LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-PROC-647639

Stability of an Embedded Mesh Method for Coupling Lagrangian and ALE Finite Element Models

M. A. Puso, E. J. Kokko, B. T. Liu, B. Simpkins

December 12, 2013

AHPCRC Workshop on Evolving Discontinuities
Stanford, CA, United States
December 3, 2013 through December 5, 2013

Stability of an Embedded Mesh Method for Coupling Lagrangian and ALE Finite Element Models

AHPCRC Workshop on Evolving Discontinuities
Stanford University, Dec 3 2013

Mike Puso, Ed Kokko, Ben Liu, Bryan Simpkins
Methods Development Group, Computational Geosciences Group
Lawrence Livermore National Laboratory
Livermore CA

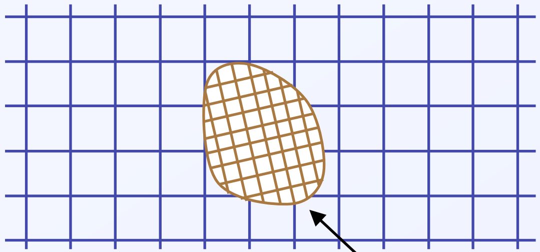
LLNL-PROC-647639

Work performed under the HPC Software Application Institute on Blast Protection for Platforms and Personnel under funding from the DoD HPC Modernization Program

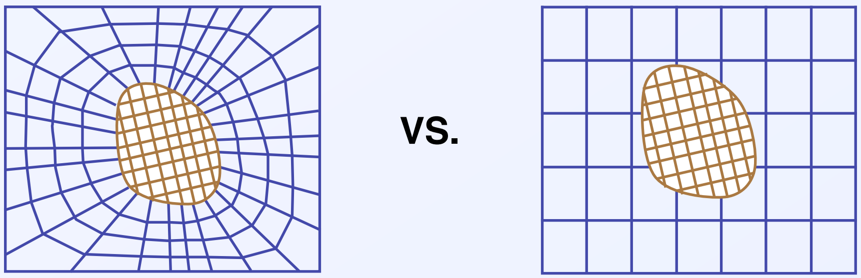
Outline

- What is embedded mesh?
- Previous works
- Two Step ALE approach based on time splitting
  - Lagrange Step
    - Mesh Coupling: *Focus on following stability issues*
      - Stability of the multiplier space
      - Condition number
      - Stable time step
  - Advection Remap Step
- Results
  - Verifications, blast, failure etc.
- Summary

Embedded Mesh technique couples overlapping meshes

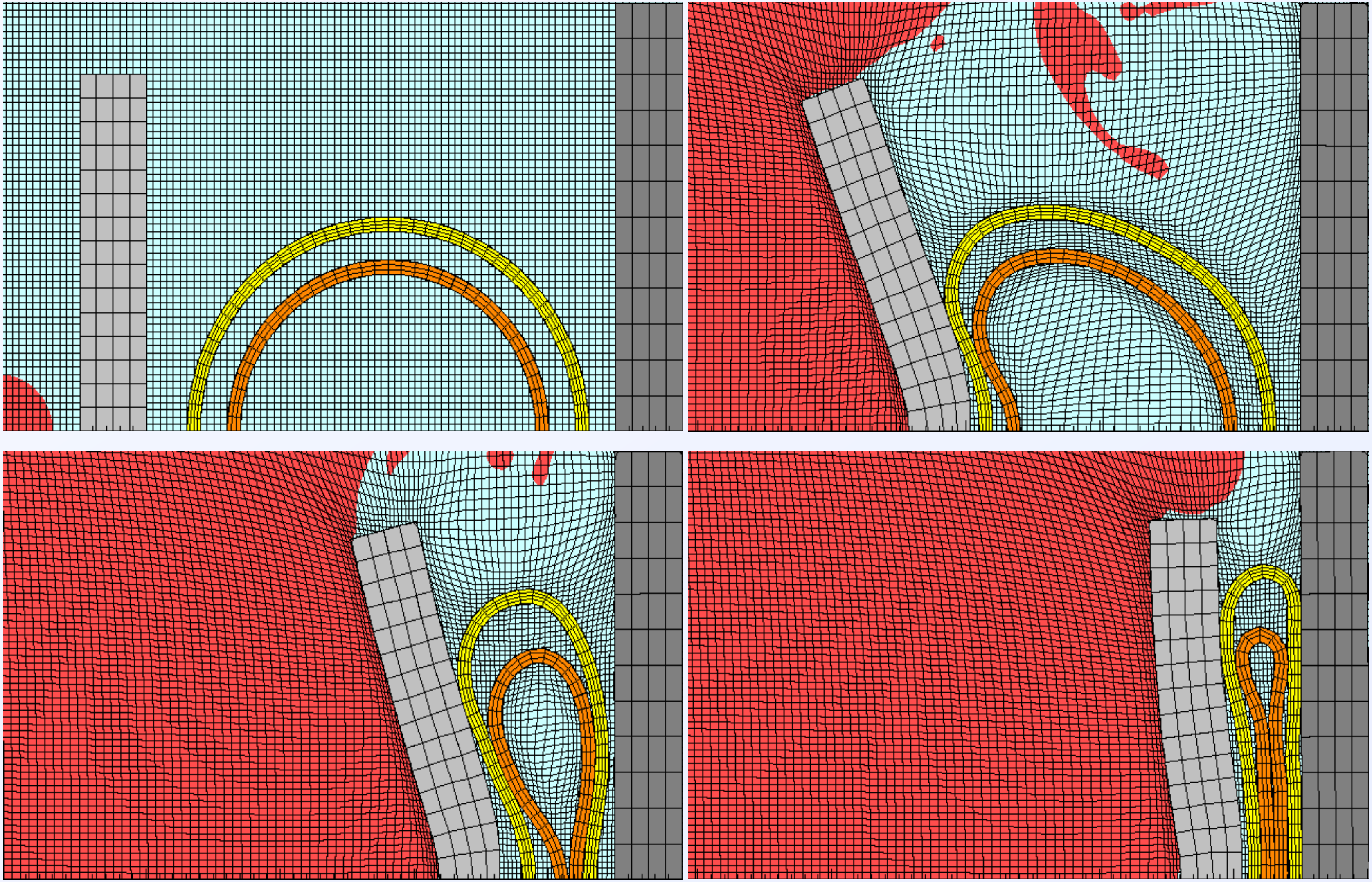


- Simple: Avoids construction of *body fitted mesh*



- Robust: Minimizes mesh tangling

For Example



Many Previous Works: to name a few

- Existing *Embedded Mesh* methods
  - *CEL method* (W.F. Noh, 1964)
  - *Immersed boundary methods* (C.S. Peskin 1977, 2002)
  - *Immersed finite element methods* (W.K. Liu 2004)
  - *Overset grid methods* (W.D. Henshaw 2006)
  - *Zapotec material insertion method* (Bessette 2002)
    - Sandia code couples CTH and Pronto
  - *LS-Dyna, ABAQUS* (commercial codes)
  - *Mortar fictitious domain methods* (Baaijens 2001)
  - *Nitsche's Method* (Hansbo and Hansbo, 2003; Sanders and Puso 2010)
  - *Ghost Fluid methods* (Fedikew et. al. 1999)
    - *DYSMAS* Gemini-PARADYN (Luton et. al. 2003)

Issues Regarding Different Methods

- Many developed for Eulerian-Finite Volume/Difference method
- Overset methods require auxiliary mesh
- Many require penalties
- Some don’t work for shells
- Some not “stable” and/or “consistent”
- Considered too inaccurate for many applications

New Approaches and Goals

More recent approaches

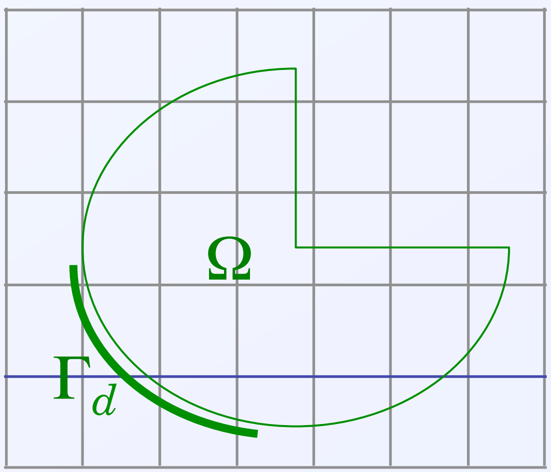
A discontinuous-Galerkin based immersed boundary method (Lew and Buscagli, 2008)

A stable Lagrange multiplier space for stiff interface conditions … (Bechet, Moes, Wohlmuth, 2009)

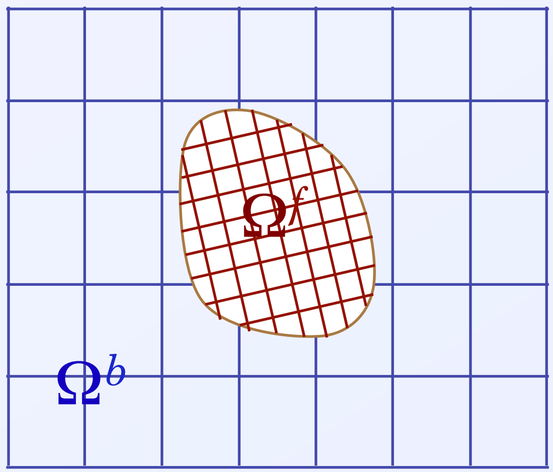
Embedded Dirichlet Method… (Gerstenberg and Wall, 2010) (Baiges et al. 2012)

Embedded Mesh… DG method (Sanders and Puso, 2012)

Fictitious domain finite element methods using cut elements… (Burman and Hansbo, 2010)



Embedded Dirichelet



Embedded Mesh

New Approaches and Goals

More recent approaches

A discontinuous-Galerkin based immersed boundary method (Lew and Buscagli, 2008)
A stable Lagrange multiplier space for stiff interface conditions … (Bechet, Moes, Wohlmuth, 2009)
Embedded Dirichlet Method… (Gerstenberg and Wall, 2010) (Baiges et al. 2012)
Embedded Mesh… DG method (Sanders and Puso, 2012)
Fictitious domain finite element methods using cut elements… (Burman and Hansbo, 2010)

Algorithmic Design Constraints

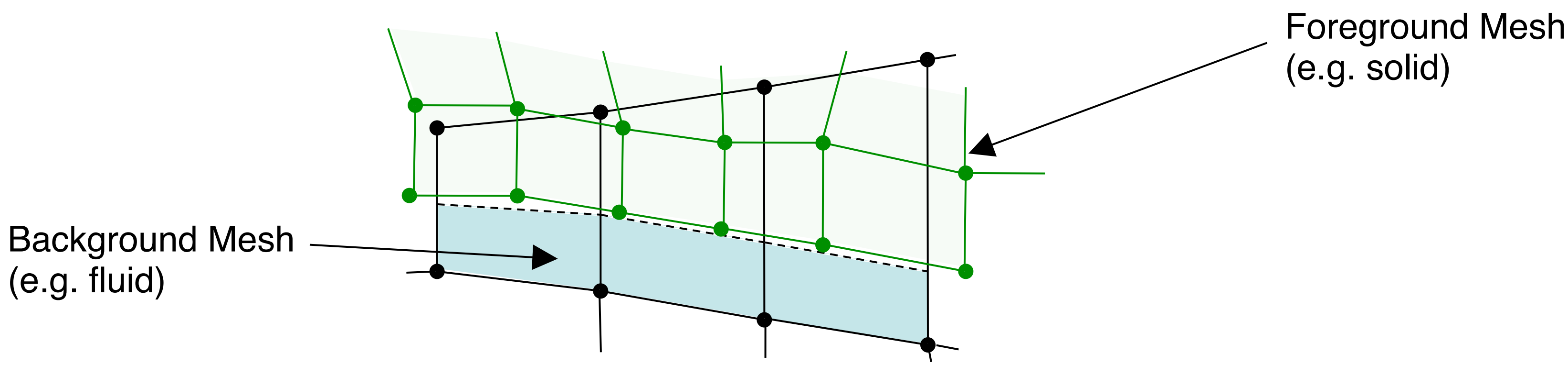
- Suitable for explicit finite element code i.e. good estimates for stable time step
- Suitable for standard finite elements i.e. use linear hexahedral, shell type elements
- Symmetric formulations
- Avoid penalties

Mesh Coupling Approach

Two basic Lagrange multiplier approaches

- Constraints (multipliers) on foreground mesh (known issues)
- Constraints (multipliers) on background mesh (not as easy)

$$\int_{\Gamma} \boldsymbol{\lambda} \cdot (\boldsymbol{v}^b - \boldsymbol{v}^f)\, d\Gamma = 0$$

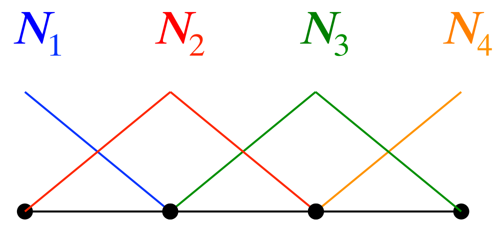


Why use Lagrange multipliers?

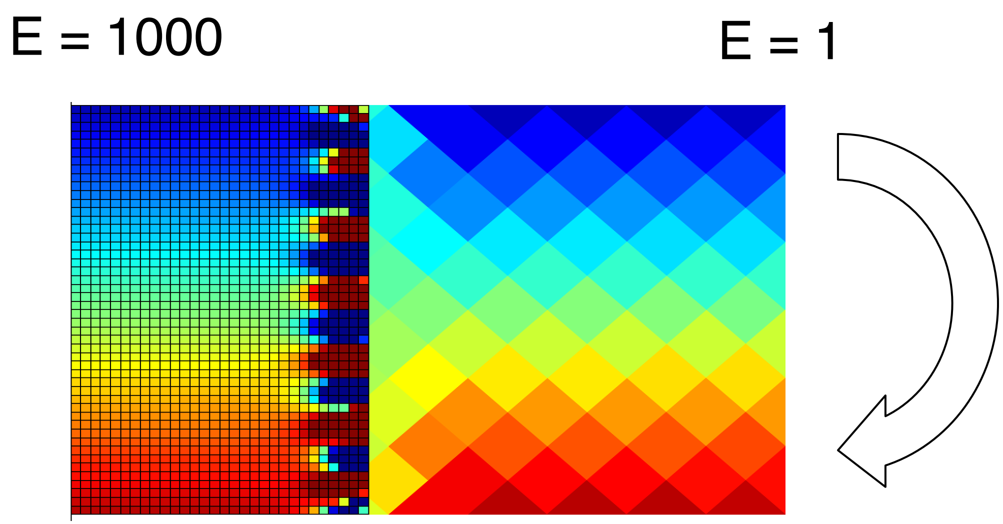
- Do a good job enforcing constraint (better than penalty)
- Easier to do get symmetric forms (e.g. with Niche)
- Natural way to incorporate “added mass” as opposed to staggered approach
- Turns out not to drastically affect performance

Constraints defined on foreground mesh

- Use standard Lagrange multipliers



$$\boldsymbol{\lambda} = \sum_{A=1}^{N} N_A(\boldsymbol{\xi}) \boldsymbol{\lambda}_A \qquad \int_\Gamma \boldsymbol{\lambda} \cdot (\boldsymbol{v}^b - \boldsymbol{v}^f) \, d\Gamma = 0$$



Other foreground constraint approaches

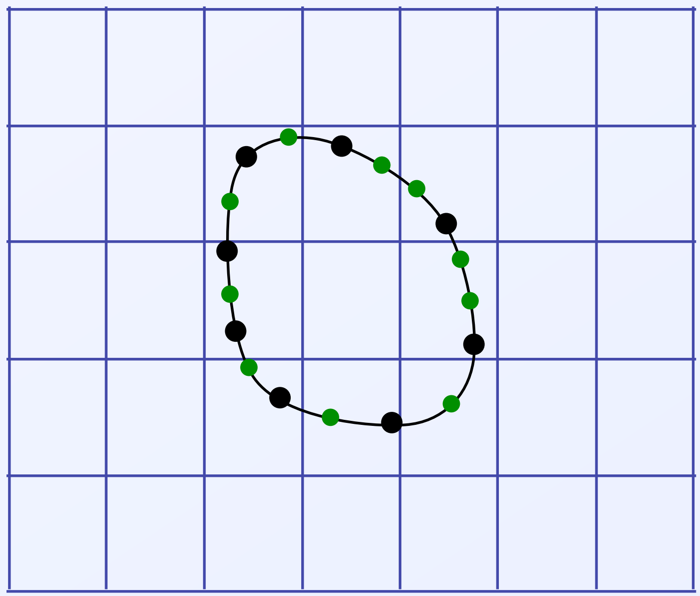
- Many commercial codes apply constraints on foreground mesh
  - Penalized constraints on tracer particles
  - Based on collocation, not integrals

● Shell Node

● Tracer Particle

$$F^b = \sum_{n}^{nodes+tracers} k\, w_n^b (v_n^f - v_n^b)$$

$$F^f = \sum_{n}^{nodes+tracers} k\, w_n^f (v_n^b - v_n^f)$$



- “Leak” if k is too small
- “Lock” if k is too big

Constraints on background mesh

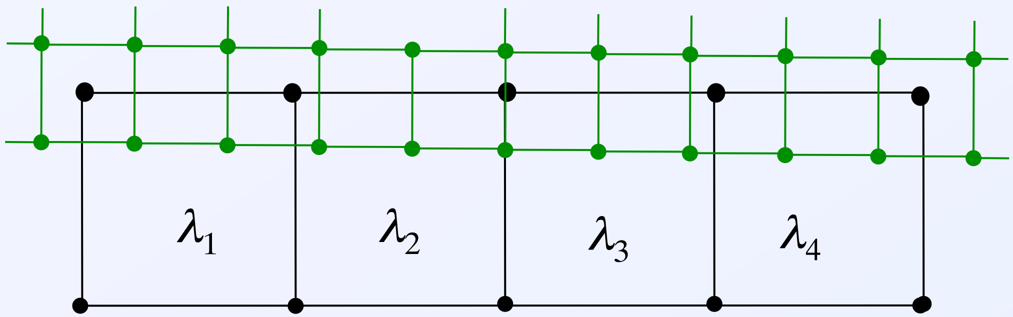
- Edge based constraints

(Bechet et al. 2009, Puso et al. 2012)

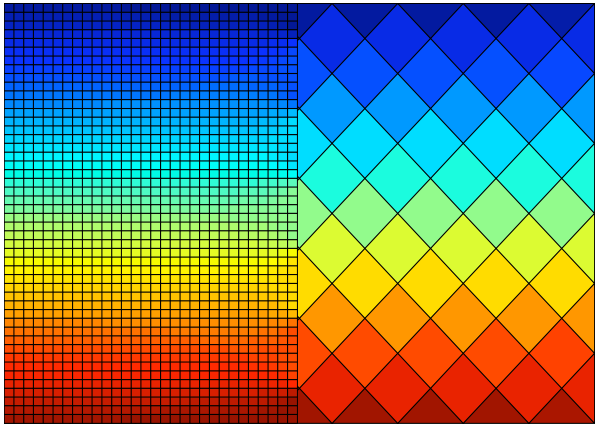
- Piecewise constants

(Embedded Dirichlet, Burman Hansbo 2011; Embedded Mesh, Puso 2012 et al.)

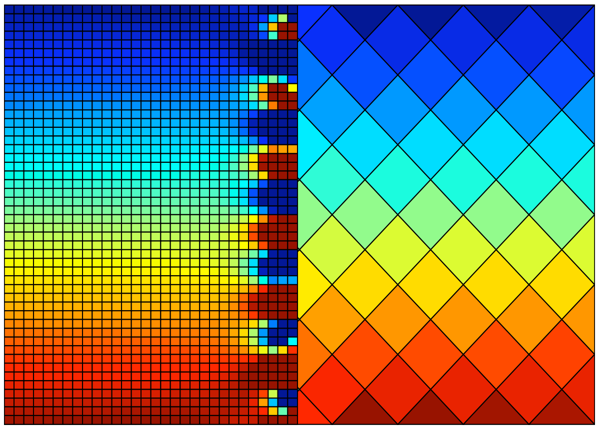
  – Simple
  – Requires stabilization



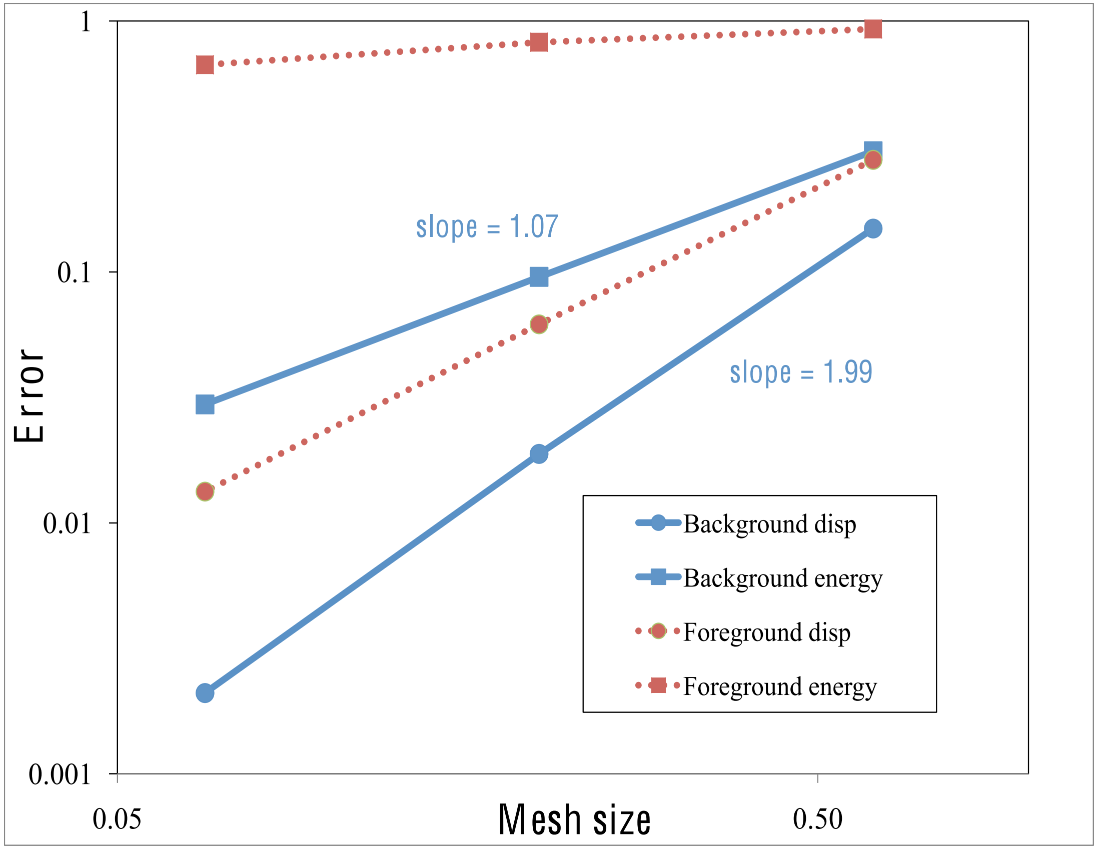
Multipliers on background mesh: Beam result



