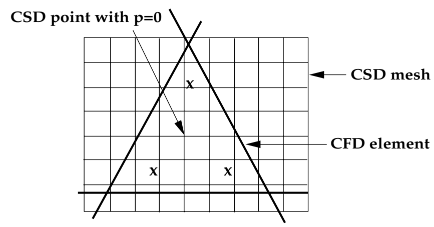
- Known: $N_s^i(\mathbf{x}_{qp})$
- Unknown: $p_f(\mathbf{x}_{qp}) = N_f^j(\mathbf{x}_{qp})p_{jf}$
- $\Rightarrow$ Interpolation From CFD to Known CSD Locations
- Every CSD Point/Face Has a Load Value
- **Non Conservative** $(h_{CFD} < h_{CSD})$

OPTION 2: LOOP OVER CFD FACES (1)

$$r^i = \int N_s^i N_f^j d\Gamma p_{jf}$$

$$= \sum_f A_f \sum_{qp} W_{qp} N_s^i(\mathbf{x}_{qp}) N_f^j(\mathbf{x}_{qp}) p_{jf}$$





OPTION 2: LOOP OVER CFD FACES (2)

- Known: $p_f(\mathbf{x}_{qp}) = N_f^j(\mathbf{x}_{qp})p_{jf}$
- Unknown: $N_s^i(\mathbf{x}_{qp})$
- $\Rightarrow$ Interpolation From CSD to Known CFD Locations
- Conservative
- **Not Every CSD Point May Have a Load Value**
  $(h_{CSD} < h_{CFD})$
  $\Rightarrow$ Adaptive Gaussian Quadrature

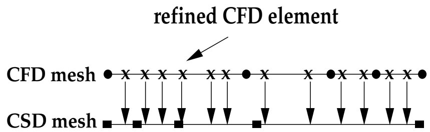
OPTION 3: LOOP OVER CFD FACES, CONSTANT P

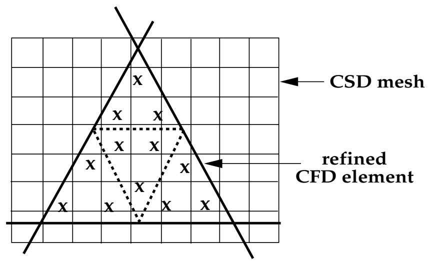
<u>Why</u>: Many CSD Codes Require Constant Face Loads

$$A_{is}p_{is} = \sum_f A_f \sum_{qp} W_{qp} P_s^i(\mathbf{x}_{qp}) p_f(\mathbf{x}_{qp})$$

- Known: $p_f(\mathbf{x}_{qp}) = N_f^j(\mathbf{x}_{qp})p_{jf}$
- Unknown: $N_s^i(\mathbf{x}_{qp})$
- $\Rightarrow$ Interpolation From CSD to Known CFD Locations
- Conservative
- 1-Point Quadrature Sufficient
- Not Every CSD Face May Have a Load Value

ADAPTIVE GAUSSIAN QUADRATURE (1)





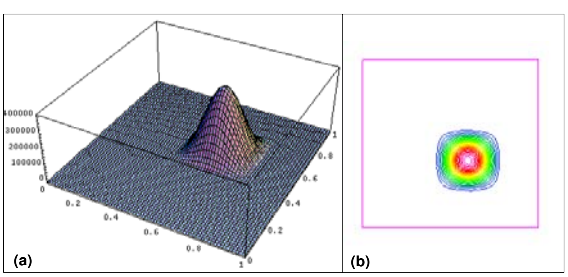
ADAPTIVE GAUSSIAN QUADRATURE (2)

- Before Projection: Monitor Size
  - Divide Accordingly
- Until Converged:
  - Project With Unit Pressure Diagnostics
  - Refine As Required
- Final Projection

ACCURACY STUDY (1)

Analytical Distribution:

$$p(x,y) = A\left[1 + cos\left(\frac{(x-x_c)\pi}{x_0}\right)\right]\left[1 + cos\left(\frac{(y-y_c)\pi}{y_0}\right)\right]$$



ACCURACY STUDY (2)

Error Measurements:

- ‘Least Squares’

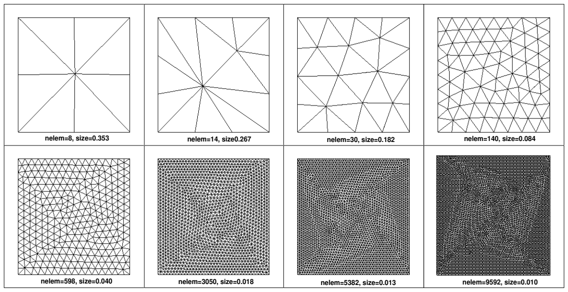
$$e_{LS} = \sqrt{\frac{\int_{\Omega_s}(p_f - p_s)^2 d\Omega}{\int_{\Omega_f} p_f^2 d\Omega}}$$

- ‘Relative’

$$e^i = \frac{|p_f^i - p_s^i|}{p_f^i}$$

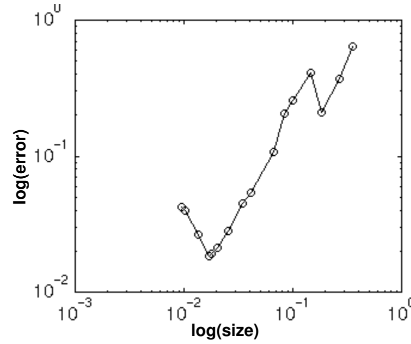
ACCURACY STUDY (3)

Grids Chosen:



ACCURACY STUDY (4)

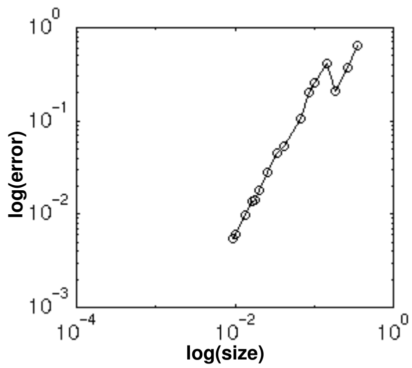
Results for 3 Gauss Points per CFD Face



No Refinement ⇒ Too Few

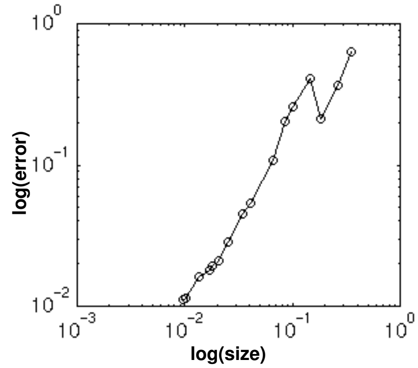
ACCURACY STUDY (5)

Results for 3 Gauss Points per CFD Face



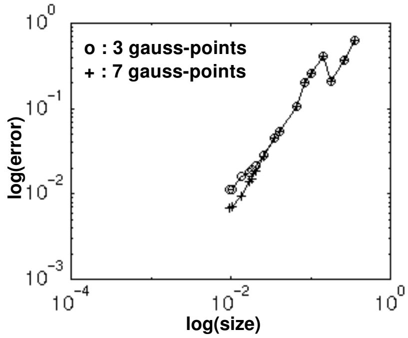
Uniform Refinement ⇒ Too Many

ACCURACY STUDY (6)



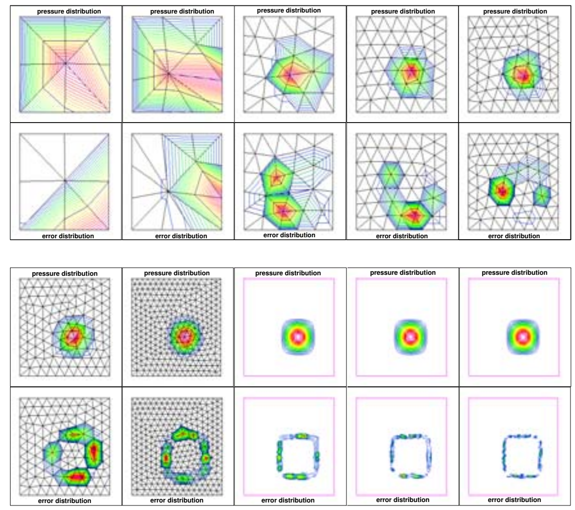
Adaptive Refinement (3QP)

ACCURACY STUDY (7)



Adaptive Refinement (3/7QP)

ACCURACY STUDY (8)



Loads and Error Distribution

MONOTONICITY PRESERVING TRANSFER

WRM Projection:

$$\mathbf{M}_c \mathbf{p}_s = \mathbf{r} = \mathbf{L}\mathbf{p}_f$$

- Consistent Mass:
  - Higher Accuracy
  - Possible Over/Undershoots
- Solved Iteratively As:

$$\mathbf{M}_l \mathbf{p}_s^{i+1} = \mathbf{r} + (\mathbf{M}_l - \mathbf{M}_c)\mathbf{p}_s^i \quad , \quad i = 1, niter$$

- $\mathbf{p}^0 = 0$
- Usually: $niter = 3$

PFCT (1)

Observations:

- Need to Iterate With M_l Anyhow
- First Pass:

$$\mathbf{M}_l \mathbf{p}^l = \mathbf{r}$$

- Lumped Mass:
  - Lower Accuracy
  - Less Over/Undershoots

Key Ideas:

- Low-Order Scheme: M_l-Projection
- High-Order Scheme: M_c-Projection
- Monotonicity Via FCT

PFCT (2)

- Compute Low-Order Projection

$$\mathbf{M}_l \mathbf{p}^l = \mathbf{r}$$

- Compute Antidiffusive Flux:

$$d = (\mathbf{M}_l - \mathbf{M}_c)\mathbf{p}^h$$

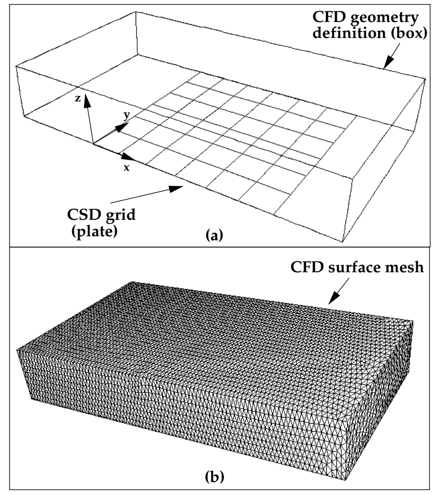
- Limit Antidiffusive Flux via FEM-FCT:

$$d' = c_l d$$

- Add Antidiffusive Flux:

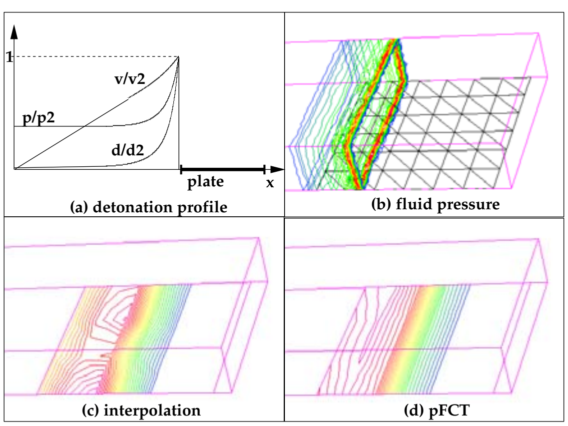
$$\mathbf{M}_l \mathbf{p} = \mathbf{M}_l \mathbf{p}^l + d'$$

