Lecture 6

Fluid-structure interaction simulation

Course *Block-structured Adaptive Finite Volume Methods in C++*

Ralf Deiterding
University of Southampton
Engineering and the Environment
Highfield Campus, Southampton SO17 1BJ, UK

E-mail: r.deiterding@soton.ac.uk

Outline

Outline

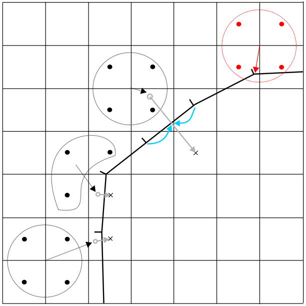
Outline

Construction of coupling data

- Moving boundary/interface is treated as a moving contact discontinuity and represented by level set [Fedkiw, 2002][Arienti et al., 2003]
- Efficient construction of level set from triangulated surface data with closest-point-transform (CPT) algorithm [Mauch, 2003]

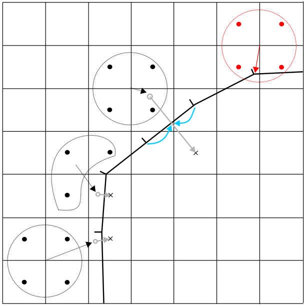
Construction of coupling data

- Moving boundary/interface is treated as a moving contact discontinuity and represented by level set [Fedkiw, 2002][Arienti et al., 2003]
- Efficient construction of level set from triangulated surface data with closest-point-transform (CPT) algorithm [Mauch, 2003]
- One-sided construction of mirrored ghost cell and new nodal point values



Construction of coupling data

- Moving boundary/interface is treated as a moving contact discontinuity and represented by level set [Fedkiw, 2002][Arienti et al., 2003]
- Efficient construction of level set from triangulated surface data with closest-point-transform (CPT) algorithm [Mauch, 2003]
- One-sided construction of mirrored ghost cell and new nodal point values
- Gathering of solid force and momentum information and solution of equations of motion on central node



Coupling conditions on interface
Viscous fluid:

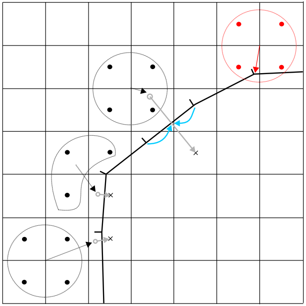
$$\left.\begin{array}{ccc} u^S & = & u^F \\ \sigma^S_{nm} & = & \sigma^F_{nm} \end{array}\right|_{\mathcal{I}}$$

with $\sigma^F_{nm} = -p^F \delta_{nm} + \Sigma^F_{nm}$

Construction of coupling data

- Moving boundary/interface is treated as a moving contact discontinuity and represented by level set [Fedkiw, 2002][Arienti et al., 2003]
- Efficient construction of level set from triangulated surface data with closest-point-transform (CPT) algorithm [Mauch, 2003]
- One-sided construction of mirrored ghost cell and new nodal point values
- Gathering of solid force and momentum information and solution of equations of motion on central node
- Stable explicit coupling possible if geometry and velocities are prescribed for compressible fluid [Specht, 2000]

$u^F := u^S(t)|_{\mathcal{I}}$
`UpdateFluid(` Δt `)`
$\sigma^S_{nm} := \sigma^F_{nm}(t+\Delta t)|_{\mathcal{I}}$
`UpdateSolid(` Δt `)`
$t := t + \Delta t$



Coupling conditions on interface
Viscous fluid:

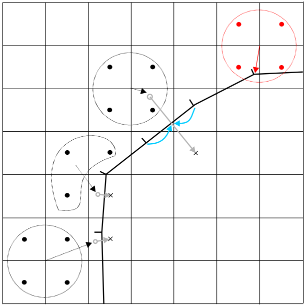
$$\left.\begin{array}{rcl} u^S & = & u^F \\ \sigma^S_{nm} & = & \sigma^F_{nm} \end{array}\right|_{\mathcal{I}}$$

with $\sigma^F_{nm} = -p^F \delta_{nm} + \Sigma^F_{nm}$

Construction of coupling data

- Moving boundary/interface is treated as a moving contact discontinuity and represented by level set [Fedkiw, 2002][Arienti et al., 2003]

- Efficient construction of level set from triangulated surface data with closest-point-transform (CPT) algorithm [Mauch, 2003]
- One-sided construction of mirrored ghost cell and new nodal point values
- Gathering of solid force and momentum information and solution of equations of motion on central node
- Stable explicit coupling possible if geometry and velocities are prescribed for compressible fluid [Specht, 2000]



$u_n^F := u_n^S(t)|_{\mathcal{I}}$
`UpdateFluid(` Δt `)`
$\sigma_{nm}^S := -p^F(t+\Delta t)\delta_{nm}|_{\mathcal{I}}$
`UpdateSolid(` Δt `)`
$t := t + \Delta t$

Coupling conditions on interface
Inviscid fluid (for FV methods):

$$\left. \begin{aligned} u_n^S &= u_n^F \\ \sigma_{nm}^S &= -p^F \delta_{nm} \end{aligned} \right|_{\mathcal{I}}$$

Usage of SAMR

- Eulerian SAMR + non-adaptive Lagrangian FEM scheme
- Exploit SAMR time step refinement for effective coupling to solid solver

Usage of SAMR

- Eulerian SAMR + non-adaptive Lagrangian FEM scheme
- Exploit SAMR time step refinement for effective coupling to solid solver
  - Lagrangian simulation is called only at level $l_c \leq l_{\max}$

Usage of SAMR

- Eulerian SAMR + non-adaptive Lagrangian FEM scheme
- Exploit SAMR time step refinement for effective coupling to solid solver
  - Lagrangian simulation is called only at level $l_c \leq l_{\max}$
  - SAMR refines solid boundary at least at level l_c
  - Additional levels can be used resolve geometric ambiguities

Usage of SAMR

- Eulerian SAMR + non-adaptive Lagrangian FEM scheme
- Exploit SAMR time step refinement for effective coupling to solid solver
  - Lagrangian simulation is called only at level $l_c \leq l_{\max}$
  - SAMR refines solid boundary at least at level l_c
  - Additional levels can be used resolve geometric ambiguities
- Nevertheless: Inserting sub-steps accommodates for time step reduction from the solid solver within an SAMR cycle

Usage of SAMR

- Eulerian SAMR + non-adaptive Lagrangian FEM scheme
- Exploit SAMR time step refinement for effective coupling to solid solver
  - Lagrangian simulation is called only at level $l_c \leq l_{\max}$
  - SAMR refines solid boundary at least at level l_c
  - Additional levels can be used resolve geometric ambiguities
- Nevertheless: Inserting sub-steps accommodates for time step reduction from the solid solver within an SAMR cycle
- Communication strategy:
  - Updated boundary info from solid solver must be received before regridding operation
  - Boundary data is sent to solid when highest level available

Usage of SAMR

- Eulerian SAMR + non-adaptive Lagrangian FEM scheme
- Exploit SAMR time step refinement for effective coupling to solid solver
  - Lagrangian simulation is called only at level $l_c \le l_{\max}$
  - SAMR refines solid boundary at least at level l_c
  - Additional levels can be used resolve geometric ambiguities
- Nevertheless: Inserting sub-steps accommodates for time step reduction from the solid solver within an SAMR cycle
- Communication strategy:
  - Updated boundary info from solid solver must be received before regridding operation
  - Boundary data is sent to solid when highest level available
- Inter-solver communication (point-to-point or globally) managed on the fly special coupling module

SAMR algorithm for FSI coupling

```
AdvanceLevel(l)

 Repeat r_l times
   Set ghost cells of Q^l(t)

   If time to regrid?
       Regrid(l)
   UpdateLevel(l)
   If level l+1 exists?
       Set ghost cells of Q^l(t+Δt_l)
       AdvanceLevel(l+1)
       Average Q^{l+1}(t+Δt_l) onto Q^l(t+Δt_l)




   t := t+Δt_l
```

SAMR algorithm for FSI coupling

```
AdvanceLevel(l)

 Repeat r_l times
   Set ghost cells of Q^l(t)
   CPT(φ^l, C^l, I, δ_l)
   If time to regrid?
       Regrid(l)
   UpdateLevel(Q^l, φ^l, C^l, u^S|_I, Δt_l)
   If level l+1 exists?
       Set ghost cells of Q^l(t+Δt_l)
       AdvanceLevel(l+1)
       Average Q^{l+1}(t+Δt_l) onto Q^l(t+Δt_l)

   t := t+Δt_l
```

- Call CPT algorithm before `Regrid(l)`
- Include also call to `CPT(·)` into `Recompose(l)` to ensure consistent level set data on levels that have changed

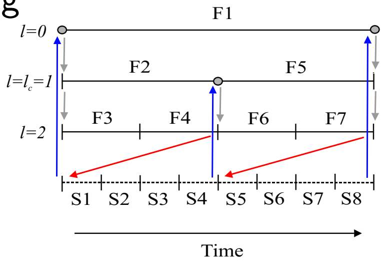
SAMR algorithm for FSI coupling

```
AdvanceLevel(l)

 Repeat r_l times
   Set ghost cells of Q^l(t)
   CPT(φ^l, C^l, I, δ_l)
   If time to regrid?
       Regrid(l)
   UpdateLevel(Q^l, φ^l, C^l, u^S|_I, Δt_l)
   If level l+1 exists?
       Set ghost cells of Q^l(t+Δt_l)
       AdvanceLevel(l+1)
       Average Q^{l+1}(t+Δt_l) onto Q^l(t+Δt_l)
   If l = l_c?
       SendInterfaceData(p^F(t+Δt_l)|_I)
       If (t+Δt_l) < (t_0+Δt_0)?
           ReceiveInterfaceData(I, u^S|_I)
   t := t+Δt_l
```

- Call CPT algorithm before `Regrid(l)`
- Include also call to `CPT(·)` into `Recompose(l)` to ensure consistent level set data on levels that have changed
- Communicate boundary data on coupling level l_c

SAMR algorithm for FSI coupling



AdvanceLevel(l)

Repeat r_l times
- Set ghost cells of $\mathbf{Q}^l(t)$
- CPT(φ^l, C^l, $\mathcal{I}$, δ_l)
- If time to regrid?
  - Regrid(l)
- UpdateLevel($\mathbf{Q}^l$, φ^l, C^l, $\mathbf{u}^S|_\mathcal{I}$, Δt_l)
- If level $l+1$ exists?
  - Set ghost cells of $\mathbf{Q}^l(t+\Delta t_l)$
  - AdvanceLevel($l+1$)
  - Average $\mathbf{Q}^{l+1}(t+\Delta t_l)$ onto $\mathbf{Q}^l(t+\Delta t_l)$
- If $l = l_c$?
  - SendInterfaceData($p^F(t+\Delta t_l)|_\mathcal{I}$)
  - If $(t+\Delta t_l) < (t_0+\Delta t_0)$?
    - ReceiveInterfaceData($\mathcal{I}$, $\mathbf{u}^S|_\mathcal{I}$)
- $t := t + \Delta t_l$

- Call CPT algorithm before Regrid(l)
- Include also call to CPT($\cdot$) into Recompose(l) to ensure consistent level set data on levels that have changed
- Communicate boundary data on coupling level l_c

Fluid and solid update / exchange of time steps

`FluidStep( )`

$\Delta\tau_F := \min_{l=0,\cdots,l_{\max}} (R_l \cdot \texttt{ StableFluidTimeStep}(l)\texttt{, } \Delta\tau_S)$

$\Delta t_l := \Delta\tau_F / R_l$ `for` $l = 0, \cdots, L$

$\texttt{ReceiveInterfaceData}(\mathcal{I}\texttt{, } \mathbf{u}^S|_{\mathcal{I}})$

`AdvanceLevel(0)`

with $R_l = \prod_{\iota=0}^{l} r_\iota$

Fluid and solid update / exchange of time steps

`FluidStep( )`

$\Delta\tau_F := \min\limits_{l=0,\cdots,l_{\max}} (R_l \cdot$ `StableFluidTimeStep(`l`)`, $\Delta\tau_S)$

$\Delta t_l := \Delta\tau_F / R_l$ for $l = 0, \cdots, L$

`ReceiveInterfaceData(`$\mathcal{I}$, $\mathbf{u}^S|_{\mathcal{I}}$`)`

`AdvanceLevel(0)`

`SolidStep( )`

$\Delta\tau_S := \min(K \cdot R_{l_c} \cdot$ `StableSolidTimeStep()`, $\Delta\tau_F)$

with $R_l = \prod_{\iota=0}^{l} r_\iota$

Fluid and solid update / exchange of time steps

`FluidStep( )`

$\Delta\tau_F := \min_{l=0,\cdots,l_{\max}} (R_l \cdot$ `StableFluidTimeStep(`l`)`, $\Delta\tau_S)$
$\Delta t_l := \Delta\tau_F / R_l$ `for` $l = 0, \cdots, L$
`ReceiveInterfaceData(`$\mathcal{I}$, $\mathbf{u}^S|_{\mathcal{I}}$`)`
`AdvanceLevel(0)`

`SolidStep( )`

$\Delta\tau_S := \min(K \cdot R_{l_c} \cdot$ `StableSolidTimeStep()`, $\Delta\tau_F)$
`Repeat` R_{l_c} `times`
$t_{\mathrm{end}} := t + \Delta\tau_S / R_{l_c}$, $\Delta t := \Delta\tau_S / (K R_{l_c})$

- Time step stays constant for R_{l_c} steps, which correponds to one fluid step at level 0

with $R_l = \prod_{\iota=0}^{l} r_\iota$

Fluid and solid update / exchange of time steps

`FluidStep( )`

$\Delta\tau_F := \min\limits_{l=0,\cdots,l_{\max}} (R_l \cdot \texttt{StableFluidTimeStep}(l)\texttt{,}\ \Delta\tau_S)$
$\Delta t_l := \Delta\tau_F / R_l$ `for` $l = 0, \cdots, L$
$\texttt{ReceiveInterfaceData}(\mathcal{I}\texttt{,}\ \mathbf{u}^S|_{\mathcal{I}})$
`AdvanceLevel(0)`

`SolidStep( )`

$\Delta\tau_S := \min(K \cdot R_{l_c} \cdot \texttt{StableSolidTimeStep()}\texttt{,}\ \Delta\tau_F)$
`Repeat` R_{l_c} `times`
$\quad t_{\rm end} := t + \Delta\tau_S / R_{l_c}$, $\Delta t := \Delta\tau_S / (K R_{l_c})$
$\quad$`While` $t < t_{\rm end}$
$\qquad \texttt{SendInterfaceData}(\mathcal{I}(t)\texttt{,}\ \vec{u}^S|_{\mathcal{I}}(t))$
$\qquad \texttt{ReceiveInterfaceData}(p^F|_{\mathcal{I}})$
$\qquad \texttt{UpdateSolid}(p^F|_{\mathcal{I}}\texttt{,}\ \Delta t)$
$\qquad t := t + \Delta t$
$\qquad \Delta t := \min(\texttt{StableSolidTimeStep()}\texttt{,}\ t_{\rm end} - t)$

with $R_l = \prod_{\iota=0}^{l} r_\iota$

- Time step stays constant for R_{l_c} steps, which correponds to one fluid step at level 0