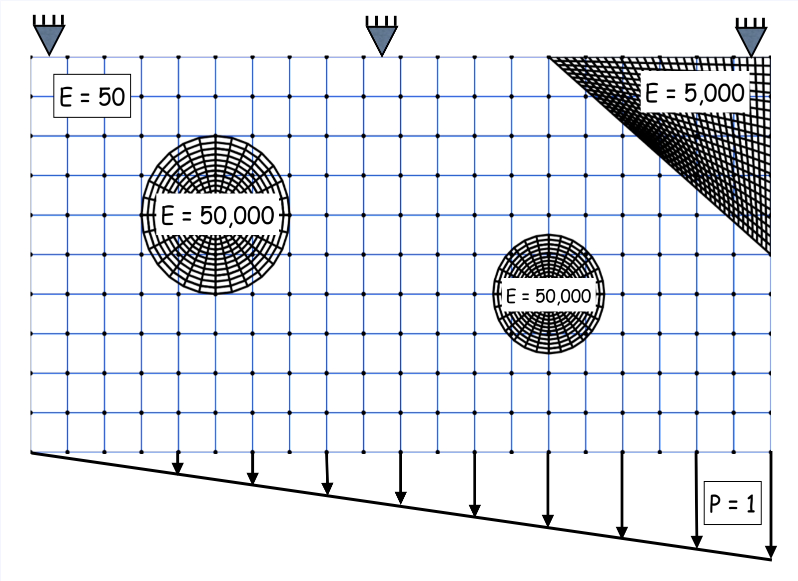
background

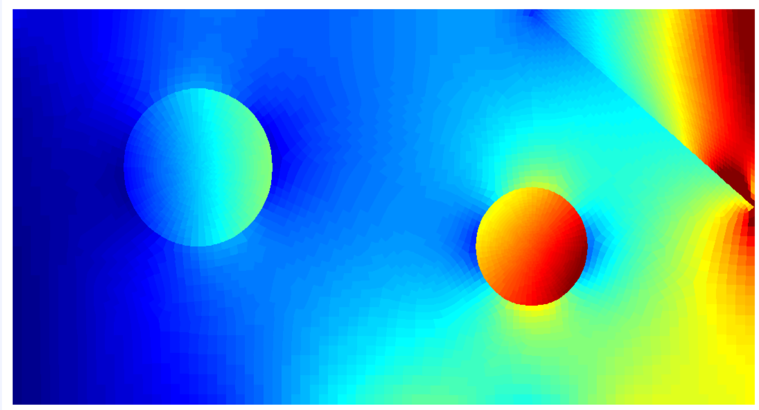


foreground

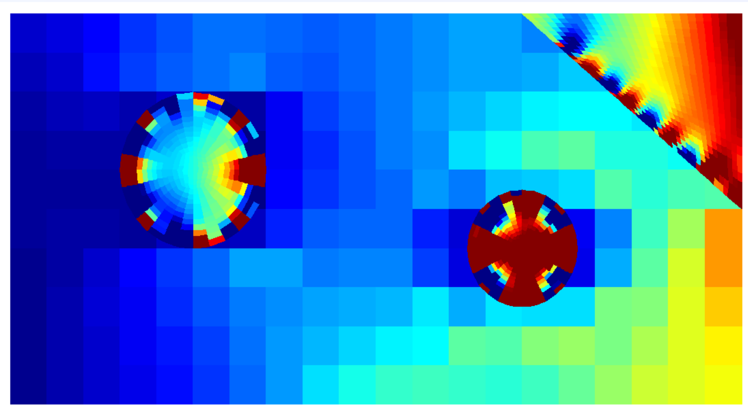


Multipliers on background mesh: 2D result

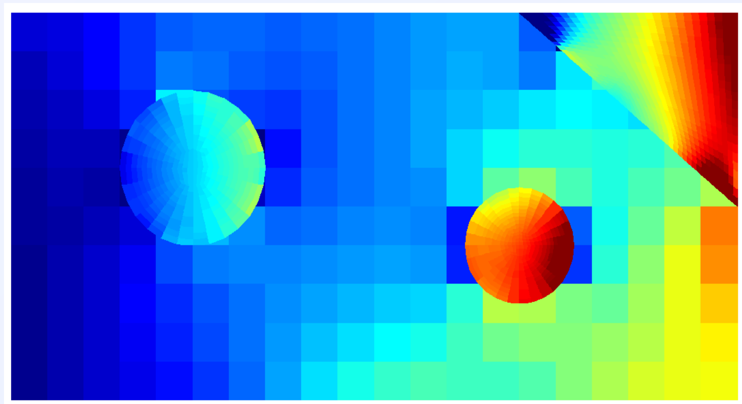




conforming

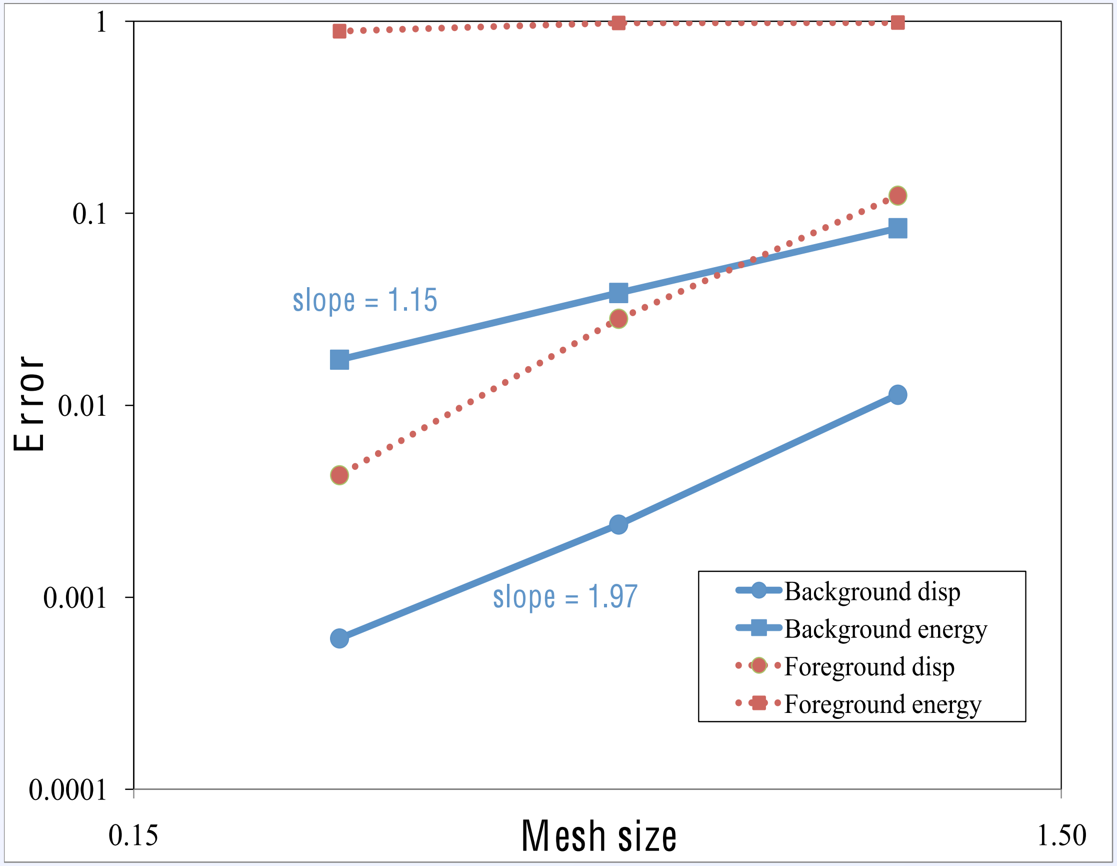


foreground



background

Multipliers on background mesh: 2D result

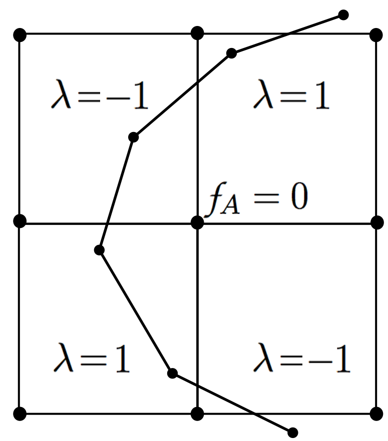


Mathematical details of approach

- Stability of multiplier space
  - Determine conditions for LBB stability
- Solution to equations of motion
  - Show that condition number is independent of mesh size
- Stability of time integrator
  - Estimate time stable time step

Stability of multiplier space

- Consider abstract form of BVP: $$a(u_h, v_h) + b(\lambda_h, u_h) = \langle f, v \rangle$$ $$b(\mu_h, u_h) - j(\mu_h, \lambda_h) = 0$$
- e.g elasticity $$a(u_h, v_h) = \int_{\Omega^b/\Omega^f} \nabla v_h^b : \mathcal{C}^b \nabla u_h^b \, d\Omega + \int_{\Omega^f} \nabla v_h^f : \mathcal{C}^f \nabla u_h^f \, d\Omega$$ $$b(\lambda_h, u_h^b - u_h^f) = \int_\Gamma \lambda \cdot (u_h^b - u_h^f) \, d\Gamma$$
- Checkerboard mode: constraint force at node $f_A = 0$, $\lambda_K \neq 0$ $a.e.$ $$f_A = \sum_{K \in \mathcal{S}_A^c} B_{KA} \lambda_K \qquad B_{KA} = \int_{\Gamma_K} \phi_A \, d\Gamma$$
- Add stabilization term j: determine conditions for stability $$\mathscr{B}[(u_h, \lambda_h), (v_h, \mu_h)] = a(u_h, v_h) + b(\lambda_h, v_h) + b(\mu_h, u_h) - j(\lambda_h, \mu_h)$$ $$\sup_{\{v_h, \mu_h\} \in \mathcal{W}_h} \frac{\mathscr{B}[(u_h, \lambda_h), (v_h, \mu_h)]}{\|v_h\|_{1,\Omega} + \|\mu\|_{-1/2,h,\Gamma}} \geq \|u_h\|_{1,\Omega} + \|\lambda\|_{-1/2,h,\Gamma}$$

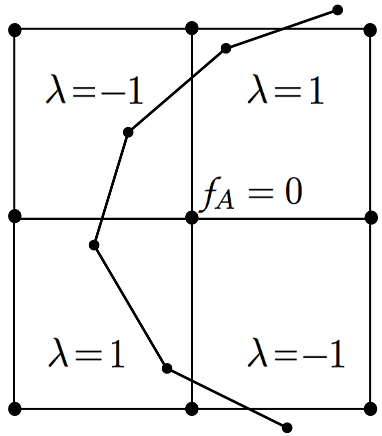


- Choosing appropriate pair (v_h, μ_h) satisfies inequality $$v_h^b = u_h^b + \alpha f_h \quad \text{where} \quad f_h = \sum_{A \in \mathcal{N}^c} \phi_A f_A \qquad v_h^f = u_h^f \quad \mu_h = -\lambda_h$$

Stability of multiplier space

- Stability condition $$\int_\Gamma \lambda_h f_h \, d\Gamma + j(\lambda_h, \lambda_h) \geq c_{min} \|\lambda_h\|^2_{-1/2,h}$$

- Stabilization form $$j(\lambda_h, \lambda_h) = \sum_{f \in \mathcal{F}^c} \gamma_f h^2 (\lambda_{K_{f1}} - \lambda_{K_{f2}})$$



- Recall from before

$$f_h = \sum_{A \in \mathcal{N}^c} \phi_A f_A \qquad f_A = \sum_{K \in \mathcal{S}^c_A} B_{KA} \lambda_K \qquad B_{KA} = \int_{\Gamma_K} \phi_A \, d\Gamma$$

- Mesh dependent L_2 norm $$\|\lambda\|^2_{-1/2,h} = h \int_\Gamma \lambda_h^2 \, d\Gamma$$

- Compute ratio, with substitution

$$\frac{\int_\Gamma \lambda_h f_h \, d\Gamma + j(\lambda_h, \lambda_h)}{\|\lambda_h\|^2_{-1/2,h}} = \frac{\sum_{A \in \mathcal{N}^c} \left[\frac{1}{4} \left(\sum_{K \in \mathcal{S}^c_A} B_{KA} \lambda_K \right)^2 + \sum_{f \in \mathcal{F}^c_A} \gamma_f h^2 (\lambda_{K_{f1}} - \lambda_{K_{f2}})^2 \right]}{\frac{h}{4} \sum_{A \in \mathcal{N}^c} \sum_{K \in \mathcal{S}^c_A} \lambda_K^2 \, A_K} \geq c_{min}$$

Solution to EOM: matrix decomposition

- Central difference equations of motion

$$M^b(v^b_{n+1/2} - v^b_{n-1/2})/\Delta t + B^{b\,T}\lambda = f^b_{n-1}$$
$$M^f(v^f_{n+1/2} - v^f_{n-1/2})/\Delta t + B^{f\,T}\lambda = f^f_{n-1}$$
$$B^b v^b_{n+1/2} + B^f v^f_{n+1/2} - J/\Delta t\lambda = 0$$

- Exploiting diagonal mass, solve for $v^b_{n+1/2}$ and $v^f_{n+1/2}$ in terms of λ e.g.

$$v^b_{n+1/2} = -\Delta t {M^b}^{-1} B^{b\,T}\lambda + {M^b}^{-1}\tilde{f}^b$$

- Substituting into bottom row yields following decomposition

$$H\lambda = f$$
$$H = \Delta t^2\left(B^b {M^b}^{-1} B^{b\,T} + B^f {M^f}^{-1} B^{f\,T}\right) + J$$
$$f = \Delta t B^b {M^b}^{-1}\tilde{f}^b + \Delta t B^f {M^f}^{-1}\tilde{f}^f$$

- Use CG solver to find λ

Solution to EOM: matrix decomposition

- Central difference equations of motion

$$\begin{bmatrix} M^b & 0 & B^{b\,T} \\ 0 & M^f & B^{f\,T} \\ B^b & B^f & -J/\Delta t^2 \end{bmatrix} \begin{Bmatrix} v^b_{n+1/2} \\ v^f_{n+1/2} \\ \lambda \Delta t \end{Bmatrix} = \begin{Bmatrix} f^b \Delta t + M^b v^b_{n-1/2} \\ f^f \Delta t + M^f v^f_{n-1/2} \\ 0 \end{Bmatrix} = \begin{Bmatrix} \tilde{f}^b \\ \tilde{f}^f \\ 0 \end{Bmatrix}$$

- Exploiting diagonal mass, solve for $v^b_{n+1/2}$ and $v^f_{n+1/2}$ in terms of λ e.g.

$$v^b_{n+1/2} = -\Delta t {M^b}^{-1} B^{b\,T} \lambda + {M^b}^{-1} \tilde{f}^b$$

- Substituting into bottom row yields following decomposition

$$H\lambda = f$$

$$H = \Delta t^2 \left(B^b {M^b}^{-1} B^{b\,T} + B^f {M^f}^{-1} B^{f\,T} \right) + J$$

$$f = \Delta t B^b {M^b}^{-1} \tilde{f}^b + \Delta t B^f {M^f}^{-1} \tilde{f}^f$$

- Use CG solver to find λ

Solution to EOM: condition number

- Condition number determines rate of convergence. Recall that

$$H = \Delta t^2 \left(B^b {M^b}^{-1} {B^b}^T + B^f {M^f}^{-1} {B^f}^T \right) + J$$

- Stable time step is based on mesh size h

$$\Delta t^2 \|M^{-1}\| \propto \left(\frac{h}{c}\right)^2 \left(\frac{1}{\rho h^3}\right) \propto \frac{1}{h} \quad \Rightarrow \quad \frac{c^\rho_{min}}{h} \leq \Delta t^2 \|M^{-1}\| \leq \frac{c^\rho_{max}}{h}$$

$$\begin{aligned} \lambda^{eig}_{min}(H) &= \min_{\lambda^T \lambda = 1} \lambda^T H \lambda \\ &\geq \min_{\lambda^T \lambda = 1} \frac{c^\rho_{min}}{h} \lambda^T B^b B^{bT} \lambda + \lambda^T J \lambda \\ &\geq c_{min} \|\lambda_h\|^2_{-1/2,h} \end{aligned}$$

$$\begin{aligned} \lambda^{eig}_{max}(H) &= \max_{\lambda^T \lambda = 1} \lambda^T H \lambda \\ &\leq \max_{\lambda^T \lambda = 1} \frac{c^\rho_{max}}{h} \lambda^T B^b B^{bT} \lambda + \lambda^T J \lambda \\ &\leq c_{max} \|\lambda_h\|^2_{-1/2,h} \end{aligned}$$

- The condition number does not change with mesh refinement

$$s_{con} = \frac{\lambda^{eig}_{max}}{\lambda^{eig}_{min}} = \frac{c_{max}}{c_{min}}$$

Stable Time Step (*WIP*)

- Compute spectral decomposition of of J

$$J = [\Phi_{m_d}^T, \Phi_{\text{ker}}^T] \begin{bmatrix} \Psi & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \Phi_{m_d} \\ \Phi_{\text{ker}} \end{Bmatrix}$$

$$\text{where} \quad \Psi = \text{diag}(\psi_1, \dots \psi_{m_d}) \quad \text{and} \quad \Phi_{m_d} \in \mathbb{R}^{m_d \times n},\ \Phi_{\text{ker}} \in \mathbb{R}^{(n-m_d)\times n}$$

- Constraints equations are now rewritten

$$\mu^T(B^b v^b_{n+1/2} + B^f v^f_{n+1/2} - J/\Delta t \lambda)$$
$$= [\mu^T_{m_d}, \mu^T_{\text{ker}}] \left(\begin{bmatrix} \Phi_{m_d} \\ \Phi_{\text{ker}} \end{bmatrix} (B^b v^b_{n+1/2} + B^f v^f_{n+1/2}) - \frac{1}{\Delta t} \begin{bmatrix} \Psi & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \lambda_{m_d} \\ \lambda_{\text{ker}} \end{Bmatrix} \right) = 0$$

- For sake of analysis, add penalty term to un-damped part $\varepsilon \to 0$

$$\Phi_{m_d}(B^b v^b_{n+1/2} + B^f v^f_{n+1/2})\Delta t - \Psi \lambda_{m_d} = 0$$
$$\Phi_{\text{ker}}(B^b v^b_{n+1/2} + B^f v^f_{n+1/2})\Delta t - \varepsilon \lambda_{\text{ker}} = 0$$

Stable Time Step

- Solving for λ_{m_d} and λ_{ker} in terms of velocity leads to damped EOM

$$M\left\{\begin{array}{c} v^b_{n+1/2} \\ v^f_{n+1/2} \end{array}\right\} + C\left\{\begin{array}{c} v^b_{n+1/2} \\ v^f_{n+1/2} \end{array}\right\} = \left\{\begin{array}{c} \tilde{f}^b \\ \tilde{f}^f \end{array}\right\}$$

where the damping matrix C is defined

$$C = \left[\begin{array}{c} B^{b\,T} \\ B^{f\,T} \end{array}\right]\left[\Phi^T_{m_d}\Psi^{-1}\Phi_{m_d} + \frac{1}{\varepsilon}\Phi^T_{\text{ker}}\Phi_{\text{ker}}\right][B^b, B^f]\Delta t$$

Stable Time Step

- Develop amplitude matrix from the set of equations

$$d_n = d_{n-1} + v_{n-1/2}\Delta t$$
$$(M + C)v_{n+1/2} = -Kd_n\Delta t + Mv_{n-1/2}$$

and rewriting

$$\left\{ \begin{array}{c} d_n \\ v_{n+1/2} \end{array} \right\} = \left[\begin{array}{cc} I & 0 \\ 0 & D \end{array} \right] \left[\begin{array}{cc} I & I\Delta t \\ -M^{-1}K\Delta t & -M^{-1}K\Delta t^2 + I \end{array} \right] \left\{ \begin{array}{c} d_{n-1} \\ v_{n-1/2} \end{array} \right\} = I^d A^u \left\{ \begin{array}{c} d_n \\ v_{n-1/2} \end{array} \right\}$$

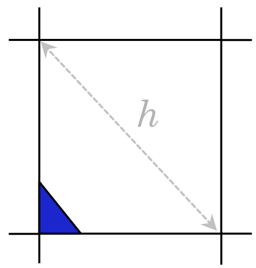
where $D = (I + M^{-1}C)^{-1} \quad \lambda^{eig}_{max}(D) \leq 1 \quad \Rightarrow \quad \lambda^{eig}_{max}(I^d) \leq 1$

$$\lambda^{eig}_{max}(A^u) \leq 1 \quad \text{since} \quad \Delta t < \frac{2}{\omega_{max}}$$

Stable Time Step: Maximum Frequency Estimation

- Stable time step based on Max frequency computed from Rayleigh quotient

$$\omega_{max}^2 \leq \sup_{u^b, u^f} \frac{{u^b}^T K^b u^b + {u^f}^T K^f u^f}{{u^b}^T M^b u^b + {u^f}^T M^f u^f} \leq \frac{k_{e_c}^{bulk} \sum_A^8 (B_{e_c,A} \cdot u_A^{e_c})^2 V_{e_c}}{(1/8)\rho_{e_c} V_{e_c} \sum_B^8 u_B^{e_c} \cdot u_B^{e_c}} \approx \left(\frac{c}{h}\right)_{e_c}^2$$

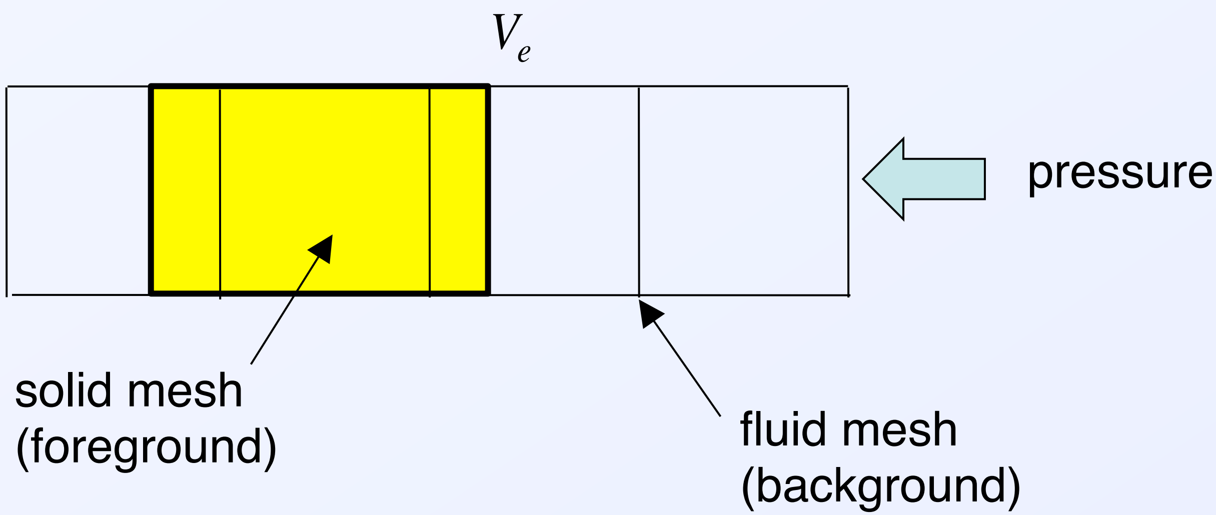


- “Cut elements” have no negative effect on time step since mass lumping distributes equal mass to all nodes

ALE implementation: with foreground Lagrange Mesh

- Use central difference explicit 2 step ALE approach

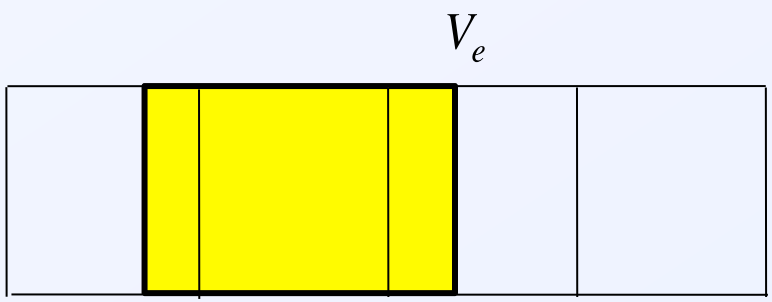
Load step
Lagrange step
Advection remap step
Advance time step $t_n \rightarrow t_{n+1}$



ALE implementation: with foreground Lagrange Mesh

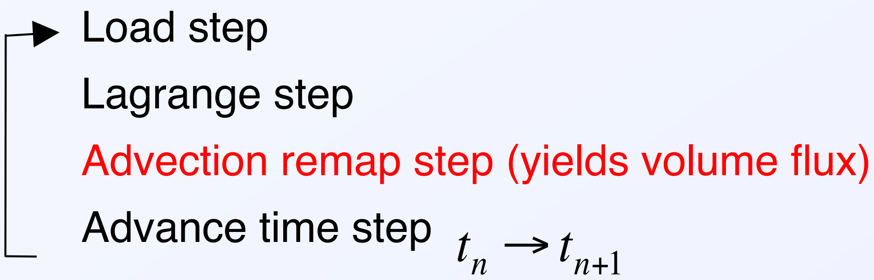
- Use central difference explicit 2 step ALE approach

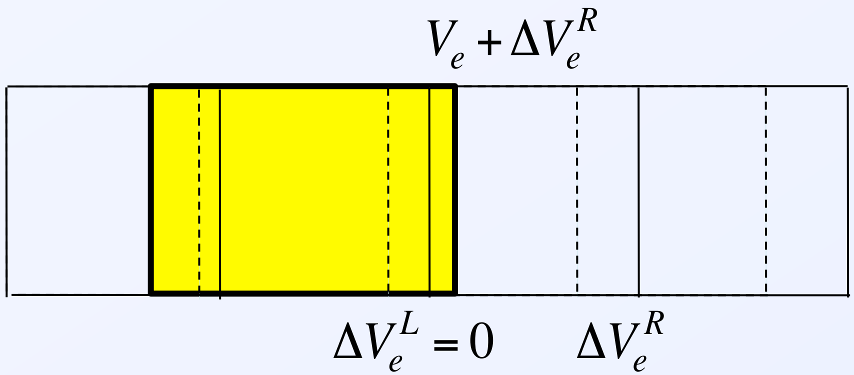
  - Load step
  - Lagrange step (velocity constraints applied)
  - Advection remap step
  - Advance time step $t_n \rightarrow t_{n+1}$



ALE implementation: with foreground Lagrange Mesh

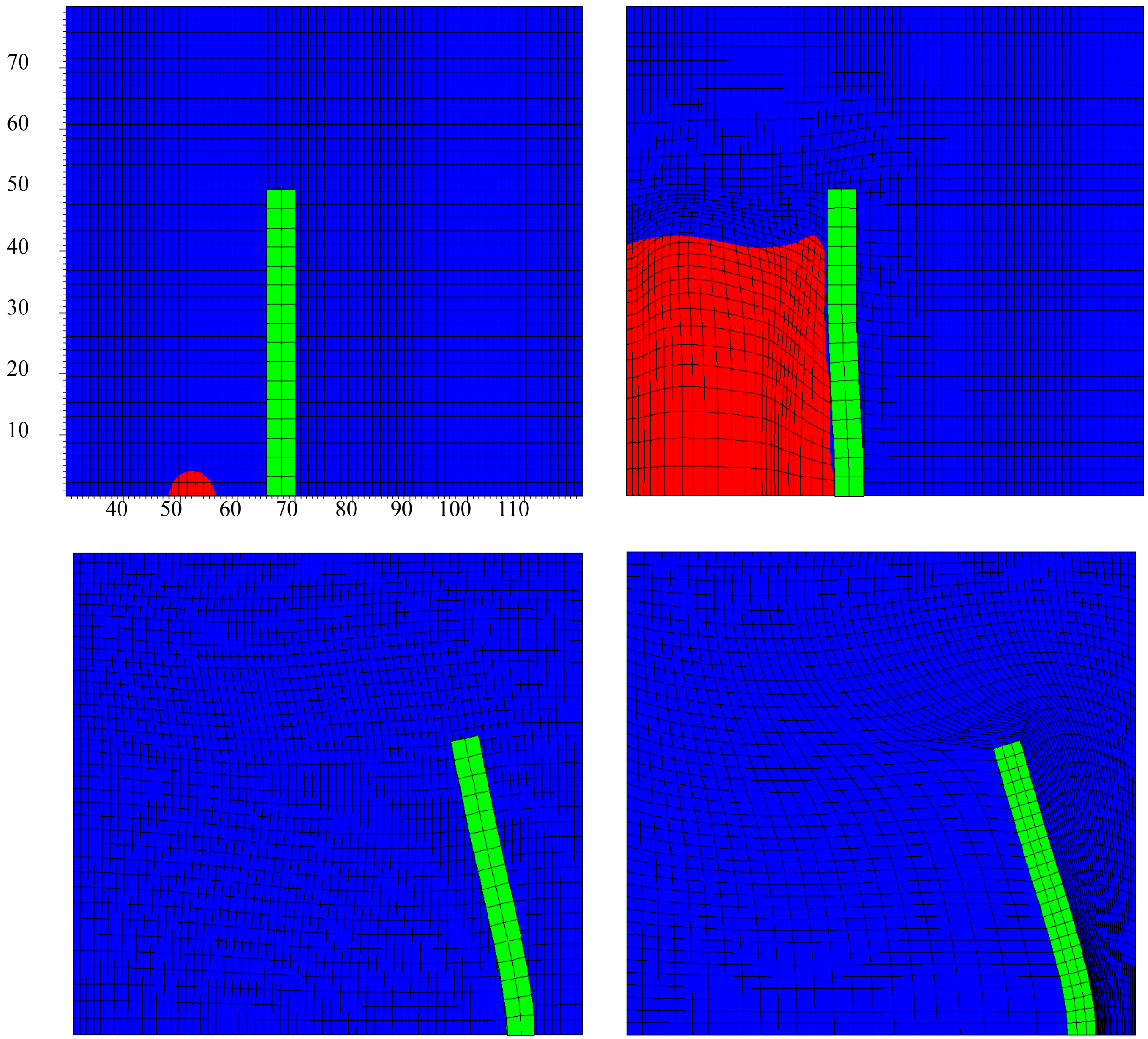
- Use central difference explicit 2 step ALE approach