SHOCK-PLATE INTERACTION (1)



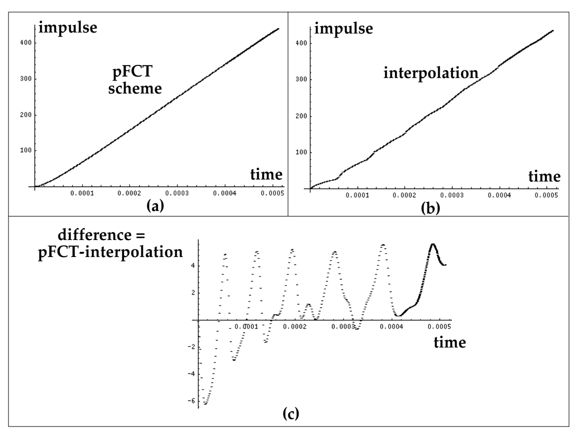
Problem Definition and Surface Grids

SHOCK-PLATE INTERACTION (2)



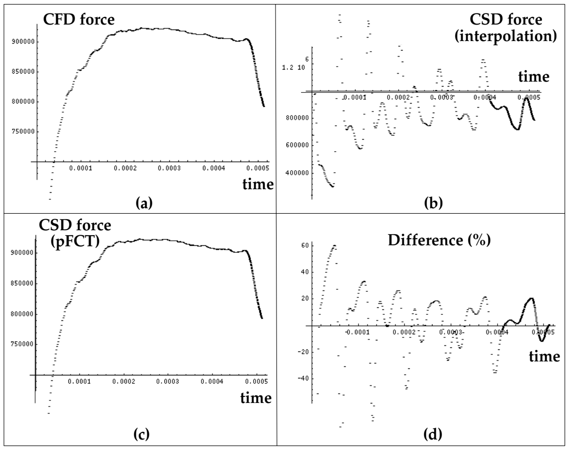
Results Obtained

SHOCK-PLATE INTERACTION (3)



Comparison of Impulses

SHOCK-PLATE INTERACTION (4)



Comparison of Forces

LOAD/FLUX TRANSFER ISSUES

<u>Desired</u>:

- Accuracy

$$\sigma_s(\mathbf{x}) \approx \sigma_f(\mathbf{x})$$

- Conservation

$$\mathbf{f} = \int \sigma_s \mathbf{n} d\Gamma = \int \sigma_f \mathbf{n} d\Gamma$$

- Speed [Interpolation, Adaptive Quadrature]
- Generality
- Ability to Deal With Lower Dimensionality Abstractions

POSITION/VELOCITY TRANSFER ISSUES

<u>Desired</u>:

- Geometric Fidelity

$$\mathbf{x}_f \approx \mathbf{x}_s \quad ; \quad \mathbf{v}_f \approx \mathbf{v}_s$$

  - Glued (Linear, Quadratic, Least Squares,..)
  - Separated (Initial Distance, Normal Rotation,..)
- Speed
- Generality
- Ability to Deal With Lower Dimensionality Abstractions
- Error Indicators

ACCURACY + CONSERVATION OF WHAT ?

<u>Accuracy</u>:

$$\mathbf{v}_f(\mathbf{x}) \approx \mathbf{v}_s(\mathbf{x})$$

$$\sigma_s(\mathbf{x}) \approx \sigma_f(\mathbf{x})$$

<u>Energy</u>:

$$W = \int \mathbf{v}_s^t \cdot \sigma_s \cdot \mathbf{n} d\Gamma = \int \mathbf{v}_f^t \cdot \sigma_f \cdot \mathbf{n} d\Gamma$$

ACCURACY + CONSERVATION OF WHAT ?

Forces:

$$\mathbf{f} = \int \sigma_s \cdot \mathbf{n} d\Gamma = \int \sigma_f \cdot \mathbf{n} d\Gamma$$

Moments:

$$\mathbf{m} = \int \mathbf{r} \times \sigma_s \cdot \mathbf{n} d\Gamma = \int \mathbf{r} \times \sigma_f \cdot \mathbf{n} d\Gamma$$

FORCE CONSERVATION (1)

- For Clarity: Use p
- WRM
- $\sum_i N_s^i(\mathbf{x}) = 1$

$$p_s = N_s^i \hat{p}_{is} \quad , \quad p_f = N_f^j \hat{p}_{jf}$$

$$\int N_s^i N_s^j \mathbf{n} d\Gamma \hat{p}_{js} = \int N_s^i N_f^j \mathbf{n} d\Gamma \hat{p}_{jf}$$

$$\mathbf{M}_{ss} \hat{\mathbf{p}}_s = \mathbf{r} = \mathbf{L}_{sf} \hat{\mathbf{p}}_f$$

FORCE CONSERVATION (2)

$$\int p_s \mathbf{n} d\Gamma = \int N_s^j \mathbf{n} d\Gamma \hat{p}_{js} = \int \sum_i N_s^i N_s^j \mathbf{n} d\Gamma \hat{p}_{js}$$

$$= \sum_i \int N_s^i N_s^j \mathbf{n} d\Gamma \hat{p}_{js} = \sum_i \int N_s^i N_f^j \mathbf{n} d\Gamma \hat{p}_{jf} =$$

$$\int \sum_i N_s^i N_f^j \mathbf{n} d\Gamma \hat{p}_{jf} = \int N_f^j \mathbf{n} d\Gamma \hat{p}_{jf} = \int p_f \mathbf{n} d\Gamma$$

Remarks:

- May Violate Moment Conservation
- May Violate Energy Conservation

ENERGY CONSERVATION (1)

Continuous:

$$W = \int \mathbf{v}_s^t \cdot \sigma_s \cdot \mathbf{n} d\Gamma = \int \mathbf{v}_f^t \cdot \sigma_f \cdot \mathbf{n} d\Gamma$$

Discrete:

$$W = \hat{\mathbf{v}}_{is}^t \cdot \int N_s^i N_s^j \mathbf{n} d\Gamma \hat{\sigma}_{js} = \hat{\mathbf{v}}_{if}^t \cdot \int N_f^i N_f^j \mathbf{n} d\Gamma \hat{\sigma}_{jf}$$

$\Rightarrow$

$$\hat{\mathbf{v}}_s^t \mathbf{M}_{ss} \hat{\sigma}_s = \hat{\mathbf{v}}_f^t \mathbf{M}_{ff} \hat{\sigma}_f$$

ENERGY CONSERVATION (2)

Given Velocity:

$$\hat{\mathbf{v}}_f = \mathbf{I}_{fs}\hat{\mathbf{v}}_s$$

$$\mathbf{M}_{ss}\hat{\sigma}_s = \mathbf{I}^t_{fs}\mathbf{M}_{ff}\hat{\sigma}_f$$

Remarks:

- May Violate Force Conservation
- May Not be Locally Accurate

ENERGY CONSERVATION (3)

Given Forces:

$$\hat{\sigma}_s = \mathbf{L}_{sf}\hat{\sigma}_f$$

$$\mathbf{M}_{ff}\hat{\mathbf{v}}_f = \mathbf{L}^t_{sf}\mathbf{M}_{ss}\hat{\mathbf{v}}_f$$

Remarks:

- May Lead to Non-Smooth Deformations
- May Not be Locally Accurate

INTERPOLATION ALGORITHMS

Why:

- Remeshing
- Multi-Disciplinary Code Coupling
- Boundary Conditions from Discrete Data
- Visualization

INTERPOLATION

Basic Idea: Shape-Function Values of Coordinates

Given:

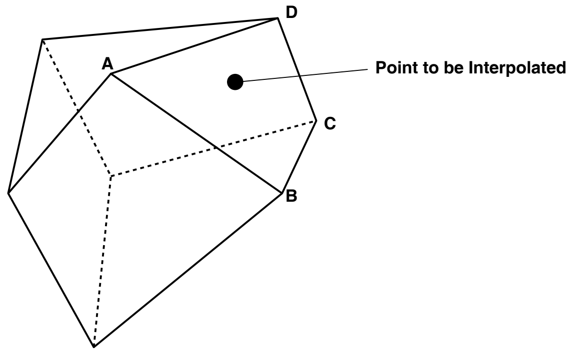
- Element *el* With Shape-Functions/Nodes: N^i, $\mathbf{x}_i$
- Point With Coordinates $\mathbf{x}_p$

Then:

$$\mathbf{x}_p = \sum_i N^i \mathbf{x}_i$$

Point In Element Iff:

$$min(N^i, 1 - N^i) > 0 \quad , \quad \forall i$$



Interpolation

BRUTE FORCE

Given:

- Elements and Points of Known Grid
- Point of Unknown Grid
- Other Info
    - None

Then:

- Vector-Loop Over the Elements of Known Grid, Checking

Remarks:

- Fastest 1-Time Start Algorithm

OCTREE SEARCH

Given:

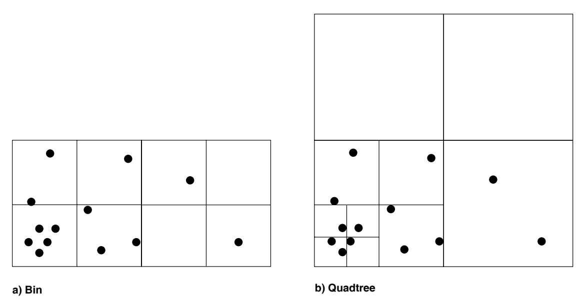
- Elements and Points of Known Grid
- Point of Unknown Grid
- Other Info:
    - Octree of Points of Known Grid
    - List of Elements Surr. Points of Known Grid

Then:

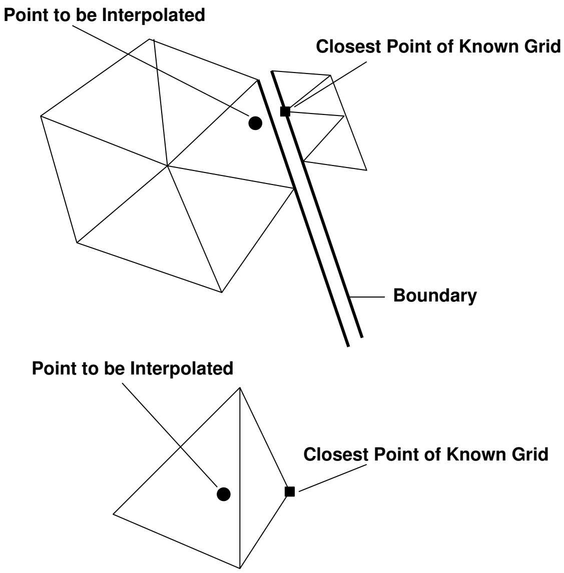
- Get Close Points of Known Grid From Octree
- Check Elements Surr. Close Points