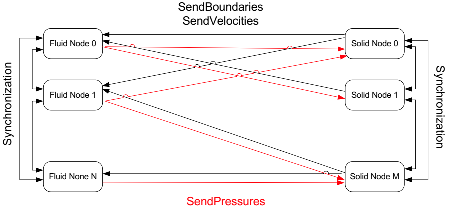
Parallelization strategy for coupled simulations

Coupling of an Eulerian FV fluid Solver and a Lagrangian FEM Solver:

Parallelization strategy for coupled simulations

Coupling of an Eulerian FV fluid Solver and a Lagrangian FEM Solver:

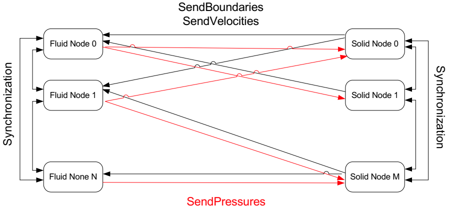
- Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes
- Setting of ghost cell values becomes strictly local operation



Parallelization strategy for coupled simulations

Coupling of an Eulerian FV fluid Solver and a Lagrangian FEM Solver:

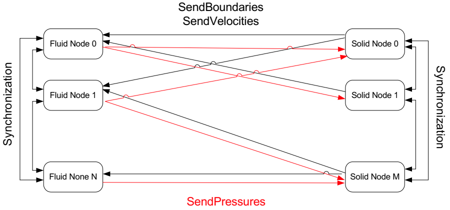
- Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes
- Setting of ghost cell values becomes strictly local operation
- Construct new nodal values strictly local on fluid nodes and transfer them back to solid nodes
- Only surface data is transfered



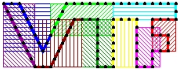
Parallelization strategy for coupled simulations

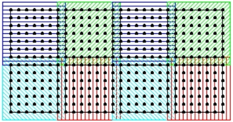
Coupling of an Eulerian FV fluid Solver and a Lagrangian FEM Solver:

- Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes
- Setting of ghost cell values becomes strictly local operation
- Construct new nodal values strictly local on fluid nodes and transfer them back to solid nodes
- Only surface data is transfered
- Asynchronous communication ensures scalability
- Generic encapsulated implementation guarantees reusability



Eulerian/Lagrangian communication module

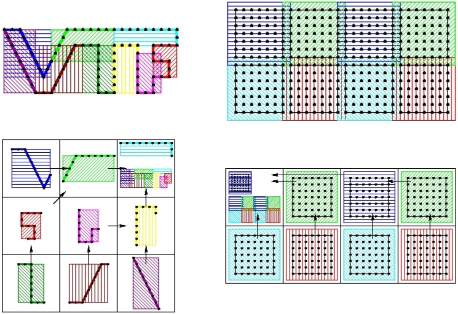




1. Put bounding boxes around each solid processors piece of the boundary and around each fluid processors grid

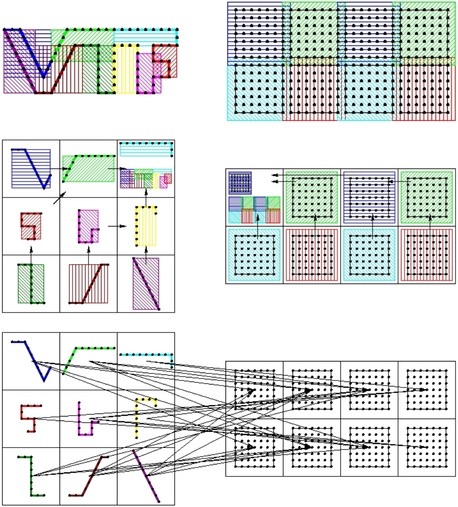
Eulerian/Lagrangian communication module

1. Put bounding boxes around each solid processors piece of the boundary and around each fluid processors grid
2. Gather, exchange and broadcast of bounding box information

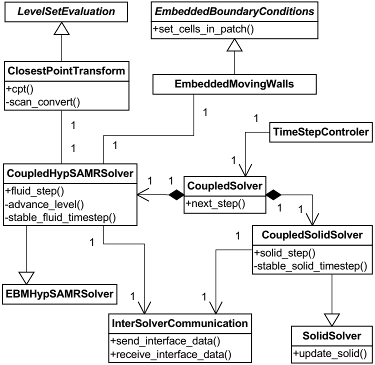


Eulerian/Lagrangian communication module

1. Put bounding boxes around each solid processors piece of the boundary and around each fluid processors grid
2. Gather, exchange and broadcast of bounding box information
3. Optimal point-to-point communication pattern, non-blocking

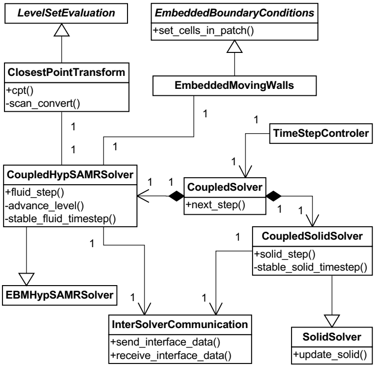


FSI coupling



- Coupling algorithm implemented in further derived HypSAMRSolver class
- Level set evaluation always with CPT algorithm
- Parallel communication through efficient non-blocking communication module ELC
- Time step selection for both solvers through CoupledSolver class

FSI coupling



- Coupling algorithm implemented in further derived HypSAMRSolver class
- Level set evaluation always with CPT algorithm
- Parallel communication through efficient non-blocking communication module ELC
- Time step selection for both solvers through CoupledSolver class

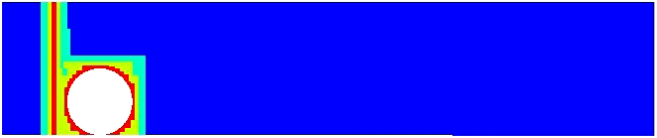
- **AMRELCGFMSolver**$<$**VectorType, FixupType, FlagType, dim** $>$ is the derived AMRSolver$<>$class. `code/amroc/doc/html/amr/classAMRELCGFMSolver.html`
- Uses the Eulerian interface of the Lagrangian communication routines `code/stlib/doc/html/elc/elc__page.html`
- and the closest point transform algorithm `code/stlib/doc/html/cpt/cpt__page.html` through the **CPTLevelSet**$<$**DataType, dim** $>$ `code/amroc/doc/html/amr/classCPTLevelSet.html`

Lift-up of a spherical body

Cylindrical body hit by Mach 3 shockwave, 2D test case by [Falcovitz et al., 1997]

Schlieren plot of density

Refinement levels



code/amroc/doc/html/apps/clawpack_2applications_2euler_22d_2SphereLiftOff_2src_2Problem_8h_source.html

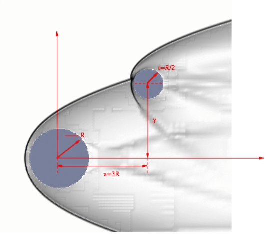
Proximal bodies in hypersonic flow

Flow modeled by Euler equations for a single polytropic gas with $p = (\gamma - 1)\,\rho e$

$$\partial_t \rho + \partial_{x_n}(\rho u_n) = 0\ , \quad \partial_t(\rho u_k) + \partial_{x_n}(\rho u_k u_n + \delta_{kn} p) = 0\ , \quad \partial_t(\rho E) + \partial_{x_n}(u_n(\rho E + p)) = 0$$

Numerical approximation with

- Finite volume flux-vector splitting scheme with MUSCL reconstruction, dimensional splitting



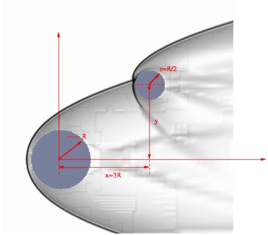
Proximal bodies in hypersonic flow

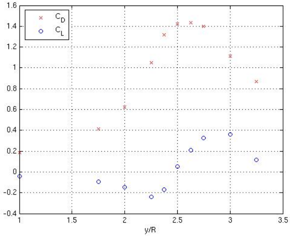
Flow modeled by Euler equations for a single polytropic gas with $p = (\gamma - 1)\,\rho e$

$$\partial_t \rho + \partial_{x_n}(\rho u_n) = 0\ , \quad \partial_t(\rho u_k) + \partial_{x_n}(\rho u_k u_n + \delta_{kn} p) = 0\ , \quad \partial_t(\rho E) + \partial_{x_n}(u_n(\rho E + p)) = 0$$

Numerical approximation with

- Finite volume flux-vector splitting scheme with MUSCL reconstruction, dimensional splitting
- Spherical bodies, force computation with overlaid lattitude-longitude mesh to obtain drag and lift coefficients $C_{D,L} = \dfrac{2F_{D,L}}{\rho v^2 \pi r^2}$
- inflow $M = 10$, C_D and C_L on secondary sphere, lateral position varied, no motion





Verification and validation

Static force measurements, $M = 10$:
[Laurence et al., 2007]

- Refinement study: $40 \times 40 \times 32$ base grid , up to without AMR up to $\sim 209.7 \cdot 10^6$ cells, largest run $\sim 35,000\,\mathrm{h}$ CPU

$l_{\max}$	C_D	ΔC_D	C_L	ΔC_L
1	1.264		-0.176	
2	1.442	0.178	-0.019	0.157
3	1.423	-0.019	0.052	0.071
4	1.408	-0.015	0.087	0.035

Verification and validation

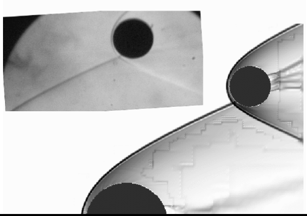
Static force measurements, $M = 10$:
[Laurence et al., 2007]

- Refinement study: $40 \times 40 \times 32$ base grid , up to without AMR up to $\sim 209.7 \cdot 10^6$ cells, largest run $\sim 35,000\,\mathrm{h}$ CPU

l_{max}	C_D	ΔC_D	C_L	ΔC_L
1	1.264		-0.176	
2	1.442	0.178	-0.019	0.157
3	1.423	-0.019	0.052	0.071
4	1.408	-0.015	0.087	0.035

- Comparison with experimental results: 3 additional levels, $\sim 2000\,\mathrm{h}$ CPU

	Experimental	Computational
C_D	1.11 ± 0.08	1.01
C_L	0.29 ± 0.05	0.28



Verification and validation

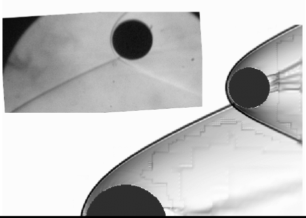
Static force measurements, $M = 10$:
[Laurence et al., 2007]

- Refinement study: $40 \times 40 \times 32$ base grid , up to without AMR up to $\sim 209.7 \cdot 10^6$ cells, largest run $\sim 35,000\,\mathrm{h}$ CPU

$l_{\max}$	C_D	ΔC_D	C_L	ΔC_L
1	1.264		-0.176	
2	1.442	0.178	-0.019	0.157
3	1.423	-0.019	0.052	0.071
4	1.408	-0.015	0.087	0.035

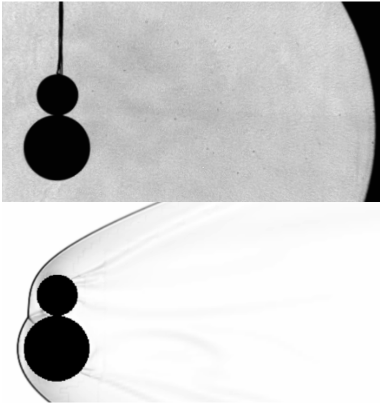
- Comparison with experimental results: 3 additional levels, $\sim 2000\,\mathrm{h}$ CPU

	Experimental	Computational
C_D	1.11 ± 0.08	1.01
C_L	0.29 ± 0.05	0.28



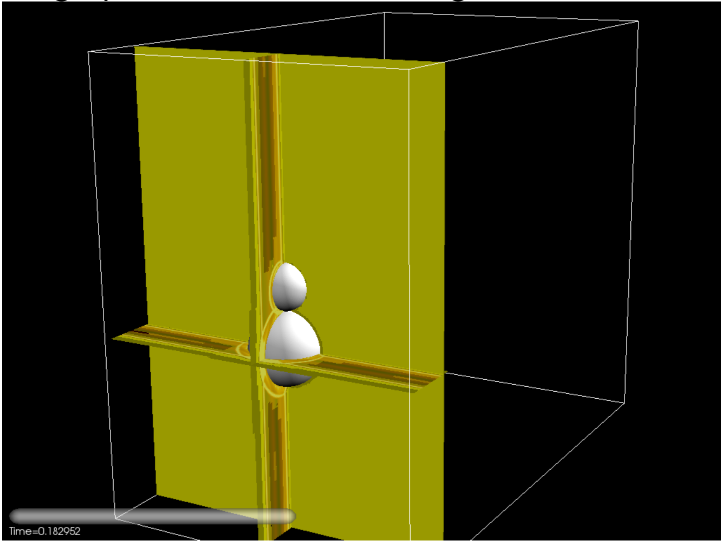
Dynamic motion, $M = 4$:

- Base grid $150 \times 125 \times 90$, two additional levels with $r_{1,2} = 2$
- 24,704 time steps, $36,808\,\mathrm{h}$ CPU on 256 cores IBM BG/P



[Laurence and Deiterding, 2011]

Schlieren graphics on refinement regions



code/amroc/doc/html/apps/clawpack_2applications_2euler_23d_2Spheres_2src_2Problem_8h_source.html

Treatment of thin structures

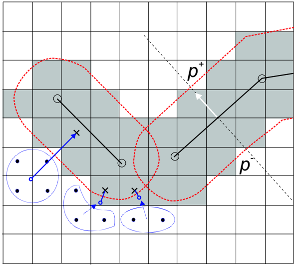
- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method

Treatment of thin structures

- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method
- Unsigned distance level set function φ

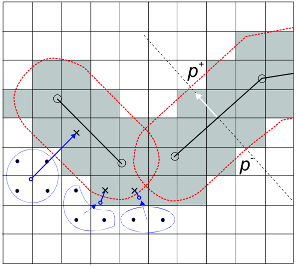
Treatment of thin structures

- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method
- Unsigned distance level set function φ
- Treat cells with $0 < \varphi < d$ as ghost fluid cells



Treatment of thin structures

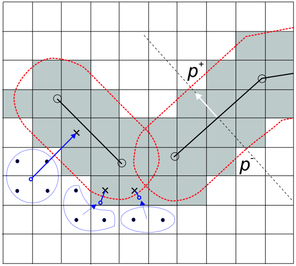
- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method
- Unsigned distance level set function φ
- Treat cells with $0 < \varphi < d$ as ghost fluid cells



- Leaving φ unmodified ensures correctness of $\nabla\varphi$

Treatment of thin structures

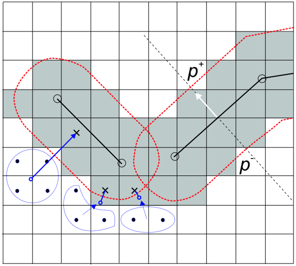
- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method
- Unsigned distance level set function φ
- Treat cells with $0 < \varphi < d$ as ghost fluid cells