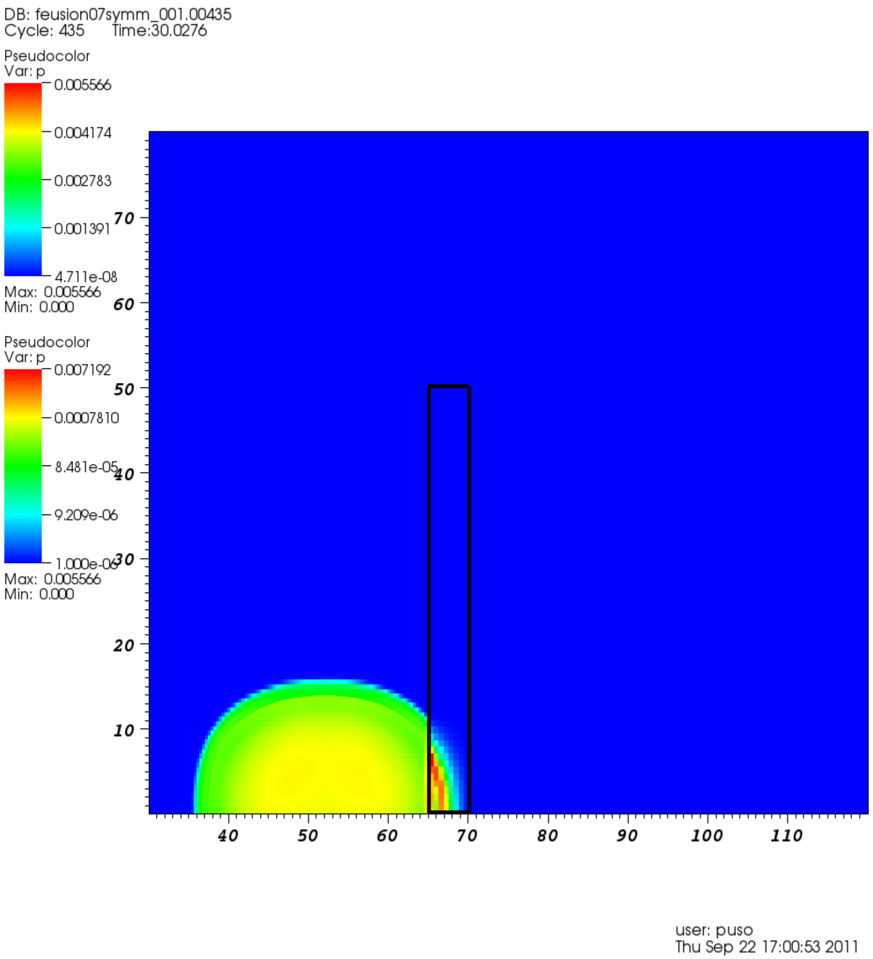


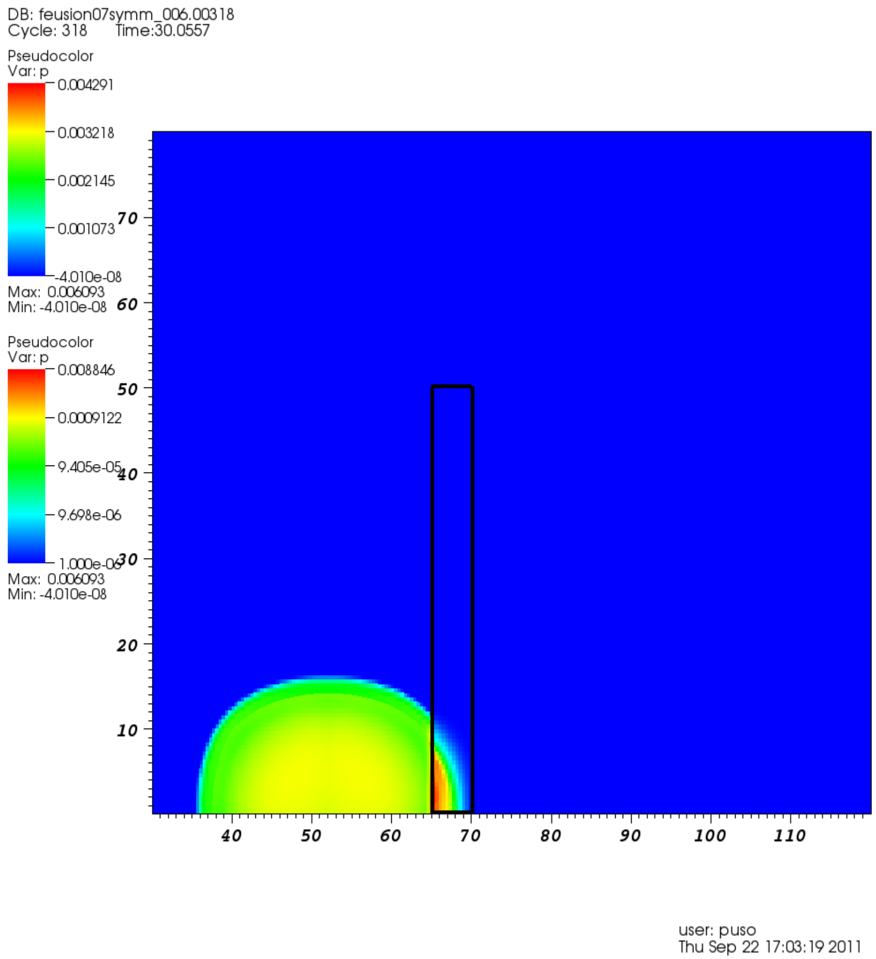


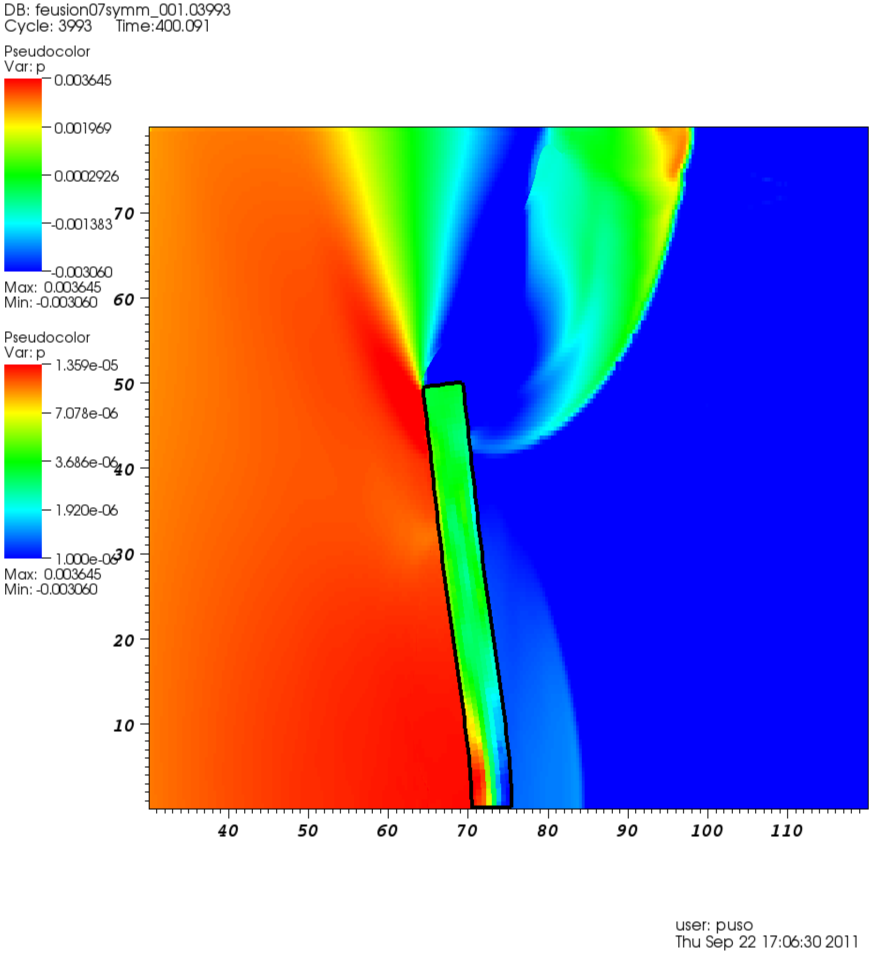
Verification: 2D blast on plate

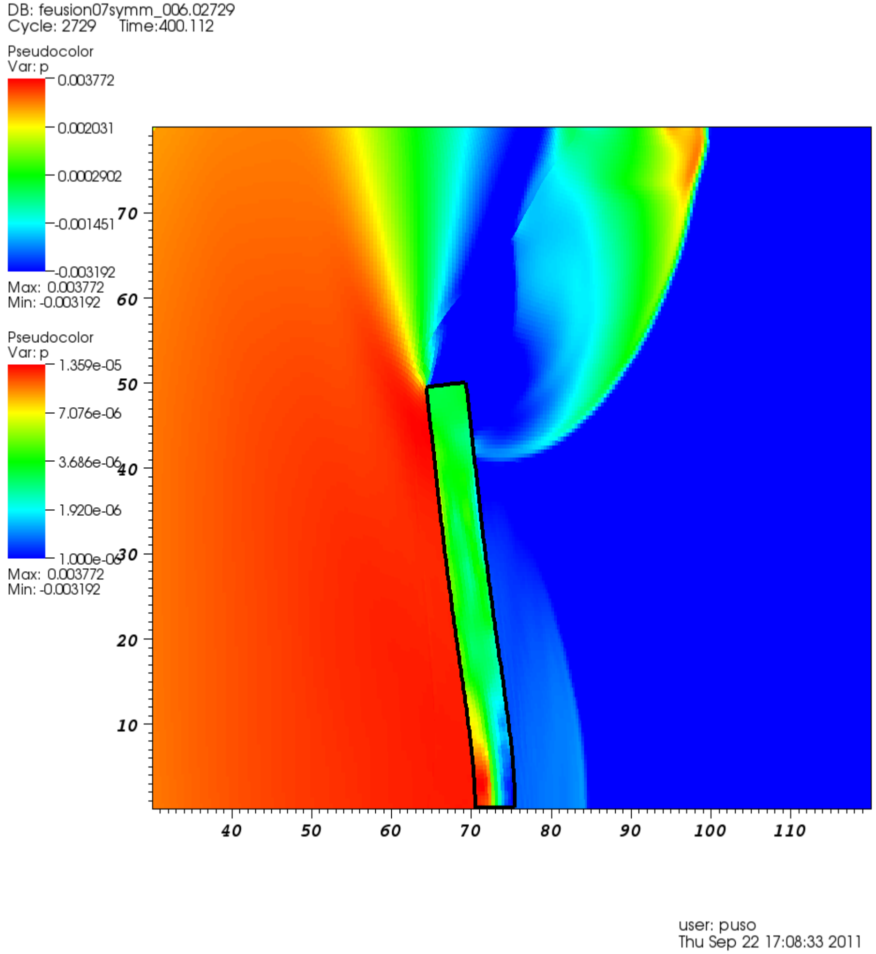
- Verify Embedded Grid with conforming ALE Model
- C4 blast loading on 5 cm. thick Al plate in air
- Compare embedded grid method to conforming mesh: 3 mesh densities

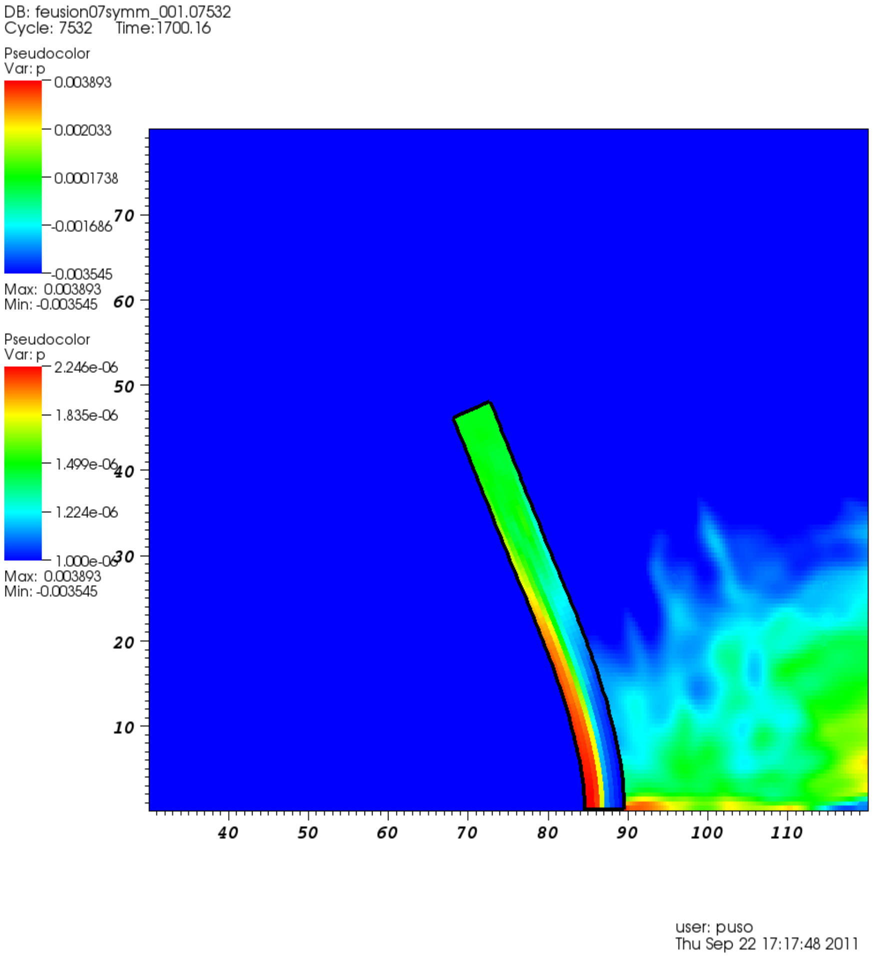


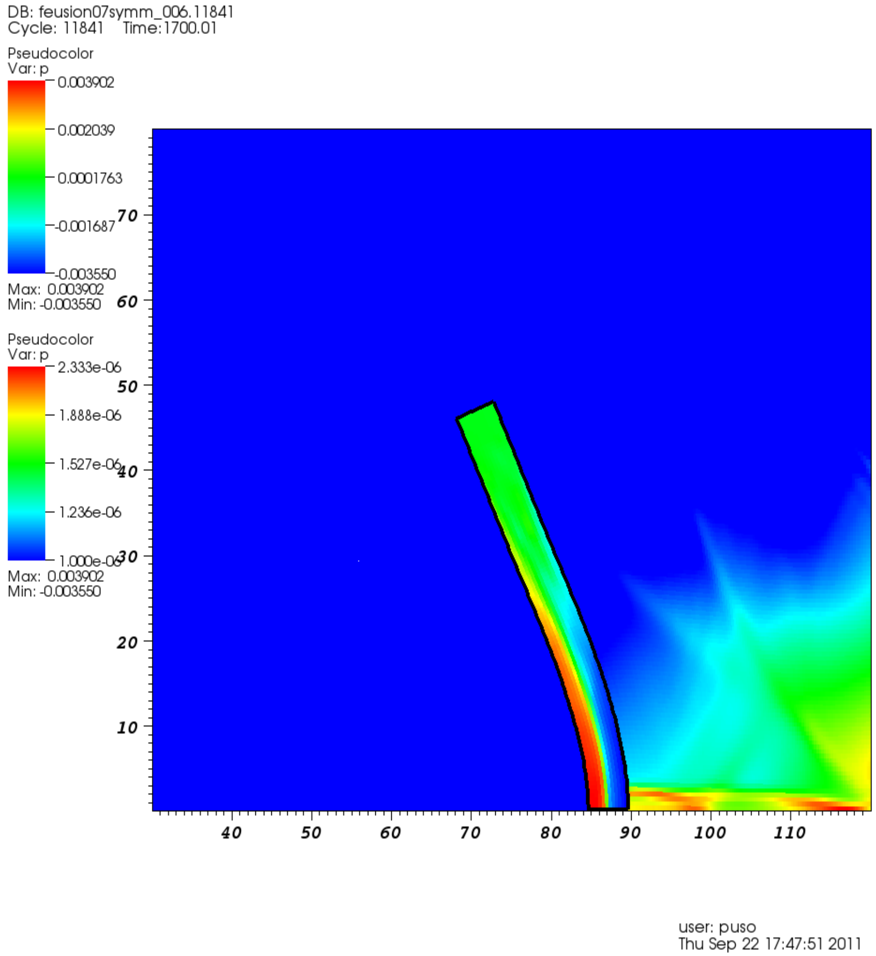

DB: feusion07symm_001.00435
Cycle: 435 Time:30.0276
Pseudocolor
Var: p
0.005566
0.004174
0.002783
0.001391
4.711e-08
Max: 0.005566
Min: 0.000
Pseudocolor
Var: p
0.007192
0.0007810
8.481e-05
9.209e-06
1.000e-06
Max: 0.005566
Min: 0.000
70
60
50
40
30
20
10
40 50 60 70 80 90 100 110
user: puso
Thu Sep 22 17:00:53 2011

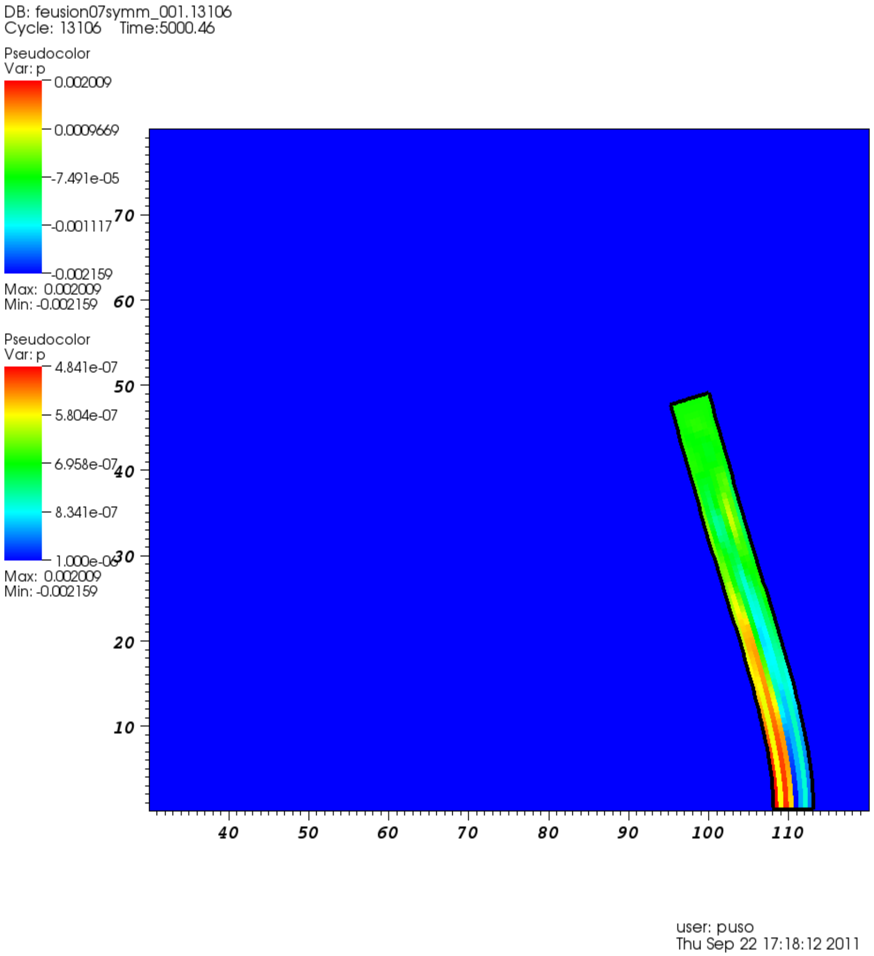

DB: feusion07symm_006.00318
Cycle: 318 Time:30.0557
Pseudocolor
Var: p
0.004291
0.003218
0.002145
0.001073
-4.010e-08
Max: 0.006093
Min: -4.010e-08
Pseudocolor
Var: p
0.008846
0.0009122
9.405e-05
9.698e-06
1.000e-06
Max: 0.006093
Min: -4.010e-08
70
60
50
40
30
20
10
40 50 60 70 80 90 100 110
user: puso
Thu Sep 22 17:03:19 2011

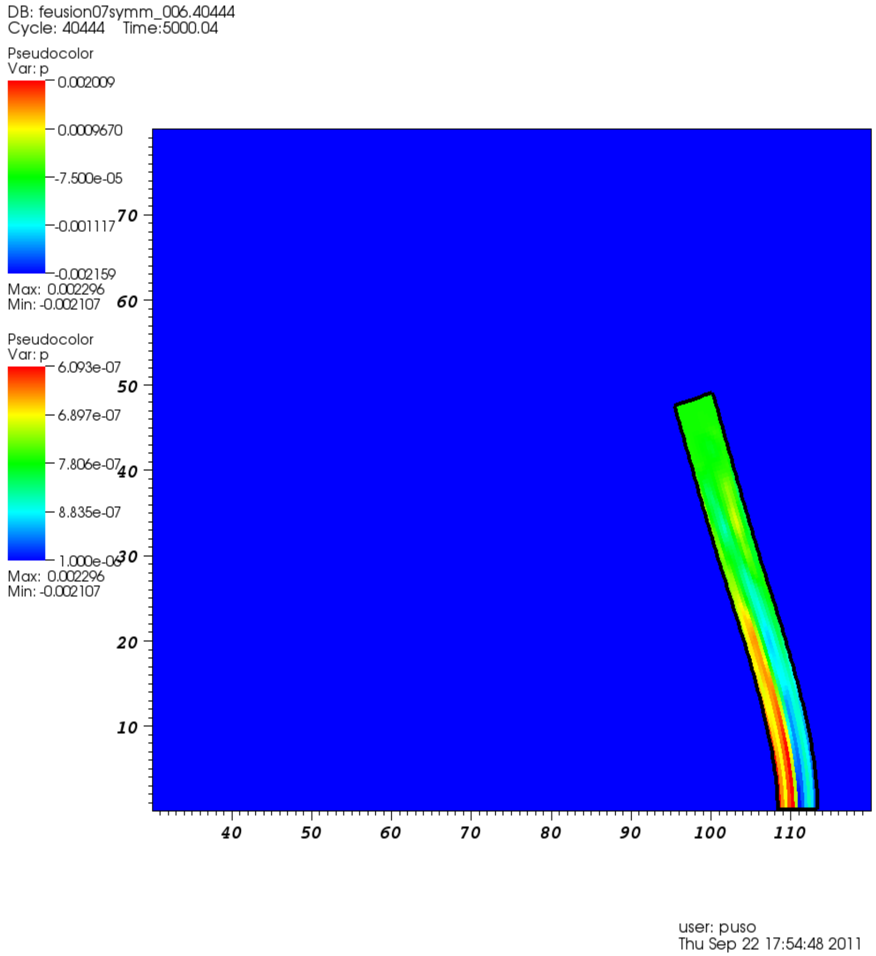

DB: feusion07symm_001.03993
Cycle: 3993 Time:400.091
Pseudocolor
Var: p
0.003645
0.001969
0.0002926
-0.001383
-0.003060
Max: 0.003645
Min: -0.003060
Pseudocolor
Var: p
1.359e-05
7.078e-06
3.686e-06
1.920e-06
1.000e-06
Max: 0.003645
Min: -0.003060
70
60
50
40
30
20
10
40
50
60
70
80
90
100
110
user: puso
Thu Sep 22 17:06:30 2011

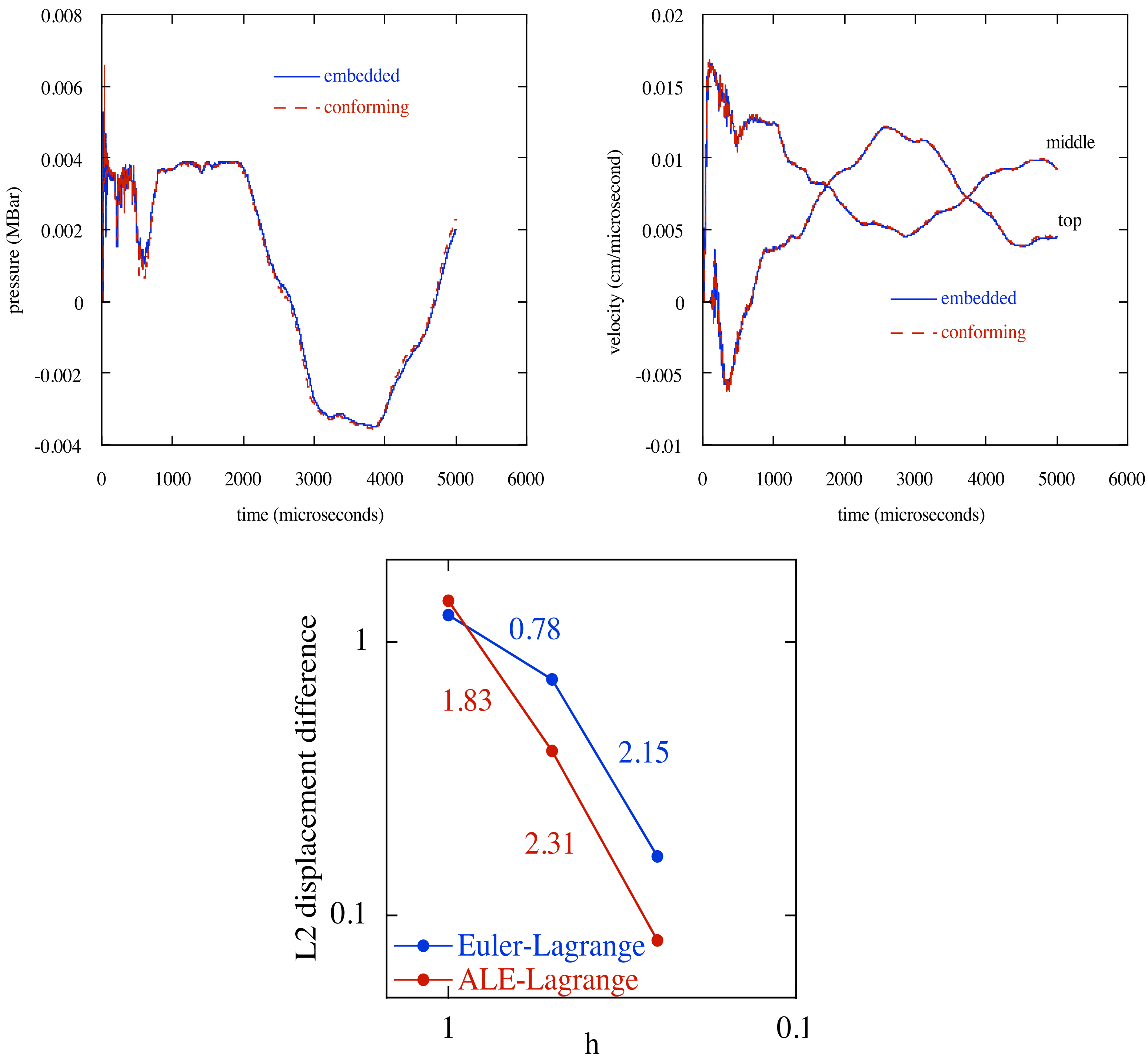

DB: feusion07symm_006.02729
Cycle: 2729 Time:400.112
Pseudocolor
Var: p
0.003772
0.002031
0.0002902
-0.001451
-0.003192
Max: 0.003772
Min: -0.003192
Pseudocolor
Var: p
1.359e-05
7.076e-06
3.686e-06
1.920e-06
1.000e-06
Max: 0.003772
Min: -0.003192
70
60
50
40
30
20
10
40
50
60
70
80
90
100
110
user: puso
Thu Sep 22 17:08:33 2011


DB: feusion07symm_001.07532
Cycle: 7532 Time:1700.16
Pseudocolor
Var: p
0.003893
0.002033
0.0001738
-0.001686
-0.003545
Max: 0.003893
Min: -0.003545
Pseudocolor
Var: p
2.246e-06
1.835e-06
1.499e-06
1.224e-06
1.000e-06
Max: 0.003893
Min: -0.003545
70
60
50
40
30
20
10
40
50
60
70
80
90
100
110
user: puso
Thu Sep 22 17:17:48 2011


