Lecture 5

Fluid-structure interaction simulation

Course *Block-structured Adaptive Finite Volume Methods for Shock-Induced Combustion Simulation*

Ralf Deiterding
German Aerospace Center (DLR)
Institute for Aerodynamics and Flow Technology
Bunsenstr. 10, Göttingen, Germany

E-mail: ralf.deiterding@dlr.de

Outline

Fluid-structure interaction

- Coupling to a solid mechanics solver
- Rigid body motion
- Thin elastic and deforming thin structures
- Deformation from water hammer
- Real-world example

Outline

Fluid-structure interaction
- Coupling to a solid mechanics solver
- Rigid body motion
- Thin elastic and deforming thin structures
- Deformation from water hammer
- Real-world example

Adaptive Lattice Boltzmann method with FSI
- Adaptive LBM
- Realistic static embedded geometries
- Simulation of wind turbines

Outline

Fluid-structure interaction
- Coupling to a solid mechanics solver
- Rigid body motion
- Thin elastic and deforming thin structures
- Deformation from water hammer
- Real-world example

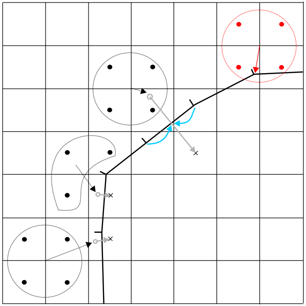
Adaptive Lattice Boltzmann method with FSI
- Adaptive LBM
- Realistic static embedded geometries
- Simulation of wind turbines

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]

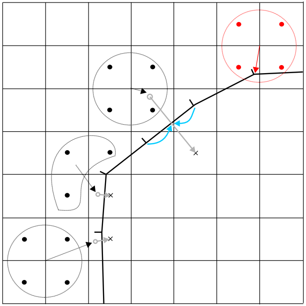
Construction of coupling data

- Moving boundary/interface is treated as a moving contact discontinuity and represented by level set [Fedkiw, 2002][Arienti et al., 2003]
- One-sided construction of mirrored ghost cell and new FEM nodal point values



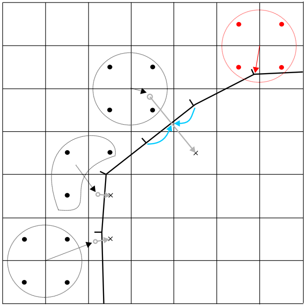
Construction of coupling data

- Moving boundary/interface is treated as a moving contact discontinuity and represented by level set [Fedkiw, 2002][Arienti et al., 2003]
- One-sided construction of mirrored ghost cell and new FEM nodal point values
- FEM ansatz-function interpolation to obtain intermediate surface 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]
- One-sided construction of mirrored ghost cell and new FEM nodal point values
- FEM ansatz-function interpolation to obtain intermediate surface values



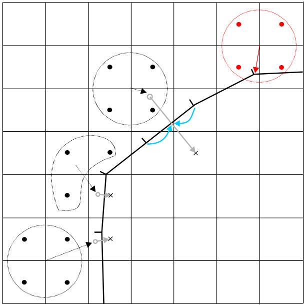
Coupling conditions on interface

$$\left. \begin{array}{rcl} u_n^S & = & u_n^F \\ \sigma_{nn}^S & = & p^F \\ \sigma_{nm}^S & = & 0 \end{array} \right|_{\mathcal{I}}$$

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]
- One-sided construction of mirrored ghost cell and new FEM nodal point values
- FEM ansatz-function interpolation to obtain intermediate surface values
- Explicit coupling possible if geometry and velocities are prescribed for the more compressible medium [Specht, 2000]

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



Coupling conditions on interface

$$\left.\begin{array}{rcl} u_n^S & = & u_n^F \\ \sigma_{nn}^S & = & p^F \\ \sigma_{nm}^S & = & 0 \end{array}\right|_{\mathcal{I}}$$

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 $l_c \leq l_{\mathrm{max}}$

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 $l_c \leq l_{\mathrm{max}}$
  - 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 $l_c \le l_{\max}$
  - 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