Remarks:

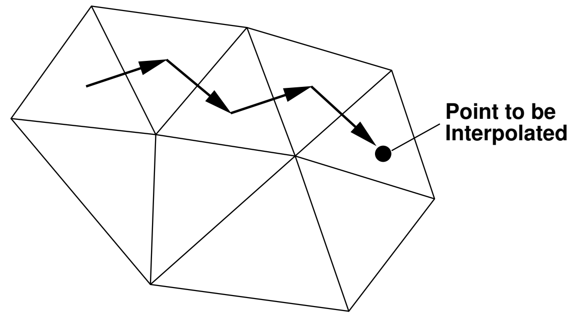
- Fastest N-Time Start Algorithm
- ⇒ Fastest Algorithm for Points



Spatial Ordering of Point Data



Possible Problems With Neighbour Data



Nearest Neighbour Search

NEIGHBOUR-TO-NEIGHBOUR (NNS)

Given:

- Elements and Points of Known Grid
- Point of Unknown Grid
- Other Info:
  - List of Neighbour Elements of Known Grid
  - Start-Element in Neighbourhood `IESTA`

Then:

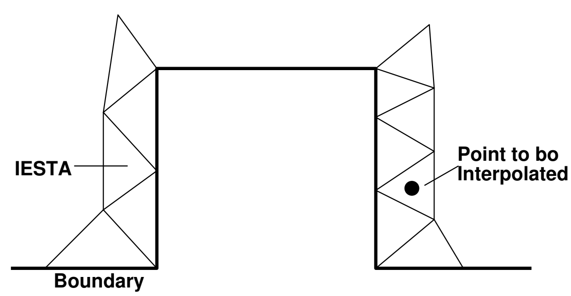
N.1 Compute Shape-Functions for `IESTA`

N.2 If Not In `IESTA`:
  - Set `IESTA` To Neighbour Associated With $min(N^i)$;
  - Goto N.1

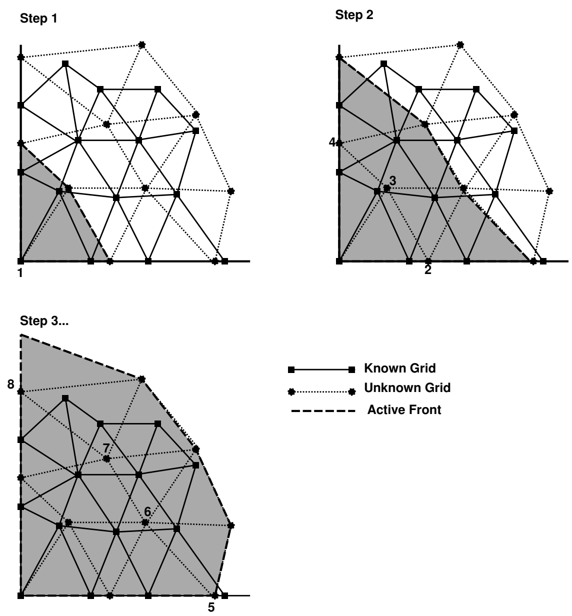
Endif

Remarks:

- Fastest Algorithm for Known Vicinity
- $\Rightarrow$ Fastest Algorithm for Grid-Grid Transfer



Possible Problems With NNS



Advancing Front Vicinity Algorithm

ADVANCING FRONT VICINITY ALGORITHM (1)

Idea: If Mesh to Mesh Interpolation Desired:

Advance Interpolation Front Using NNS

Given:

- Elements and Points of Known Grid
- Elements and Points of Unknown Grid
- Other Info:
  - List of Neighbour Elements of Known Grid
  - List of Elements Surr. Points for Unknown Grid

ADVANCING FRONT VICINITY ALGORITHM (2)

Then:

A.1 Mark Elements and Points of Unknown Grid as Untouched

A.2 Initialize List of Front Points for Unknown Grid

A.3 Select Next Non-Interpolated Point From Front `IPUNK`

A.4 Obtain Starting Element `IESTA` in Known Grid

A.5 Attempt Nearest Neighbour Search for `NTRY` Attempts;

- `IF` Unsuccessful: Use Brute Force
- `IF` Unsuccessful: Stop or Skip

⇒ `IEEND`

ADVANCING FRONT VICINITY ALGORITHM (3)

A.6 Store Shape-Functions and Host Elements

A.7 Loop Over Elements Surrounding `IPUNK`:

- `IF`: Element Has Not Been Marked:
  - Mark the Element
  - Loop Over the Points of This Element:
  - `IF`: the Point Has Not Been Marked:
    - Store `IEEND` as Starting Element For This Point;
    - Include This Point In Front;
  - `ENDIF`
- `ENDIF`

A.9 Mark Point `IPUNK` as Interpolated and `GOTO` A.3

IMPROVEMENTS

- Inside-Out Interpolation
  - Prefer Interior Points
- Vectorization
- Layering of Brute Force Search
  - Octree
  - Brute Force With Boundary Elements
  - Brute Force With Complete Mesh
- Concavity Measures
- Different Domains ($\mathbf{x}_2$ Not in Ω_1)

INSIDE-OUT INTERPOLATION

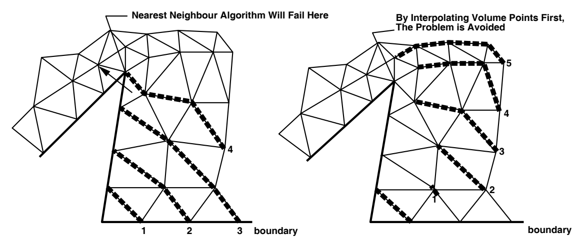
Problem:

- Brute Force Search at Edges/Ridges/Corners

Key Ideas:

- Interpolate First Interior Points
- From Interior → Boundary Points

  ⇒ Know Proximity



Inside-Out Interpolation

VECTORIZATION (1)

Key Ideas

- Interpolate Many Points At the Same Time
- Advance in Layers
- Work on ‘Remaining List of Points’
- Refresh List When Depleted

VECTORIZATION (2)

V.0 Set Remaining Number of Points `NPREM=NPFRT`

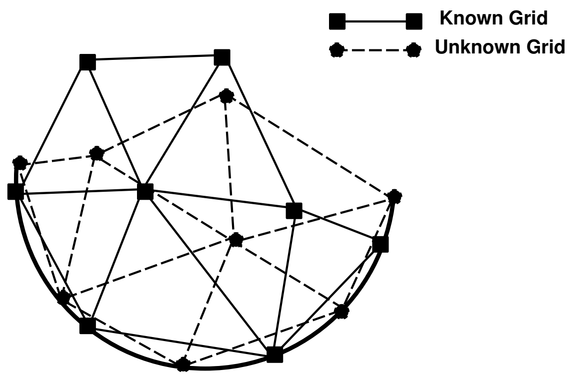
V.1 Perform Interpolation Checks in Vector Mode for `NPREM`

V.2 Write the `NPNXT` Points with No Host Elements into `LPCUR(1:NPNXT)`; If `NPNXT=0`: Stop.

V.3 Write the `NPREM-NPNXT` Points with Host Elements into `LPCUR(NPNXT+1:NPREM)`

V.4 Reorder Arrays with `LPCUR` ⇒ Remaining Points in `1:NPNXT`

V.5 Set `NPREM=NPNXT` and go to V.1.



Concave Boundaries

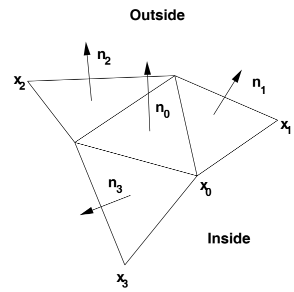
MEASURING CONCAVITY

Why: Concave Surfaces ⇒ $O(N_b^2)$-Search

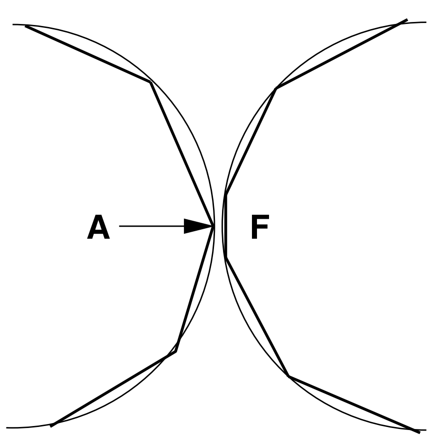
Idea: Measure Concavity and Proximity

- Measure Used: Visibility of Neighbouring Faces

$$d = \alpha |min(0, \mathbf{n} \cdot (\mathbf{x}_0 - \mathbf{x}_i))| \ \ , \ \ 0.5 < \alpha < 1.5$$



Measuring Concavity

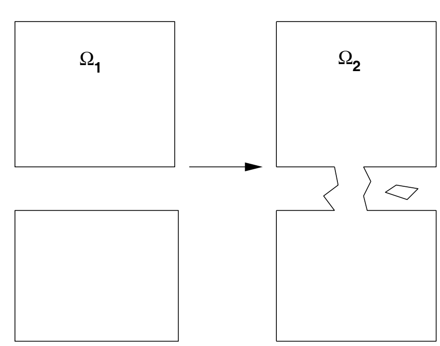


Problems...

DIFFERENT DOMAINS (1)

Problem:

- Interpolation Impossible
  ⇒ Brute Force Search Triggered



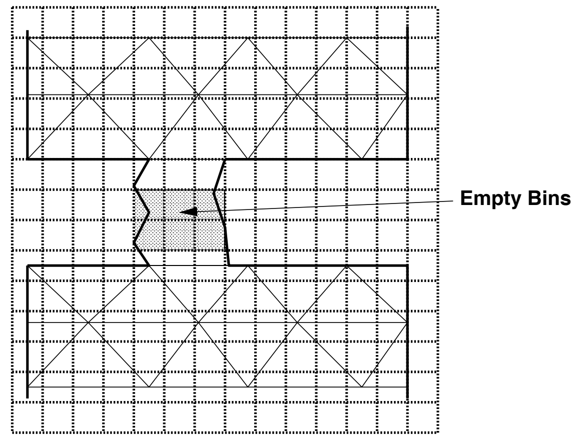
Interpolation With Different Domains

DIFFERENT DOMAINS (2)

Key Ideas:

- Form Bins
- Mark Bins Covered by Elements of Known Grid
- Mark Points of Unknown Grid in Bins Not Covered by Known Grid
- Do Recursively ('Telescoping')
  - Obtain Min/Max Bins of Marked Points
  - Redo Bin

DIFFERENT DOMAINS (3)



Marking Impossible Points

DIFFERENT DOMAINS (3)

Interpolate Impossible Points:

- Advancing Layers (+Isotropic)
- Upstream ($Ma > 1$)
- User-Prescribed Subroutine
- Closest Known Point