DB: feusion07symm_006.11841
Cycle: 11841 Time:1700.01
Pseudocolor
Var: p
0.003902
0.002039
0.0001763
-0.001687
-0.003550
Max: 0.003902
Min: -0.003550
Pseudocolor
Var: p
2.333e-06
1.888e-06
1.527e-06
1.236e-06
1.000e-06
Max: 0.003902
Min: -0.003550
70
60
50
40
30
20
10
40
50
60
70
80
90
100
110
user: puso
Thu Sep 22 17:47:51 2011


DB: feusion07symm_001.13106
Cycle: 13106 Time:5000.46
Pseudocolor
Var: p
0.002009
0.0009669
-7.491e-05
-0.001117
-0.002159
Max: 0.002009
Min: -0.002159
Pseudocolor
Var: p
4.841e-07
5.804e-07
6.958e-07
8.341e-07
1.000e-06
Max: 0.002009
Min: -0.002159
user: puso
Thu Sep 22 17:18:12 2011

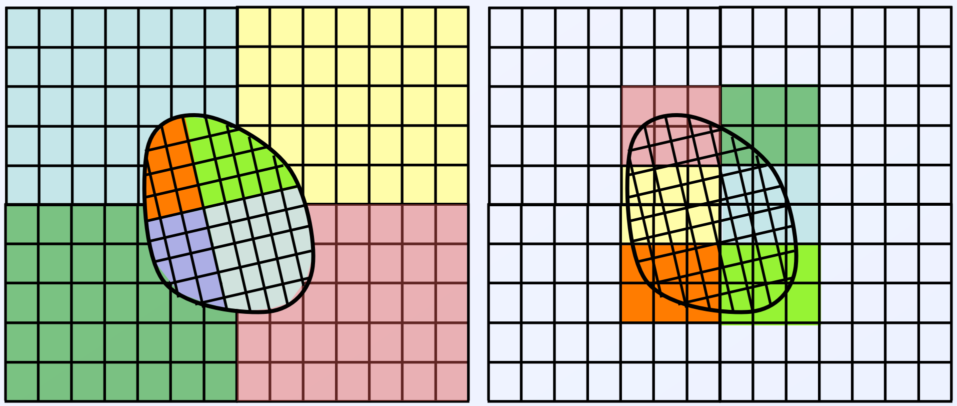

DB: feusion07symm_006.40444
Cycle: 40444 Time:5000.04
Pseudocolor
Var: p
0.002009
0.0009670
-7.500e-05
-0.001117
-0.002159
Max: 0.002296
Min: -0.002107
Pseudocolor
Var: p
6.093e-07
6.897e-07
7.806e-07
8.835e-07
1.000e-06
Max: 0.002296
Min: -0.002107
user: puso
Thu Sep 22 17:54:48 2011



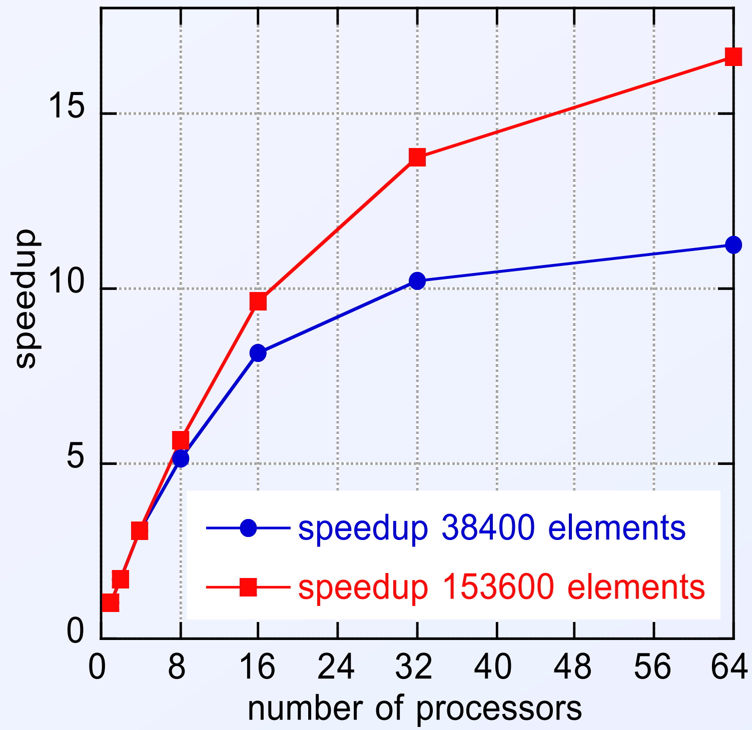
pressure (MBar)
time (microseconds)
embedded
conforming
velocity (cm/microsecond)
time (microseconds)
middle
top
embedded
conforming
L2 displacement difference
h
0.78
1.83
2.15
2.31
Euler-Lagrange
ALE-Lagrange

Parallel Issues

- Use dynamic domain decomposition
  - Build every time step
    - Move data to new decomposition,
    - Do parallel calculation
      - Compute “cut” background cells
      - Do parallel solve
    - Move data back
- Problem took 28~30 iterations for every mesh refinement



Static Decomposition Dynamic Decomposition



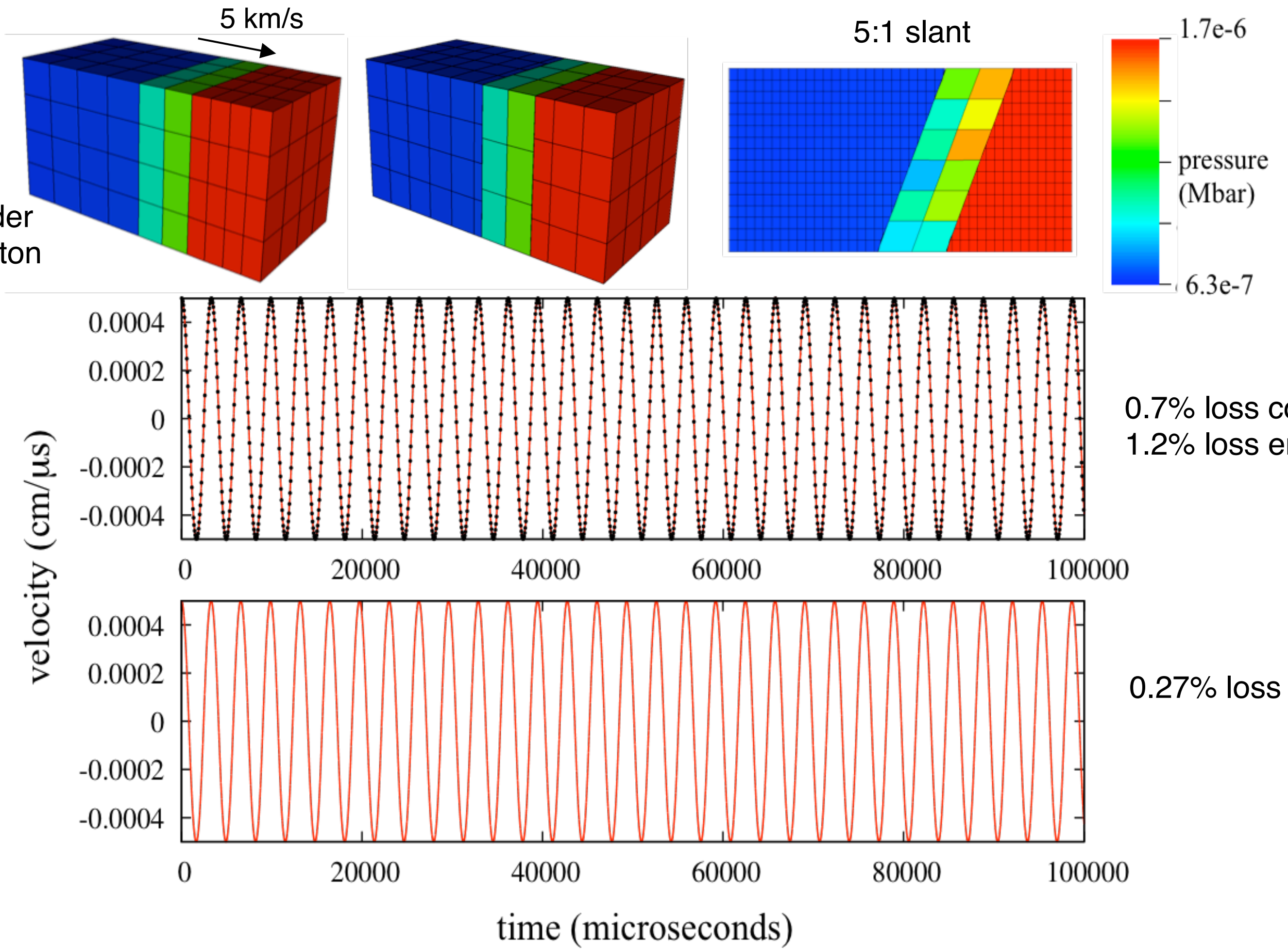
Parallel speedup versus number of processors.

Piston problem: check energy conservation

- Considered 2 piston problems: 3D rectangular cylinder and 2D slanted piston
- Compare energy loss for conforming and embedded meshes

1x1x2 cm cylinder
0.4 cm thick piston

oscillates at ~300 Hz

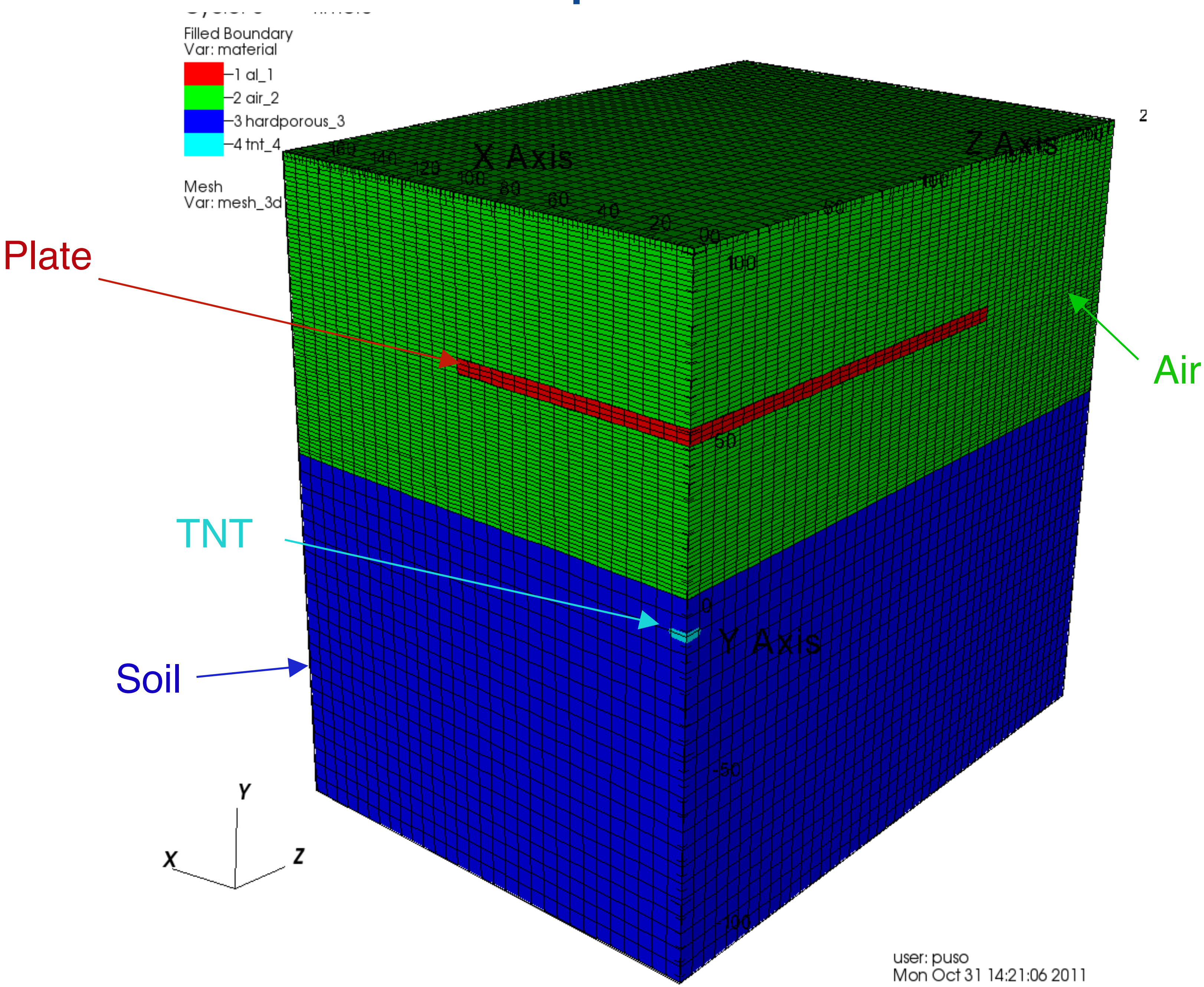


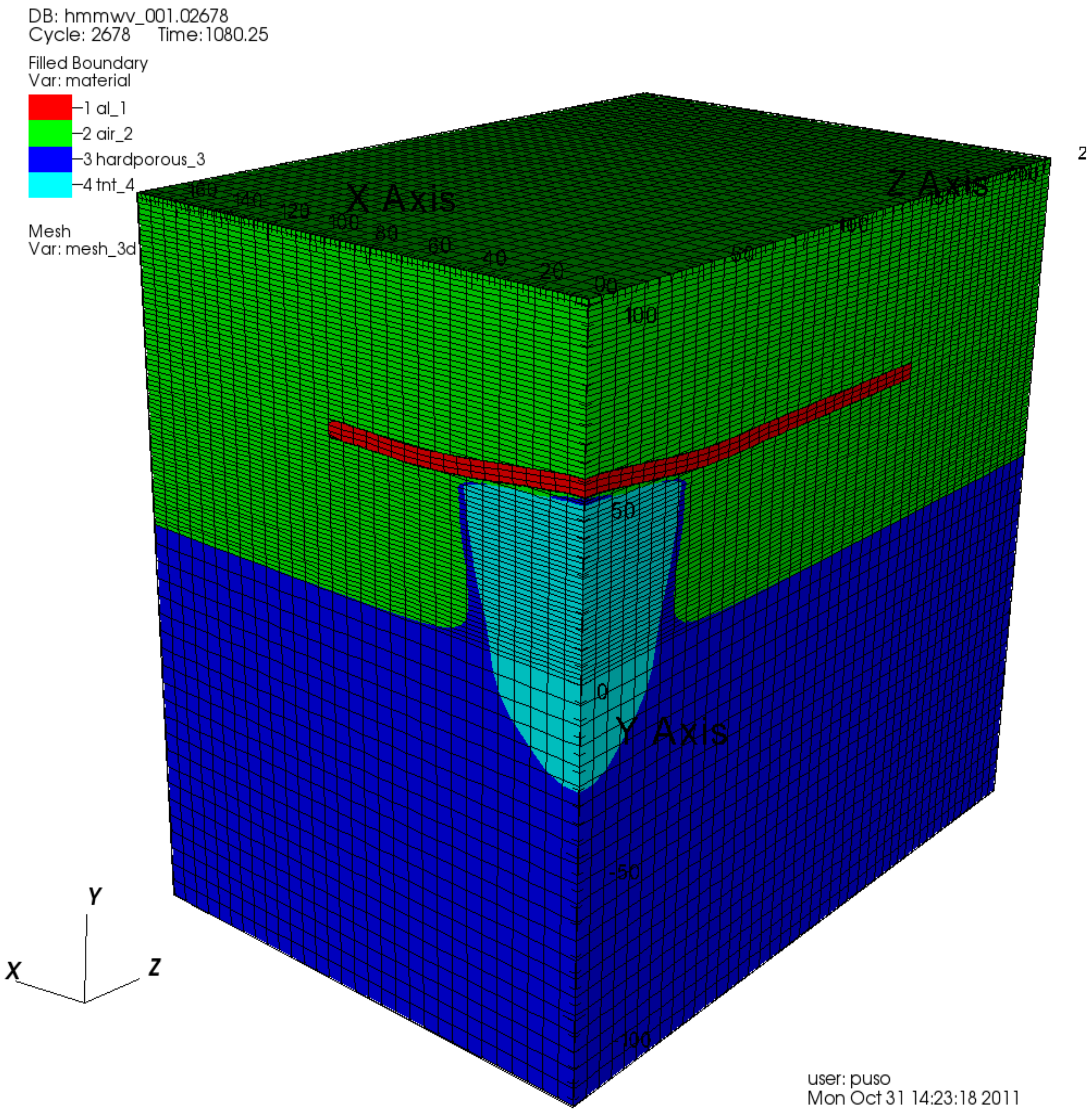
0.7% loss conforming
1.2% loss embedded

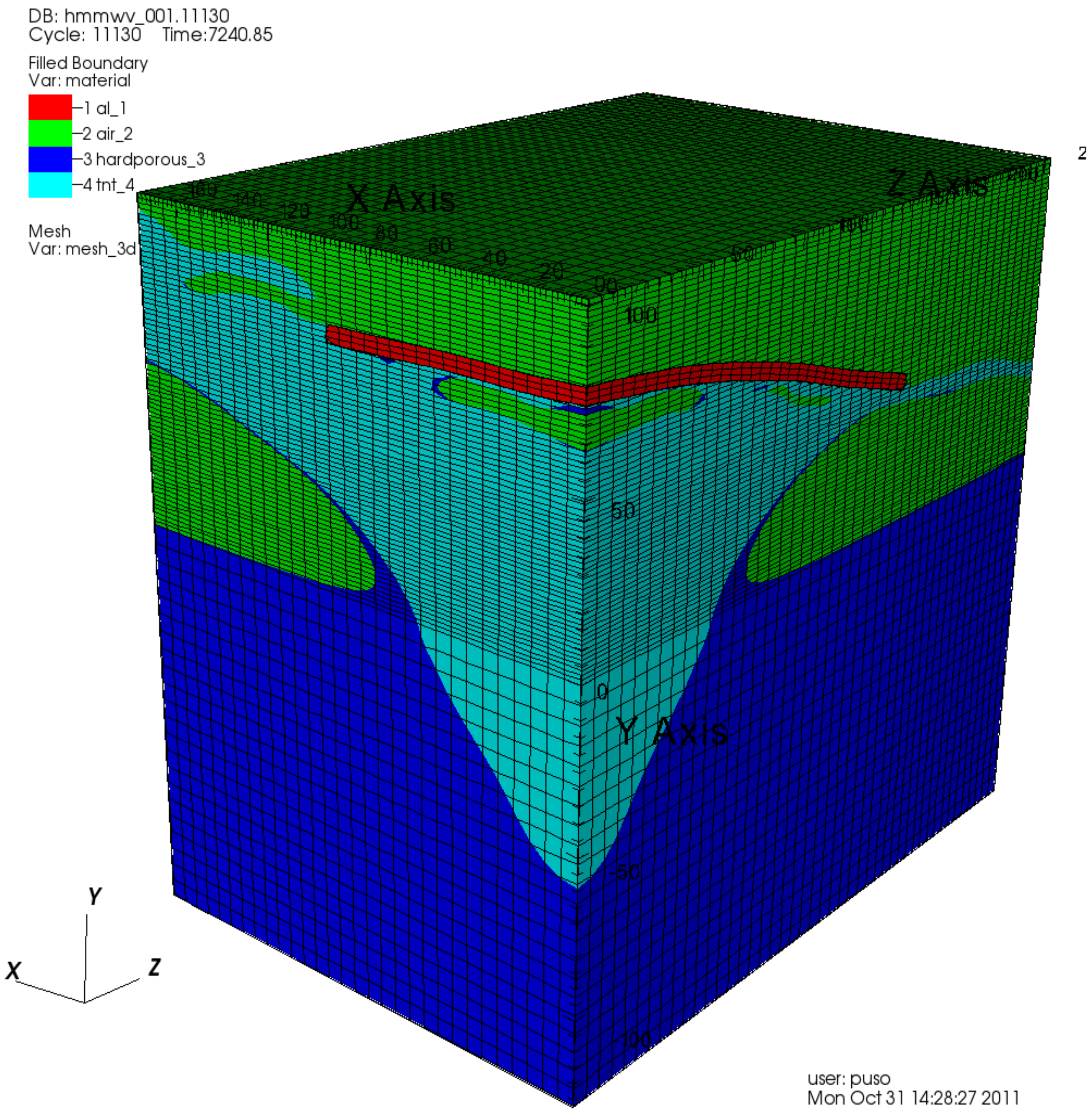
0.27% loss embedded

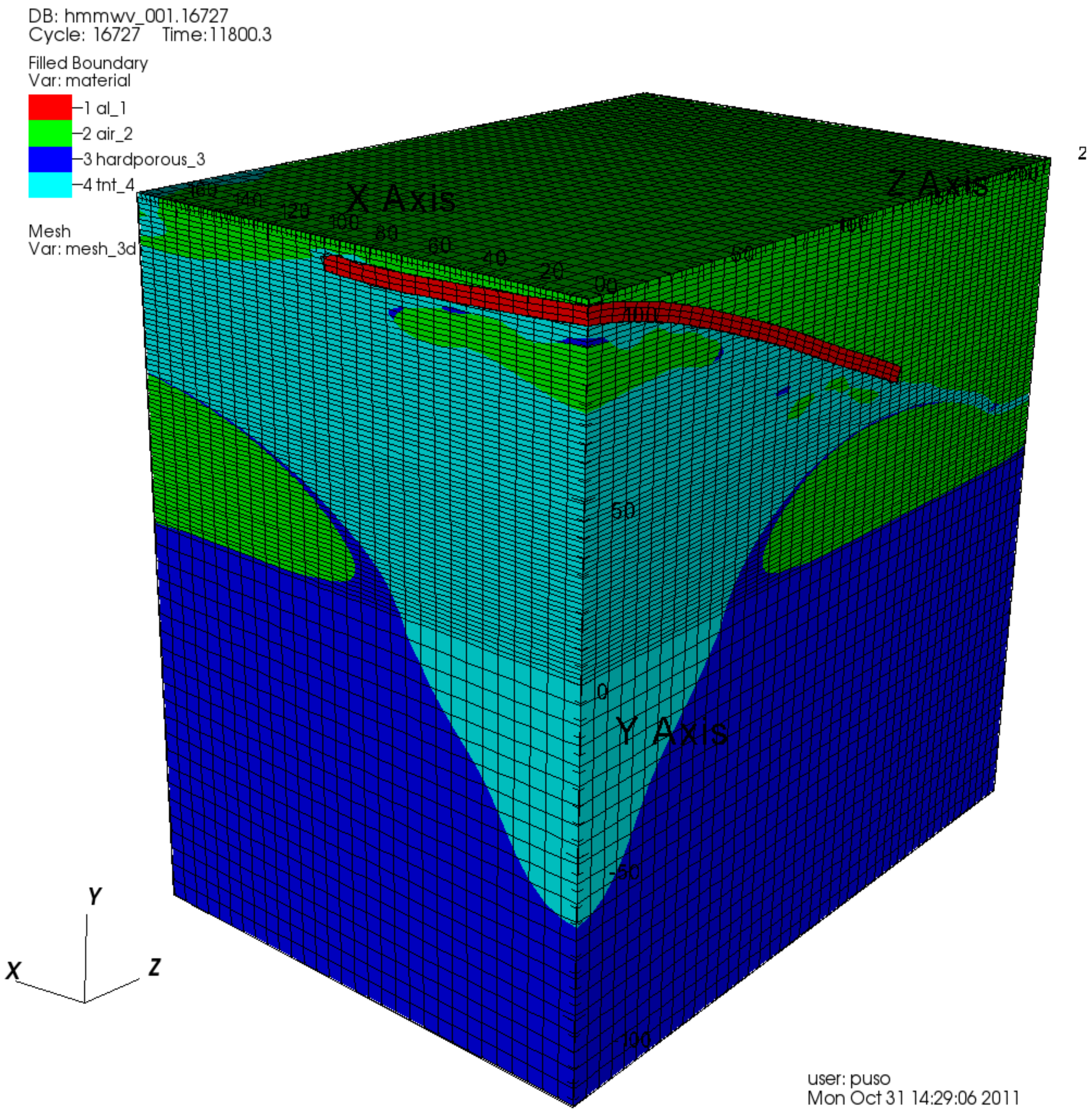
- After 1 million time steps (1 s)
  - 3D: 6% and10% loss for conforming and embedded meshes respectively
  - 2D: 1.6% loss for embedded mesh

Buried mine blast on 3D plate



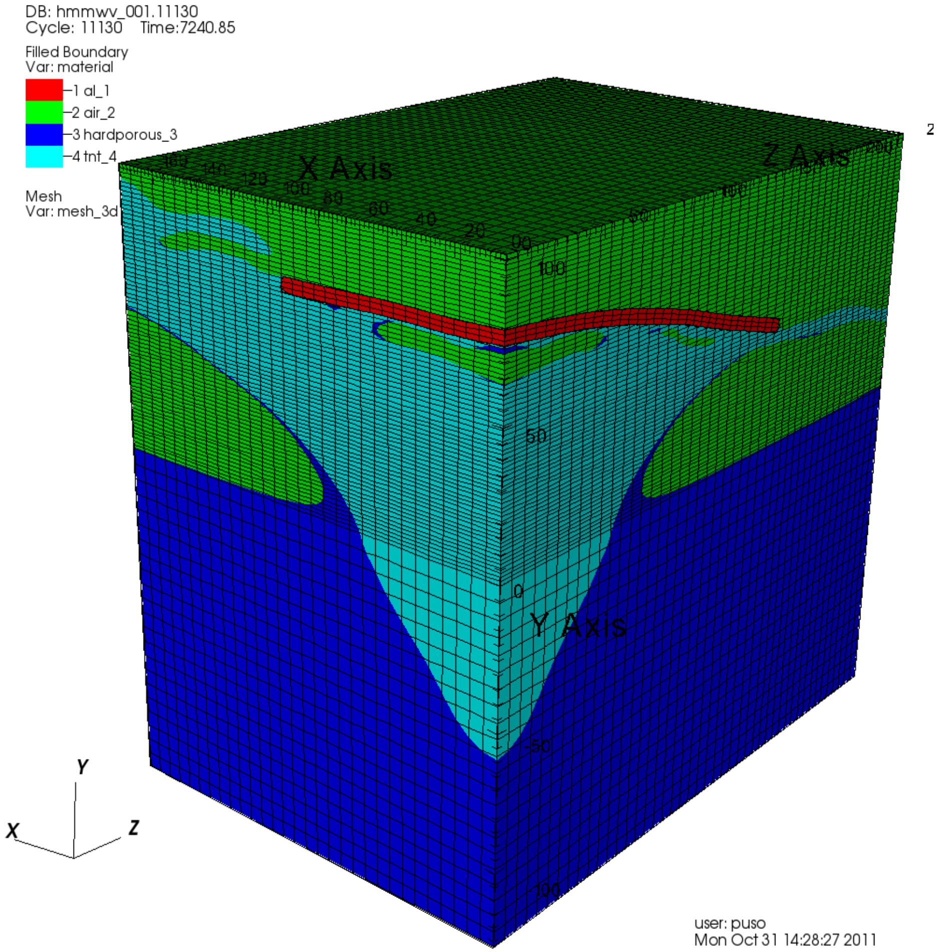

DB: hmmwv_001.02678
Cycle: 2678 Time:1080.25
Filled Boundary
Var: material
1 al_1
2 air_2
3 hardporous_3
4 tnt_4
Mesh
Var: mesh_3d
X Axis
Z Axis
Y Axis
Y
X
Z
user: puso
Mon Oct 31 14:23:18 2011