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 $l_c \leq l_{\max}$
  - 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
- 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)
```

$t := t + \Delta t_l$

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)

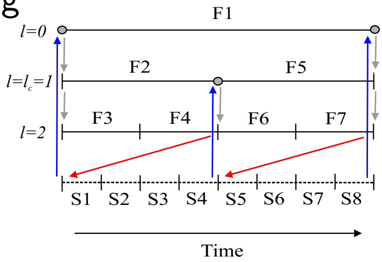
Repeat r_l times
  Set ghost cells of $\mathbf{Q}^l(t)$
  CPT(φ^l, C^l, $\mathcal{I}$, δ_l)
  If time to regrid?
    Regrid(l)
  UpdateLevel($\mathbf{Q}^l$, φ^l, C^l, $\mathbf{u}^S|_{\mathcal{I}}$, Δt_l)
  If level $l+1$ exists?
    Set ghost cells of $\mathbf{Q}^l(t+\Delta t_l)$
    AdvanceLevel($l+1$)
    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

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

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` $l = 0, \cdots, L$

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

`AdvanceLevel(0)`

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

Fluid and solid update / exchange of time steps

FluidStep( )

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

$\Delta\tau_F := \min_{l=0,\cdots,l_{\max}} (R_l \cdot \texttt{StableFluidTimeStep}(l)\texttt{,}\ \Delta\tau_S)$
$\Delta t_l := \Delta\tau_F / R_l$ 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
$\quad t_{\mathrm{end}} := t + \Delta\tau_S / R_{l_c}\texttt{,}\ \ \Delta t := \Delta\tau_S / (K R_{l_c})$

- 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( )

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

SolidStep( )

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

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

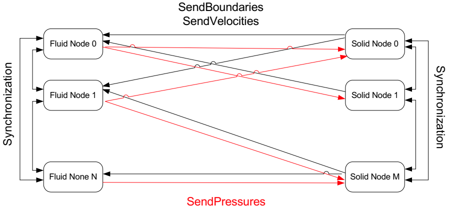
Parallelization strategy for coupled simulations

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:

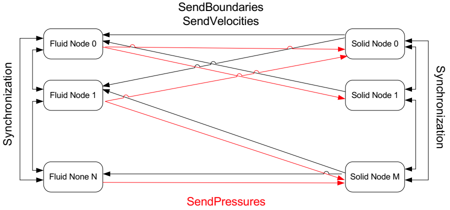
- Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes
- Setting of ghost cell values becomes strictly local operation



Parallelization strategy for coupled simulations

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

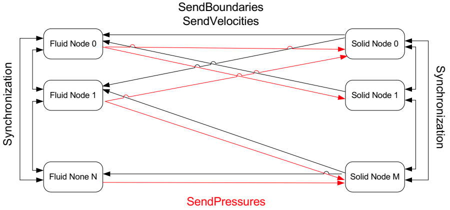
- Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes
- Setting of ghost cell values becomes strictly local operation