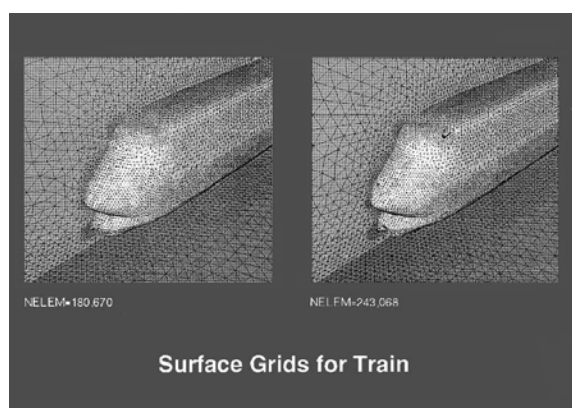


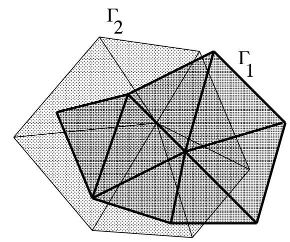
Table 1 Interpolation Timings

Case	NELEM$_1$	NELEM$_2$	# BFS	CPU-scalar	CPU-
Cube$_1$	34,661	30,801	0	0.1399	0.028
Cube$_2$	34,661	160,355	0	0.5360	0.110
Train$_1$	180,670	243,068	31	1.1290	0.340
Train$_2$	180,670	243,068	0	0.9905	0.202

SURFACE-GRID TO SURFACE-GRID

<u>Given</u>: Two Surface Triangulations

- Treat Topology as 2-D
- Treat Neighbour Search as 2-D
- Use Relative Distance Criterion for Normal Distance



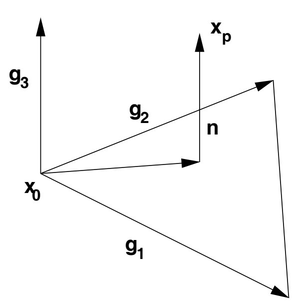
Surface–Surface Interpolation

Surface-Surface Interpolation

BASIC PROCEDURE(1)

$$\mathbf{x}_p = \mathbf{x}_0 + \sum_i \alpha^i \mathbf{g}_i$$

$$\alpha^4 = 1 - \alpha^1 - \alpha^2$$



Surface-Surface Interpolation

BASIC PROCEDURE(2)

Point On Face Iff:

a)

$$min(\alpha^i, 1-\alpha^i) \geq 0 \quad , \quad \forall i = 1,2,4 \qquad (1.a)$$

b)

$$d_n = \left|\alpha^3 \mathbf{g}_3\right| \leq \delta_n \qquad (1.b)$$

ISSUES (1)

a) Proper Choice of δ_n:

- What is Close Enough ?
- Relative/Absolute Distance ?
- Used Successfully:

$$\delta_n < c_n \cdot |\mathbf{g}_1 \times \mathbf{g}_2|^{0.5} \quad , \quad c_n = 0.05$$

c_n: May be Problem Dependent

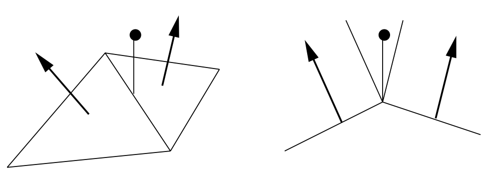
ISSUES (2)

b) Convex Ridges/Corners:

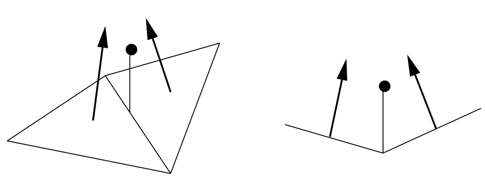
- Closeness Criterion May Never Be Satisfied
- $\Rightarrow$ Take Closest Face

c) Concave Ridges/Corners:

- Closeness Criterion May Be Satisfied By More Than One Face
- $\Rightarrow$ Compare And Take Smallest d_n



a) Concave Ridge (No Host Face)



b) Convex Ridge (Multiple Host Faces)

EXHAUSTIVE SEARCH

Triggered at Corners, Multi-Body Configurations

Consider all Faces Satisfying:

a)

$$min(\alpha^i, 1-\alpha^i) \geq \alpha_{es} < 0 \quad , \quad \forall i = 1,2,4 \qquad (2.a)$$

b)

$$d_n = \left|\alpha^3 \mathbf{g}_3\right| \leq \delta_{es} \qquad (2.b)$$

$\Rightarrow$

- Relax Closeness Criterion ($\alpha_{es} = -1$, $c_{es} = 0.5$)
- Keep the Face Closest to the Point
  - If Eqn.(1) Satisfied $\Rightarrow \delta = d_n$
  - If Eqn.(1) Not Satisfied $\Rightarrow$ Take Closest Distance to Face

$$\delta = min_{ij} \left|\mathbf{x}_p - (1-\beta_{ij})\mathbf{x}_i - \beta_{ij}\mathbf{x}_j\right|$$

$$\beta_{ij} = \frac{(\mathbf{x}_p - \mathbf{x}_i) \cdot (\mathbf{x}_j - \mathbf{x}_i)}{(\mathbf{x}_j - \mathbf{x}_i) \cdot (\mathbf{x}_j - \mathbf{x}_i)}$$

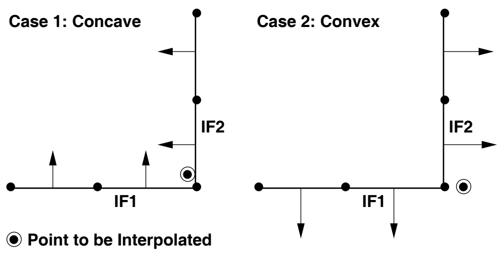
LOCAL EXHAUSTIVE SEARCH (1)

Triggered at Corners, Ridges

a) Case 1: `IF1` Satisfies Eqn.(1), But `IF2` Better

b) Case 2: `IF1, IF2` Have Same Distance

- $\Rightarrow$ Prefer Face With Eqn.(1a) Satisfied (`IF2`)

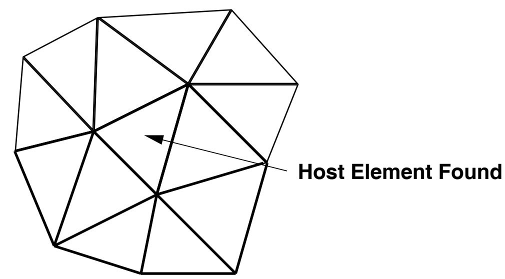


LOCAL EXHAUSTIVE SEARCH (2)

$\Rightarrow$

After Finding Face:

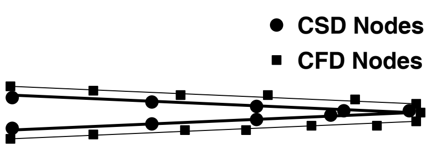
- Perform Local Exhaustive Search
- Consider All Faces Surrounding the Points of the Face



Local Exhaustive Search

TREATMENT OF THIN SURFACES

Why: For Shells



Interpolation With Thin Surfaces

Problem: Face on the 'Wrong Side' May be Closest

Solution: For Smooth Portions of the Surface

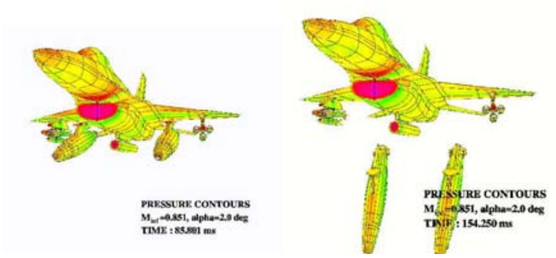
- Define Point Normal $\mathbf{n}_p$
- Only Consider Faces Aligned With Point Normal

$$\mathbf{n}_f \cdot \mathbf{n}_p > c_s \quad , \quad c_s = 0.5$$

F-16 FUEL TANKS (1)

- CFD Code: FEFLO
  - Compressible Euler
  - ALE, Mesh Movement/Remeshing
- Coupling:
  - Compute Loads
  - Move Fuel Tanks
- Machine: SGIO2K
  - 1-8 Procs
- Year: 1996
- Ref: AIAA-97-0166 (1997)

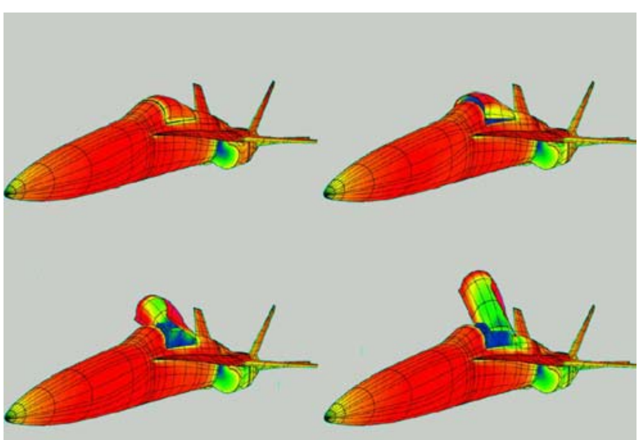
F-16 FUEL TANKS



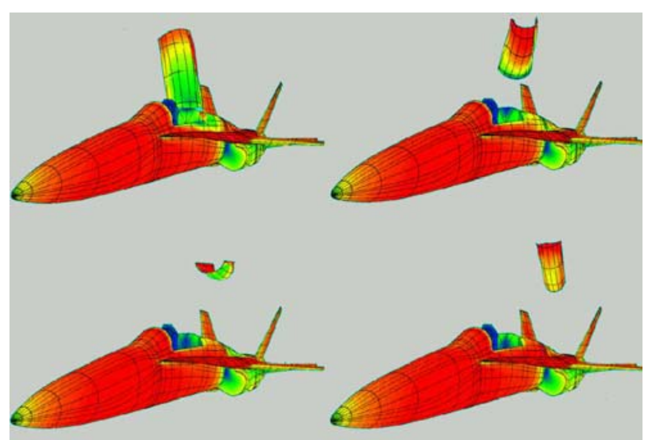
CANOPY EJECTION (1)

- CFD Code: FEFLO
  - Compressible Euler
  - ALE, Mesh Movement/Remeshing
- Coupling:
  - Compute Loads
  - Move Canopy
- Machine: SGIO2K
  - 1-8 Procs
- Year: 1996
- Ref: AIAA-97-1885 (1997)

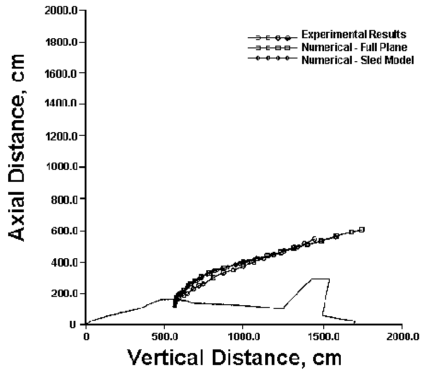
CANOPY EJECTION (2)



CANOPY EJECTION (3)

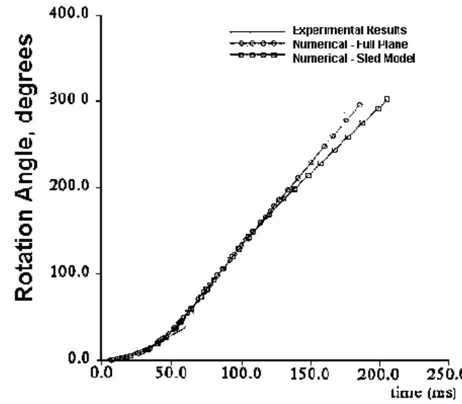


CANOPY EJECTION (4)