- Leaving φ unmodified ensures correctness of $\nabla\varphi$
- Use face normal in shell element to evaluate in $\Delta p = p^+ - p^-$

Treatment of thin structures

- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method
- Unsigned distance level set function φ
- Treat cells with $0 < \varphi < d$ as ghost fluid cells



- Leaving φ unmodified ensures correctness of $\nabla\varphi$
- Use face normal in shell element to evaluate in $\Delta p = p^+ - p^-$
- Utilize finite difference solver using the beam equation

$$\rho_s h \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial \bar{x}^4} = p^F$$

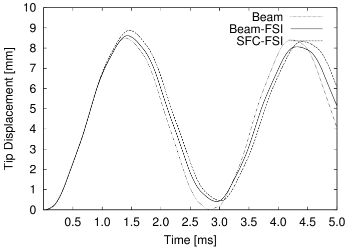
to verify FSI algorithms

FSI verification by elastic vibration

- Thin steel plate (thickness $h = 1\,\mathrm{mm}$, length $50\,\mathrm{mm}$), clamped at lower end
- $\rho_s = 7600\,\mathrm{kg/m^3}$, $E = 220\,\mathrm{GPa}$, $I = h^3/12$, $\nu = 0.3$
- Modeled with beam solver (101 points) and thin-shell FEM solver (325 triangles) by F. Cirak

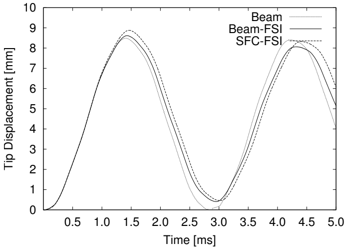
FSI verification by elastic vibration

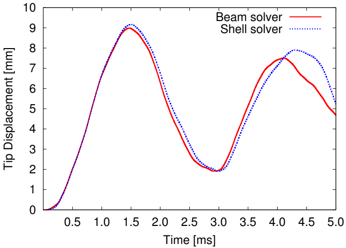
- Thin steel plate (thickness $h = 1\,\mathrm{mm}$, length $50\,\mathrm{mm}$), clamped at lower end
- $\rho_s = 7600\,\mathrm{kg/m^3}$, $E = 220\,\mathrm{GPa}$, $I = h^3/12$, $\nu = 0.3$
- Modeled with beam solver (101 points) and thin-shell FEM solver (325 triangles) by F. Cirak
- Left: Coupling verification with constant instantenous loading by $\Delta p = 100\,\mathrm{kPa}$



FSI verification by elastic vibration

- Thin steel plate (thickness $h = 1\,\mathrm{mm}$, length $50\,\mathrm{mm}$), clamped at lower end
- $\rho_s = 7600\,\mathrm{kg/m^3}$, $E = 220\,\mathrm{GPa}$, $I = h^3/12$, $\nu = 0.3$
- Modeled with beam solver (101 points) and thin-shell FEM solver (325 triangles) by F. Cirak
- Left: Coupling verification with constant instantenous loading by $\Delta p = 100\,\mathrm{kPa}$
- Right: FSI verification with Mach 1.21 shockwave in air ($\gamma = 1.4$)

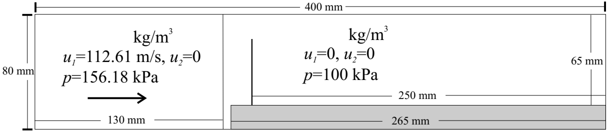




Shock-driven elastic panel motion

Test case suggested by [Giordano et al., 2005]

- Forward facing step geometry, fixed walls everywhere except at inflow

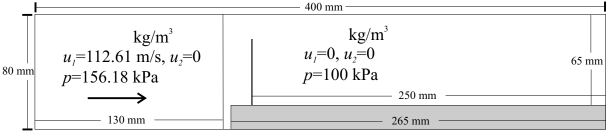


- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$

Shock-driven elastic panel motion

Test case suggested by [Giordano et al., 2005]

- Forward facing step geometry, fixed walls everywhere except at inflow

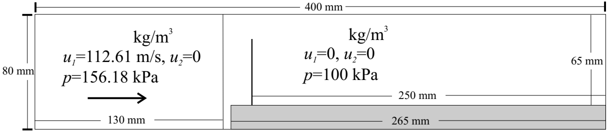


- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$
- Intel 3.4GHz Xeon dual processors, GB Ethernet interconnect
  - Beam-FSI: 12.25 h CPU on 3 fluid CPU + 1 solid CPU
    code/doc/html/capps/beam-amroc_2VibratingBeam_2src_2FluidProblem_8h_source.html, code/doc/html/capps/beam-amroc_2VibratingBeam_2src_2SolidProblem_8h_source.html
  - FEM-FSI: 322 h CPU on 14 fluid CPU + 2 solid CPU
    code/doc/html/capps/sfc-amroc_2VibratingPanel_2src_2FluidProblem_8h_source.html, code/doc/html/capps/VibratingPanel_2src_2ShellManagerSpecific_8h_source.html

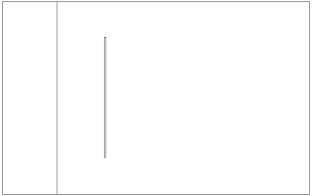
Shock-driven elastic panel motion

Test case suggested by [Giordano et al., 2005]

- Forward facing step geometry, fixed walls everywhere except at inflow



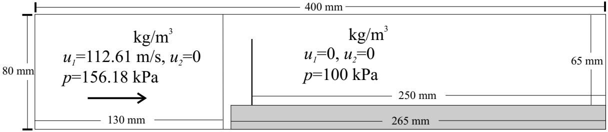
- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$
- Intel 3.4GHz Xeon dual processors, GB Ethernet interconnect
  - Beam-FSI: 12.25 h CPU on 3 fluid CPU + 1 solid CPU
    code/doc/html/capps/beam-amroc_2VibratingBeam_2src_2FluidProblem_8h_source.html,
    code/doc/html/capps/beam-amroc_2VibratingBeam_2src_2SolidProblem_8h_source.html
  - FEM-FSI: 322 h CPU on 14 fluid CPU + 2 solid CPU
    code/doc/html/capps/sfc-amroc_2VibratingPanel_2src_2FluidProblem_8h_source.html,
    code/doc/html/capps/VibratingPanel_2src_2ShellManagerSpecific_8h_source.html



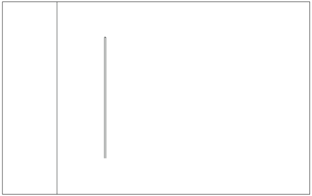
Shock-driven elastic panel motion

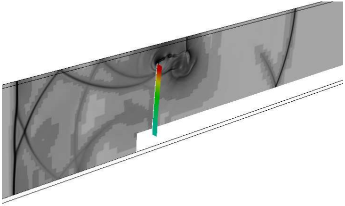
Test case suggested by [Giordano et al., 2005]

- Forward facing step geometry, fixed walls everywhere except at inflow



- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$
- Intel 3.4GHz Xeon dual processors, GB Ethernet interconnect
  - Beam-FSI: 12.25 h CPU on 3 fluid CPU + 1 solid CPU
    code/doc/html/capps/beam-amroc_2VibratingBeam_2src_2FluidProblem_8h_source.html, code/doc/html/capps/beam-amroc_2VibratingBeam_2src_2SolidProblem_8h_source.html
  - FEM-FSI: 322 h CPU on 14 fluid CPU + 2 solid CPU
    code/doc/html/capps/sfc-amroc_2VibratingPanel_2src_2FluidProblem_8h_source.html, code/doc/html/capps/VibratingPanel_2src_2ShellManagerSpecific_8h_source.html



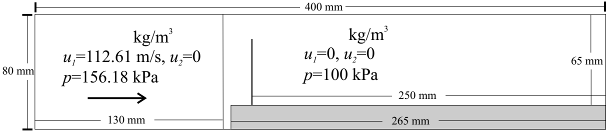


$t = 0.43$ ms after impact

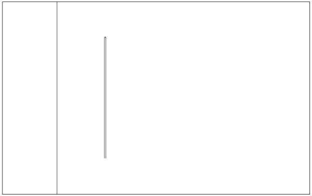
Shock-driven elastic panel motion

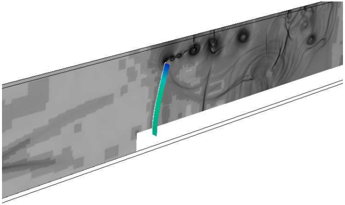
Test case suggested by [Giordano et al., 2005]

- Forward facing step geometry, fixed walls everywhere except at inflow



- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$
- Intel 3.4GHz Xeon dual processors, GB Ethernet interconnect
  - Beam-FSI: $12.25\,\mathrm{h}$ CPU on 3 fluid CPU + 1 solid CPU
    code/doc/html/capps/beam-amroc_2VibratingBeam_2src_2FluidProblem_8h_source.html, code/doc/html/capps/beam-amroc_2VibratingBeam_2src_2SolidProblem_8h_source.html
  - FEM-FSI: $322\,\mathrm{h}$ CPU on 14 fluid CPU + 2 solid CPU
    code/doc/html/capps/sfc-amroc_2VibratingPanel_2src_2FluidProblem_8h_source.html, code/doc/html/capps/VibratingPanel_2src_2ShellManagerSpecific_8h_source.html





$t = 1.56\,\mathrm{ms}$ after impact

Detonation-driven plastic deformation

Chapman-Jouguet detonation in a tube filled with a stoichiometric ethylene and oxygen ($\mathrm{C_2H_4} + 3\,\mathrm{O_2}$, 295 K) mixture. Euler equations with single exothermic reaction $A \longrightarrow B$

$$\partial_t \rho + \partial_{x_n}(\rho u_n) = 0 \;, \quad \partial_t(\rho u_k) + \partial_{x_n}(\rho u_k u_n + \delta_{kn} p) = 0 \;, k = 1, \ldots, d$$

$$\partial_t(\rho E) + \partial_{x_n}(u_n(\rho E + p)) = 0 \;, \quad \partial_t(Y\rho) + \partial_{x_n}(Y\rho u_n) = \psi$$

with

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho u_n u_n - \rho Y q_0) \quad \text{and} \quad \psi = -kY\rho \exp\left(\frac{-E_\mathrm{A}\rho}{p}\right)$$

Detonation-driven plastic deformation

Chapman-Jouguet detonation in a tube filled with a stoichiometric ethylene and oxygen ($\mathrm{C_2H_4} + 3\,\mathrm{O_2}$, 295 K) mixture. Euler equations with single exothermic reaction $A \longrightarrow B$

$$\partial_t \rho + \partial_{x_n}(\rho u_n) = 0\ , \quad \partial_t(\rho u_k) + \partial_{x_n}(\rho u_k u_n + \delta_{kn} p) = 0\ , k = 1, \dots, d$$

$$\partial_t(\rho E) + \partial_{x_n}(u_n(\rho E + p)) = 0\ , \quad \partial_t(Y\rho) + \partial_{x_n}(Y\rho u_n) = \psi$$

with

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho u_n u_n - \rho Y q_0) \quad \text{and} \quad \psi = -kY\rho \exp\left(\frac{-E_{\mathrm{A}}\rho}{p}\right)$$

modeled with heuristic detonation model by [Mader, 1979]

$V := \rho^{-1},\ V_0 := \rho_0^{-1},\ V_{\mathrm{CJ}} := \rho_{\mathrm{CJ}}$
$Y' := 1 - (V - V_0)/(V_{\mathrm{CJ}} - V_0)$
`If` $0 \le Y' \le 1$ `and` $Y > 10^{-8}$ `then`
  `If` $Y < Y'$ `and` $Y' < 0.9$ `then` $Y' := 0$
  `If` $Y' < 0.99$ `then` $p' := (1 - Y')p_{\mathrm{CJ}}$
    `else` $p' := p$
  $\rho_{\mathrm{A}} := Y'\rho$
  $E := p'/(\rho(\gamma - 1)) + Y' q_0 + \frac{1}{2} u_n u_n$

Detonation-driven plastic deformation

Chapman-Jouguet detonation in a tube filled with a stoichiometric ethylene and oxygen ($\mathrm{C_2H_4} + 3\,\mathrm{O_2}$, 295 K) mixture. Euler equations with single exothermic reaction $A \longrightarrow B$

$$\partial_t \rho + \partial_{x_n}(\rho u_n) = 0\ , \quad \partial_t(\rho u_k) + \partial_{x_n}(\rho u_k u_n + \delta_{kn} p) = 0\ , k = 1, \dots, d$$
$$\partial_t(\rho E) + \partial_{x_n}(u_n(\rho E + p)) = 0\ , \quad \partial_t(Y\rho) + \partial_{x_n}(Y\rho u_n) = \psi$$

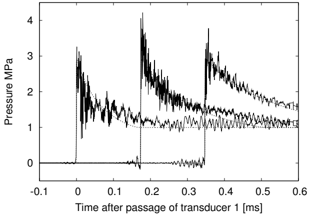
with

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho u_n u_n - \rho Y q_0) \quad \text{and} \quad \psi = -kY\rho \exp\left(\frac{-E_\mathrm{A}\rho}{p}\right)$$

modeled with heuristic detonation model by [Mader, 1979]

$V := \rho^{-1},\ V_0 := \rho_0^{-1},\ V_\mathrm{CJ} := \rho_\mathrm{CJ}$
$Y' := 1 - (V - V_0)/(V_\mathrm{CJ} - V_0)$
If $0 \le Y' \le 1$ and $Y > 10^{-8}$ then
  If $Y < Y'$ and $Y' < 0.9$ then $Y' := 0$
  If $Y' < 0.99$ then $p' := (1 - Y')p_\mathrm{CJ}$
    else $p' := p$
  $\rho_\mathrm{A} := Y'\rho$
  $E := p'/(\rho(\gamma - 1)) + Y' q_0 + \frac{1}{2}u_n u_n$

Comparison of the pressure traces in the experiment and in a 1d simulation



Tube with flaps

- Fluid: VanLeer FVS
  - Detonation model with $\gamma = 1.24$, $p_{\mathrm{CJ}} = 3.3\,\mathrm{MPa}$, $D_{\mathrm{CJ}} = 2376\,\mathrm{m/s}$
  - AMR base level: $104 \times 80 \times 242$, $r_{1,2} = 2$, $r_3 = 4$
  - $\sim 4 \cdot 10^7$ cells instead of $7.9 \cdot 10^9$ cells (uniform)
  - Tube and detonation fully refined
  - Thickening of 2D mesh: $0.81\,\mathrm{mm}$ on both sides (real $0.445\,\mathrm{mm}$)