- Construct new nodal values strictly local on fluid nodes and transfer them back to solid nodes
- Only surface data is transfered



Parallelization strategy for coupled simulations

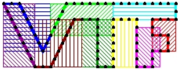
Coupling of an Eulerian FV fluid Solver and a Lagrangian FEM Solver:

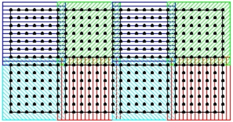
- Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes
- Setting of ghost cell values becomes strictly local operation
- Construct new nodal values strictly local on fluid nodes and transfer them back to solid nodes
- Only surface data is transfered
- Asynchronous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



Eulerian/Lagrangian communication module

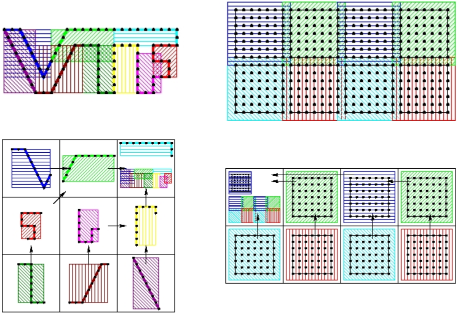
1. Put bounding boxes around each solid processors piece of the boundary and around each fluid processors grid





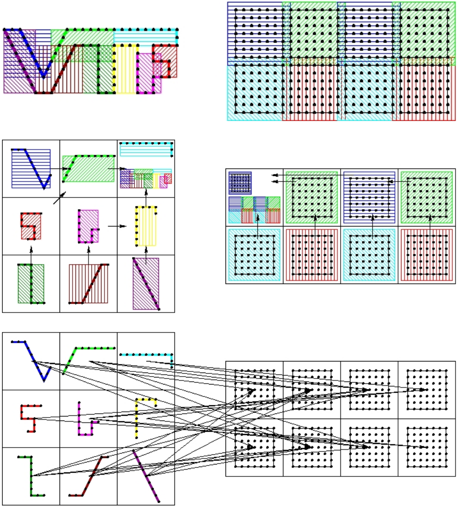
Eulerian/Lagrangian communication module

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

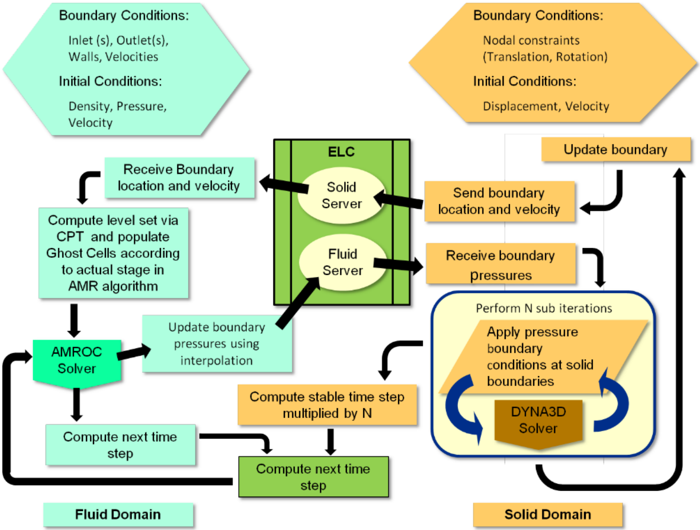


Eulerian/Lagrangian communication module

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



Coupling elements

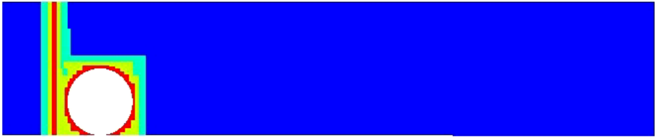


Lift-up of a spherical body

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

Schlieren plot of density

Refinement levels



vtf/amroc/clawpack/applications/euler/2d/SphereLiftOff

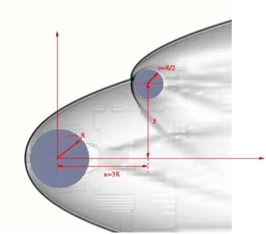
Proximal bodies in hypersonic flow

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



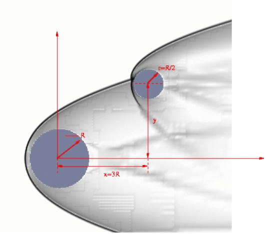
Proximal bodies in hypersonic flow

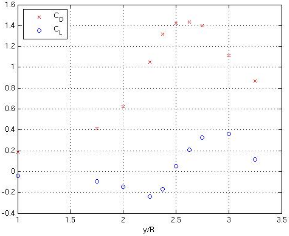
Flow modeled by Euler equations for a single polytropic gas with $p = (\gamma - 1)\,\rho e$

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

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

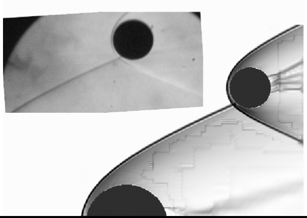
Static force measurements, $M = 10$:
[Laurence et al., 2007]

- 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

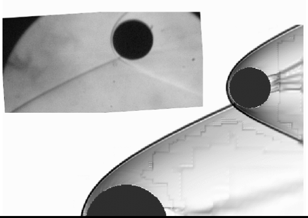
Static force measurements, $M = 10$:
[Laurence et al., 2007]

- 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

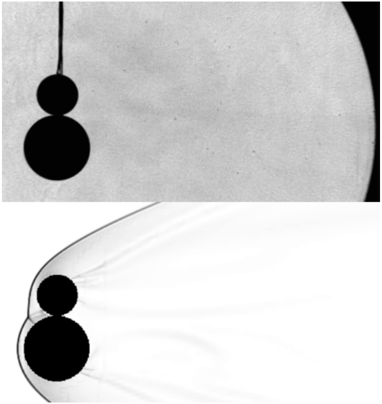