Canopy Ejection: Comparison of Runs

CANOPY EJECTION (5)



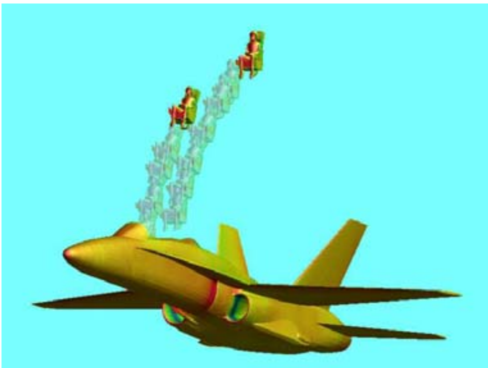
Canopy Ejection: Comparison of Runs

CANOPY + 2 PILOT EJECTION (1)

- CFD Code: FEFLO
  - Compressible Euler
  - ALE, Mesh Movement/Remeshing
- Coupling:
  - Compute Loads
    Force for Rocket Thrusters
  - Move Canopy + Pilots
- Machine: SGIO3K
  - 8-16 Procs
- Year: 1996
- Ref: Proc. 1st ICCFD, Kyoto, 387-392 (2000)

CANOPY + 2 PILOT EJECTION (2)

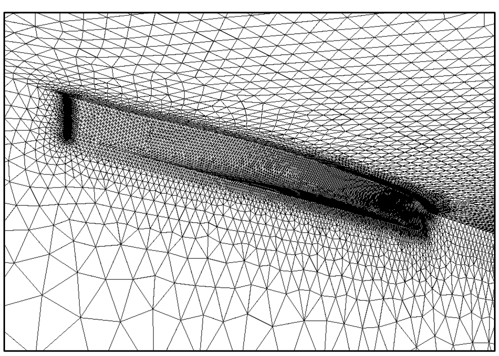
CANOPY + 2 PILOT EJECTION (3)



SINK+TRIM FOR SHIPS (1)

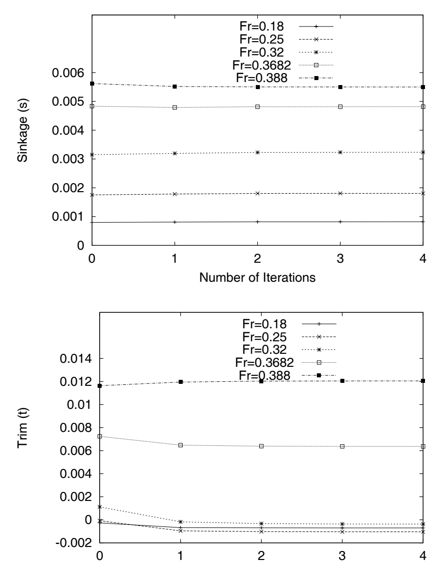
- CFD Code: FEFLO
  - Incompressible Euler
  - Mesh Movement/Interface Tracking
  - 400Ktet
- Coupling:
  - Run to Flow to Steady-State
  - Compute Loads
  - Move Ship
- Machine: SGIO2K
  - 1-8 Procs
- Year: 2001
- Ref: IJCFD 16, 3, 217-227 (2002)

SINK+TRIM FOR SHIPS (2)



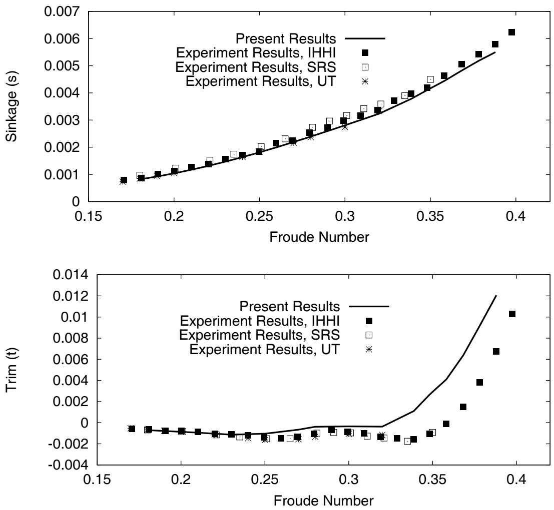
Series 60 Hull: Surface Mesh

SINK+TRIM FOR SHIPS (3)



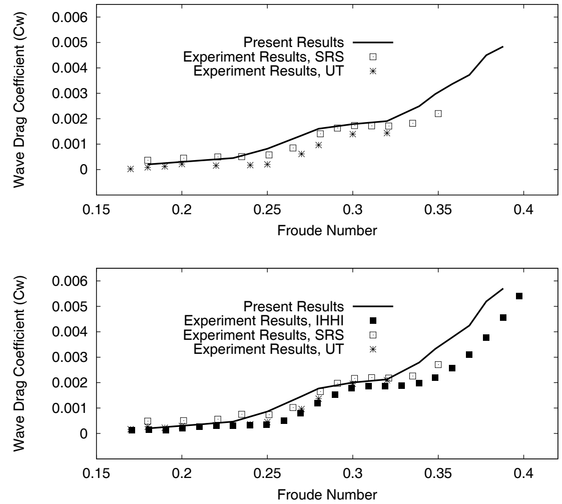
Series 60 Hull: Convergence of Sinkage and Trim

SINK+TRIM FOR SHIPS (4)



Series 60 Hull: Sinkage and Trim vs. Froude-Nr.

SINK+TRIM FOR SHIPS (5)

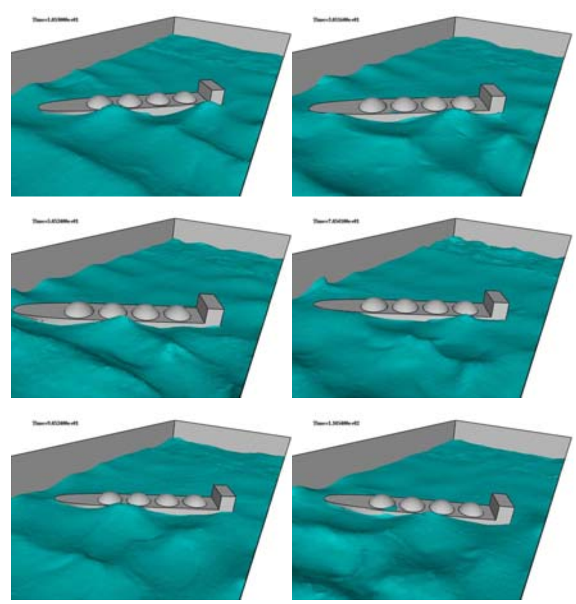


Series 60 Hull: Wavedrag for Fixed and Free Model

SHIP ADRIFT IN WAVES (1)

- CFD Code: FEFLO
  - Incompressible Euler
  - Mesh Movement/VOF
  - 3.4Mtet
- Coupling:
  - Wave Generator Upstream
  - Compute Loads
  - Move Ship
  - Full 6DOF
- Machine: SGI Altix
  - 4 Procs
- Year: 2005
- Ref: AIAA-06-0291 (2006)

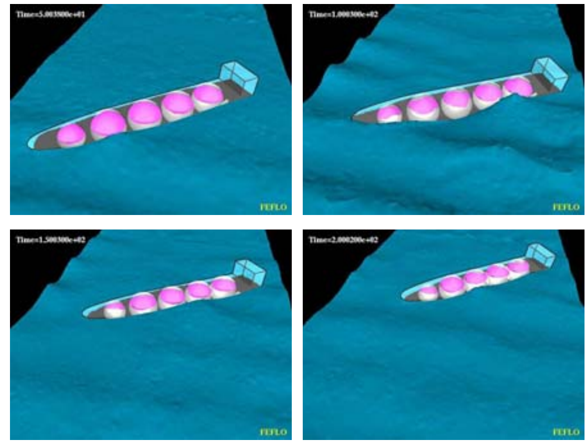
SHIP ADRIFT IN WAVES (2)



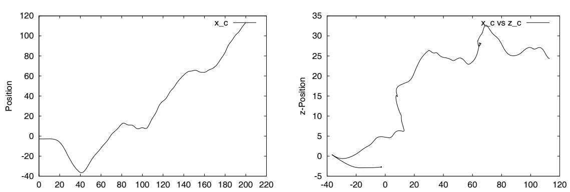
LNG TANKER ADRIFT IN WAVES (1)

- CFD Code: FEFLO
  - Incompressible Euler
  - Mesh Movement/VOF
  - Wall Contact
  - 2.7Mtet
- Coupling:
  - Tanks 80% Full (Sloshing)
  - Wave Generator Upstream
  - Compute Loads
  - Move Ship
  - Full 6DOF
- Machine: Dell P4
  - 3.2Ghz, 2Gbyte, Intel F, Linux OS
- Year: 2005
- Ref: AIAA-06-0291 (2006)

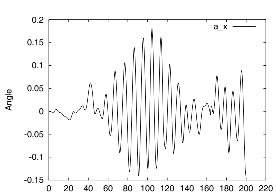
LNG TANKER ADRIFT IN WAVES (2)



LNG TANKER ADRIFT IN WAVES (3)



Position of Center of Mass

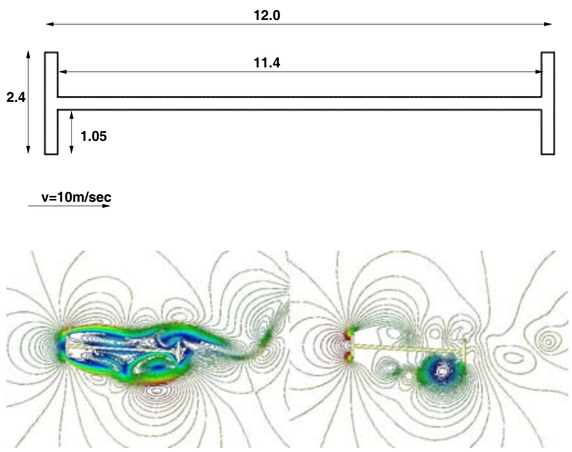


Roll Angle vs. Time

TACOMA BRIDGE SECTION (1)