Tube with flaps

- Fluid: VanLeer FVS
  - Detonation model with $\gamma = 1.24$, $p_{\rm CJ} = 3.3\,\mathrm{MPa}$, $D_{\rm CJ} = 2376\,\mathrm{m/s}$
  - AMR base level: $104 \times 80 \times 242$, $r_{1,2} = 2$, $r_3 = 4$
  - $\sim 4 \cdot 10^7$ cells instead of $7.9 \cdot 10^9$ cells (uniform)
  - Tube and detonation fully refined
  - Thickening of 2D mesh: $0.81\,\mathrm{mm}$ on both sides (real $0.445\,\mathrm{mm}$)
- Solid: thin-shell solver by F. Cirak
  - Aluminum, J2 plasticity with hardening, rate sensitivity, and thermal softening
  - Mesh: 8577 nodes, 17056 elements

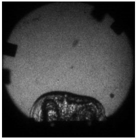
Tube with flaps

- Fluid: VanLeer FVS
  - Detonation model with $\gamma = 1.24$, $p_{\rm CJ} = 3.3\,\mathrm{MPa}$, $D_{\rm CJ} = 2376\,\mathrm{m/s}$
  - AMR base level: $104 \times 80 \times 242$, $r_{1,2} = 2$, $r_3 = 4$
  - $\sim 4 \cdot 10^7$ cells instead of $7.9 \cdot 10^9$ cells (uniform)
  - Tube and detonation fully refined
  - Thickening of 2D mesh: $0.81\,\mathrm{mm}$ on both sides (real $0.445\,\mathrm{mm}$)
- Solid: thin-shell solver by F. Cirak
  - Aluminum, J2 plasticity with hardening, rate sensitivity, and thermal softening
  - Mesh: 8577 nodes, 17056 elements
- 16+2 nodes 2.2 GHz AMD Opteron quad processor, PCI-X 4x Infiniband network, $\sim 4320\,\mathrm{h}$ CPU to $t_{end} = 450\,\mu\mathrm{s}$

Tube with flaps

- Fluid: VanLeer FVS
  - Detonation model with $\gamma = 1.24$, $p_{\mathrm{CJ}} = 3.3\,\mathrm{MPa}$, $D_{\mathrm{CJ}} = 2376\,\mathrm{m/s}$
  - AMR base level: $104 \times 80 \times 242$, $r_{1,2} = 2$, $r_3 = 4$
  - $\sim 4 \cdot 10^7$ cells instead of $7.9 \cdot 10^9$ cells (uniform)
  - Tube and detonation fully refined
  - Thickening of 2D mesh: $0.81\,\mathrm{mm}$ on both sides (real $0.445\,\mathrm{mm}$)
- Solid: thin-shell solver by F. Cirak
  - Aluminum, J2 plasticity with hardening, rate sensitivity, and thermal softening
  - Mesh: 8577 nodes, 17056 elements
- 16+2 nodes 2.2 GHz AMD Opteron quad processor, PCI-X 4x Infiniband network, $\sim 4320\,\mathrm{h}$ CPU to $t_{end} = 450\,\mu\mathrm{s}$

0.032 ms

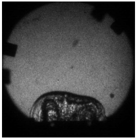


0.030 ms

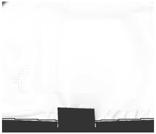
Tube with flaps

- Fluid: VanLeer FVS
  - Detonation model with $\gamma = 1.24$, $p_{\mathrm{CJ}} = 3.3\,\mathrm{MPa}$, $D_{\mathrm{CJ}} = 2376\,\mathrm{m/s}$
  - AMR base level: $104 \times 80 \times 242$, $r_{1,2} = 2$, $r_3 = 4$
  - $\sim 4 \cdot 10^7$ cells instead of $7.9 \cdot 10^9$ cells (uniform)
  - Tube and detonation fully refined
  - Thickening of 2D mesh: $0.81\,\mathrm{mm}$ on both sides (real $0.445\,\mathrm{mm}$)
- Solid: thin-shell solver by F. Cirak
  - Aluminum, J2 plasticity with hardening, rate sensitivity, and thermal softening
  - Mesh: 8577 nodes, 17056 elements
- 16+2 nodes 2.2 GHz AMD Opteron quad processor, PCI-X 4x Infiniband network, $\sim 4320\,\mathrm{h}$ CPU to $t_{end} = 450\,\mu\mathrm{s}$

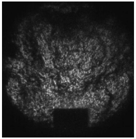
0.032 ms



0.030 ms

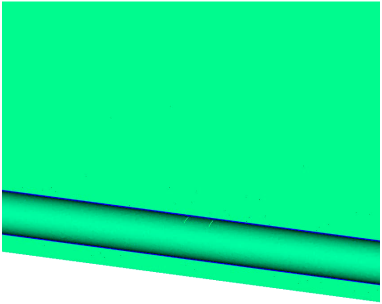


0.212 ms



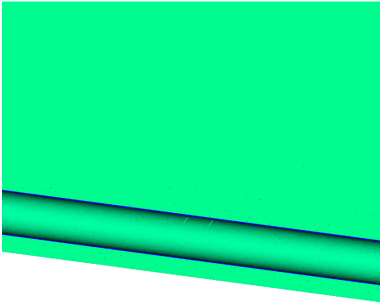
0.210 ms

Tube with flaps: results

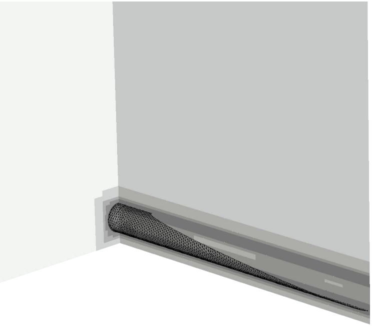


Fluid density and diplacement in y-direction in solid

Tube with flaps: results



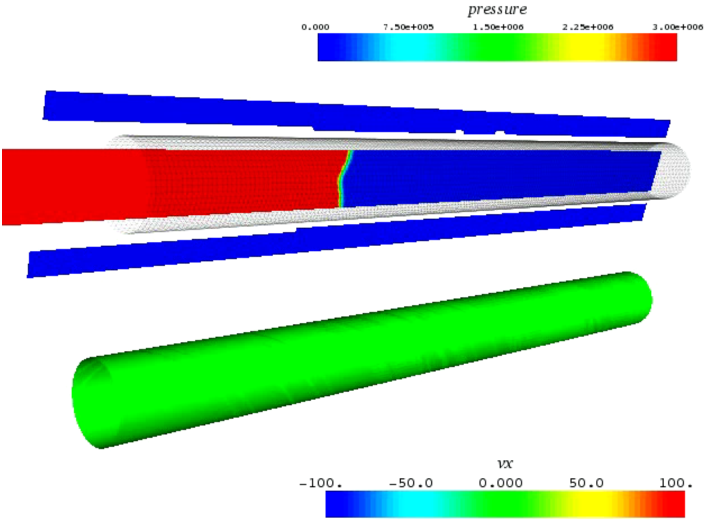
Fluid density and diplacement in y-direction in solid



Schlieren plot of fluid density on refinement levels

[Cirak et al., 2007]
code/doc/html/capps/sfc-amroc_2TubeCJBurnFlaps_2src_2FluidProblem_8h_source.html,
code/doc/html/capps/TubeCJBurnFlaps_2src_2ShellManagerSpecific_8h_source.html

Coupled fracture simulation



code/doc/html/capps/sfc-amroc_2TubeCJBurnFrac_2src_2FluidProblem_8h_source.html,
code/doc/html/capps/TubeCJBurnFrac_2src_2ShellManagerSpecific_8h_source.html

Underwater explosion modeling

Volume fraction based two-component model with $\sum_{i=1}^m \alpha^i = 1$, that defines mixture quantities as

$$\rho = \sum_{i=1}^m \alpha^i \rho^i \;, \quad \rho u_n = \sum_{i=1}^m \alpha^i \rho^i u_n^i \;, \quad \rho e = \sum_{i=1}^m \alpha^i \rho^i e^i$$

Underwater explosion modeling

Volume fraction based two-component model with $\sum_{i=1}^m \alpha^i = 1$, that defines mixture quantities as

$$\rho = \sum_{i=1}^m \alpha^i \rho^i \; , \quad \rho u_n = \sum_{i=1}^m \alpha^i \rho^i u_n^i \; , \quad \rho e = \sum_{i=1}^m \alpha^i \rho^i e^i$$

Assuming total pressure $p = (\gamma - 1)\,\rho e - \gamma p_\infty$ and speed of sound $c = (\gamma\,(p + p_\infty)/\rho)^{1/2}$ yields

$$\frac{p}{\gamma - 1} = \sum_{i=1}^m \frac{\alpha^i p^i}{\gamma^i - 1} \; , \quad \frac{\gamma p_\infty}{\gamma - 1} = \sum_{i=1}^m \frac{\alpha^i \gamma^i p_\infty^i}{\gamma^i - 1}$$

Underwater explosion modeling

Volume fraction based two-component model with $\sum_{i=1}^m \alpha^i = 1$, that defines mixture quantities as

$$\rho = \sum_{i=1}^m \alpha^i \rho^i \;, \quad \rho u_n = \sum_{i=1}^m \alpha^i \rho^i u_n^i \;, \quad \rho e = \sum_{i=1}^m \alpha^i \rho^i e^i$$

Assuming total pressure $p = (\gamma - 1)\, \rho e - \gamma p_\infty$ and speed of sound $c = (\gamma\, (p + p_\infty)/\rho)^{1/2}$ yields

$$\frac{p}{\gamma - 1} = \sum_{i=1}^m \frac{\alpha^i p^i}{\gamma^i - 1} \;, \quad \frac{\gamma p_\infty}{\gamma - 1} = \sum_{i=1}^m \frac{\alpha^i \gamma^i p_\infty^i}{\gamma^i - 1}$$

and the overall set of equations [Shyue, 1998]

$$\partial_t \rho + \partial_{x_n}(\rho u_n) = 0 \;, \quad \partial_t(\rho u_k) + \partial_{x_n}(\rho u_k u_n + \delta_{kn} p) = 0 \;, \quad \partial_t(\rho E) + \partial_{x_n}(u_n(\rho E + p)) = 0$$

$$\frac{\partial}{\partial t}\left(\frac{1}{\gamma - 1}\right) + u_n \frac{\partial}{\partial x_n}\left(\frac{1}{\gamma - 1}\right) = 0 \;, \quad \frac{\partial}{\partial t}\left(\frac{\gamma p_\infty}{\gamma - 1}\right) + u_n \frac{\partial}{\partial x_n}\left(\frac{\gamma p_\infty}{\gamma - 1}\right) = 0$$

Underwater explosion modeling

Volume fraction based two-component model with $\sum_{i=1}^m \alpha^i = 1$, that defines mixture quantities as

$$\rho = \sum_{i=1}^m \alpha^i \rho^i \;, \quad \rho u_n = \sum_{i=1}^m \alpha^i \rho^i u_n^i \;, \quad \rho e = \sum_{i=1}^m \alpha^i \rho^i e^i$$

Assuming total pressure $p = (\gamma - 1)\,\rho e - \gamma p_\infty$ and speed of sound $c = (\gamma\,(p + p_\infty)/\rho)^{1/2}$ yields

$$\frac{p}{\gamma - 1} = \sum_{i=1}^m \frac{\alpha^i p^i}{\gamma^i - 1} \;, \quad \frac{\gamma p_\infty}{\gamma - 1} = \sum_{i=1}^m \frac{\alpha^i \gamma^i p_\infty^i}{\gamma^i - 1}$$

and the overall set of equations [Shyue, 1998]

$$\partial_t \rho + \partial_{x_n}(\rho u_n) = 0 \;, \quad \partial_t(\rho u_k) + \partial_{x_n}(\rho u_k u_n + \delta_{kn} p) = 0 \;, \quad \partial_t(\rho E) + \partial_{x_n}(u_n(\rho E + p)) = 0$$

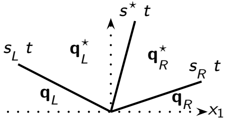
$$\frac{\partial}{\partial t}\left(\frac{1}{\gamma - 1}\right) + u_n \frac{\partial}{\partial x_n}\left(\frac{1}{\gamma - 1}\right) = 0 \;, \quad \frac{\partial}{\partial t}\left(\frac{\gamma p_\infty}{\gamma - 1}\right) + u_n \frac{\partial}{\partial x_n}\left(\frac{\gamma p_\infty}{\gamma - 1}\right) = 0$$

Oscillation free at contacts: [Abgrall and Karni, 2001][Shyue, 2006]

Approximate Riemann solver

Use HLLC approach because of robustness and positivity preservation

$$\mathbf{q}^{HLLC}(x_1, t) = \begin{cases} \mathbf{q}_L \,, & x_1 < s_L\, t, \\ \mathbf{q}_L^\star \,, & s_L\, t \le x_1 < s^\star\, t, \\ \mathbf{q}_R^\star \,, & s^\star\, t \le x_1 \le s_R\, t, \\ \mathbf{q}_R \,, & x_1 > s_R\, t, \end{cases}$$

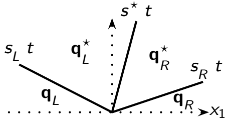


Wave speed estimates [Davis, 1988] $s_L = \min\{u_{1,L} - c_L, u_{1,R} - c_R\}$, $s_R = \max\{u_{1,L} + c_L, u_{1,R} + c_R\}$

Approximate Riemann solver

Use HLLC approach because of robustness and positivity preservation

$$\mathbf{q}^{HLLC}(x_1,t) = \begin{cases} \mathbf{q}_L\,, & x_1 < s_L\,t, \\ \mathbf{q}_L^\star\,, & s_L\,t \le x_1 < s^\star\,t, \\ \mathbf{q}_R^\star\,, & s^\star\,t \le x_1 \le s_R\,t, \\ \mathbf{q}_R\,, & x_1 > s_R\,t, \end{cases}$$



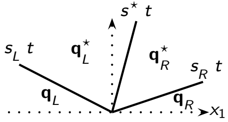
Wave speed estimates [Davis, 1988] $s_L = \min\{u_{1,L} - c_L, u_{1,R} - c_R\}$, $s_R = \max\{u_{1,L} + c_L, u_{1,R} + c_R\}$
Unkown state [Toro et al., 1994]

$$s^\star = \frac{p_R - p_L + s_L u_{1,L}(s_L - u_{1,L}) - \rho_R u_{1,R}(s_R - u_{1,R})}{\rho_L(s_L - u_{1,L}) - \rho_R(s_R - u_{1,R})}$$

Approximate Riemann solver

Use HLLC approach because of robustness and positivity preservation

$$\mathbf{q}^{HLLC}(x_1,t) = \begin{cases} \mathbf{q}_L\,, & x_1 < s_L\, t, \\ \mathbf{q}_L^\star\,, & s_L\, t \le x_1 < s^\star\, t, \\ \mathbf{q}_R^\star\,, & s^\star\, t \le x_1 \le s_R\, t, \\ \mathbf{q}_R\,, & x_1 > s_R\, t, \end{cases}$$



Wave speed estimates [Davis, 1988] $s_L = \min\{u_{1,L} - c_L, u_{1,R} - c_R\}$,
$s_R = \max\{u_{1,L} + c_L, u_{1,R} + c_R\}$
Unkown state [Toro et al., 1994]

$$s^\star = \frac{p_R - p_L + s_L u_{1,L}(s_L - u_{1,L}) - \rho_R u_{1,R}(s_R - u_{1,R})}{\rho_L(s_L - u_{1,L}) - \rho_R(s_R - u_{1,R})}$$