- Comparison with experimental results: 3 additional levels, $\sim 2000\,\mathrm{h}$ CPU

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



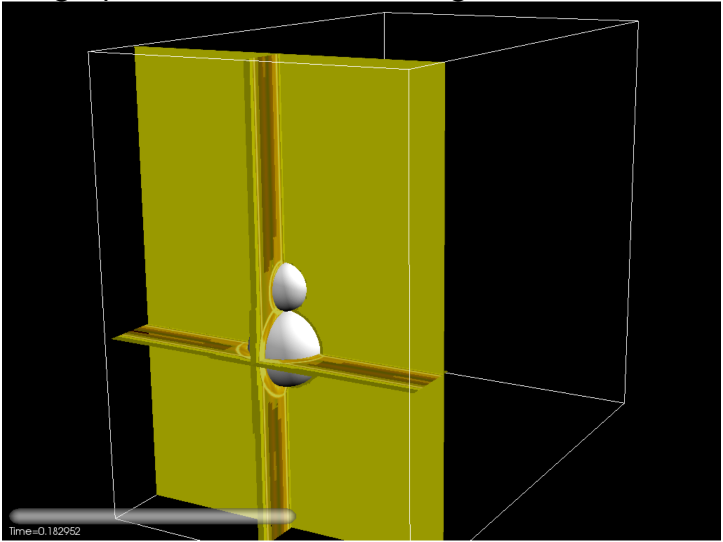
Dynamic motion, $M = 4$:

- 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



vtf/amroc/clawpack/applications/euler/3d/Spheres

Treatment of thin structures

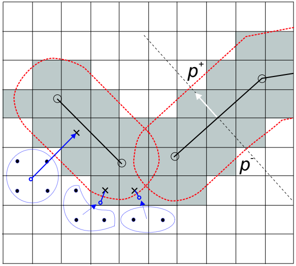
- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method

Treatment of thin structures

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

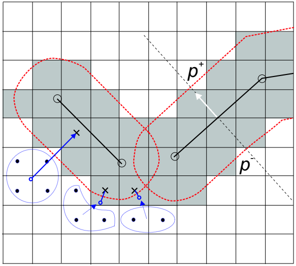
Treatment of thin structures

- 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

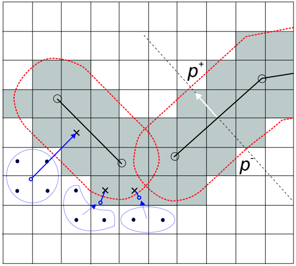
- Thin boundary structures or lower-dimensional shells require “thickening” to apply embedded boundary method
- Unsigned distance level set function φ
- Treat cells with $0 < \varphi < d$ as ghost fluid cells



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

Treatment of thin structures

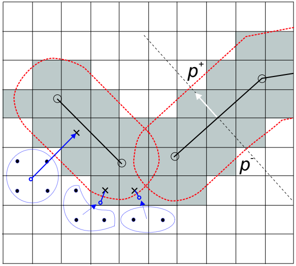
- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method
- Unsigned distance level set function φ
- Treat cells with $0 < \varphi < d$ as ghost fluid cells



- 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

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

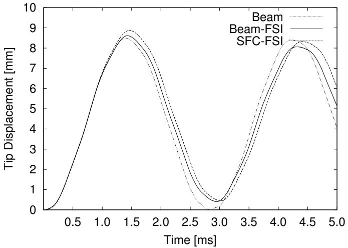
to verify FSI algorithms

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

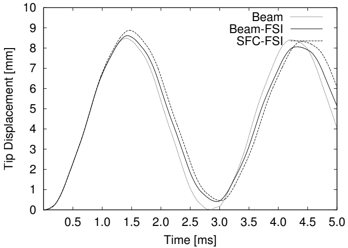
FSI verification by elastic vibration

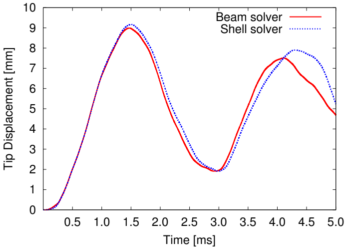
- Thin steel plate (thickness $h = 1\,\mathrm{mm}$, length $50\,\mathrm{mm}$), clamped at lower end
- $\rho_s = 7600\,\mathrm{kg/m^3}$, $E = 220\,\mathrm{GPa}$, $I = h^3/12$, $\nu = 0.3$
- Modeled with beam solver (101 points) and thin-shell FEM solver (325 triangles) by F. Cirak
- Left: Coupling verification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

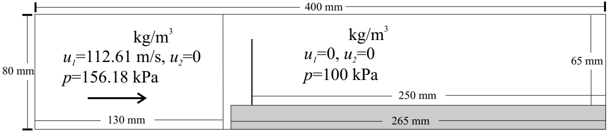




Shock-driven elastic panel motion

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

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

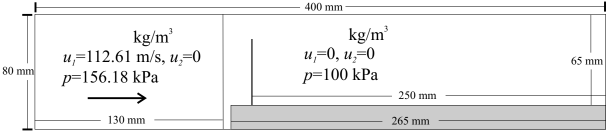


- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$