DB: hmmwv_001.11130
Cycle: 11130 Time:7240.85
Filled Boundary
Var: material
1 al_1
2 air_2
3 hardporous_3
4 tnt_4
Mesh
Var: mesh_3d
X Axis
Z Axis
Y Axis
user: puso
Mon Oct 31 14:28:27 2011

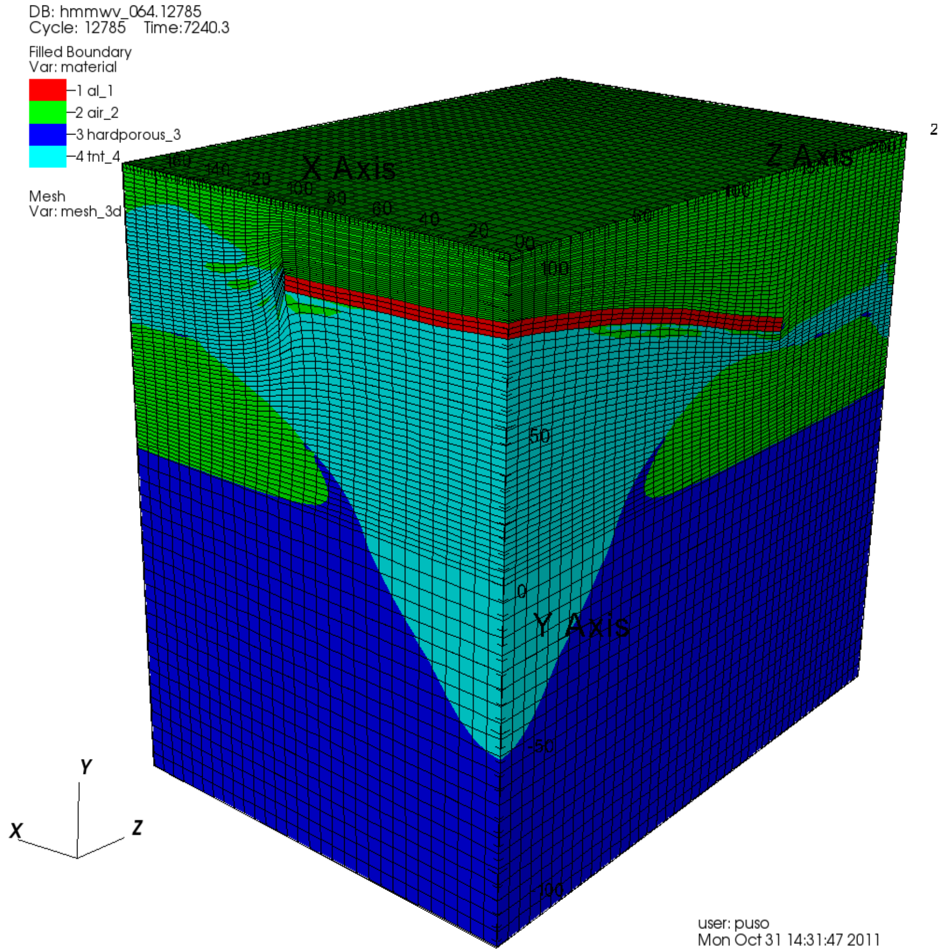


DB: hmmwv_001.16727
Cycle: 16727 Time:11800.3
Filled Boundary
Var: material
1 al_1
2 air_2
3 hardporous_3
4 tnt_4
Mesh
Var: mesh_3d
X Axis
Y Axis
Z Axis
user: puso
Mon Oct 31 14:29:06 2011

Compare embedded grid to conforming

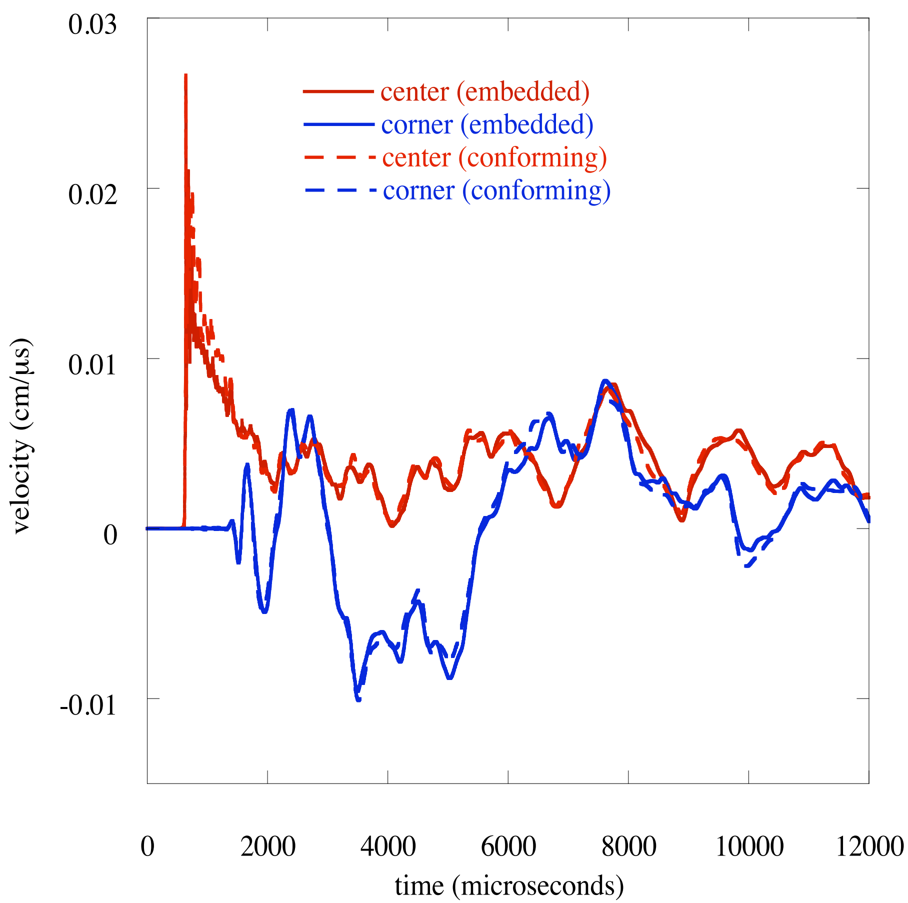


Embedded Grid at 7.24 ms



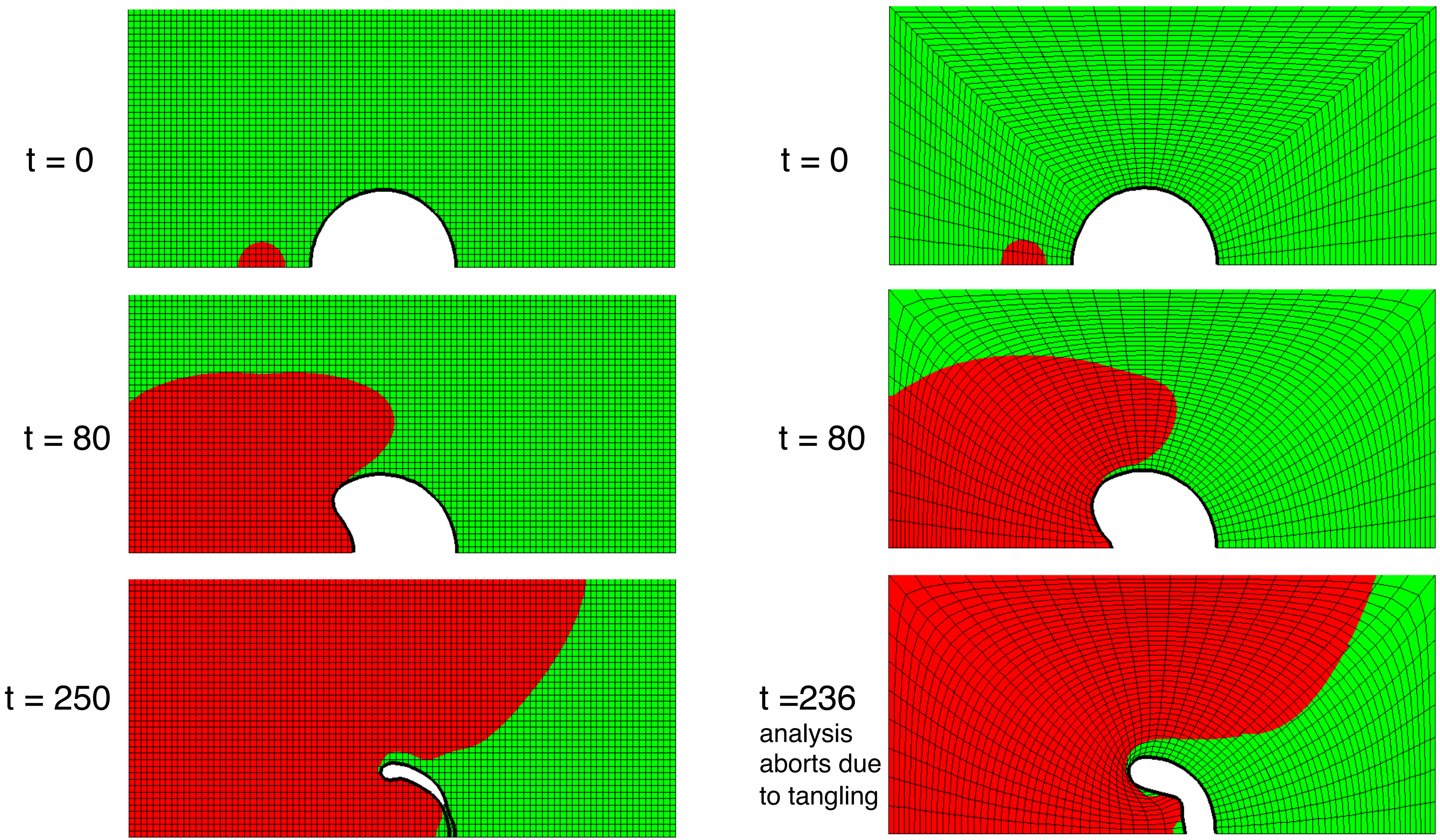
Conforming ALE Grid at 7.24 ms

Buried mine blast on 3D plate: velocity



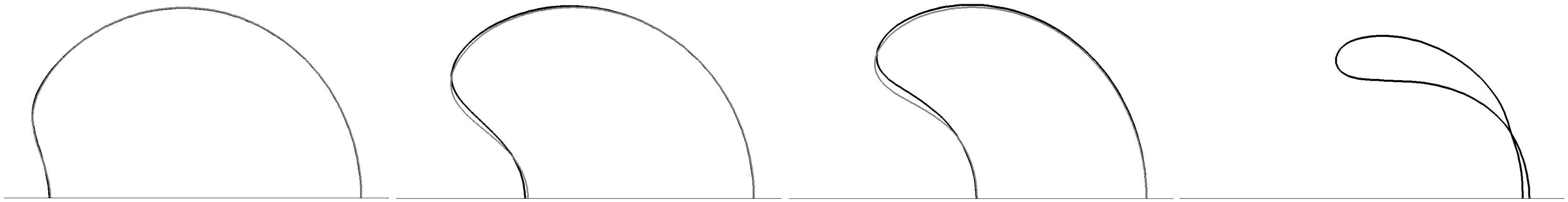
Mesh Study: Blast on structural shell element pipe

- Verify Embedded Grid with conforming ALE Model using shell elements
- C4 blast loading on 2 mm thick Al pipe in air
- Compare embedded grid method to conforming mesh: 4 mesh densities



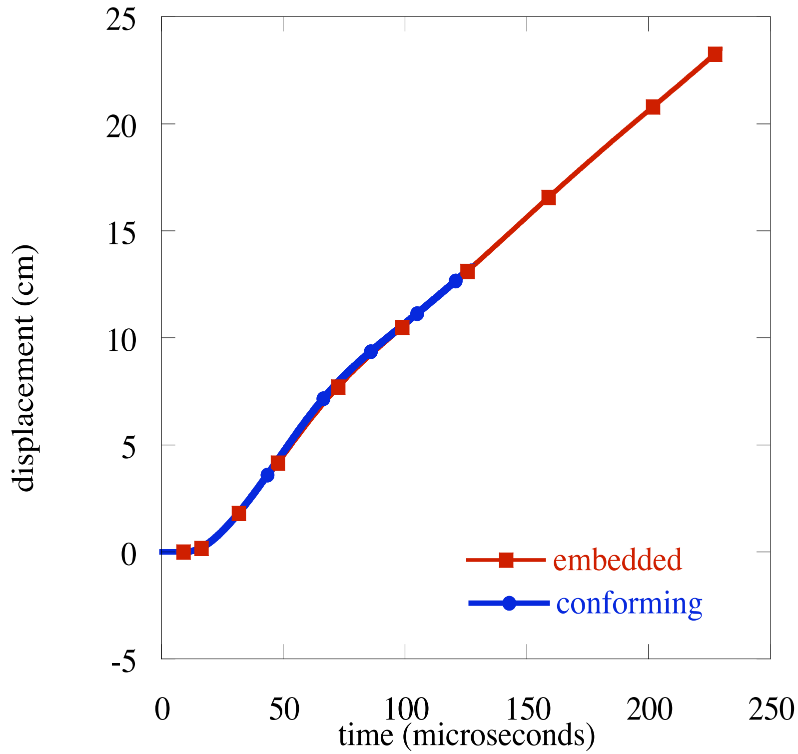
Mesh Study: Blast on structural shell element pipe

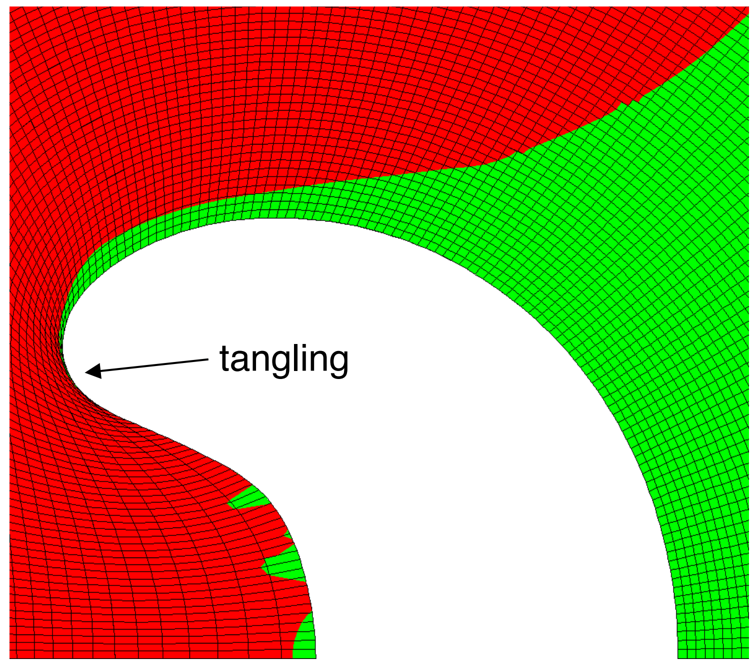
- Compare pipe displacement from *finest* mesh
  - Black line: embedded mesh results
  - Grey line: conforming ALE results