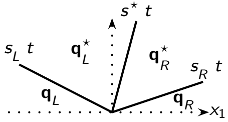
$$\mathbf{q}_\tau^\star = \left[\eta, \eta s^\star, \eta u_2, \eta\left[\frac{(\rho E)_\tau}{\rho_\tau} + (s^\star - u_{1,\tau})\left(s_\tau + \frac{p_\tau}{\rho_\tau(s_\tau - u_{1,\tau})}\right)\right], \frac{1}{\gamma_\tau - 1}, \frac{\gamma_\tau p_{\infty,\tau}}{\gamma_\tau - 1}\right]^T$$

$$\eta = \rho_\tau \frac{s_\tau - u_{1,\tau}}{s_\tau - s^\star}\,, \quad \tau = \{L, R\}$$

Approximate Riemann solver

Use HLLC approach because of robustness and positivity preservation

$$\mathbf{q}^{HLLC}(x_1,t) = \begin{cases} \mathbf{q}_L\,, & x_1 < s_L\, t, \\ \mathbf{q}^\star_L\,, & s_L\, t \le x_1 < s^\star\, t, \\ \mathbf{q}^\star_R\,, & s^\star\, t \le x_1 \le s_R\, t, \\ \mathbf{q}_R\,, & x_1 > s_R\, t, \end{cases}$$



Wave speed estimates [Davis, 1988] $s_L = \min\{u_{1,L} - c_L, u_{1,R} - c_R\}$,
$s_R = \max\{u_{1,L} + c_L, u_{1,R} + c_R\}$
Unkown state [Toro et al., 1994]

$$s^\star = \frac{p_R - p_L + s_L u_{1,L}(s_L - u_{1,L}) - \rho_R u_{1,R}(s_R - u_{1,R})}{\rho_L(s_L - u_{1,L}) - \rho_R(s_R - u_{1,R})}$$

$$\mathbf{q}^\star_\tau = \left[\eta, \eta s^\star, \eta u_2, \eta\left[\frac{(\rho E)_\tau}{\rho_\tau} + (s^\star - u_{1,\tau})\left(s_\tau + \frac{p_\tau}{\rho_\tau(s_\tau - u_{1,\tau})}\right)\right], \frac{1}{\gamma_\tau - 1}, \frac{\gamma_\tau p_{\infty,\tau}}{\gamma_\tau - 1}\right]^T$$

$$\eta = \rho_\tau \frac{s_\tau - u_{1,\tau}}{s_\tau - s^\star}\,, \quad \tau = \{L, R\}$$

Evaluate waves as $\mathcal{W}_1 = \mathbf{q}^\star_L - \mathbf{q}_L$, $\mathcal{W}_2 = \mathbf{q}^\star_R - \mathbf{q}^\star_L$, $\mathcal{W}_3 = \mathbf{q}_R - \mathbf{q}^\star_R$ and $\lambda_1 = s_L$, $\lambda_2 = s^\star$, $\lambda_3 = s_R$ to compute the fluctuations $\mathcal{A}^-\Delta = \sum_{\lambda_\nu < 0} \lambda_\nu\, \mathcal{W}_\nu$, $\mathcal{A}^+\Delta = \sum_{\lambda_\nu \ge 0} \lambda_\nu\, \mathcal{W}_\nu$ for $\nu = \{1,2,3\}$

Overall scheme: Wave Propagation method [Shyue, 2006]

Underwater explosion FSI simulations

- Air: $\gamma^A = 1.4$, $p_\infty^A = 0$, $\rho^A = 1.29\,\mathrm{kg/m^3}$
- Water: $\gamma^W = 7.415$, $p_\infty^W = 296.2\,\mathrm{MPa}$, $\rho^W = 1027\,\mathrm{kg/m^3}$

Underwater explosion FSI simulations

- Air: $\gamma^A = 1.4$, $p_\infty^A = 0$, $\rho^A = 1.29\,\mathrm{kg/m^3}$
- Water: $\gamma^W = 7.415$, $p_\infty^W = 296.2\,\mathrm{MPa}$, $\rho^W = 1027\,\mathrm{kg/m^3}$
- Cavitation modeling with pressure cut-off model at $p = -1\,\mathrm{MPa}$

Underwater explosion FSI simulations

- Air: $\gamma^A = 1.4$, $p_\infty^A = 0$, $\rho^A = 1.29\,\mathrm{kg/m^3}$
- Water: $\gamma^W = 7.415$, $p_\infty^W = 296.2\,\mathrm{MPa}$, $\rho^W = 1027\,\mathrm{kg/m^3}$
- Cavitation modeling with pressure cut-off model at $p = -1\,\mathrm{MPa}$
- 3D simulation of deformation of air backed aluminum plate with $r = 85\,\mathrm{mm}$, $h = 3\,\mathrm{mm}$ from underwater explosion
  - Water basin [Ashani and Ghamsari, 2008] $2\,\mathrm{m} \times 1.6\,\mathrm{m} \times 2\,\mathrm{m}$
  - Explosion modeled as energy increase ($m_{\mathrm{C4}} \cdot 6.06\,\mathrm{MJ/kg}$) in sphere with r=5mm
  - $\rho_s = 2719\,\mathrm{kg/m3}$, $E = 69\,\mathrm{GPa}$, $\nu = 0.33$, J2 plasticity model, yield stress $\sigma_y = 217.6\,\mathrm{MPa}$

Underwater explosion FSI simulations

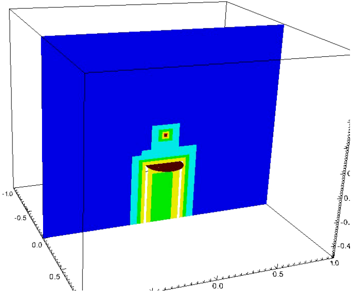
- Air: $\gamma^A = 1.4$, $p_\infty^A = 0$, $\rho^A = 1.29\,\mathrm{kg/m^3}$
- Water: $\gamma^W = 7.415$, $p_\infty^W = 296.2\,\mathrm{MPa}$, $\rho^W = 1027\,\mathrm{kg/m^3}$
- Cavitation modeling with pressure cut-off model at $p = -1\,\mathrm{MPa}$
- 3D simulation of deformation of air backed aluminum plate with $r = 85\,\mathrm{mm}$, $h = 3\,\mathrm{mm}$ from underwater explosion
  - Water basin [Ashani and Ghamsari, 2008] $2\,\mathrm{m} \times 1.6\,\mathrm{m} \times 2\,\mathrm{m}$
  - Explosion modeled as energy increase ($m_{\mathrm{C4}} \cdot 6.06\,\mathrm{MJ/kg}$) in sphere with r=5mm
  - $\rho_s = 2719\,\mathrm{kg/m3}$, $E = 69\,\mathrm{GPa}$, $\nu = 0.33$, J2 plasticity model, yield stress $\sigma_y = 217.6\,\mathrm{MPa}$
- 3D simulation of copper plate $r = 32\,\mathrm{mm}$, $h = 0.25\,\mathrm{mm}$ rupturing due to water hammer
  - Water-filled shocktube $1.3\,\mathrm{m}$ with driver piston [Deshpande et al., 2006]
  - Piston simulated with separate level set, see [Deiterding et al., 2009] for pressure wave
  - $\rho_s = 8920\,\mathrm{kg/m3}$, $E = 130\,\mathrm{GPa}$, $\nu = 0.31$, J2 plasticity model, $\sigma_y = 38.5\,\mathrm{MPa}$, cohesive interface model, max. tensile stress $\sigma_c = 525\,\mathrm{MPa}$

Underwater explosion simulation

- AMR base grid $50 \times 40 \times 50$, $r_{1,2,3} = 2$, $r_4 = 4$, $l_c = 3$, highest level restricted to initial explosion center, 3rd and 4th level to plate vicinity
- Triangular mesh with 8148 elements
- Computations of 1296 coupled time steps to $t_{end} = 1\,\mathrm{ms}$
- 10+2 nodes 3.4 GHz Intel Xeon dual processor, $\sim 130\,\mathrm{h}$ CPU

Maximal deflection $[\mathrm{mm}]$

	Exp.	Sim.
$20\,\mathrm{g}$, $d = 25\,\mathrm{cm}$	28.83	25.88
$30\,\mathrm{g}$, $d = 30\,\mathrm{cm}$	30.09	27.31

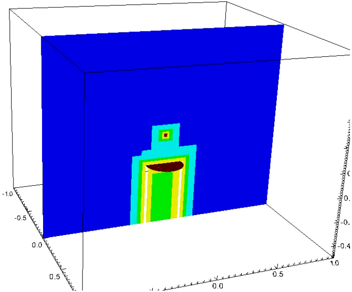


Underwater explosion simulation

- AMR base grid $50 \times 40 \times 50$, $r_{1,2,3} = 2$, $r_4 = 4$, $l_c = 3$, highest level restricted to initial explosion center, 3rd and 4th level to plate vicinity
- Triangular mesh with 8148 elements
- Computations of 1296 coupled time steps to $t_{end} = 1\,\mathrm{ms}$
- 10+2 nodes 3.4 GHz Intel Xeon dual processor, $\sim 130\,\mathrm{h}$ CPU

Maximal deflection $[\mathrm{mm}]$

	Exp.	Sim.
$20\,\mathrm{g}$, $d = 25\,\mathrm{cm}$	28.83	25.88
$30\,\mathrm{g}$, $d = 30\,\mathrm{cm}$	30.09	27.31



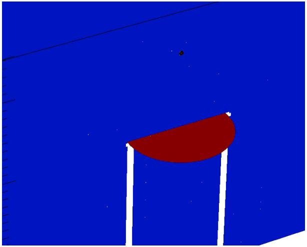
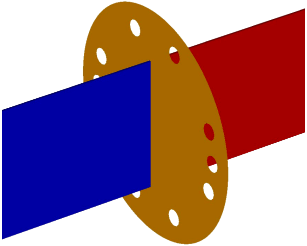


Plate in underwater shocktube

- AMR base mesh $374 \times 20 \times 20$, $r_{1,2} = 2$, $l_c = 2$, solid mesh: 8896 triangles
- ~ 1250 coupled time steps to $t_{end} = 1\,\mathrm{ms}$
- 6+6 nodes 3.4 GHz Intel Xeon dual processor, $\sim 800\,\mathrm{h}$ CPU
  code/doc/html/capps/sfc-amroc_2WaterBlastFracture_2src_2FluidProblem_8h_source.html,
  code/doc/html/capps/WaterBlastFracture_2src_2ShellManagerSpecific_8h_source.html

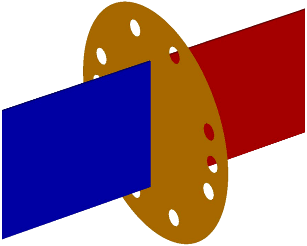


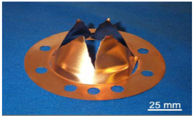
$p_0 = 64\,\mathrm{MPa}$

Plate in underwater shocktube

- AMR base mesh $374 \times 20 \times 20$, $r_{1,2} = 2$, $l_c = 2$, solid mesh: 8896 triangles
- $\sim$ 1250 coupled time steps to $t_{end} = 1\,\mathrm{ms}$
- 6+6 nodes 3.4 GHz Intel Xeon dual processor, $\sim 800\,\mathrm{h}$ CPU

code/doc/html/capps/sfc-amroc_2WaterBlastFracture_2src_2FluidProblem_8h_source.html, code/doc/html/capps/WaterBlastFracture_2src_2ShellManagerSpecific_8h_source.html

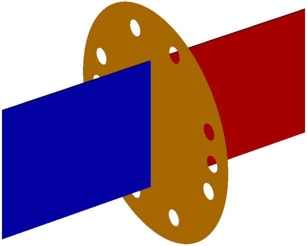


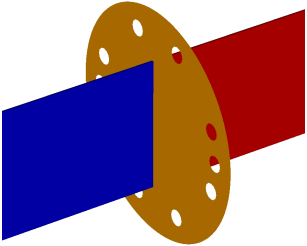


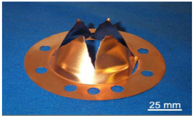
$p_0 = 64\,\mathrm{MPa}$

Plate in underwater shocktube

- AMR base mesh $374 \times 20 \times 20$, $r_{1,2} = 2$, $l_c = 2$, solid mesh: 8896 triangles
- ~ 1250 coupled time steps to $t_{end} = 1\,\mathrm{ms}$
- 6+6 nodes 3.4 GHz Intel Xeon dual processor, $\sim 800\,\mathrm{h}$ CPU
  `code/doc/html/capps/sfc-amroc_2WaterBlastFracture_2src_2FluidProblem_8h_source.html`,
  `code/doc/html/capps/WaterBlastFracture_2src_2ShellManagerSpecific_8h_source.html`







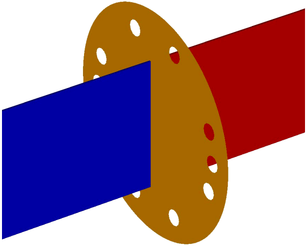
$p_0 = 64\,\mathrm{MPa}$

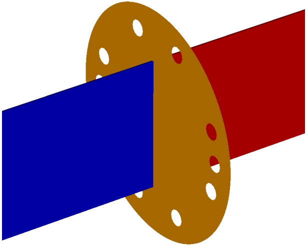
$p_0 = 173\,\mathrm{MPa}$

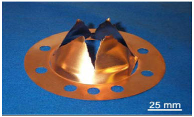
Plate in underwater shocktube

- AMR base mesh $374 \times 20 \times 20$, $r_{1,2} = 2$, $l_c = 2$, solid mesh: 8896 triangles
- ~ 1250 coupled time steps to $t_{end} = 1\,\mathrm{ms}$
- 6+6 nodes 3.4 GHz Intel Xeon dual processor, $\sim 800\,\mathrm{h}$ CPU

code/doc/html/capps/sfc-amroc_2WaterBlastFracture_2src_2FluidProblem_8h_source.html,
code/doc/html/capps/WaterBlastFracture_2src_2ShellManagerSpecific_8h_source.html





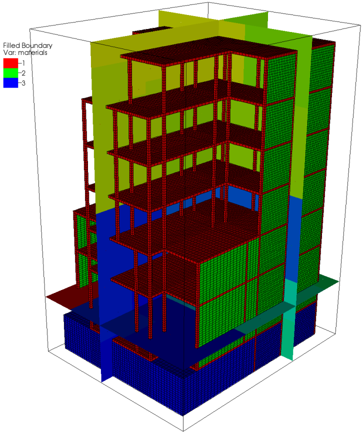


$p_0 = 64\,\mathrm{MPa}$

$p_0 = 173\,\mathrm{MPa}$

Blast explosion in a multistory building

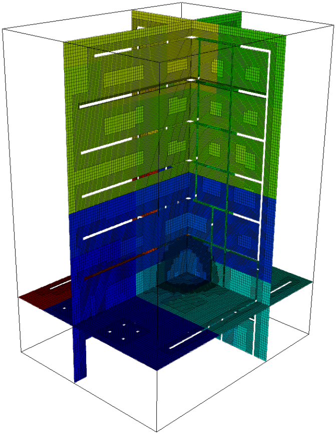
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: 1,070 coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim$ 8,000,000 cells instead of 55,296,000 (uniform)
- 69,709 hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