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

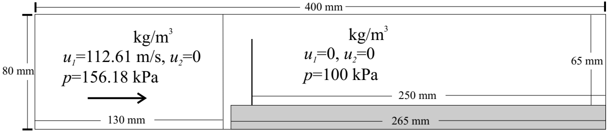
    vtf/fsi/beam-amroc/VibratingBeam - Fluid, Solid
  - FEM-FSI: 322 h CPU on 14 fluid CPU + 2 solid CPU

    vtf/fsi/sfc-amroc/VibratingPanel - Fluid, Solid

Shock-driven elastic panel motion

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

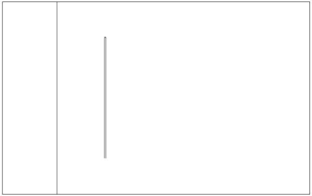
- Forward facing step geometry, fixed walls everywhere except at inflow



- 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

    vtf/fsi/beam-amroc/VibratingBeam - Fluid, Solid
  - FEM-FSI: 322 h CPU on 14 fluid CPU + 2 solid CPU

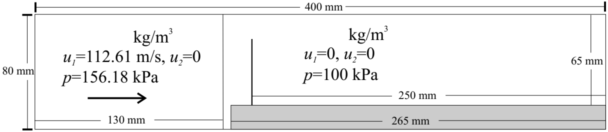
    vtf/fsi/sfc-amroc/VibratingPanel - Fluid, Solid



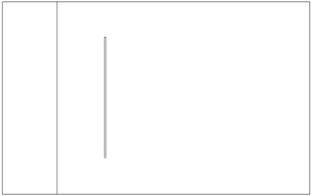
Shock-driven elastic panel motion

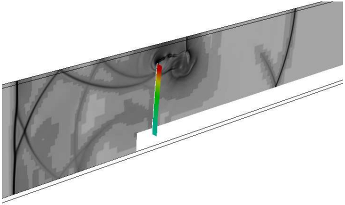
Test case suggested by [Giordano et al., 2005]

- 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
    vtf/fsi/beam-amroc/VibratingBeam - Fluid, Solid
  - FEM-FSI: $322\,\mathrm{h}$ CPU on 14 fluid CPU + 2 solid CPU
    vtf/fsi/sfc-amroc/VibratingPanel - Fluid, Solid



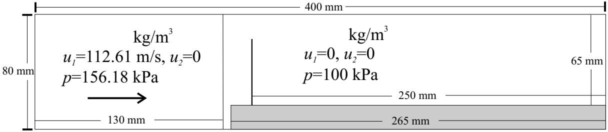


$t = 0.43\ \mathrm{ms}$ after impact

Shock-driven elastic panel motion

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

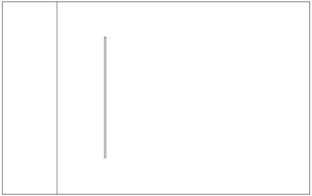
- Forward facing step geometry, fixed walls everywhere except at inflow

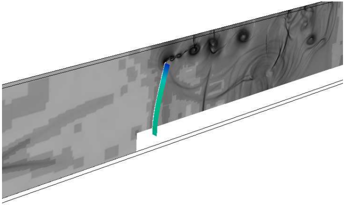


- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$
- Intel 3.4GHz Xeon dual processors, GB Ethernet interconnect
  - Beam-FSI: 12.25 h CPU on 3 fluid CPU + 1 solid CPU

    vtf/fsi/beam-amroc/VibratingBeam - Fluid, Solid
  - FEM-FSI: 322 h CPU on 14 fluid CPU + 2 solid CPU

    vtf/fsi/sfc-amroc/VibratingPanel - Fluid, Solid





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

Detonation-driven plastic deformation

Chapman-Jouguet detonation in a tube filled with a stoichiometric ethylene and oxygen ($\mathrm{C_2H_4} + 3\,\mathrm{O_2}$, $295\,\mathrm{K}$) 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 ($\mathrm{C_2H_4} + 3\,\mathrm{O_2}$, $295\,\mathrm{K}$) 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