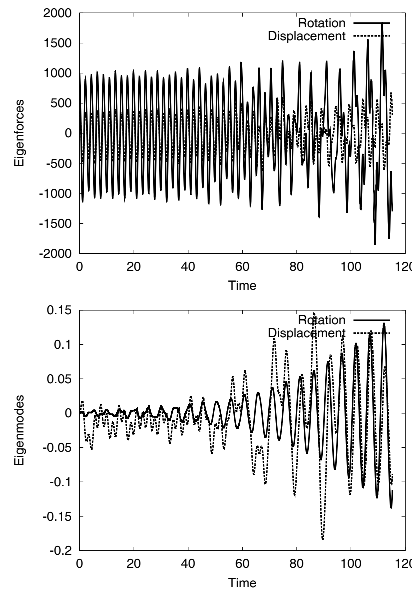
- CFD Code: FEFLO
  - Incompressible (381Ktet)
  - ALE, Mesh Movement
  - Laminar/MILES
  - $\rho = 1.25 kg/m^3, \mu = 0.1 kg/m/sec$
  - $\mathbf{v}_\infty = (10.0, 0.0, 0.0) m/sec$
- CSD Code: FEEIGEN
  - Quad Shells
  - 2 Modes (Heave, Torsion)
- Coupling:
  - Position: Linear, Glued
  - Loads: Projection of Stresses
- Machine: Dell P4
  - 3.2Ghz, 1Gbyte, Intel F, Linux OS
- Year: 2003
- Ref: AIAA-05-1093 (2005)

TACOMA BRIDGE SECTION (2)



Bridge Section: Dimensions and Typical Flowfield

TACOMA BRIDGE SECTION (3)

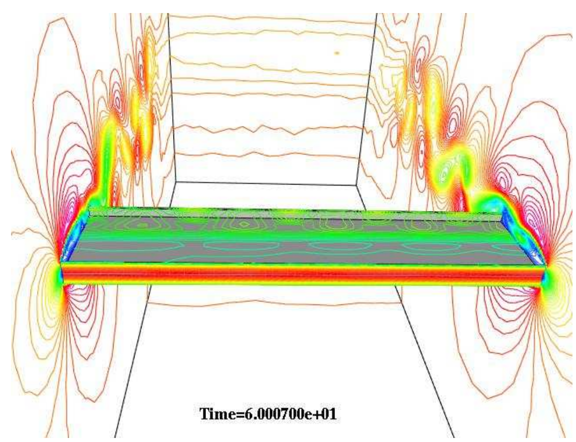


Bridge Section: Eigenforces and Eigenmodes

TACOMA BRIDGE (1)

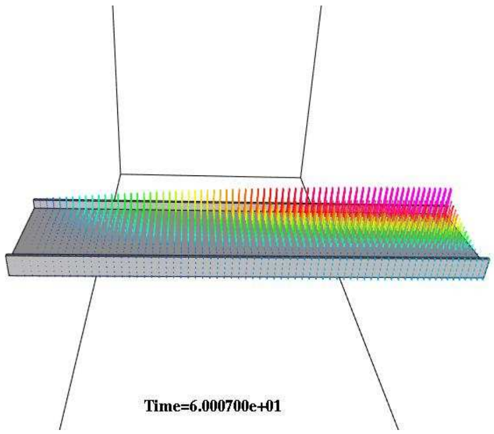
- CFD Code: FEFLO
  - Incompressible (8.5Mtet)
  - ALE, Mesh Movement
  - Laminar/MILES
  - $\rho = 1.25kg/m^3, \mu = 0.1kg/m/sec$
  - $\mathbf{v}_\infty = (10.0, 0.0, 0.0)m/sec$
- CSD Code: FEEIGEN
  - Quad Shells
  - 2 Modes (Heave, Torsion)
- Coupling:
  - Position: Linear, Glued
  - Loads: Projection of Stresses
- Machine: SGIO3K
  - 16 Procs
- Year: 2003
- Ref: AIAA-05-1093 (2005)

TACOMA BRIDGE (2)



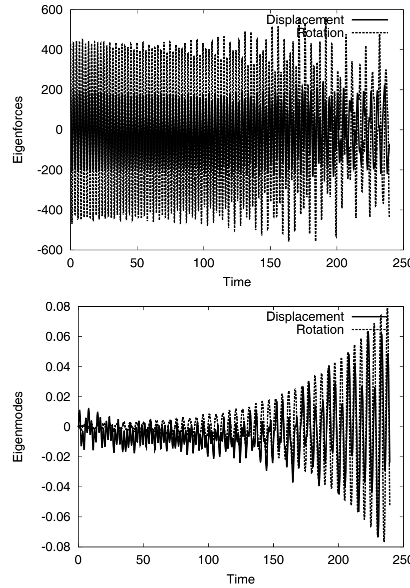
Bridge: Typical Flowfield

TACOMA BRIDGE (3)



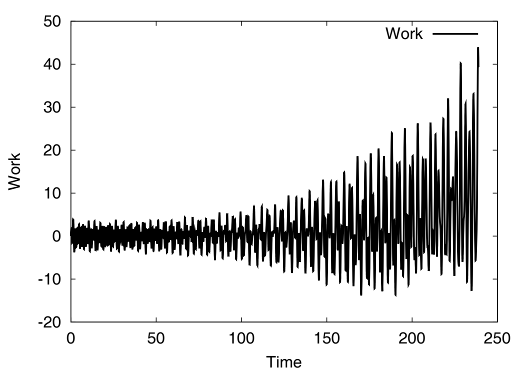
Bridge: Velocity of Structure

TACOMA BRIDGE (4)



Bridge: Eigenforces and Eigenmodes

TACOMA BRIDGE (5)



Bridge: Work Exchanged Between Fluid and Solid

SHOCK-CYLINDER INTERACTION

- Mild Steel (Elastic/ Ideal Plastic)
- Strong Shock
- CSD FEM Model: 2.3 Kqshells
- CFD FEM Model: 20 Kpts, 100 Ktets

4-ROOM EXPERIMENT

- Reinforced Concrete
- Explosion in Room 1
- CSD FEM Model 1
  - Average Quantities (No Re-Bars)
  - From Split Tetrahedra
  - 50 Kbricks
- CSD FEM Model 2
  - Detailed Quantities
  - Re-Bars
  - Regular Bricks (Hand-Made)
  - 50 Kbricks
- CFD FEM Model: 260 Kpts, 1.3 Mtets
- Ref:

TRUCK (1)

- Good Example For Geometrical Complexity
- AutoCad Data
- Simplifying Assumptions for CSD:
  - Engine, Transmission, Shafts: Rigid Solids
  - Springs and Tires: Elasto/Plastic/ Viscoplastic Solids
  - No C^3 Info
  - No Door Handles, Mirrors, etc.

TRUCK (2)

- Final CAD Model For CSD (1 Week):
  - 5,928 Points
  - 3,000 Lines
  - 1,386 Surfaces
- CAD Model for CFD: Automatic Step
  - Retain CSD Wetted Surfaces
  - Unwrap Doubly Defined Surfaces
- Final CAD Model For CFD (1/2 Day):
  - 6,306 Points
  - 3,718 Lines
  - 1,604 Surfaces

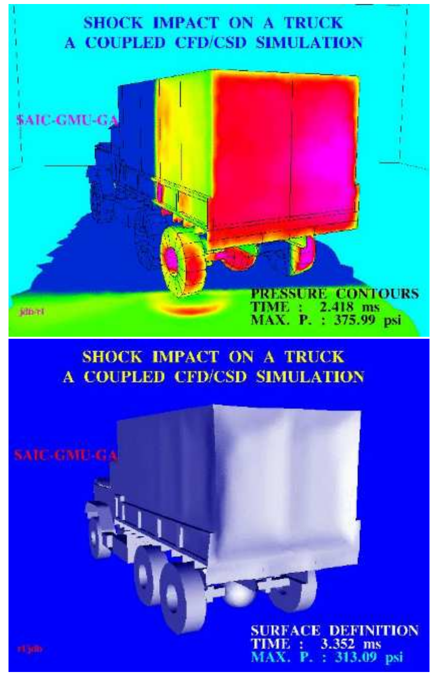
TRUCK (3)

- Same Mesh Generator for CSD and CFD
- CSD FEM Model:
  - 1 K beams
  - 50 Kqshells
  - 50 Kbricks
  - 22 Materials
- CFD FEM Model:
  - 200 Kpoints
  - 1 Mtets

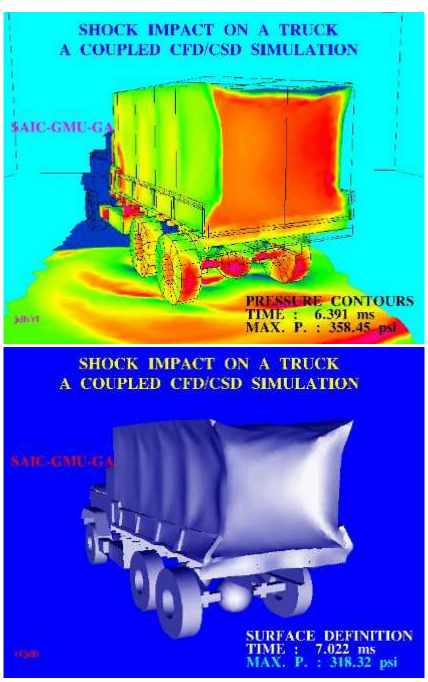
TRUCK (4)

- CFD Code: FEFLO
  - FEM-FCT
  - ALE/Remeshing
- CSD Code: DYNA3D
  - Beams, Quad Shells, Hex Solids
  - Elasto-Plastic
- Coupling:
  - Position: Linear, Glued
  - Loads: Conservative Projection of Pressures
- Computer: CRAY-C90 [What Else ?]
- Year: 1995
- Ref: AIAA-96-0795 (1996)

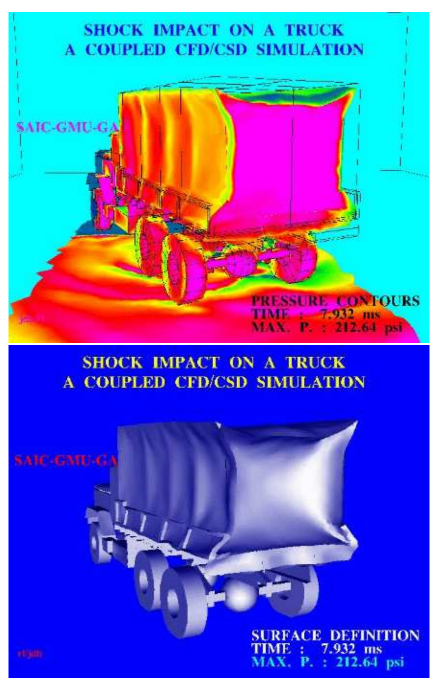
TRUCK (5)



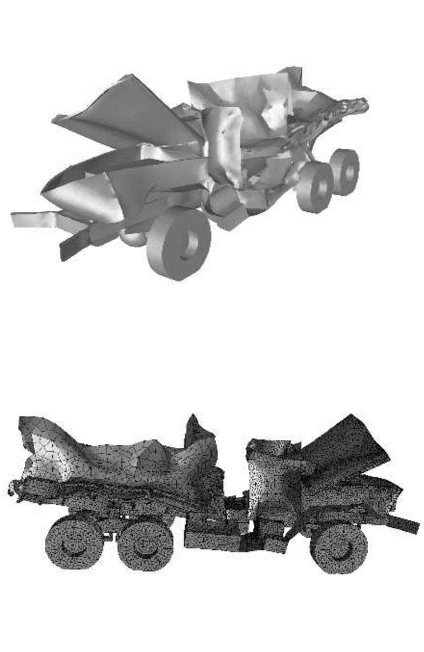
TRUCK (6)



TRUCK (7)