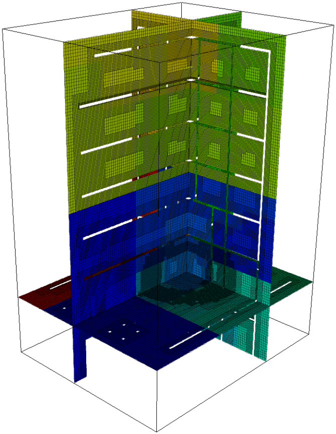
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

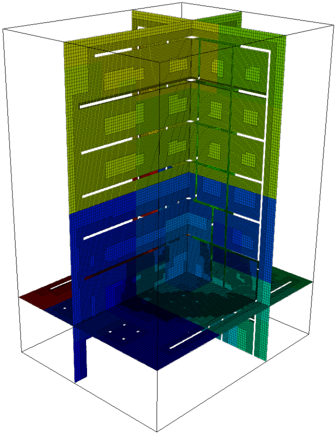
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

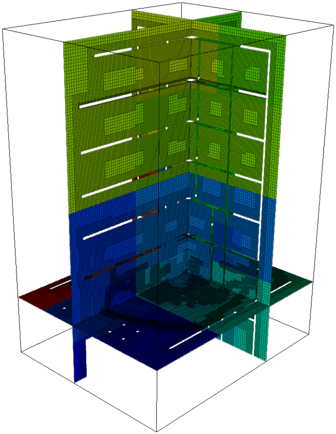
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: 1,070 coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- 69,709 hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

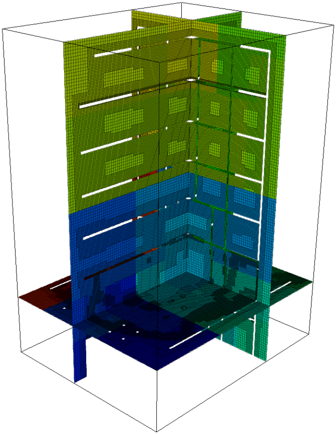
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

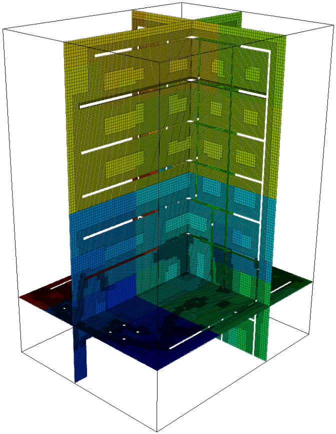
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

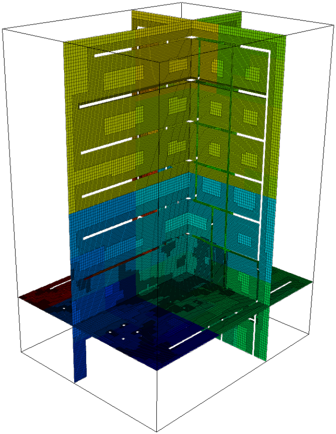
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

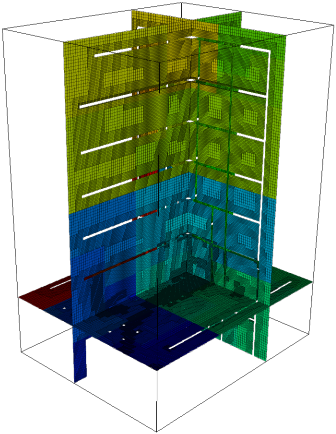
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

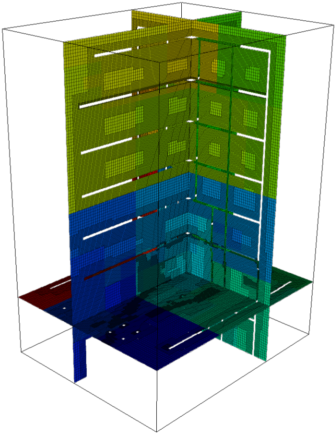
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

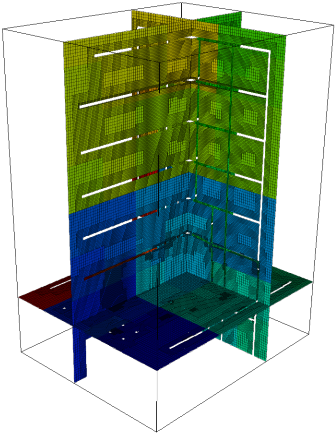
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

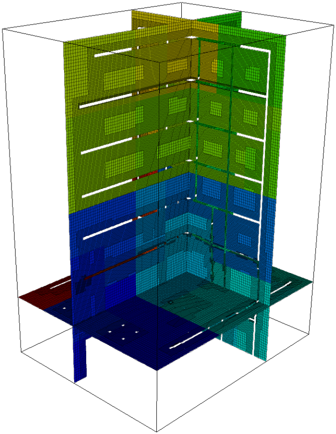
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

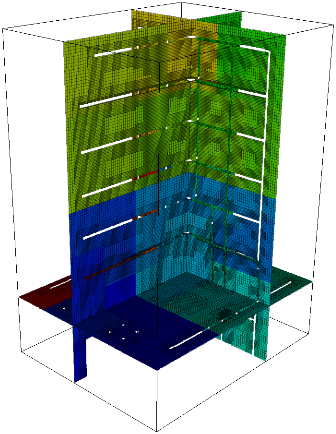
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: 1,070 coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- 69,709 hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building

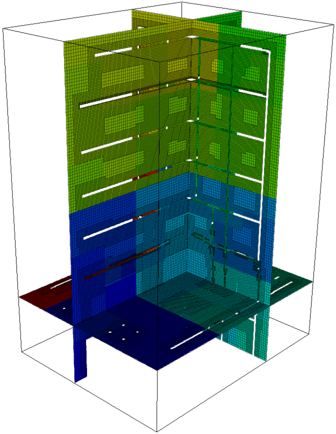
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

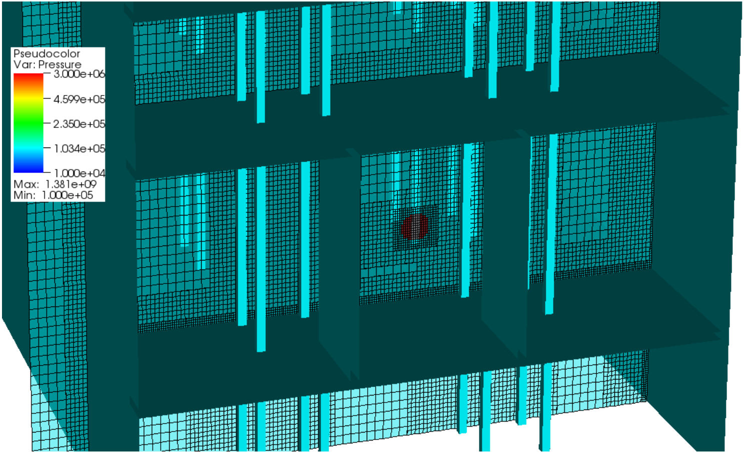
Blast explosion in a multistory building

- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- Simulation with ground: $1,070$ coupled time steps, $830\,\mathrm{h}$ CPU ($\sim 25.9\,\mathrm{h}$ wall time) on 31+1 cores
- $\sim 8,000,000$ cells instead of $55,296,000$ (uniform)
- $69,709$ hexahedral elements and with material parameters. [Deiterding and Wood, 2013]



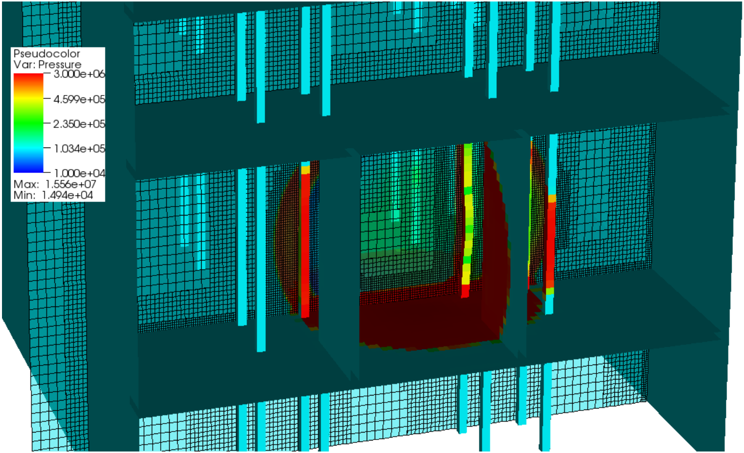
	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building – II

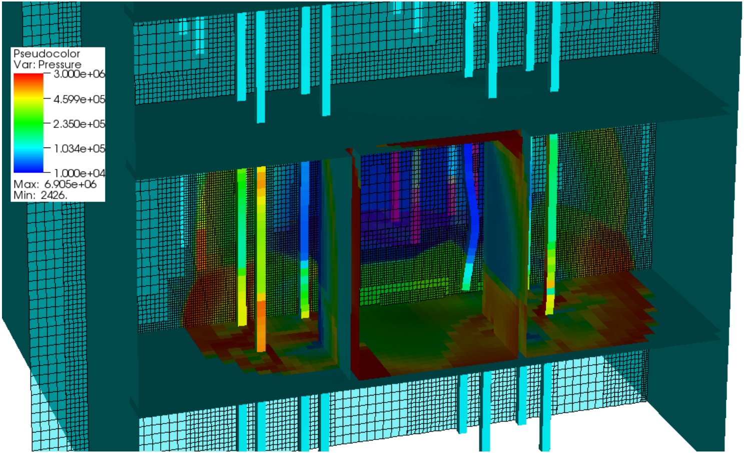


$t = 0$

Blast explosion in a multistory building – II

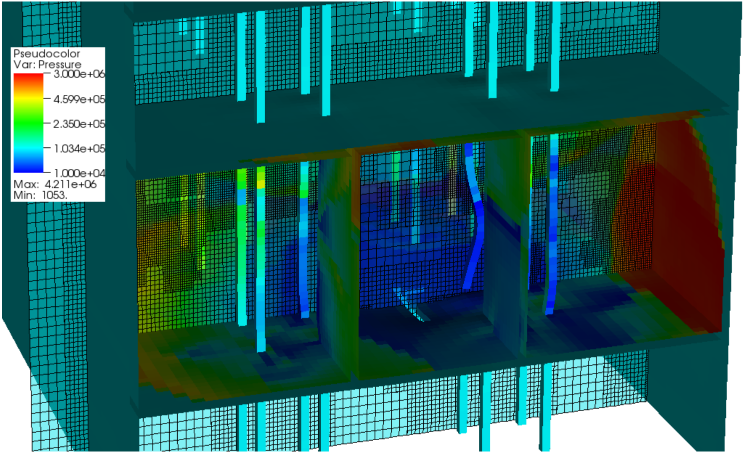


Blast explosion in a multistory building – II

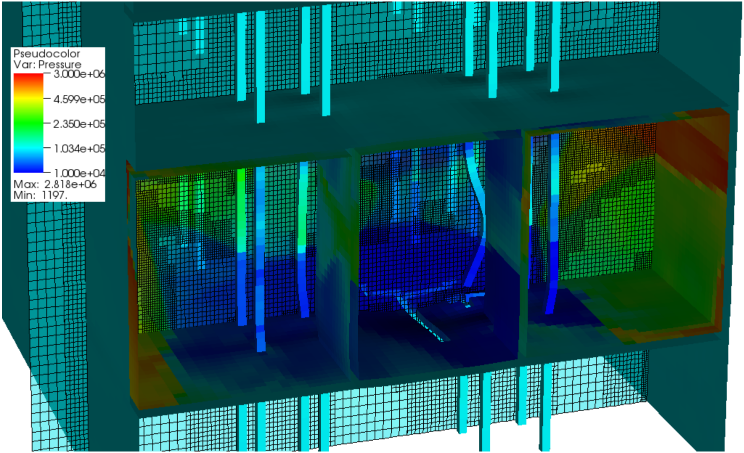


$t = 6.1\,\mathrm{ms}$

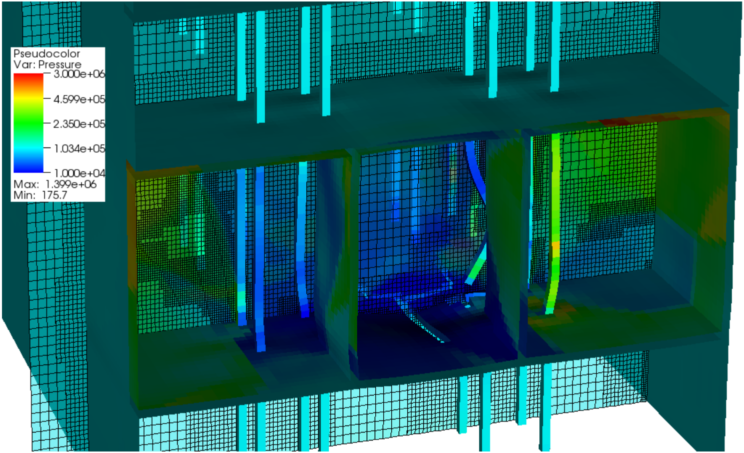
Blast explosion in a multistory building – II



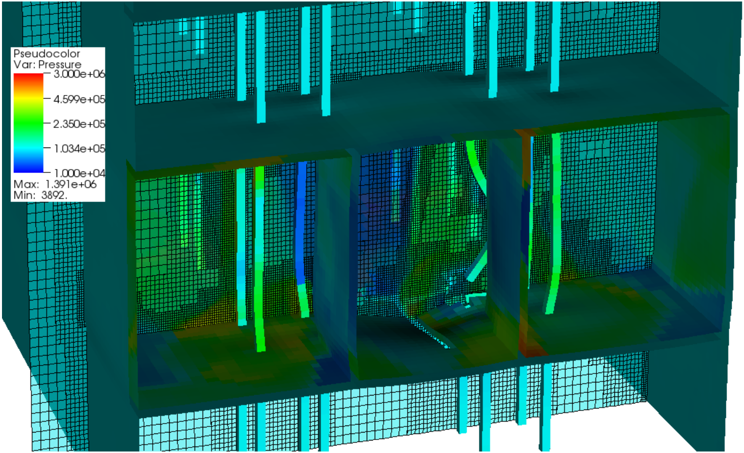
Blast explosion in a multistory building – II