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

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 ($\mathrm{C_2H_4} + 3\,\mathrm{O_2}$, $295\,\mathrm{K}$) mixture. Euler equations with single exothermic reaction $A \longrightarrow B$

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

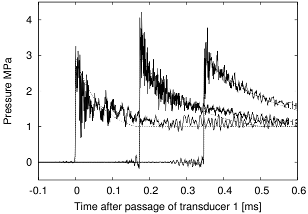
with

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

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 $\gamma = 1.24$, $p_{\mathrm{CJ}} = 3.3\,\mathrm{MPa}$, $D_{\mathrm{CJ}} = 2376\,\mathrm{m/s}$
  - 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

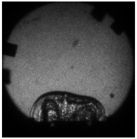
Tube with flaps

- 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

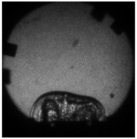


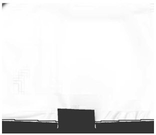
0.030 ms

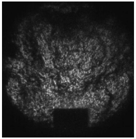
Tube with flaps

- Fluid: VanLeer FVS
  - Detonation model with $\gamma = 1.24$, $p_{\rm CJ} = 3.3\,\mathrm{MPa}$, $D_{\rm CJ} = 2376\,\mathrm{m/s}$
  - 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 $t_{end} = 450\,\mu\mathrm{s}$

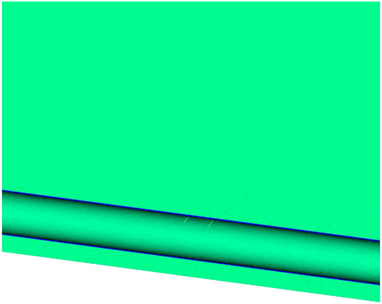

0.032 ms


0.030 ms


0.212 ms

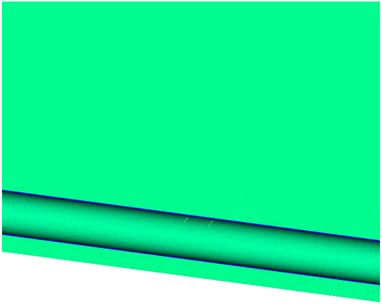

0.210 ms

Tube with flaps: results

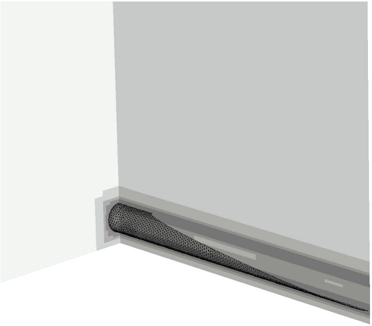


Fluid density and diplacement in y-direction in solid

Tube with flaps: results



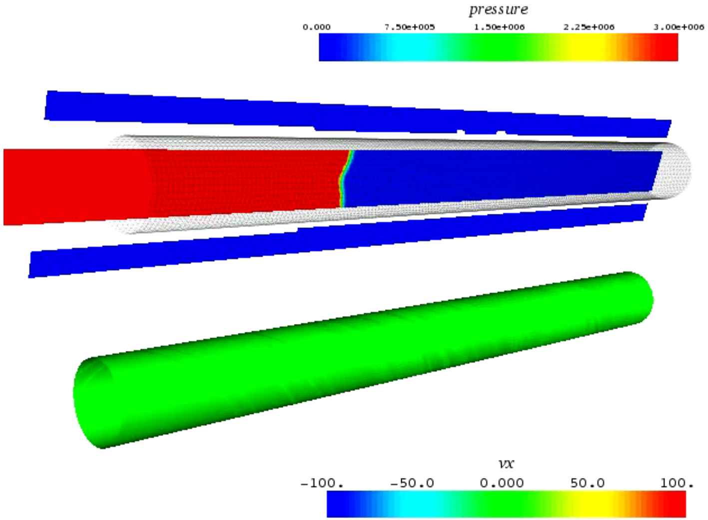
Fluid density and diplacement in y-direction in solid



Schlieren plot of fluid density on refinement levels

[Cirak et al., 2007] vtf/fsi/sfc-amroc/TubeCJBurnFlaps - Fluid, Solid

Coupled fracture simulation



vtf/fsi/sfc-amroc/TubeCJBurnFrac - Fluid, Solid

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]

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

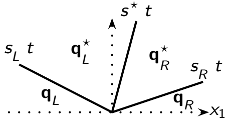
$$\frac{\partial}{\partial t}\left(\frac{1}{\gamma - 1}\right) + u_n \frac{\partial}{\partial x_n}\left(\frac{1}{\gamma - 1}\right) = 0 \,, \quad \frac{\partial}{\partial t}\left(\frac{\gamma p_\infty}{\gamma - 1}\right) + u_n \frac{\partial}{\partial x_n}\left(\frac{\gamma p_\infty}{\gamma - 1}\right)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