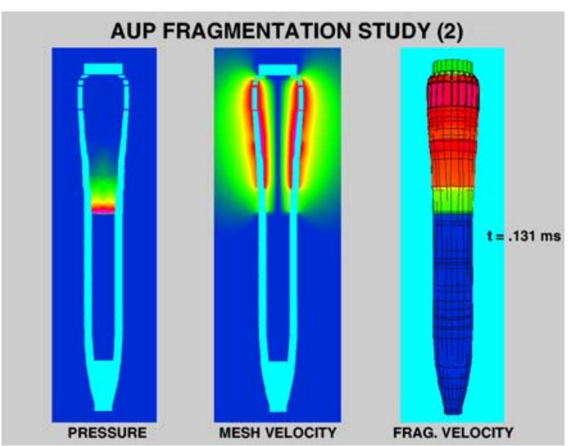
TRUCK (8)



WEAPON FRAGMENTATION (1)

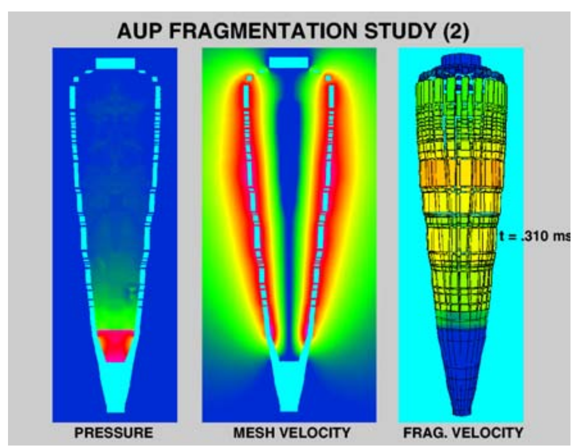
- CFD Code: FEFLO
  - FEM-FCT
  - JWL EOS for HE
  - H-Refinement and Remeshing
- CSD Code: GA-DYNA
  - Failure Criterion: Plastic Energy/Work
- Coupling:
  - Position: Linear, Glued
  - Loads: Conservative Projection of Pressures
- Machine: SGIO3K
  - 16-48 Procs
- Year: 1999
- Ref: IJNMF 31, 113-120 (1999)

WEAPON FRAGMENTATION (2)



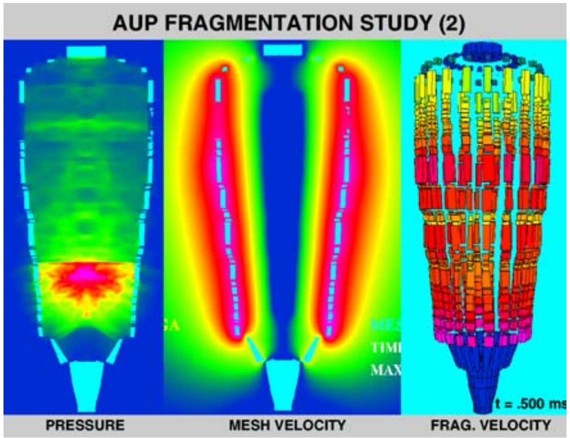
Fragmenting Weapon at 131msec

WEAPON FRAGMENTATION (3)



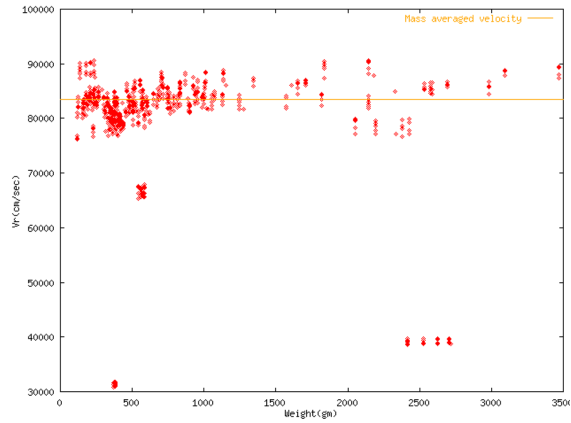
Fragmenting Weapon at 310msec

WEAPON FRAGMENTATION (4)



Fragmenting Weapon at 500msec

WEAPON FRAGMENTATION (5)



Radial Velocity as a Function of Fragment Weight

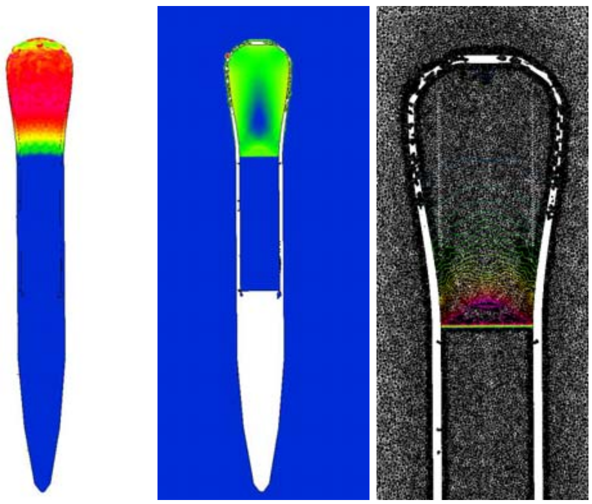
WEAPON FRAGMENTATION (1)

- CFD Code: FEFLO
  - FEM-FCT
  - JWL EOS for HE
  - H-Refinement, Embedding
- CSD Code: GA-DYNA
  - Failure Criterion: Average Element Strain > 60%
- Coupling:
  - Position: Linear, Embedded
  - Loads: Interpolation of Pressures
- Machine: SGIO3K
  - 16-48 Procs
- Year: 2003
- Ref: IJNME 60, 641-660 (2004)

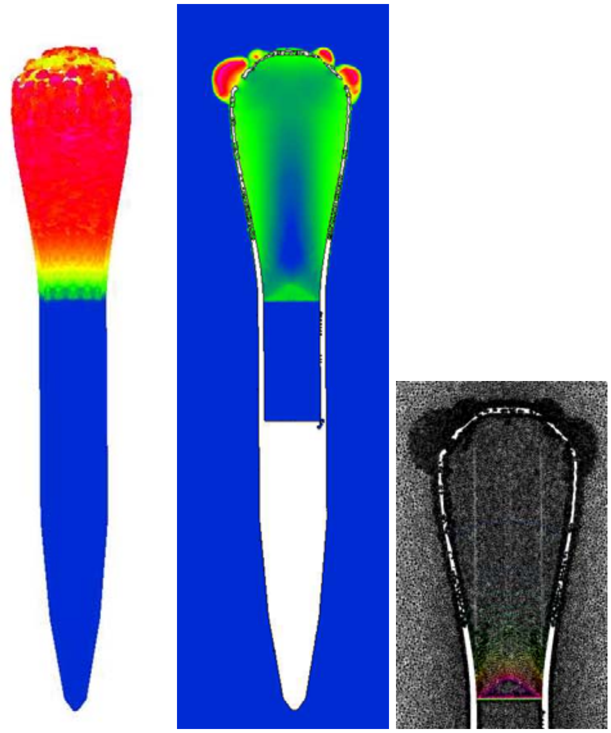
WEAPON FRAGMENTATION (2)

- CFD Mesh
  - Start: 39 Mtet
  - End: 72 Mtet
- CSD Mesh
  - 66 Khex
  - 1,555 Fragments

WEAPON FRAGMENTATION (3)



CSD/Flow Velocity and Pressure/Mesh at 68 ms

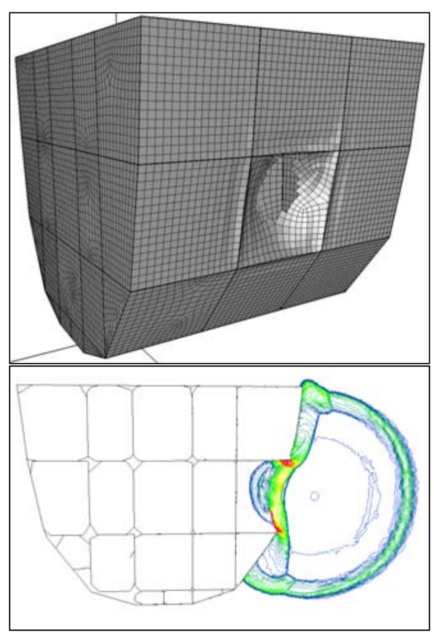


CSD/Flow Velocity and Pressure/Mesh at 102 ms

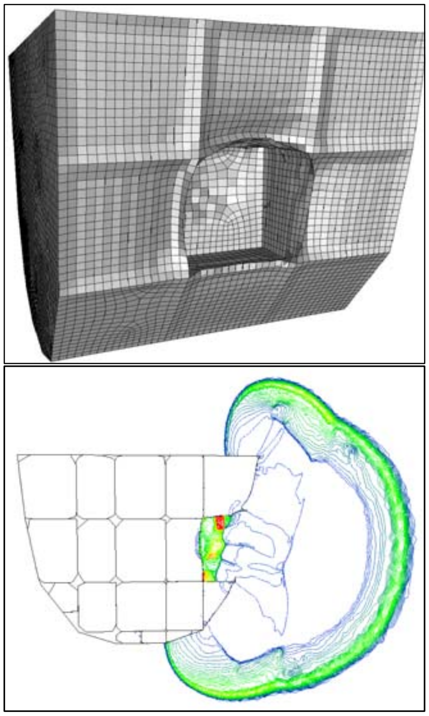
EXPLOSION CLOSE TO MODEL SHIP (1)

- CFD Code: FEFLO
  - FEM-FCT
  - JWL EOS for HE
  - H-Refinement, Embedding
- CSD Code: GA-DYNA
  - Quad Shells
  - Failure Criterion: Plastic Energy/Work
- Coupling:
  - Position: Linear, Embedded
  - Loads: Interpolation of Pressures
- Machine: SGIO3K
  - 16-48 Procs
- Year: 2001
- Ref: IJNME 60, 641-660 (2004)

EXPLOSION CLOSE TO MODEL SHIP (2)



Surface and Pressure in Cut Plane at 20msec

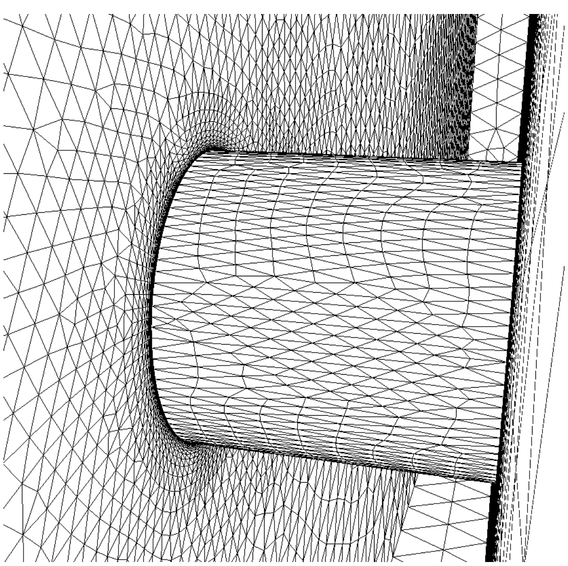


Surface and Pressure in Cut Plane at 50msec

HEATED CYLINDER (1)