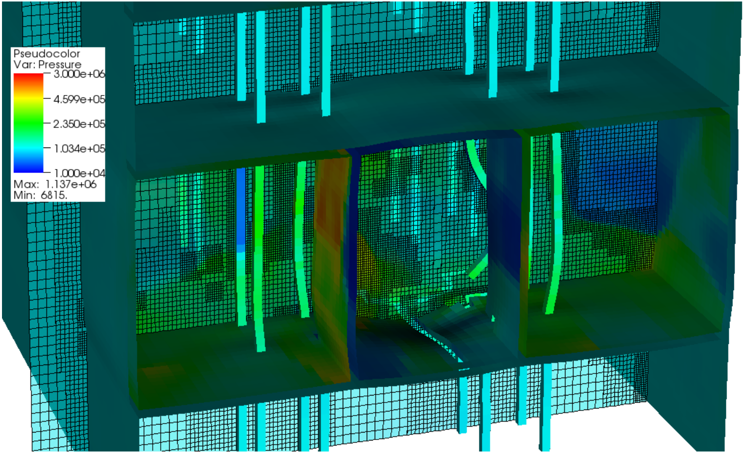
Blast explosion in a multistory building – II



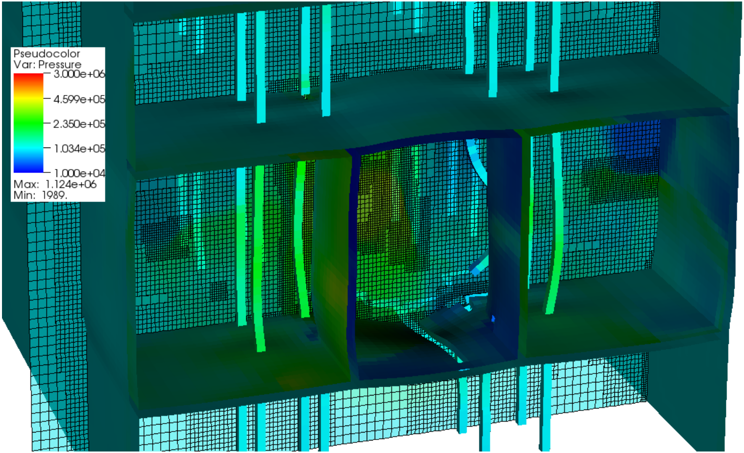
Blast explosion in a multistory building – II



Blast explosion in a multistory building – II

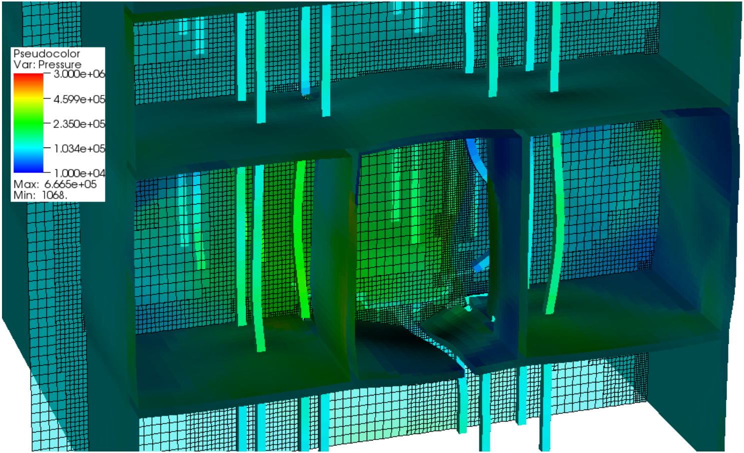


Blast explosion in a multistory building – II

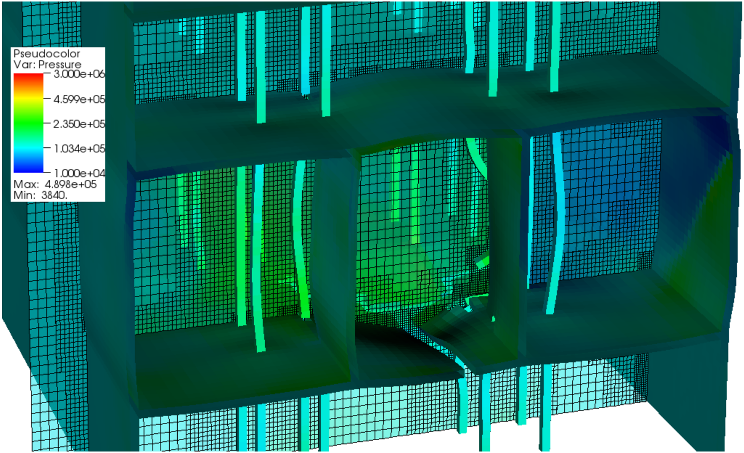


$t = 29.2\,\mathrm{ms}$

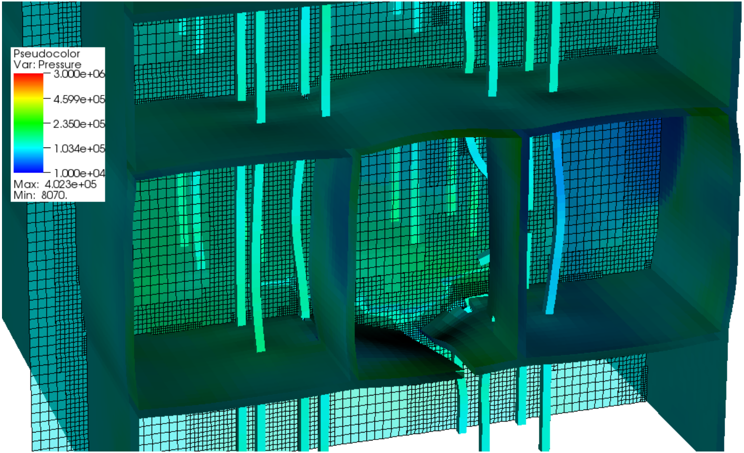
Blast explosion in a multistory building – II



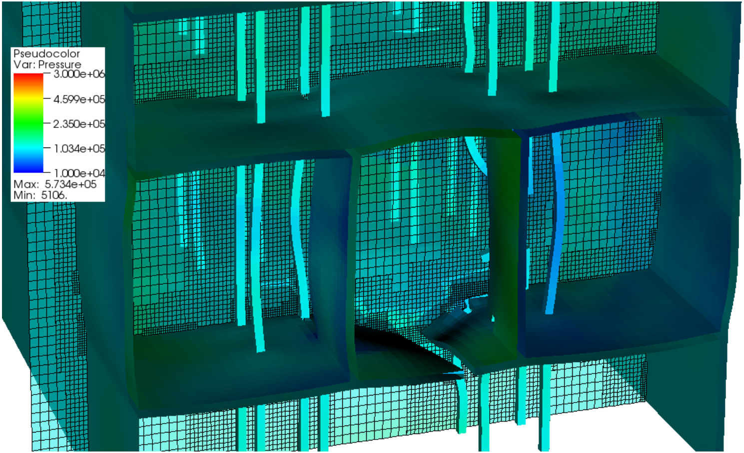
Blast explosion in a multistory building – II



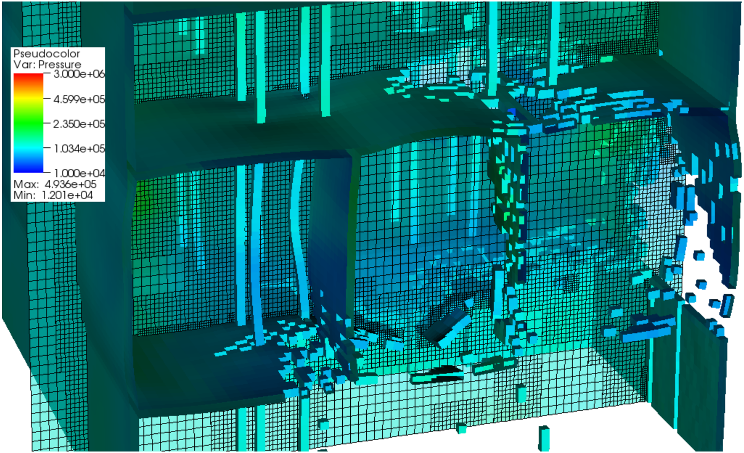
Blast explosion in a multistory building – II



Blast explosion in a multistory building – II



Blast explosion in a multistory building – II



$t = 48.7\,\mathrm{ms}$

Outline

Parallelized construction of space-filling curve

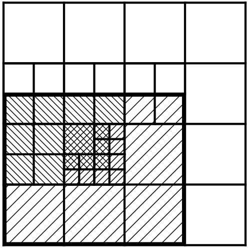
Computation of space filling curve

- Partition-Init

Parallelized construction of space-filling curve

Computation of space filling curve

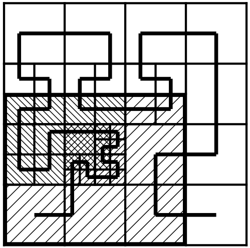
- Partition-Init
    1. Compute aggregated workload for new grid hierarchy and project result onto level 0



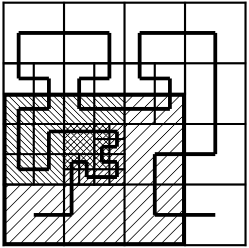
Parallelized construction of space-filling curve

Computation of space filling curve

- Partition-Init
    1. Compute aggregated workload for new grid hierarchy and project result onto level 0
    2. Construct recursively SFC-units until work in each unit is homogeneous, GuCFactor defines minimal coarseness relative to level-0 grid



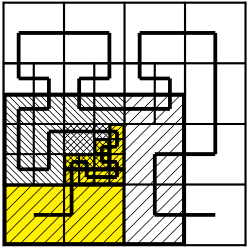
Parallelized construction of space-filling curve



Computation of space filling curve

- Partition-Init
  1. Compute aggregated workload for new grid hierarchy and project result onto level 0
  2. Construct recursively SFC-units until work in each unit is homogeneous, GuCFactor defines minimal coarseness relative to level-0 grid
- Partition-Calc
  1. Compute entire workload and new work for each processor

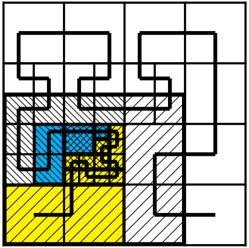
Parallelized construction of space-filling curve



Computation of space filling curve

- Partition-Init
  1. Compute aggregated workload for new grid hierarchy and project result onto level 0
  2. Construct recursively SFC-units until work in each unit is homogeneous, GuCFactor defines minimal coarseness relative to level-0 grid
- Partition-Calc
  1. Compute entire workload and new work for each processor
  2. Go sequentially through SFC-ordered list of partitioning units and assign units to processors, refine partition if necessary and possible

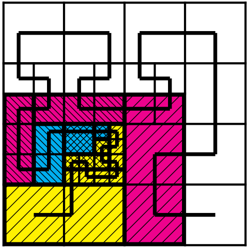
Parallelized construction of space-filling curve



Computation of space filling curve

- Partition-Init
  1. Compute aggregated workload for new grid hierarchy and project result onto level 0
  2. Construct recursively SFC-units until work in each unit is homogeneous, GuCFactor defines minimal coarseness relative to level-0 grid
- Partition-Calc
  1. Compute entire workload and new work for each processor
  2. Go sequentially through SFC-ordered list of partitioning units and assign units to processors, refine partition if necessary and possible

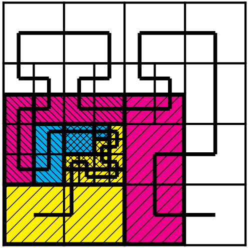
Parallelized construction of space-filling curve



Computation of space filling curve

- Partition-Init
  1. Compute aggregated workload for new grid hierarchy and project result onto level 0
  2. Construct recursively SFC-units until work in each unit is homogeneous, GuCFactor defines minimal coarseness relative to level-0 grid
- Partition-Calc
  1. Compute entire workload and new work for each processor
  2. Go sequentially through SFC-ordered list of partitioning units and assign units to processors, refine partition if necessary and possible

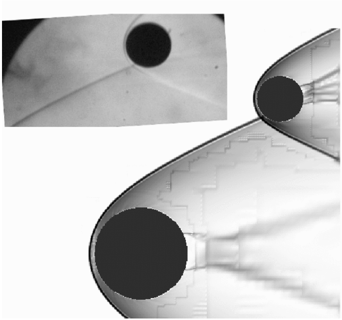
Parallelized construction of space-filling curve

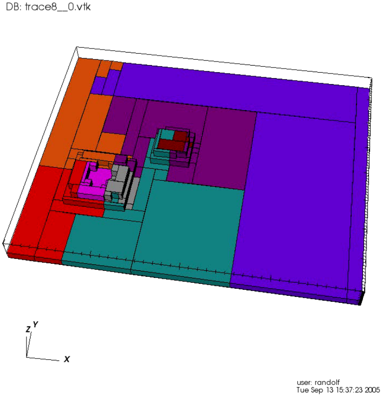


Computation of space filling curve

- Partition-Init
  1. Compute aggregated workload for new grid hierarchy and project result onto level 0
  2. Construct recursively SFC-units until work in each unit is homogeneous, GuCFactor defines minimal coarseness relative to level-0 grid
- Partition-Calc
  1. Compute entire workload and new work for each processor
  2. Go sequentially through SFC-ordered list of partitioning units and assign units to processors, refine partition if necessary and possible
- Ensure scalability of Partition-Init by creating SFC-units strictly local
- Currently still use of `MPI_allgather()` to create globally identical input for Partition-Calc (can be a bottleneck for weak scalability)

Partitioning example





- Cylinders of spheres in supersonic flow
- Predict force on secondary body
- Right: 200x160 base mesh, 3 Levels, factors 2,2,2, 8 CPUs

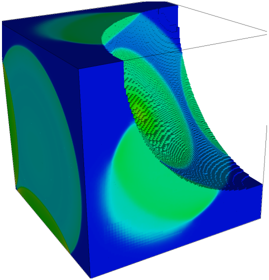
[Laurence et al., 2007]

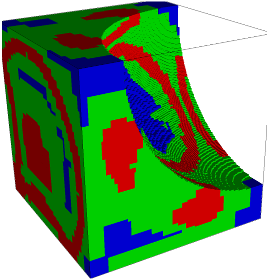
First performance assessment

- Test run on 2.2 GHz AMD Opteron quad-core cluster connected with Infiniband
- Cartesian test configuration
- Spherical blast wave, Euler equations, 3rd order WENO scheme, 3-step Runge-Kutta update
- AMR base grid 64^3, $r_{1,2} = 2$, 89 time steps on coarsest level
- With embedded boundary method: 96 time steps on coarsest level
- Redistribute in parallel every 2nd base level step
- Uniform grid $256^3 = 16.8 \cdot 10^6$ cells

Level	Grids	Cells
0	115	262,144
1	373	1,589,808
2	2282	5,907,064

Grid and cells used on 16 CPUs





Cost of SAMR and ghost-fluid method

- Flux correction is negligible
- Clustering is negligible (already local approach). For the complexities of a scalable global clustering algorithm see [Gunney et al., 2007]

CPUs	16	32	64
Time per step	32.44s	18.63s	11.87s
Uniform	59.65s	29.70s	15.15s
Integration	73.46%	64.69%	50.44%
Flux Correction	1.30%	1.49%	2.03%
Boundary Setting	13.72%	16.60%	20.44%
Regridding	10.43%	15.68%	24.25%
Clustering	0.34%	0.32%	0.26%
Output	0.29%	0.53%	0.92%
Misc.	0.46%	0.44%	0.47%