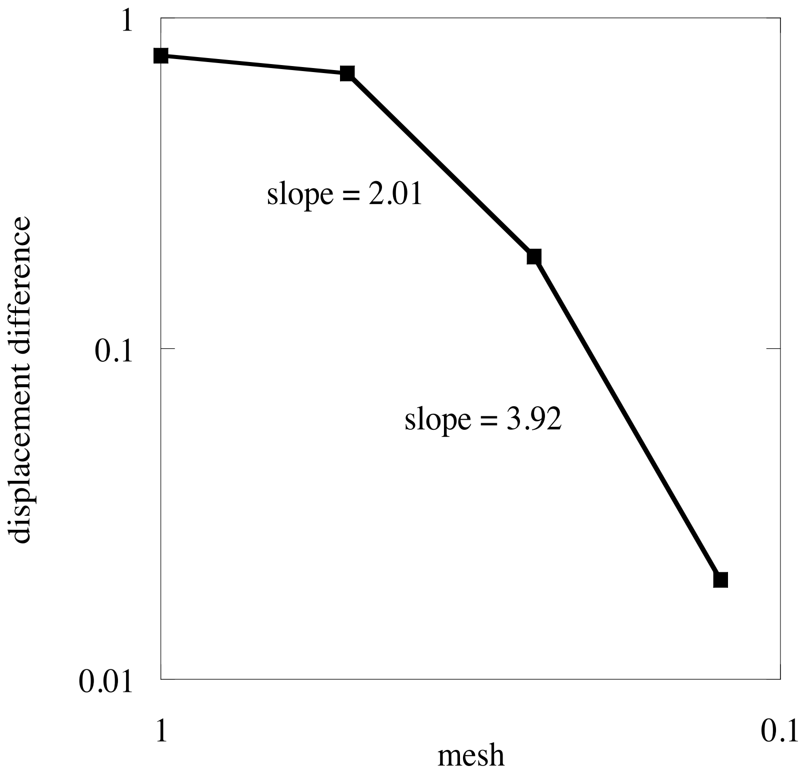
t = 40 t = 80 t = 120 t = 260

~conforming ALE analysis aborts

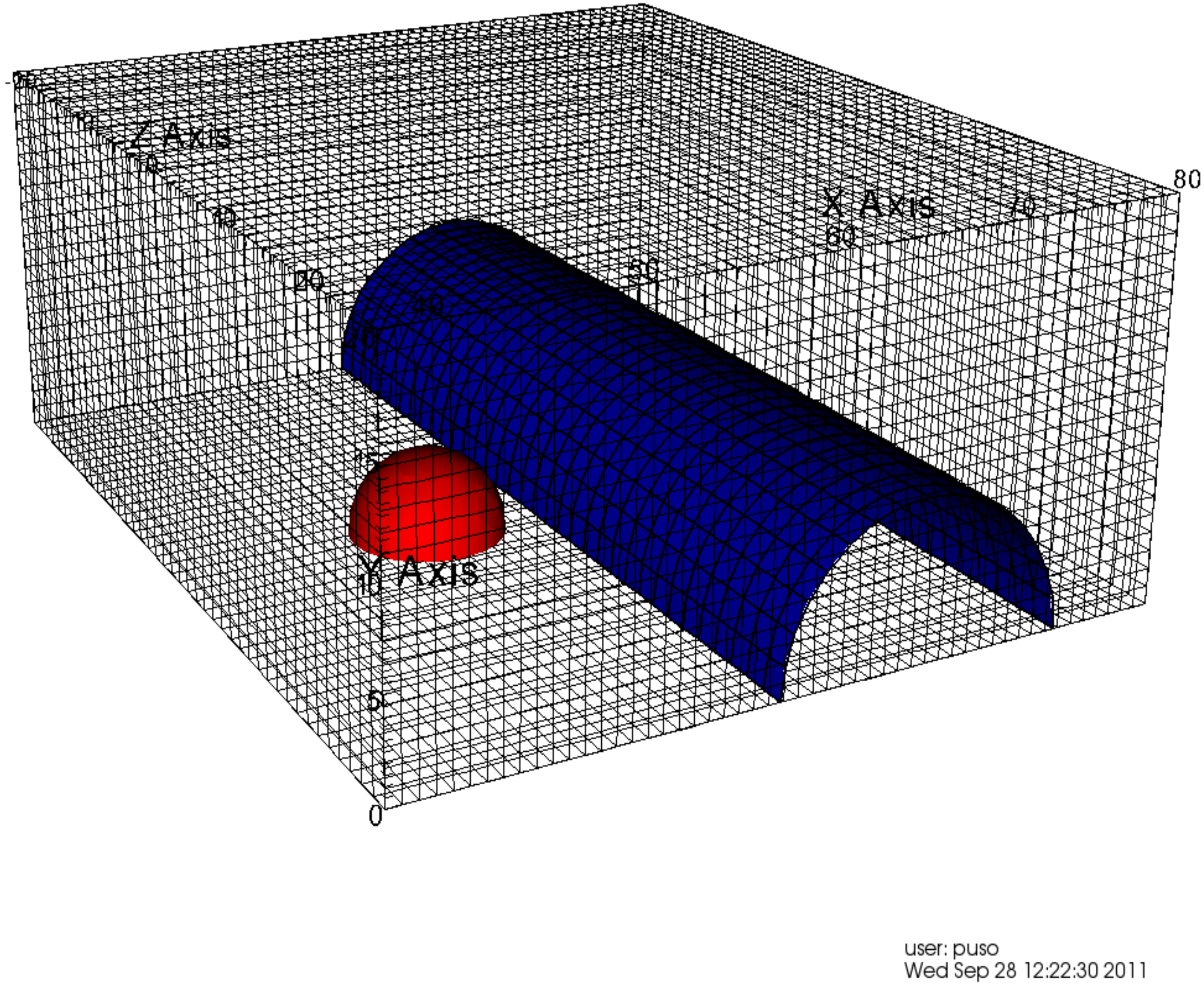


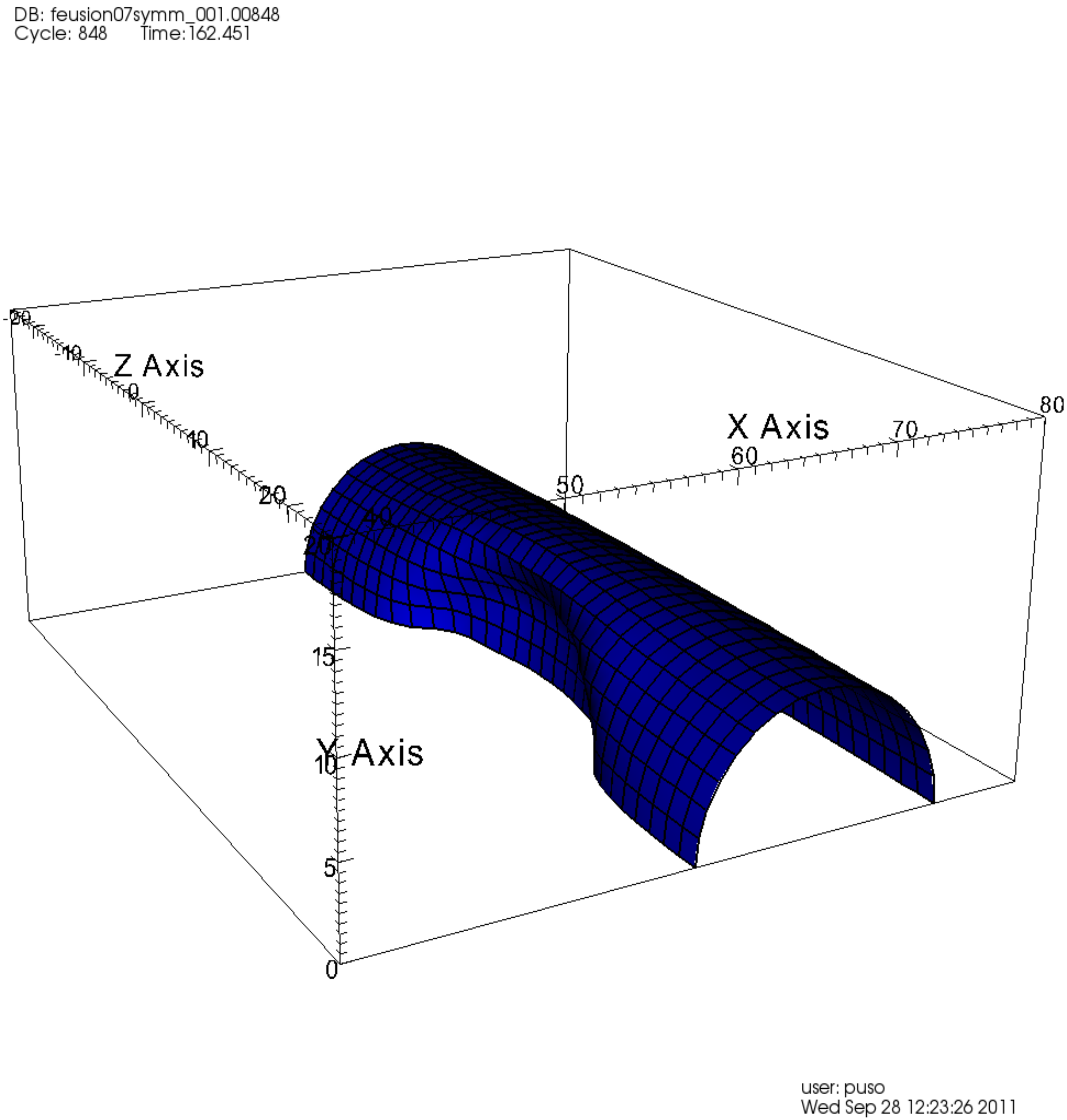


Mesh Study: Blast on structural shell element pipe

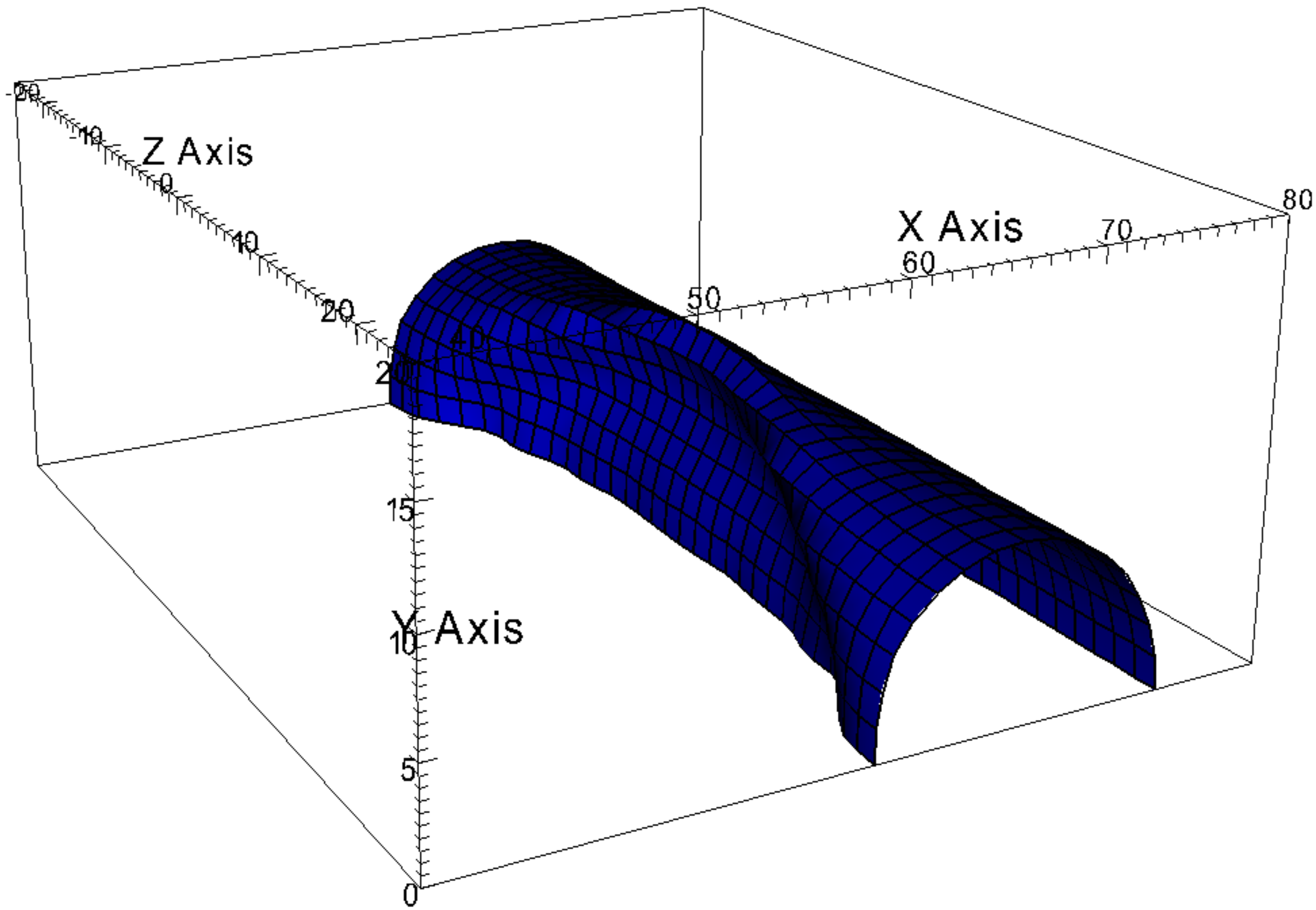


Structural shells: blast on pipe

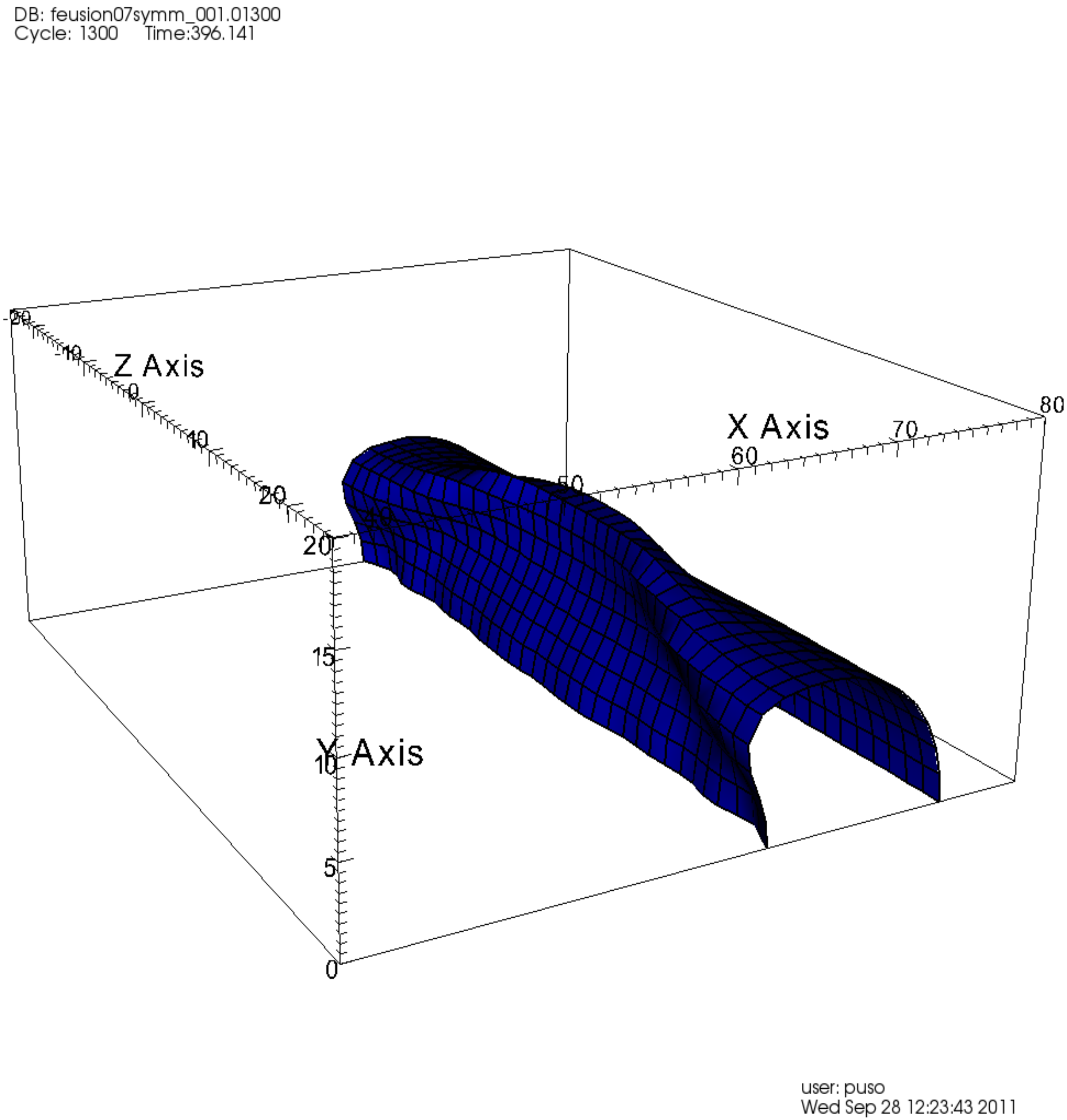


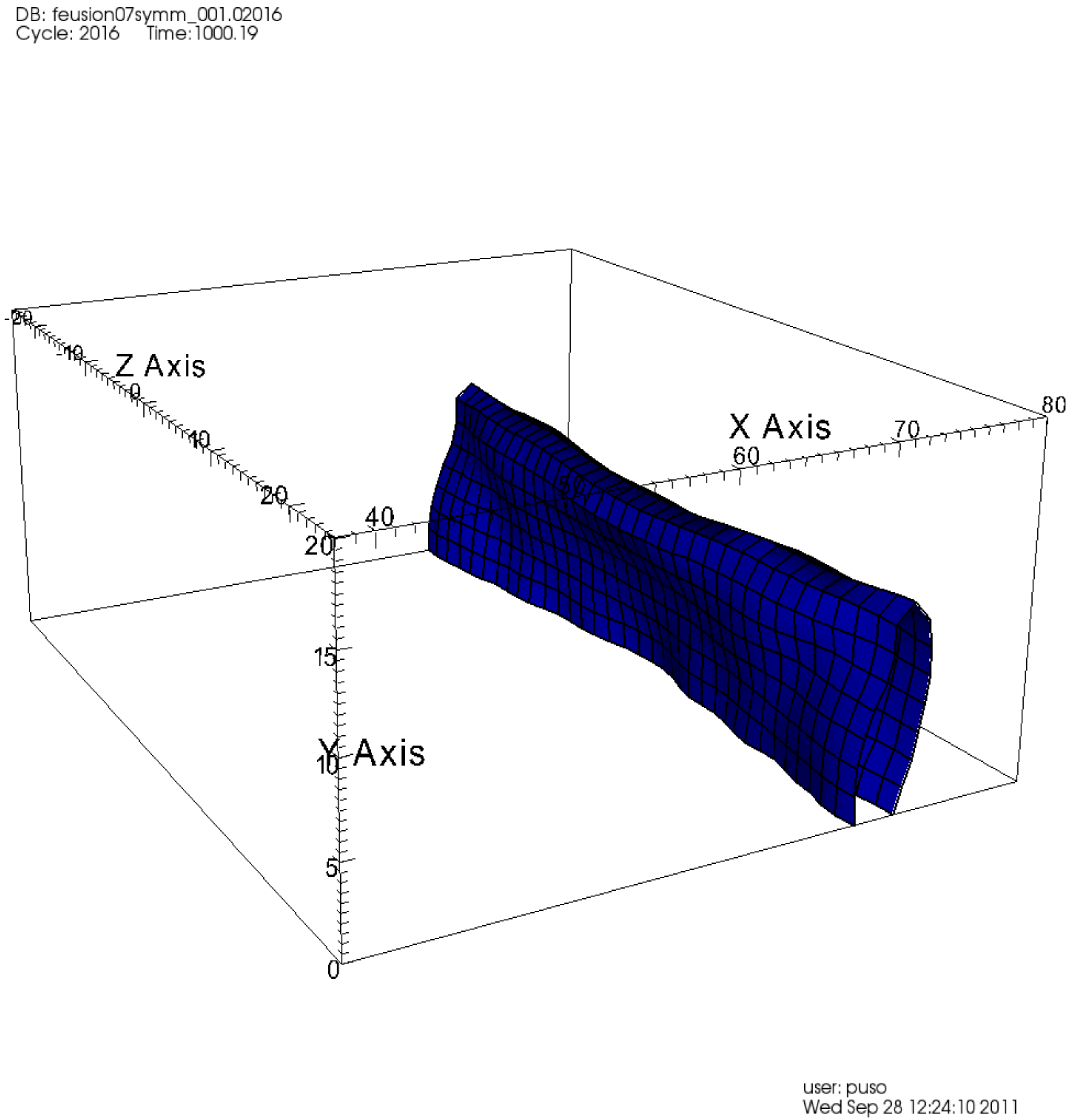

DB: feusion07symm_001.00848
Cycle: 848 Time:162.451
Z Axis
-20
-10
0
10
20
X Axis
50
60
70
80
Y Axis
0
5
10
15
user: puso
Wed Sep 28 12:23:26 2011

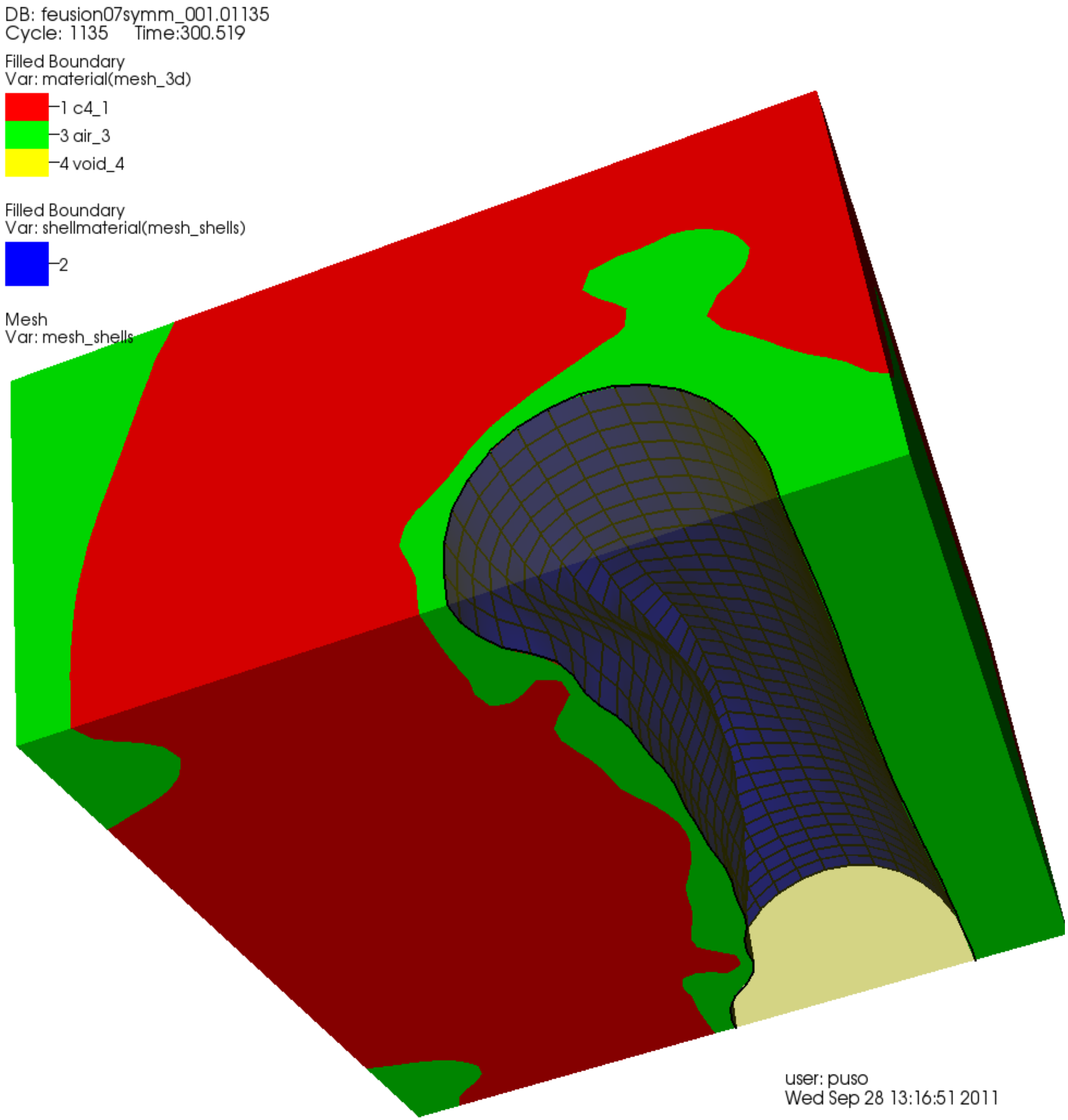
DB: feusion07symm_001.01135
Cycle: 1135 Time:300.519

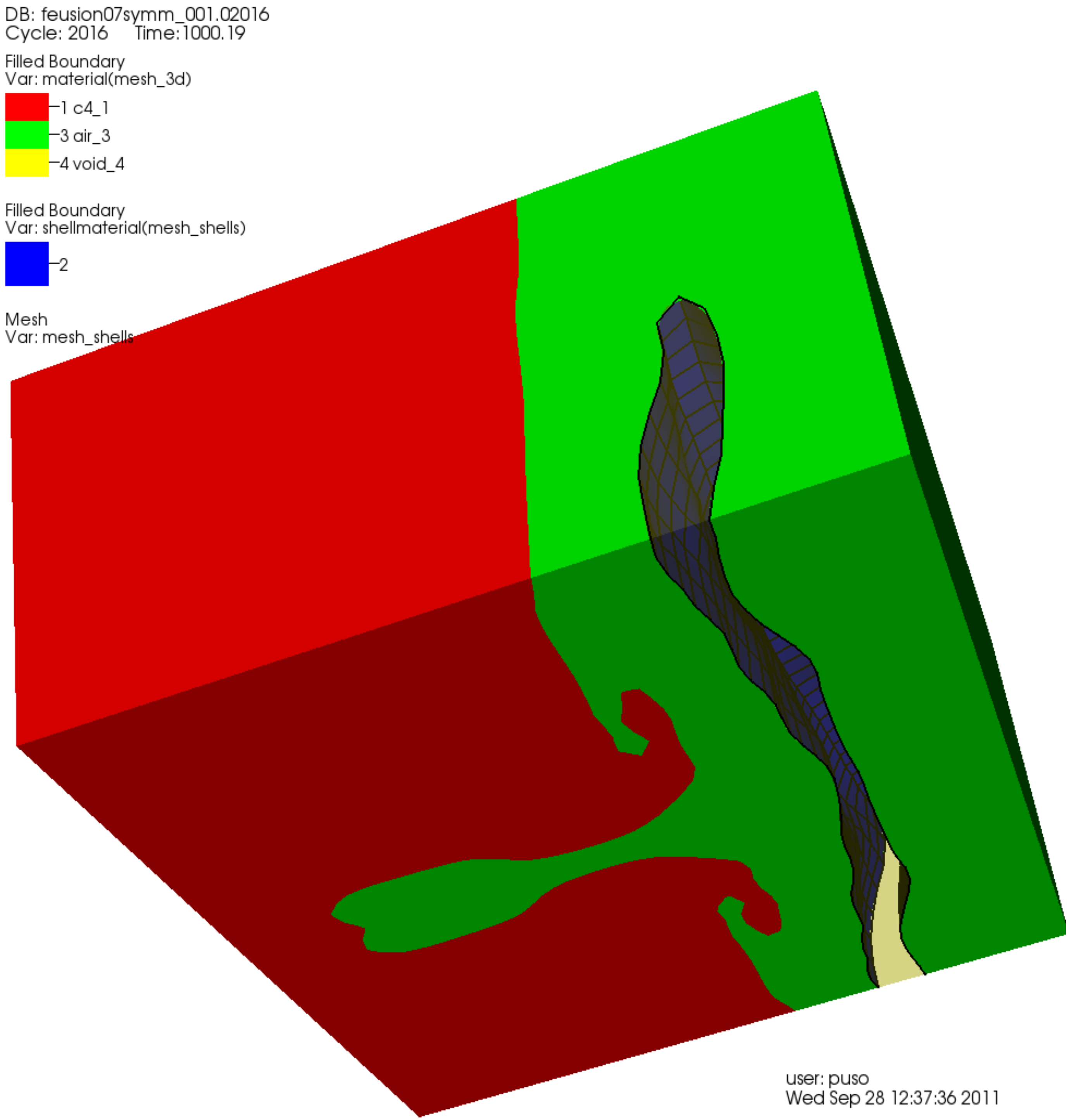


user: puso
Wed Sep 28 12:24:48 2011

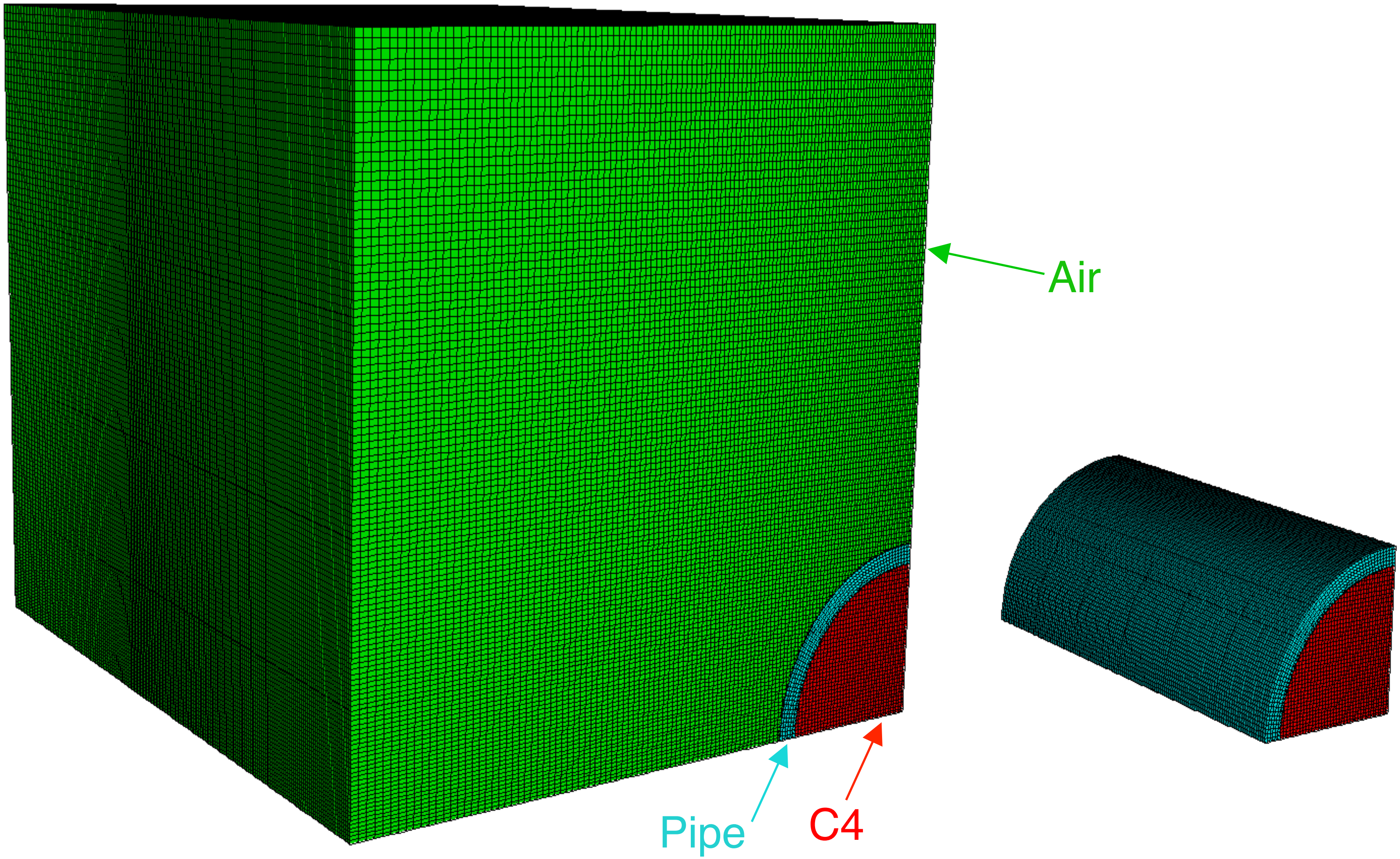

DB: feusion07symm_001.01300
Cycle: 1300 Time:396.141
Z Axis
X Axis
Y Axis
-10
0
10
20
50
60
70
80
20
15
10
5
0
user: puso
Wed Sep 28 12:23:43 2011

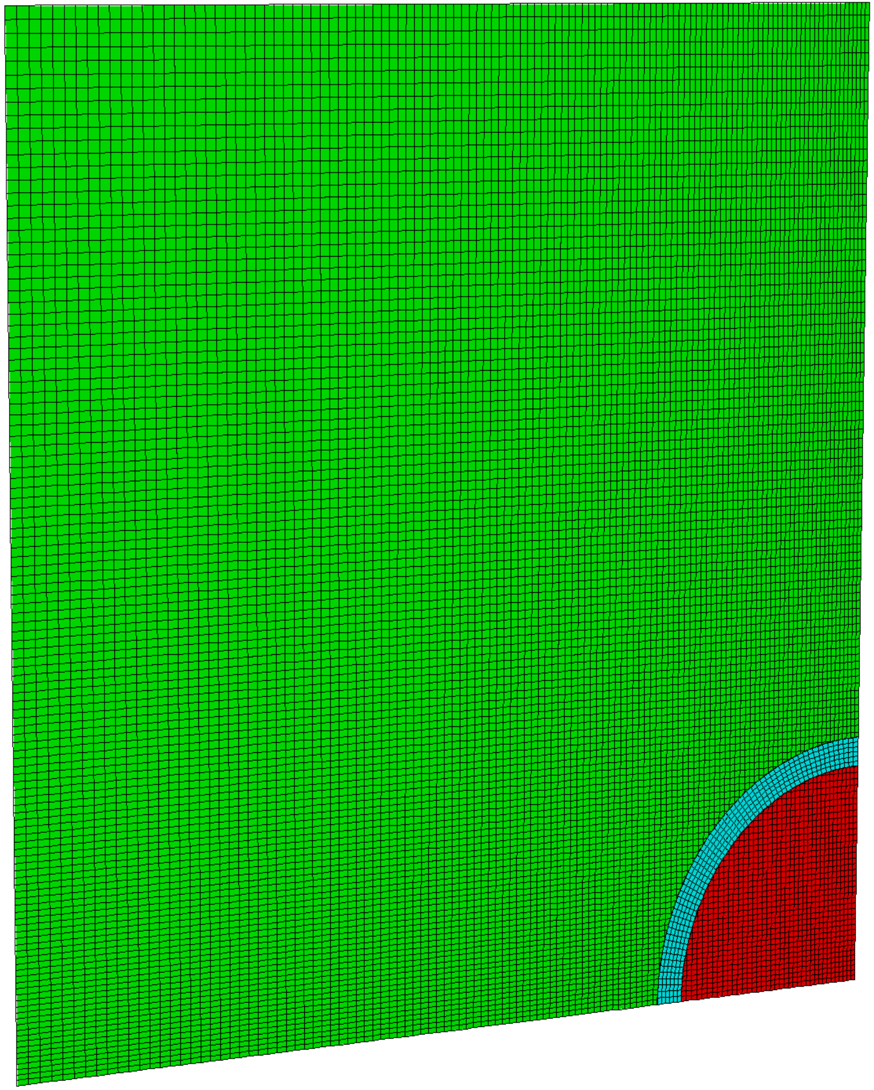

DB: feusion07symm_001.02016
Cycle: 2016 Time:1000.19
Z Axis
-10
0
10
20
X Axis
40
60
70
80
Y Axis
0
5
10
15
20
user: puso
Wed Sep 28 12:24:10 2011

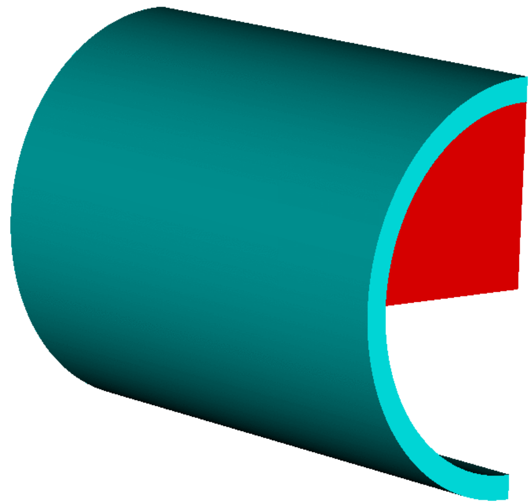

DB: feusion07symm_001.01135
Cycle: 1135 Time:300.519
Filled Boundary
Var: material(mesh_3d)
1 c4_1
3 air_3
4 void_4
Filled Boundary
Var: shellmaterial(mesh_shells)
2
Mesh
Var: mesh_shells
user: puso
Wed Sep 28 13:16:51 2011