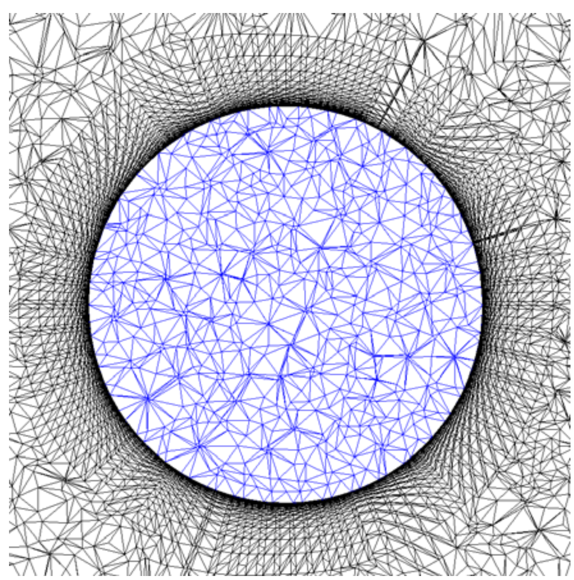
- CFD Code: FEFLO
  - Incompressible Navier-Stokes
  - $\rho = c_p = 1.0, \mu = k = 0.054$
  - $\mathbf{v} = (1.0, 0.0, 0.0)$
- CTD Code: FEHEAT
  - Tetrahedral Elements
  - $\rho = c_p = k = 1.0, s = 1.0$
- Coupling:
  - Temperature: Linear
  - Loads: Conservative Projection of Stresses/Fluxes
  - Run CFD/CTD Implicitly to Steady-State
- Machine: Dell P4
  - 3.2Ghz, 1Gbyte, Intel F, Linux OS
- Year: 2004
- Ref: AIAA-05-1093 (2005)

HEATED CYLINDER (2)



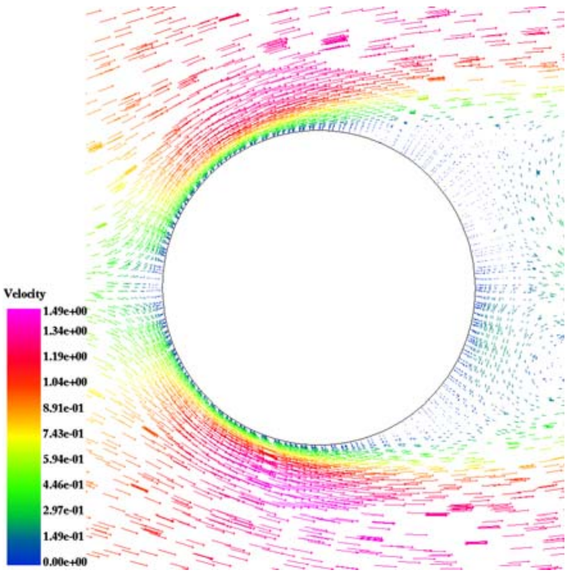
Heated Cylinder: Surface of Fluid Domain

HEATED CYLINDER (3)



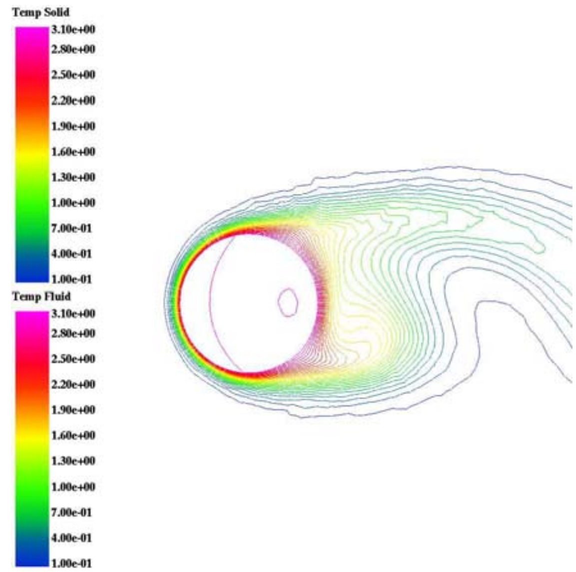
Heated Cylinder: Plane $z = 0.0$

HEATED CYLINDER (4)



Heated Cylinder: Velocity ($z = 0.0$)

HEATED CYLINDER (5)

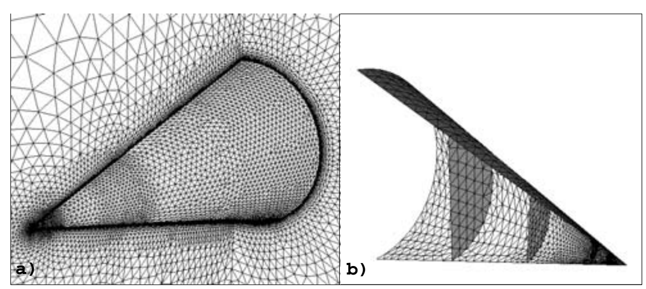


Heated Cylinder: Temperature ($z = 0.0$)

NOSE-CONE (1)

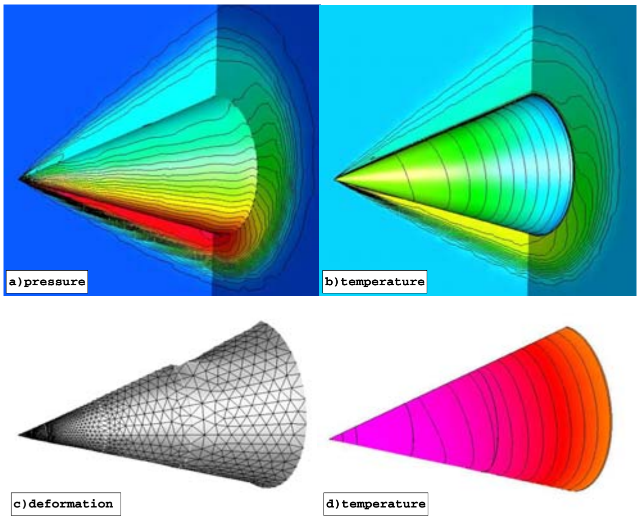
- CFD Code: FEFLO
  - Compressible RANS
  - Turbulence Model: Baldwin-Lomax
  - $M_\infty = 3.0$, $\alpha = 10^o$, $Re = 2 \cdot 10^6$
- CSD Code: COSMIC-NASTRAN
  - Quad Shells
  - Thermal Stresses
- Coupling:
  - Position: Linear, Glued
  - Loads: Conservative Projection of Stresses/Fluxes
  - Run CFD/CSD/CTD to Steady-State + Iterate
- Machine: SGIO2K
  - 1-4 Procs
- Year: 1996
- Ref: AIAA-98-2419 (1998)

NOSE-CONE (2)



Nose-Cone: Surface Grids for CFD and CSD/CTD

NOSE-CONE (3)

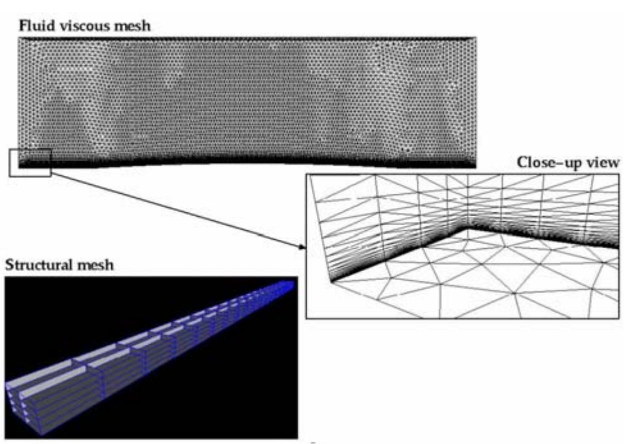


Nose-Cone: CFD/CSD/CTD Results Obtained

THERMAL PROTECTION PANEL (1)

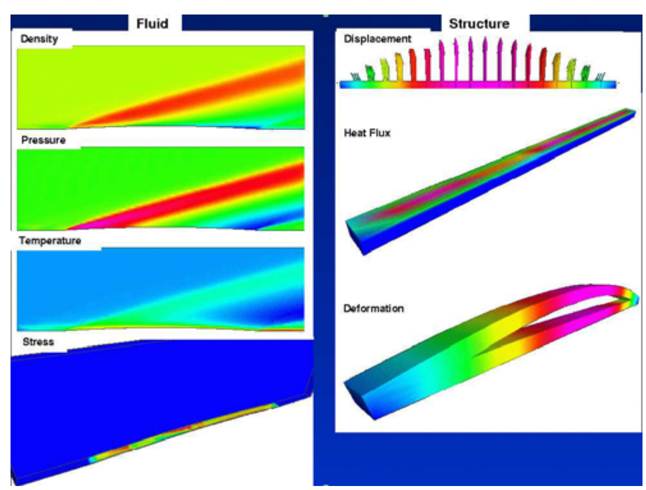
- CFD Code: FEFLO
  - Compressible RANS
  - Turbulence Model: Baldwin-Lomax
  - $M_\infty = 3.0$, $\alpha = 10^o$, $Re = 2 \cdot 10^6$
- CSD Code: COSMIC-NASTRAN
  - Hexahedral Solids
  - Thermal Stresses
- Coupling:
  - Position: Linear, Glued
  - Loads: Conservative Projection of Stresses/Fluxes
  - Run CFD/CSD/CTD to Steady-State + Iterate
- Machine: SGIO2K
  - 1-4 Procs
- Year: 1996
- Ref: AIAA-98-2419 (1998)

THERMAL PROTECTION PANEL (2)



Panel: Grids for CFD and CSD/CTD

THERMAL PROTECTION PANEL (3)



Panel: Results Obtained

PRE-PROCESSING: RESEARCH AREAS

- Multi-Format Input
- Virtual Reality Data Sets
- Surface Extraction from Voxel Data
- Fast Geometry Repair
- Very Large CAD Data Sets
- Automatic Abstraction

GRID GENERATION: RESEARCH AREAS

- Dirty Surface Meshing
- Good RANS Grids
- Wakes
- All Hex-Meshing
- Parallel Gridding

CFD: RESEARCH AREAS

- Turbulence Models
- Multigrid RANS
- High Order Schemes
- Link to High Knudsen-nrs.
- Dynamic Load Balancing
- Link to EM (Plasmas)
- Link to Surface Erosion/Etching

CSD: RESEARCH AREAS

- Tetrahedral Elements
- Triangular Plate Elements
- Multigrid/Fast Iterative Solvers
- Spalation/Breaking
- Link/Switch Continuum/Discrete
- Material Modeling
    - Concrete/Stone Failure
    - Pulverization
- Link to Atomic Scale

CTD: RESEARCH AREAS

- Multigrid/Fast Iterative Solvers
- Highly Nonlinear Materials
- Link to Atomic Scale

COUPLING: RESEARCH AREAS

- Conservation
- Treatment of Different Dimensional Abstractions