Cost of SAMR and ghost-fluid method

- Flux correction is negligible
- Clustering is negligible (already local approach). For the complexities of a scalable global clustering algorithm see [Gunney et al., 2007]
- Costs for GFM constant around $\sim$ 36%
- Main costs: `Regrid(l)` operation and ghost cell synchronization

CPUs	16	32	64
Time per step	32.44s	18.63s	11.87s
Uniform	59.65s	29.70s	15.15s
Integration	73.46%	64.69%	50.44%
Flux Correction	1.30%	1.49%	2.03%
Boundary Setting	13.72%	16.60%	20.44%
Regridding	10.43%	15.68%	24.25%
Clustering	0.34%	0.32%	0.26%
Output	0.29%	0.53%	0.92%
Misc.	0.46%	0.44%	0.47%

CPUs	16	32	64
Time per step	43.97s	25.24s	16.21s
Uniform	69.09s	35.94s	18.24s
Integration	59.09%	49.93%	40.20%
Flux Correction	0.82%	0.80%	1.14%
Boundary Setting	19.22%	25.58%	28.98%
Regridding	7.21%	9.15%	13.46%
Clustering	0.25%	0.23%	0.21%
GFM Find Cells	2.04%	1.73%	1.38%
GFM Interpolation	6.01%	10.39%	7.92%
GFM Overhead	0.54%	0.47%	0.37%
GFM Calculate	0.70%	0.60%	0.48%
Output	0.23%	0.52%	0.74%
Misc.	0.68%	0.62%	0.58%

AMROC scalability tests

Basic test configuration

- Spherical blast wave, Euler equations, 3D wave propagation method
- AMR base grid 32^3 with $r_{1,2} = 2, 4$. 5 time steps on coarsest level
- Uniform grid $256^3 = 16.8 \cdot 10^6$ cells, 19 time steps
- Flux correction deactivated
- No volume I/O operations
- Tests run IBM BG/P (mode VN)

AMROC scalability tests

Basic test configuration

- Spherical blast wave, Euler equations, 3D wave propagation method
- AMR base grid 32^3 with $r_{1,2} = 2, 4$. 5 time steps on coarsest level
- Uniform grid $256^3 = 16.8 \cdot 10^6$ cells, 19 time steps
- Flux correction deactivated
- No volume I/O operations
- Tests run IBM BG/P (mode VN)

Weak scalability test

- Reproduction of configuration each 64 CPUs
- On 1024 CPUs: $128 \times 64 \times 64$ base grid, $> 33,500$ Grids, $\sim 61 \cdot 10^6$ cells, uniform $1024 \times 512 \times 512 = 268 \cdot 10^6$ cells

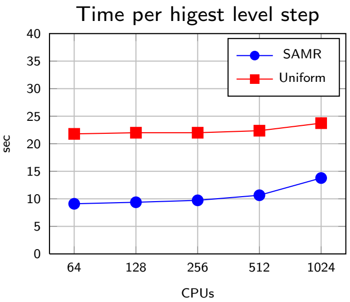
Level	Grids	Cells
0	606	32,768
1	575	135,312
2	910	3,639,040

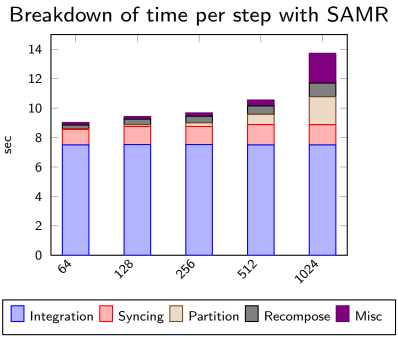
Strong scalability test

- $64 \times 32 \times 32$ base grid, uniform $512 \times 256 \times 256 = 33.6 \cdot 10^6$ cells

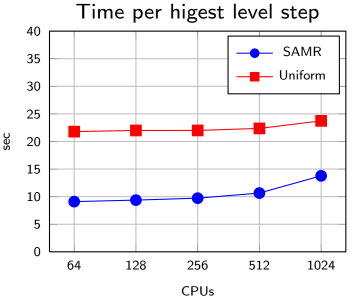
Level	Grids	Cells
0	1709	65,536
1	1735	271,048
2	2210	7,190,208

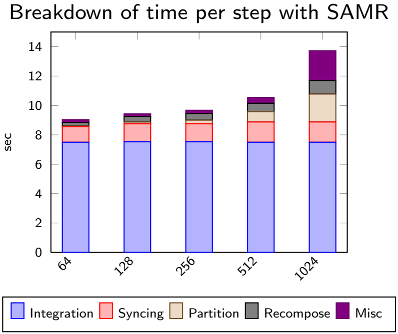
Weak scalability test





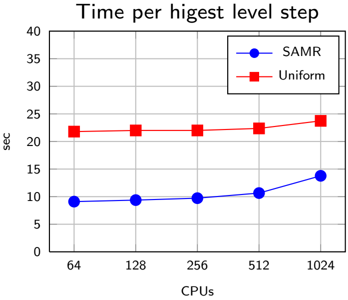
Weak scalability test

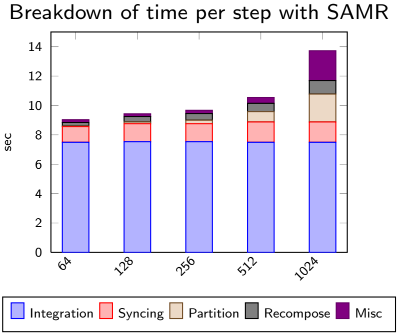




- Costs for Syncing basically constant

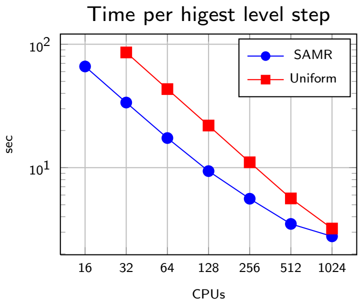
Weak scalability test

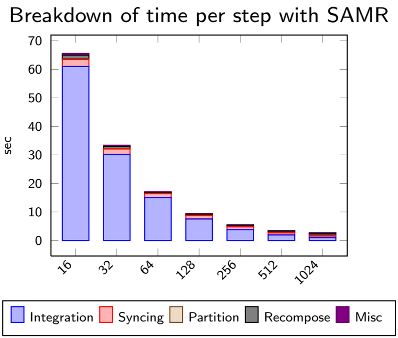




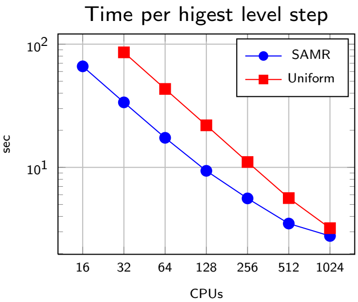
- Costs for Syncing basically constant
- Partitioning, Recompose, Misc (origin not clear) increase
- 1024 required usage of -DUAL option due to usage of global lists data structures in Partition-Calc and Recompose

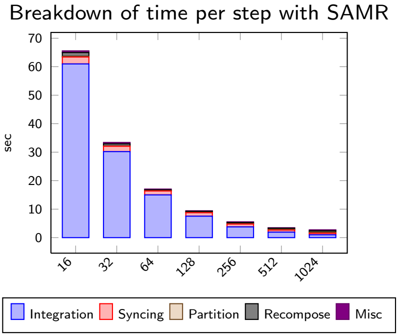
Strong scalability test





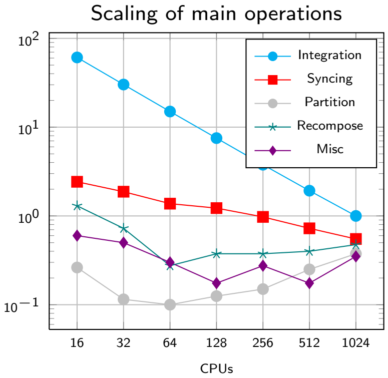
Strong scalability test

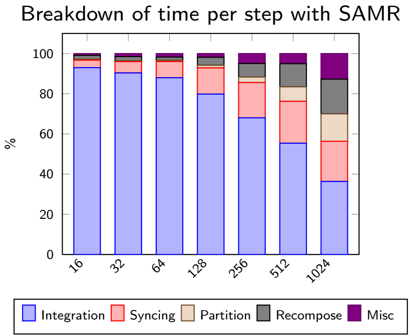




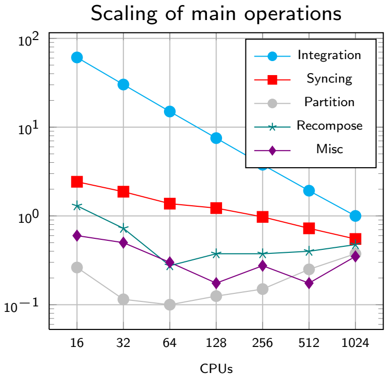
- Uniform code has basically linear scalability (explicit method)
- SAMR visibly looses efficiency for > 512 CPU, or $15,000$ finite volume cells per CPU

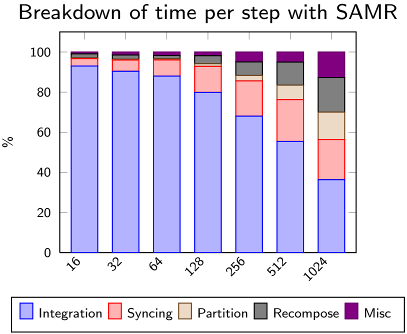
Strong scalability test - II





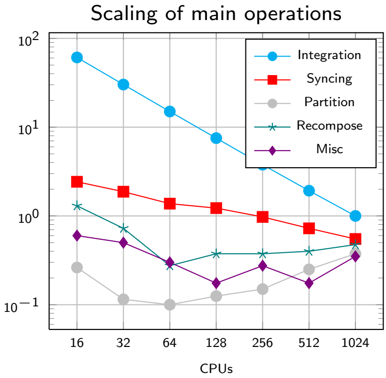
Strong scalability test - II

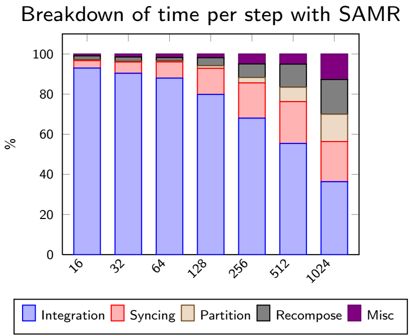




- Perfect scaling of Integration, reasonable scaling of Syncing

Strong scalability test - II





- Perfect scaling of Integration, reasonable scaling of Syncing
- Strong scalability of Partition needs to be addressed (eliminate global lists)

References I

[Abgrall and Karni, 2001] Abgrall, R. and Karni, S. (2001). Computations of compressible multifluids. *J. Comput. Phys.*, 169:594–523.

[Arienti et al., 2003] Arienti, M., Hung, P., Morano, E., and Shepherd, J. E. (2003). A level set approach to Eulerian-Lagrangian coupling. *J. Comput. Phys.*, 185:213–251.

[Ashani and Ghamsari, 2008] Ashani, J. Z. and Ghamsari, A. K. (2008). Theoretical and experimental analysis of plastic response of isotropic circular plates subjected to underwater explosion loading. *Mat.-wiss. u. Werkstofftechn.*, 39(2):171–175.

[Cirak et al., 2007] Cirak, F., Deiterding, R., and Mauch, S. P. (2007). Large-scale fluid-structure interaction simulation of viscoplastic and fracturing thin shells subjected to shocks and detonations. *Computers & Structures*, 85(11-14):1049–1065.

[Davis, 1988] Davis, S. F. (1988). Simplified second-order Godunov-type methods. *SIAM J. Sci. Stat. Comp.*, 9:445–473.

References II

[Deiterding et al., 2009] Deiterding, R., Cirak, F., and Mauch, S. P. (2009). Efficient fluid-structure interaction simulation of viscoplastic and fracturing thin-shells subjected to underwater shock loading. In Hartmann, S., Meister, A., Schäfer, M., and Turek, S., editors, *Int. Workshop on Fluid-Structure Interaction. Theory, Numerics and Applications, Herrsching am Ammersee 2008*, pages 65–80. kassel university press GmbH.

[Deiterding and Wood, 2013] Deiterding, R. and Wood, S. L. (2013). Parallel adaptive fluid-structure interaction simulations of explosions impacting on building structures. *Computers & Fluids*, 88:719–729.

[Deshpande et al., 2006] Deshpande, V. S., Heaver, A., and Fleck, N. A. (2006). An underwater shock simulator. *Royal Society of London Proceedings Series A*, 462(2067):1021–1041.

[Falcovitz et al., 1997] Falcovitz, J., Alfandary, G., and Hanoch, G. (1997). A two-dimensional conservation laws scheme for compressible flows with moving boundaries. *J. Comput. Phys.*, 138:83–102.

References III

[Fedkiw, 2002] Fedkiw, R. P. (2002). Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *J. Comput. Phys.*, 175:200–224.

[Giordano et al., 2005] Giordano, J., Jourdan, G., Burtschell, Y., Medale, M., Zeitoun, D. E., and Houas, L. (2005). Shock wave impacts on deforming panel, an application of fluid-structure interaction. *Shock Waves*, 14(1-2):103–110.

[Gunney et al., 2007] Gunney, B. T., Wissink, A. M., and Hysoma, D. A. (2007). Parallel clustering algorithms for structured AMR. *J. Parallel and Distributed Computing*, 66(11):1419–1430.

[Laurence and Deiterding, 2011] Laurence, S. J. and Deiterding, R. (2011). Shock-wave surfing. *J. Fluid Mech.*, 676:369–431.

[Laurence et al., 2007] Laurence, S. J., Deiterding, R., and Hornung, H. G. (2007). Proximal bodies in hypersonic flows. *J. Fluid Mech.*, 590:209–237.

[Luccioni et al., 2004] Luccioni, B. M., Ambrosini, R. D., and Danesi, R. F. (2004). Analysis of building collapse under blast loads. *Engineering & Structures*, 26:63–71.

References IV

[Mader, 1979] Mader, C. L. (1979). *Numerical modeling of detonations*. University of California Press, Berkeley and Los Angeles, California.

[Mauch, 2003] Mauch, S. P. (2003). *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. PhD thesis, California Institute of Technology.

[Shyue, 1998] Shyue, K.-M. (1998). An efficient shock-capturing algorithm for compressible multicomponent problems. *J. Comput. Phys.*, 142:208–242.

[Shyue, 2006] Shyue, K.-M. (2006). A volume-fraction based algorithm for hybrid barotropic and non-barotropic two-fluid flow problems. *Shock Waves*, 15:407–423.

[Specht, 2000] Specht, U. (2000). *Numerische Simulation mechanischer Wellen an Fluid-Festkörper-Mediengrenzen*. Number 398 in VDI Reihe 7. VDU Verlag, Düsseldorf.

[Toro et al., 1994] Toro, E. F., Spruce, M., and Speares, W. (1994). Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4:25–34.