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

Approximate Riemann solver

Use HLLC approach because of robustness and positivity preservation

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

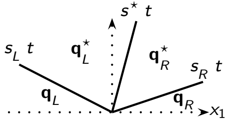


Wave speed estimates [Davis, 1988] $s_L = \min\{u_{1,L} - c_L, u_{1,R} - c_R\}$, $s_R = \max\{u_{1,L} + c_L, u_{1,R} + c_R\}$

Approximate Riemann solver

Use HLLC approach because of robustness and positivity preservation

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



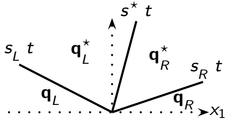
Wave speed estimates [Davis, 1988] $s_L = \min\{u_{1,L} - c_L, u_{1,R} - c_R\}$, $s_R = \max\{u_{1,L} + c_L, u_{1,R} + c_R\}$
Unkown state [Toro et al., 1994]

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

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]

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

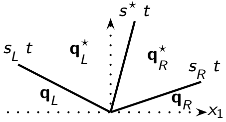
$$\mathbf{q}_\tau^\star = \left[\eta, \eta s^\star, \eta u_2, \eta\left[\frac{(\rho E)_\tau}{\rho_\tau} + (s^\star - u_{1,\tau})\left(s_\tau + \frac{p_\tau}{\rho_\tau(s_\tau - u_{1,\tau})}\right)\right], \frac{1}{\gamma_\tau - 1}, \frac{\gamma_\tau p_{\infty,\tau}}{\gamma_\tau - 1}\right]^T$$

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

Approximate Riemann solver

Use HLLC approach because of robustness and positivity preservation

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



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]

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

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

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

Evaluate waves as $\mathcal{W}_1 = \mathbf{q}_L^\star - \mathbf{q}_L$, $\mathcal{W}_2 = \mathbf{q}_R^\star - \mathbf{q}_L^\star$, $\mathcal{W}_3 = \mathbf{q}_R - \mathbf{q}_R^\star$ 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

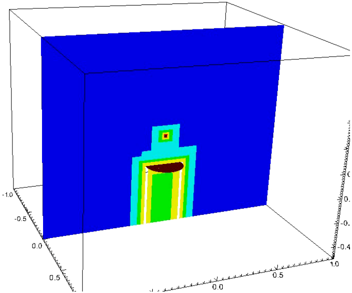
- Air: $\gamma^A = 1.4$, $p_\infty^A = 0$, $\rho^A = 1.29\,\mathrm{kg/m^3}$
- Water: $\gamma^W = 7.415$, $p_\infty^W = 296.2\,\mathrm{MPa}$, $\rho^W = 1027\,\mathrm{kg/m^3}$
- Cavitation modeling with pressure cut-off model at $p = -1\,\mathrm{MPa}$