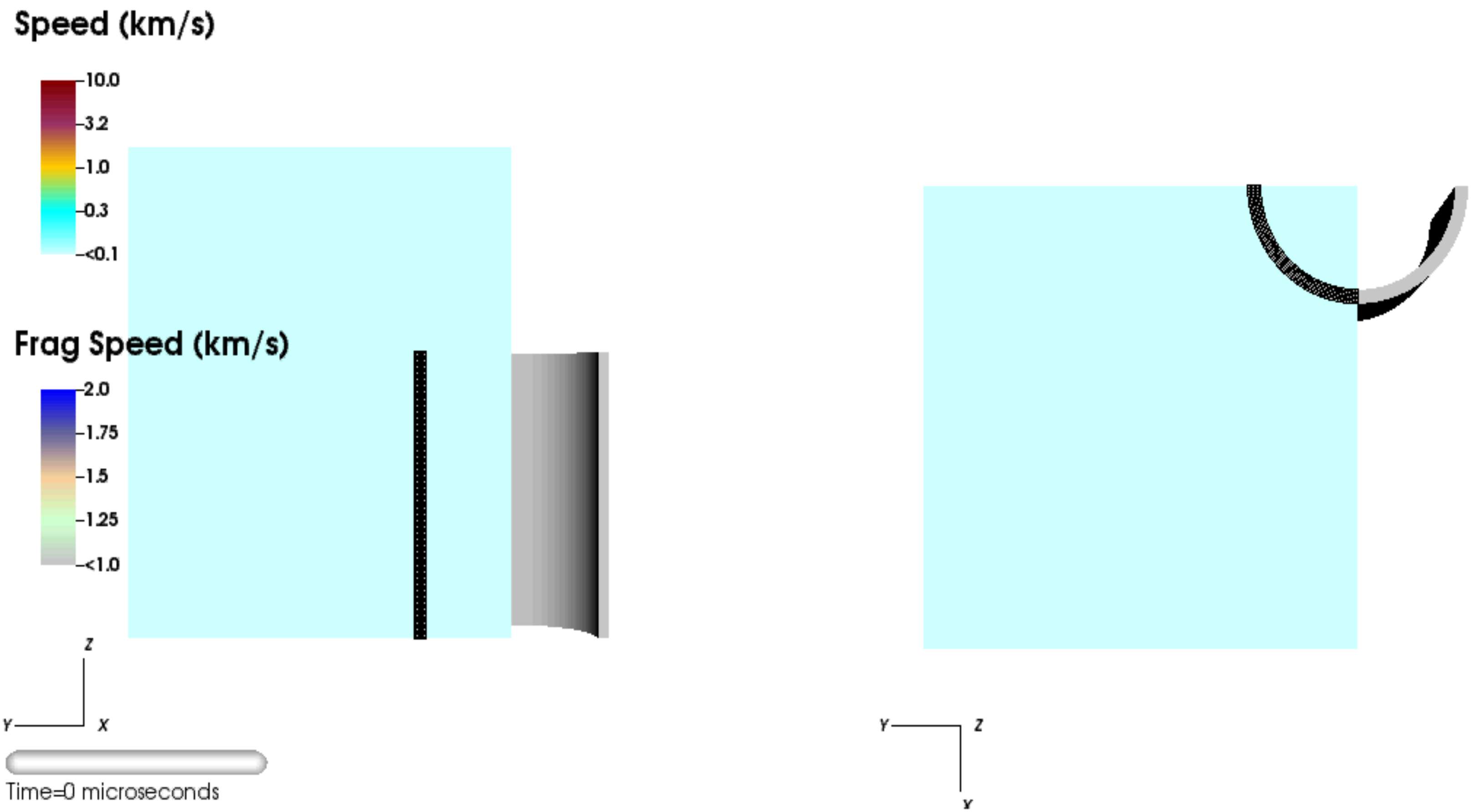

DB: feusion07symm_001.02016
Cycle: 2016 Time:1000.19
Filled Boundary
Var: material(mesh_3d)
1 c4_1
3 air_3
4 void_4
Filled Boundary
Var: shellmaterial(mesh_shells)
2
Mesh
Var: mesh_shells
user: puso
Wed Sep 28 12:37:36 2011

Pipe Bomb: Fragmentation

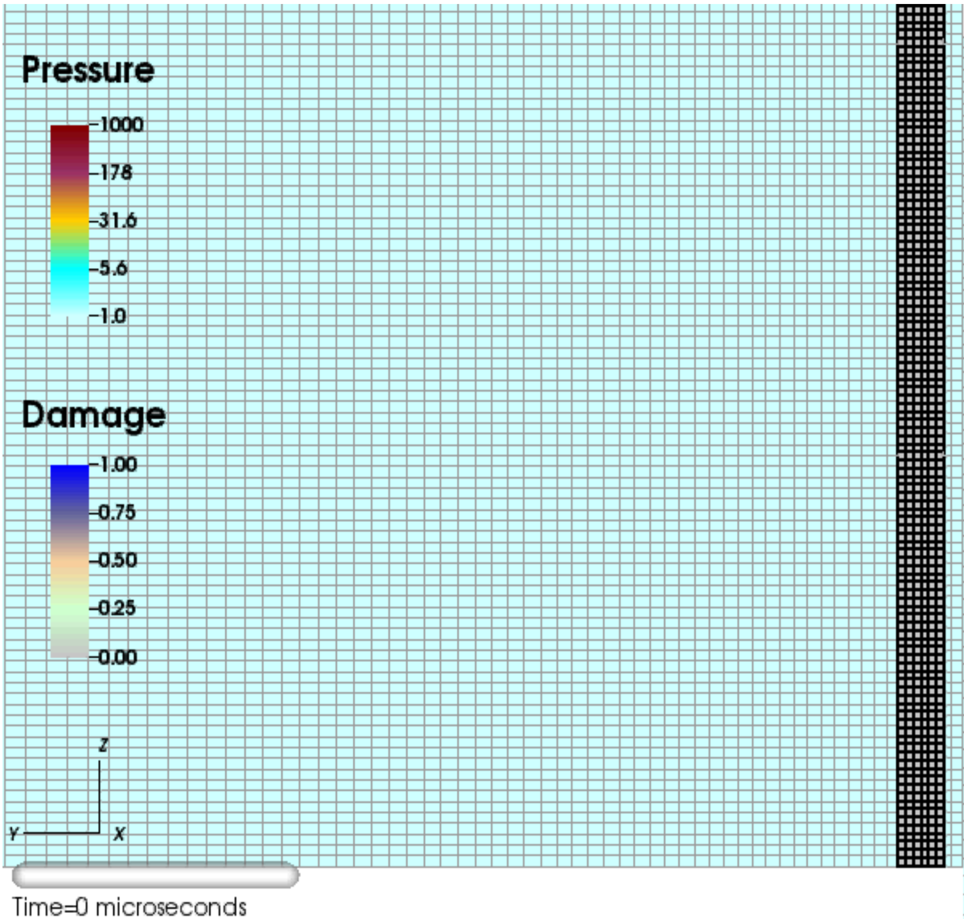


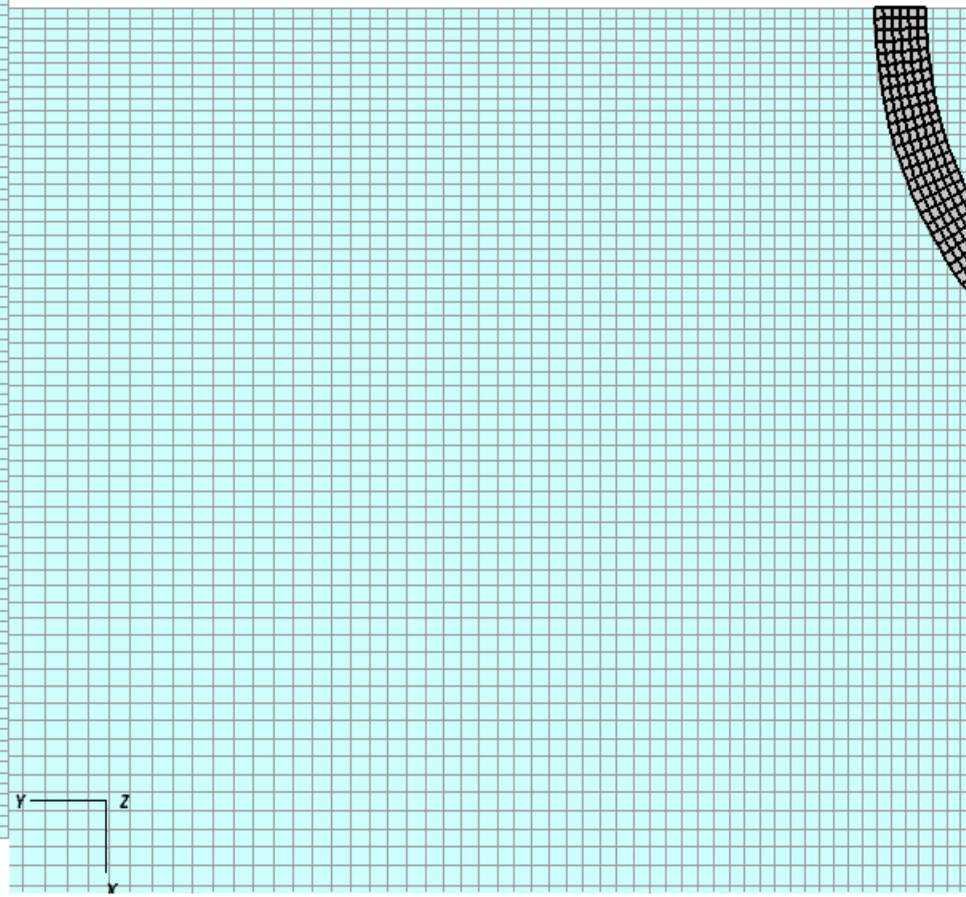


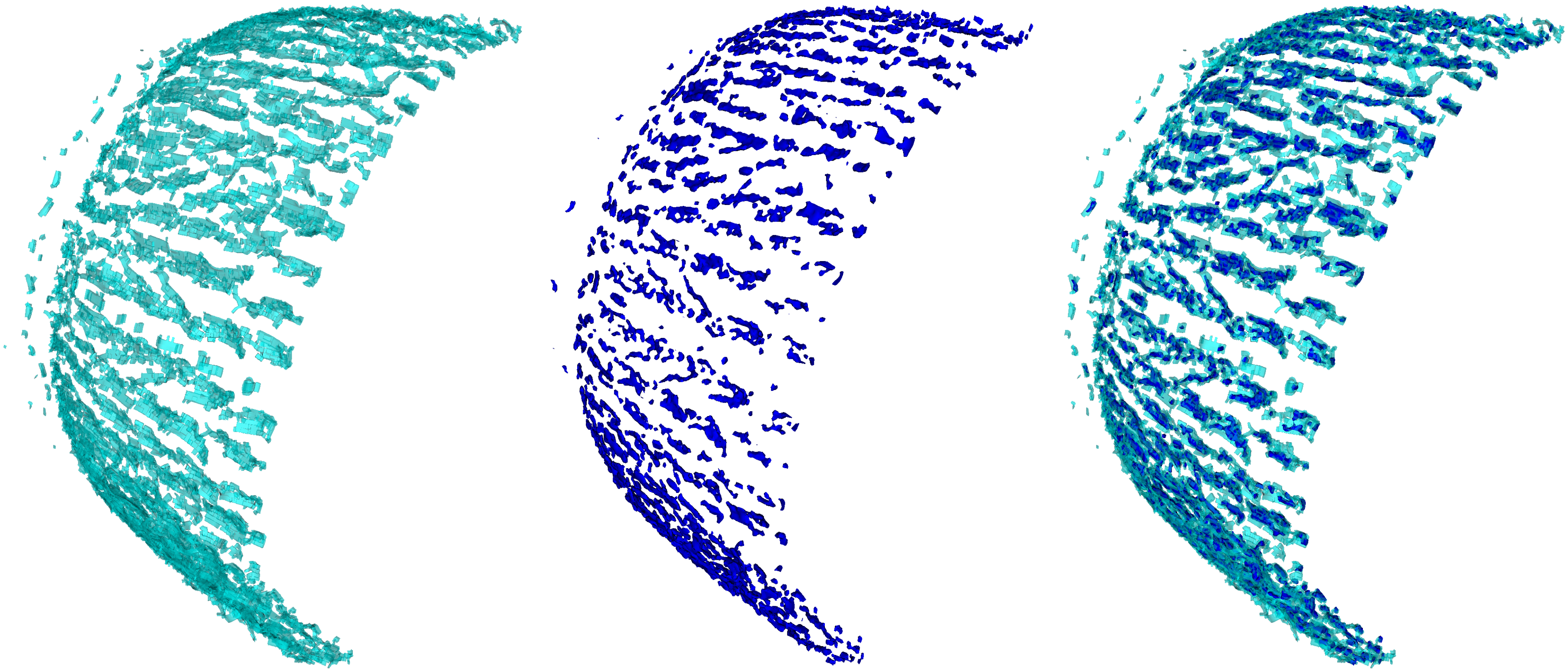




Speed (km/s)
10.0
3.2
1.0
0.3
<0.1
Frag Speed (km/s)
2.0
1.75
1.5
1.25
<1.0
Z
Y
X
Time=0 microseconds
Y
Z
Y


Pressure
1000
178
31.6
5.6
1.0
Damage
1.00
0.75
0.50
0.25
0.00
Z
Y
X
Time=0 microseconds


Y
Z
X

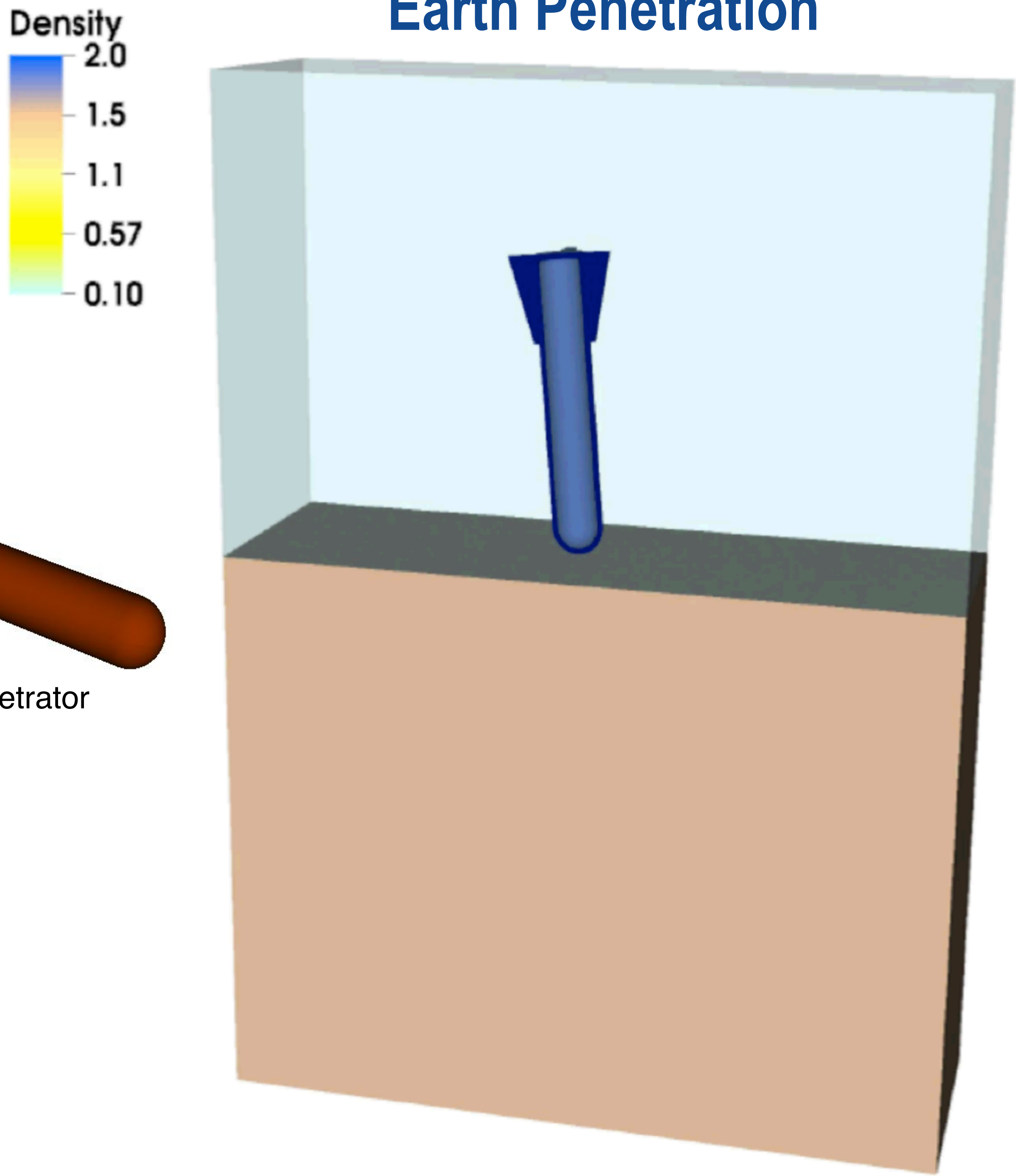


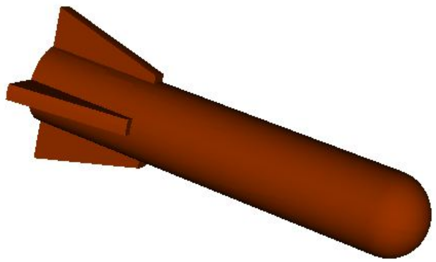
pipe fragments

void material “under” pipe fragments

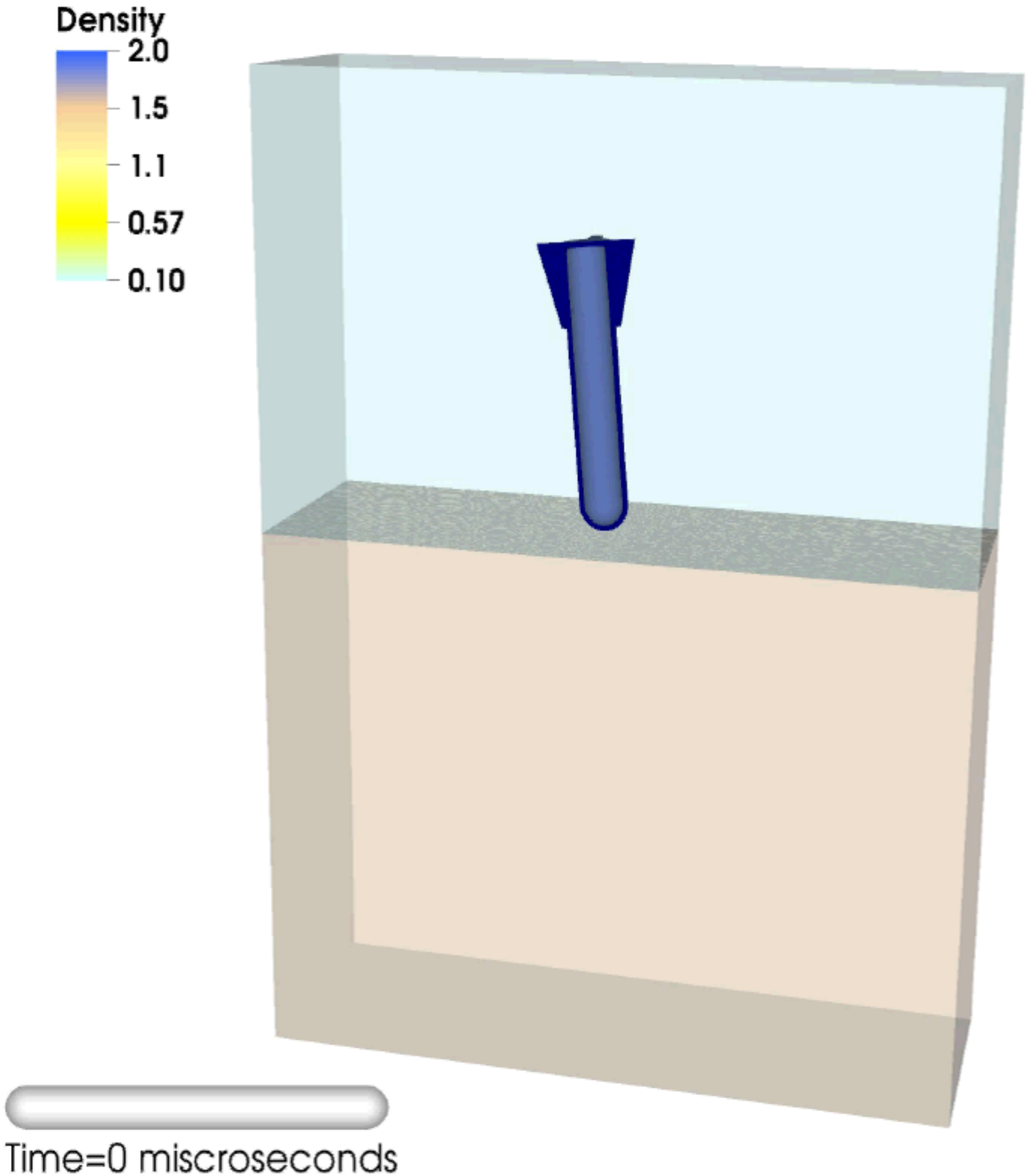
superimposed

Earth Penetration

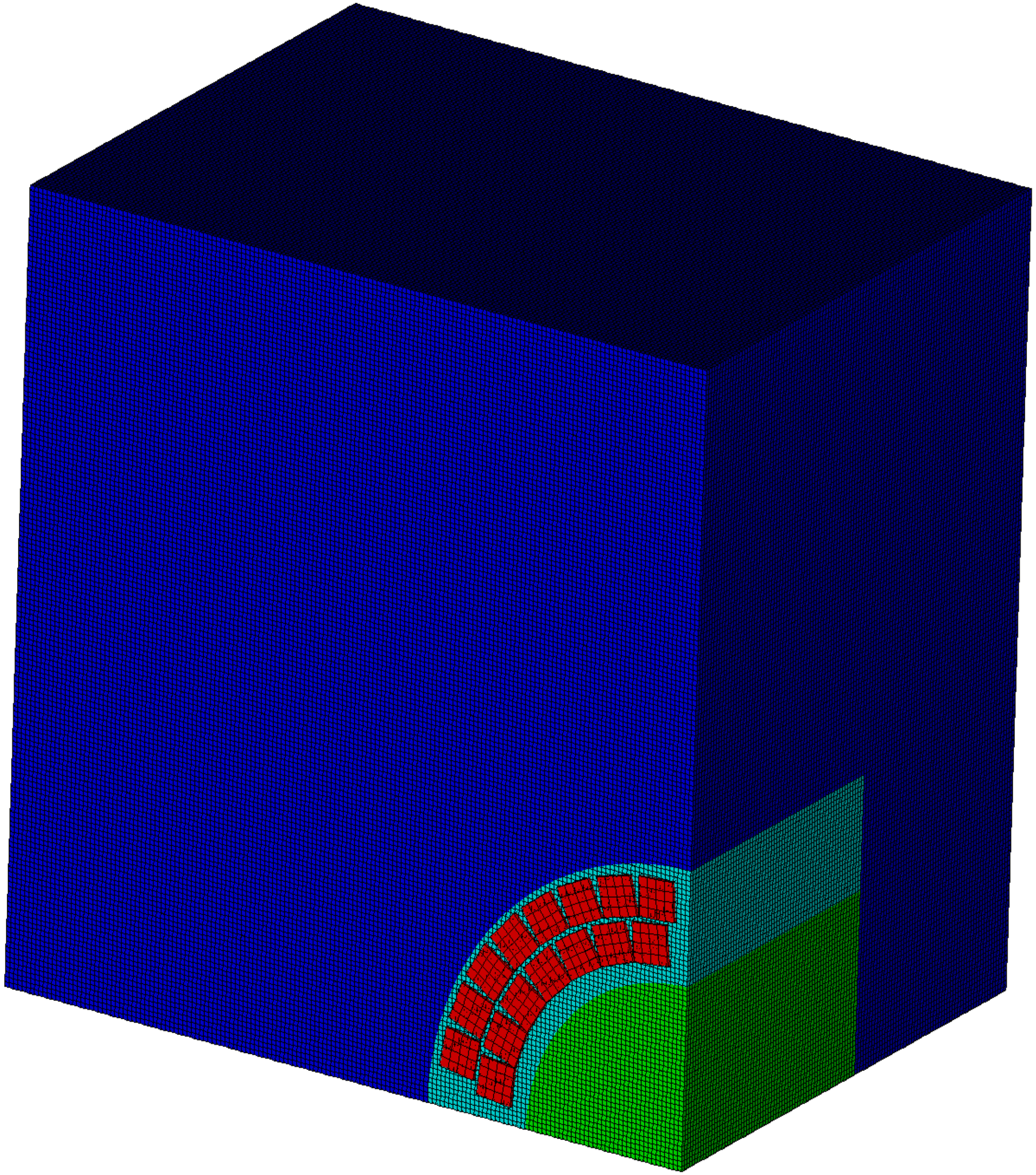


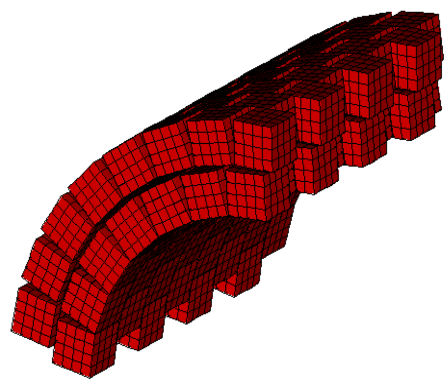


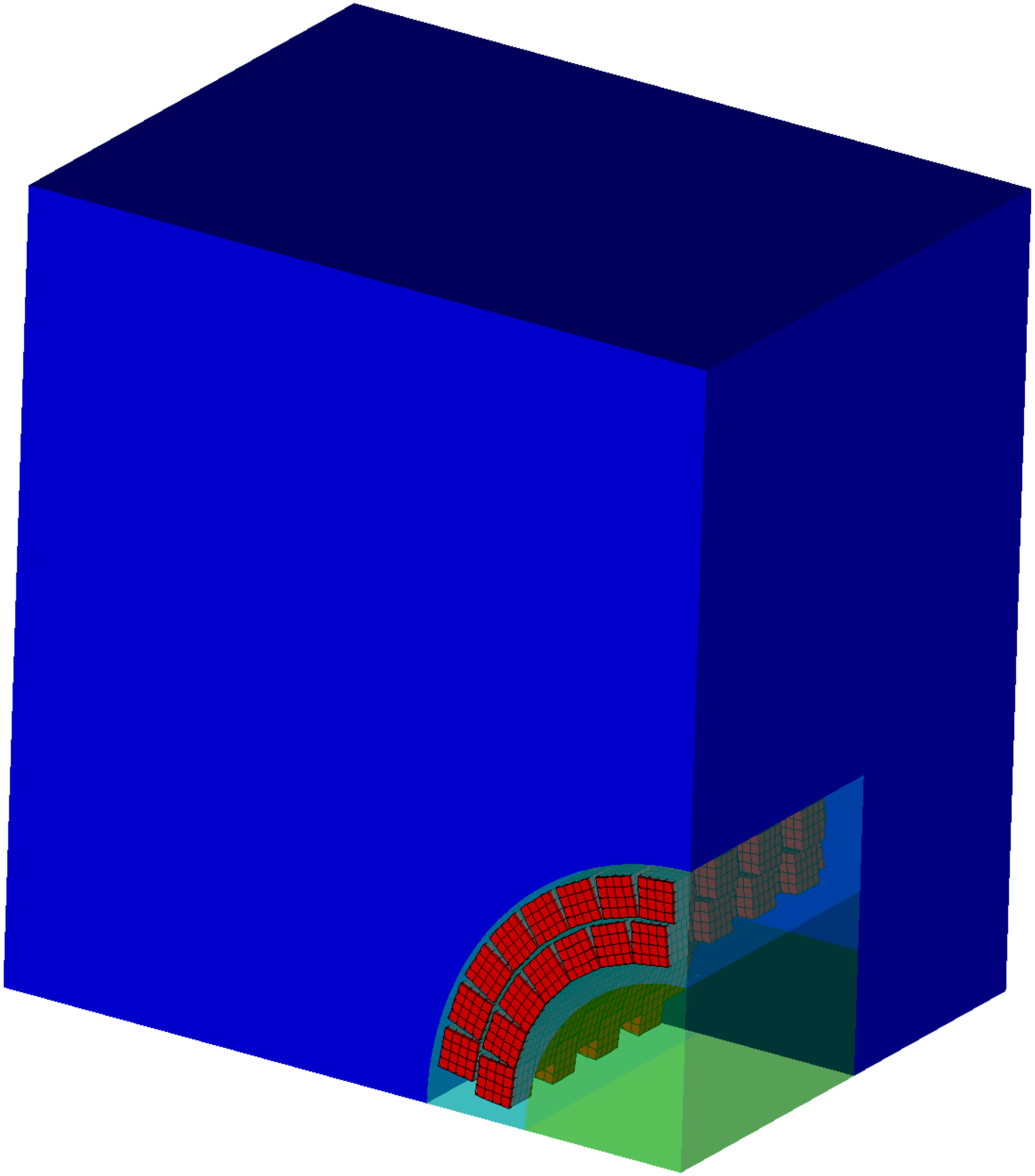
Hollow Penetrator

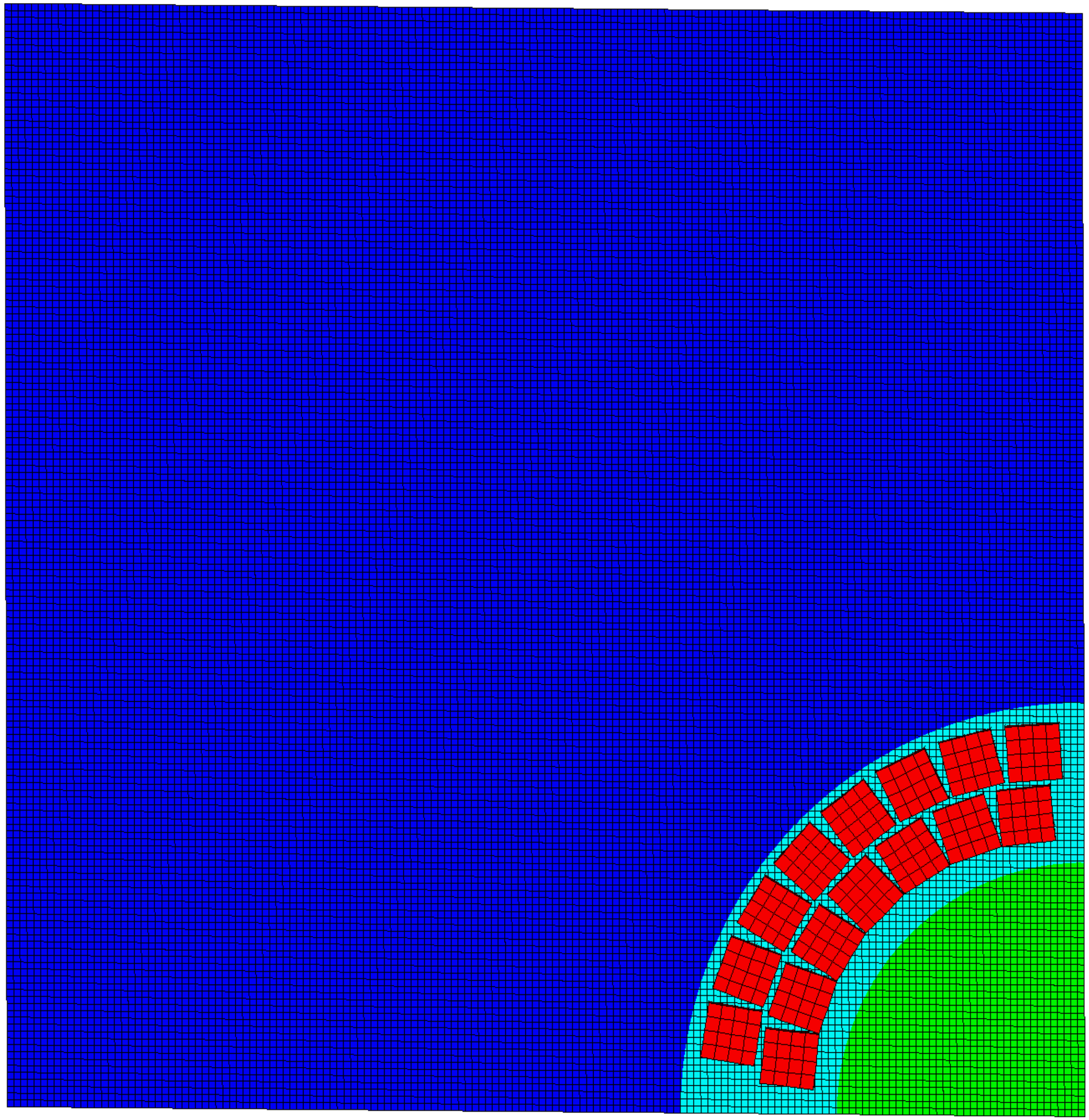


Density
2.0
1.5
1.1
0.57
0.10
Time=0 miscroseconds









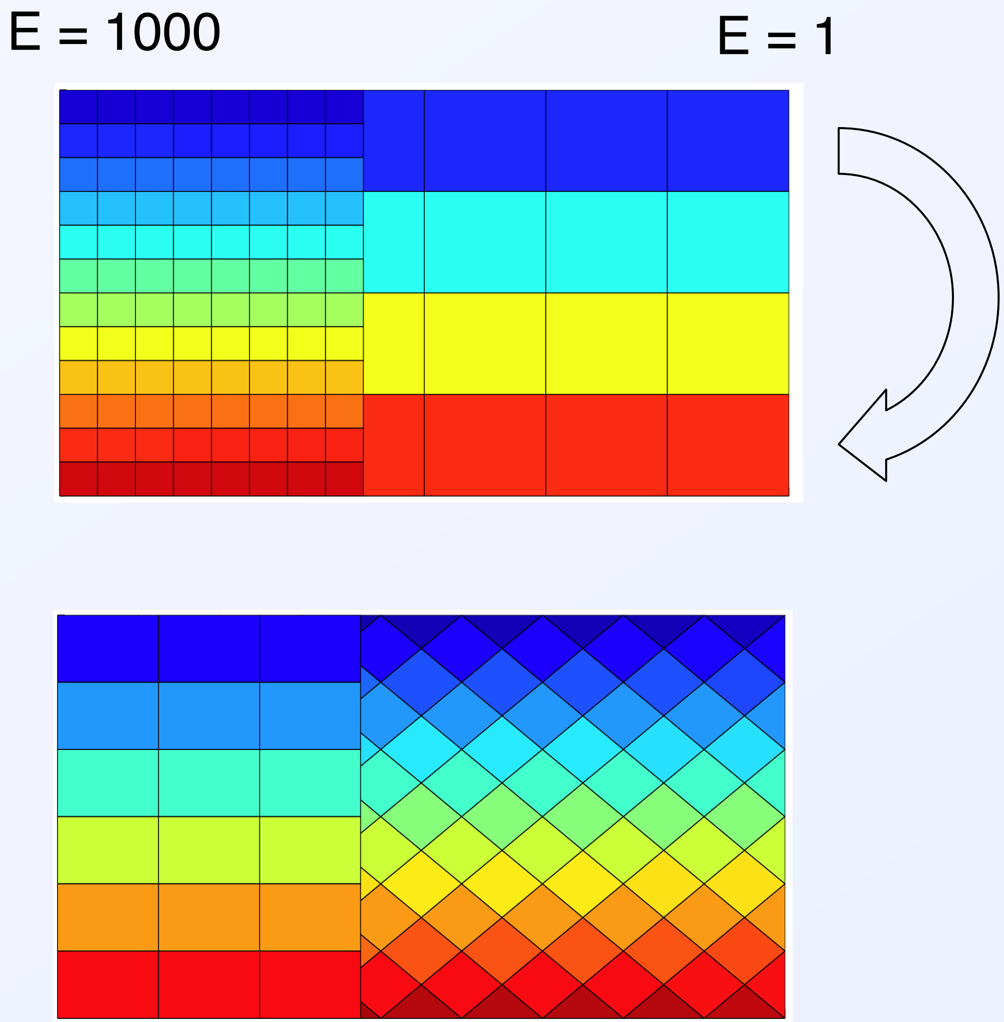
Summary

- Develop embedded mesh method using piecewise constant Lagrange multipliers with stabilization
- Demonstrate stability and convergence of method
- CG solution of Lagrange multipliers solved on decomposed system
- Condition number of system is independent of mesh refinement
- Time step not affected by embedded mesh
- Reasonable energy conservation
- Incorporate method in 2 step ALE approach with *r* adaptivity
- Verify and validate method

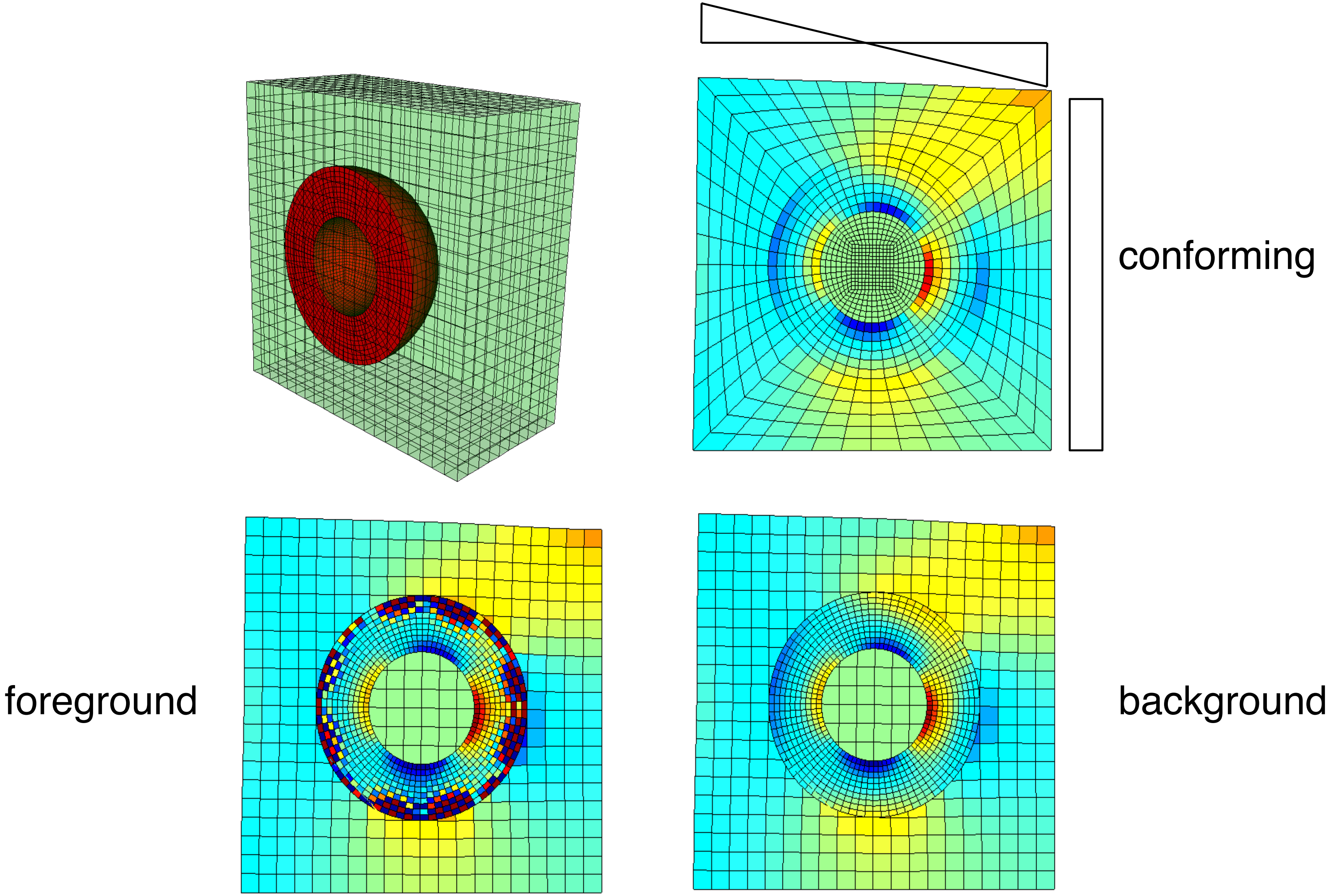
Current/Future work

- Continue verification and validation of approach
- Improve parallel scaling
  - Better dynamic domain decomposition
- Provide analysis for estimates of convergence rates
- Currently adding XFEM into foreground mesh

Results good for many cases, consider beam



Multipliers on background mesh: 3D result



ALE implementation

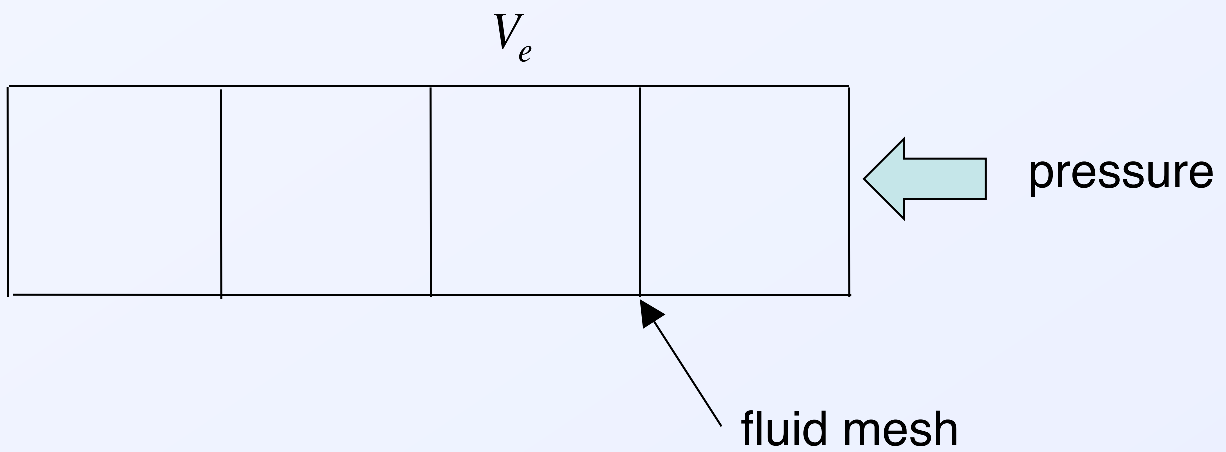
- Use central difference explicit 2 step ALE approach

Load step

Lagrange step

Advection remap step

Advance time step $t_n \rightarrow t_{n+1}$

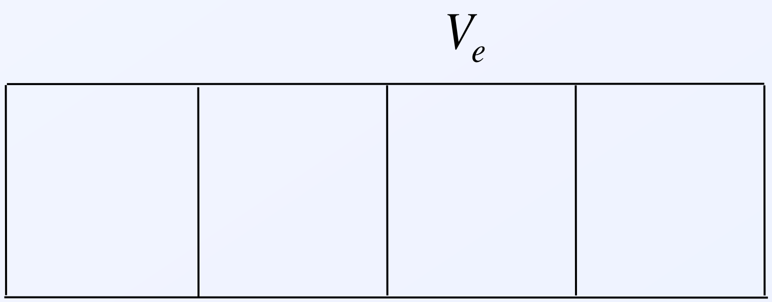


$V_e \equiv$ Reference volume of fluid (i.e. element volume/density)

ALE implementation

- Use central difference explicit 2 step ALE approach

  - Load step
  - Lagrange step
  - Advection remap step
  - Advance time step $t_n \rightarrow t_{n+1}$

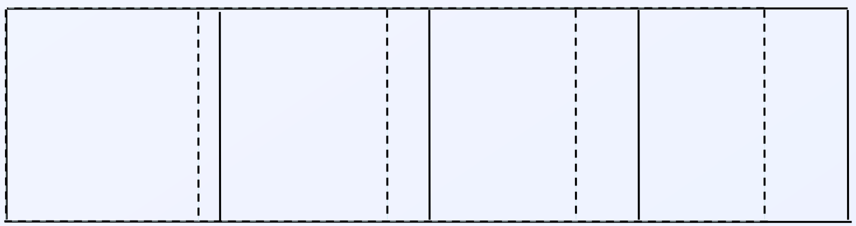


ALE implementation

- Use central difference explicit 2 step ALE approach

  - Load step
  - Lagrange step
  - Advection remap step (yields volume flux)
  - Advance time step $t_n \rightarrow t_{n+1}$

$$V_e - \Delta V_e^L + \Delta V_e^R$$



ΔV_e^L ΔV_e^R

ALE implementation: Multiple Materials

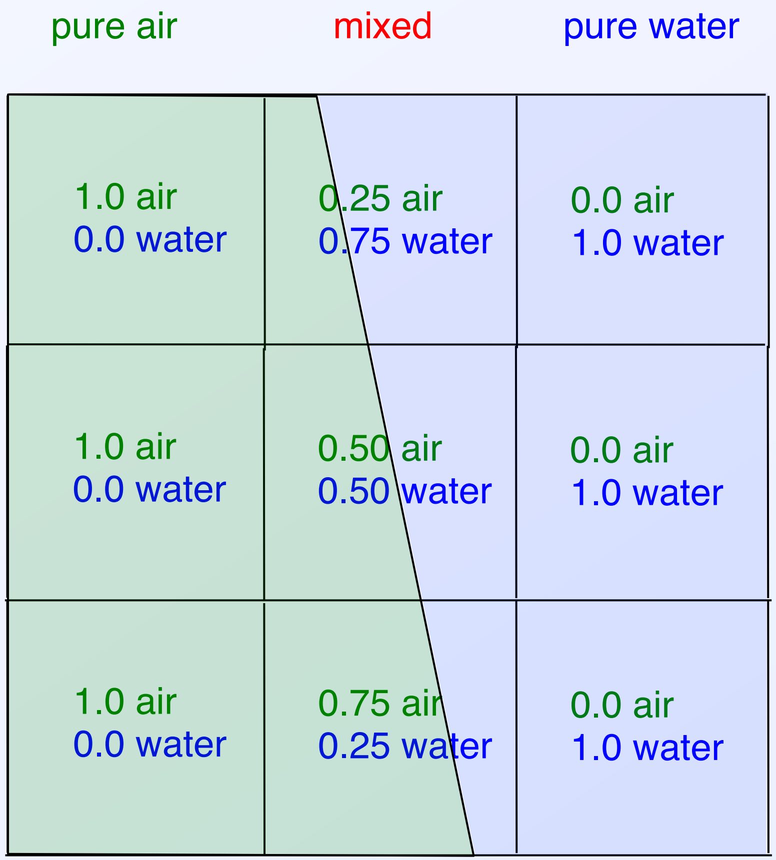
- Young's Interface Reconstruction to compute PWL approximation of interface

Consider Cell Volume Fractions (VOF)

pure air	mixed	pure water
1.0 air 0.0 water	0.25 air 0.75 water	0.0 air 1.0 water
1.0 air 0.0 water	0.50 air 0.50 water	0.0 air 1.0 water
1.0 air 0.0 water	0.75 air 0.25 water	0.0 air 1.0 water

ALE implementation: Multiple Materials

- Can typically recover a reasonable approximation of interface when smooth

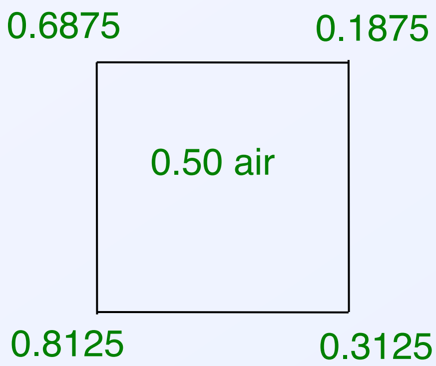


ALE implementation: Young's Interface Reconstruction

- Compute nodal VOF from cell VOF
- Assume a linear approximation to material VOF and apply Least Squares for 2D case below

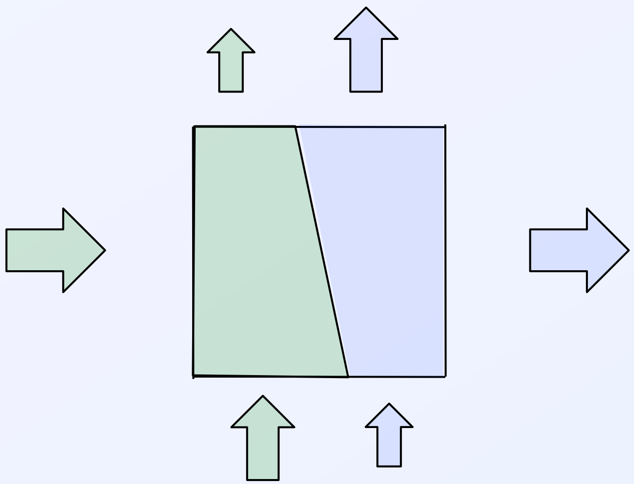
$$VOF(\boldsymbol{x}) \approx ax + by - d$$

$$E = \sum_{i=1}^{4} [VOF(\boldsymbol{x}_i) - (VOF_i - 0.5)]^2$$



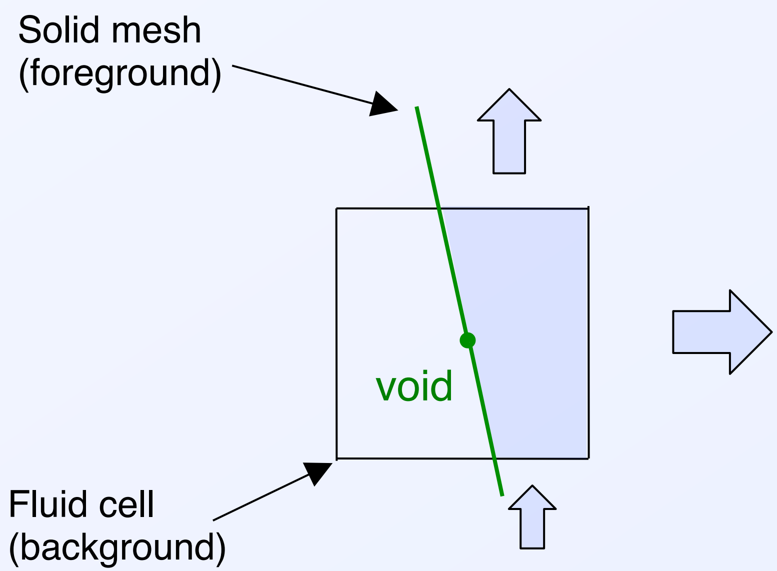
ALE implementation: Young's Interface Reconstruction

- Advection remap step determines flux volumes of each face ΔV_e^f
- Interface determines fractions of material in flux volume



ALE implementation: Young's Interface Reconstruction

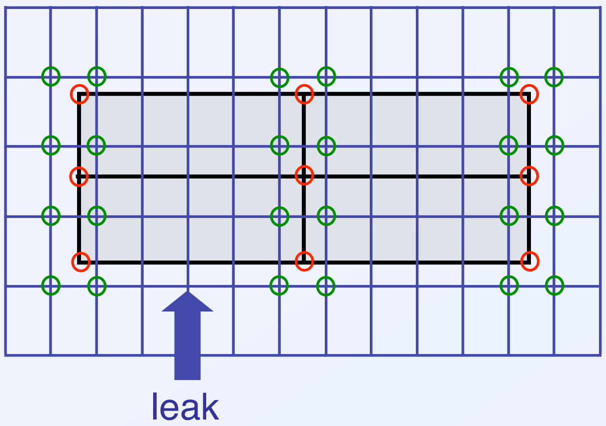
- Modification for embedded mesh: interface determined from solid mesh
- No volume fluxes on "void" side



Previous Works

- Existing *Embedded Mesh* methods for *moving meshes*
  - *Immersed boundary methods* (C.S. Peskin 1977, 2002)
    - Finite difference fluid with membranes
    - $h_s << h_f$ otherwise leaks i.e., not consistent
  - *Immersed finite element methods* (W.K. Liu 2004)
    - Enforces constraints "point-wise" between solid mesh fluid mesh
    - $h_s << h_f$ otherwise leaks i.e., not consistent

e.g. $h_s > h_f$



Previous Works

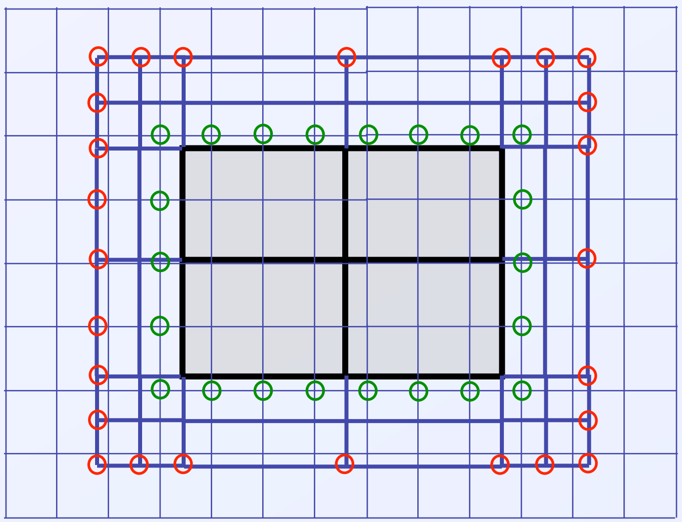
- Existing *Embedded Mesh* methods for *moving meshes*
  - *Overset grid methods* (W.D. Henshaw 2006)
    - Finite differences/volume for structured grids
    - Momentum/Flux not conserved across boundary
    - Lose symmetry of [K] where [K]$u = f$
    - Difficult to implement for shells

Attach conforming overset fluid grid to solid and tie fluid grids

Red “ghost” points collocated to outside grid

Green “ghost” points collocated to inside grid

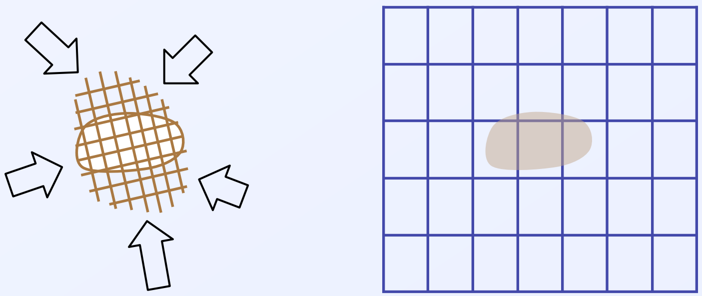
Ignore force balance at “ghost” points



Previous Works

- Existing *Embedded Mesh* methods for *moving meshes*
  - *Zapotec method* (Bessette 2002)
    - Based on material insertion/force compution
    - Mainly designed for Lagrangian-Eulerian coupling
    - Scheme is not monolithic
    - Cumbersome for Lagrangian shell
    - Requires material insertion everywhere

1. Update Lagrangian body based on current surface loads
2. “Insert” Lagrangian material, velocities, densities etc. into Eulerian cells
3. Update Eulerian velocities
4. Project Eulerian fluid stress onto Lagrangian surface



Previous Works

- Existing *Embedded Mesh* methods for *moving meshes*
  - *Mortar fictitious domain methods* (Baaijens 2001)
    - 2D implementation for finite elements
    - No Leaks, Conserves momentum, Retains symmetry
    - Requires surface integral (difficult, but tractable)
    - Requires solution to system of equations for constraint (bad for explicit!)

Apply surface integral to constrain fluid and solid surface velocities

$$\int_{\Gamma} \boldsymbol{\lambda} \cdot (\boldsymbol{v}^s - \boldsymbol{v}^f)\, d\Gamma = 0$$