- 3D simulation of deformation of air backed aluminum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

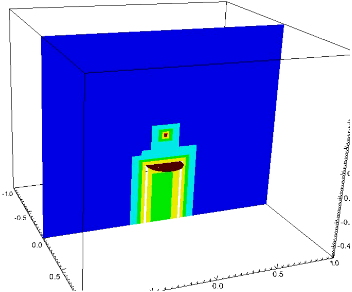


Underwater explosion simulation

- 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



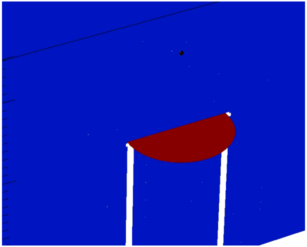
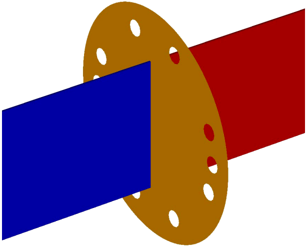


Plate in underwater shocktube

- 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

vtf/fsi/sfc-amroc/WaterBlastFracture - Fluid, Solid

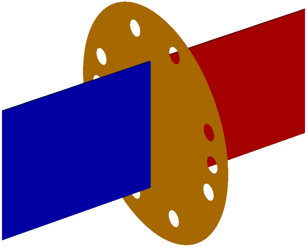


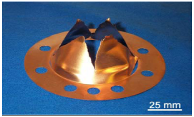
$p_0 = 64\,\mathrm{MPa}$

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

vtf/fsi/sfc-amroc/WaterBlastFracture - Fluid, Solid



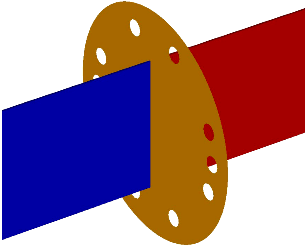


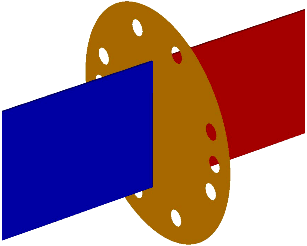
$p_0 = 64\,\mathrm{MPa}$

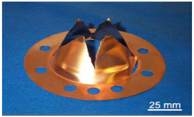
Plate in underwater shocktube

- 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

vtf/fsi/sfc-amroc/WaterBlastFracture - Fluid, Solid







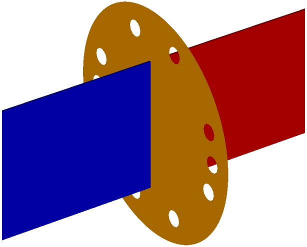
$p_0 = 64\,\mathrm{MPa}$

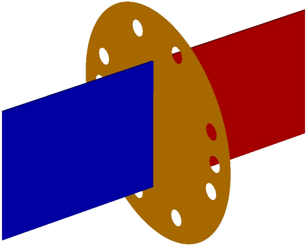
$p_0 = 173\,\mathrm{MPa}$

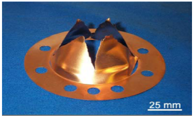
Plate in underwater shocktube

- 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

vtf/fsi/sfc-amroc/WaterBlastFracture - Fluid, Solid





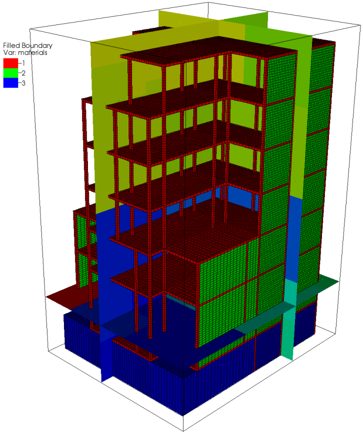


$p_0 = 64\,\mathrm{MPa}$

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

Blast explosion in a multistory building

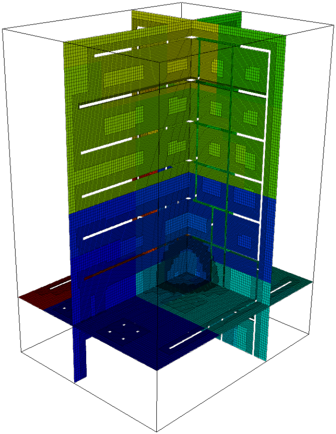
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

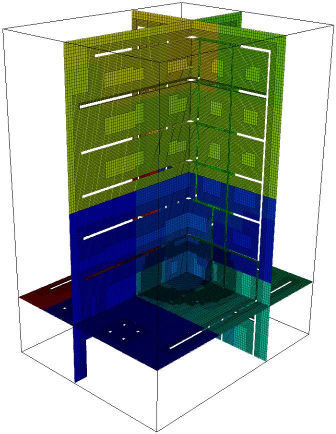
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\rm fsi} = 1$, $k = 1$
- 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

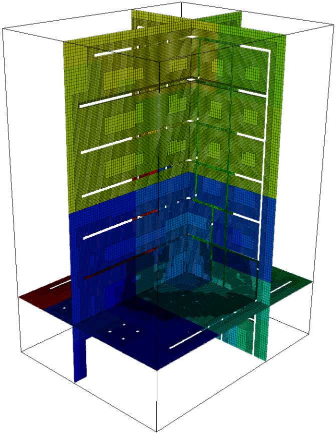
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

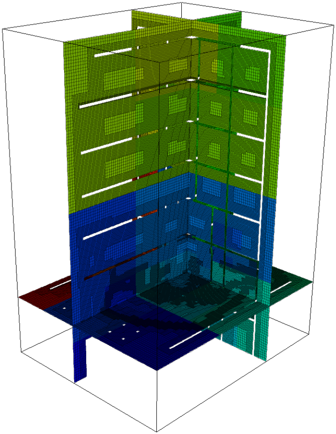
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

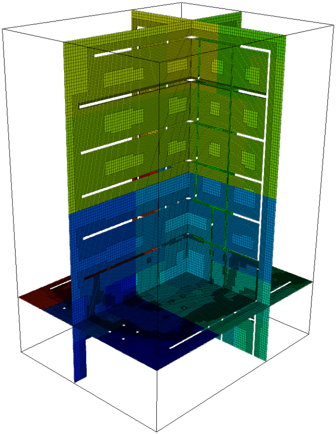
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

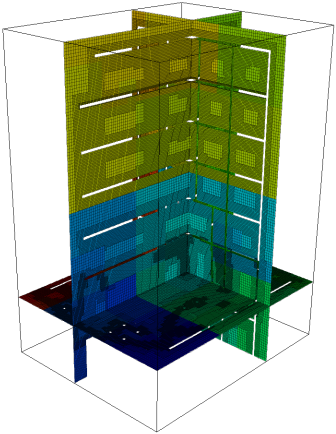
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

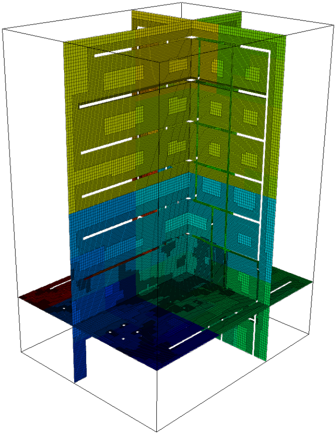
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

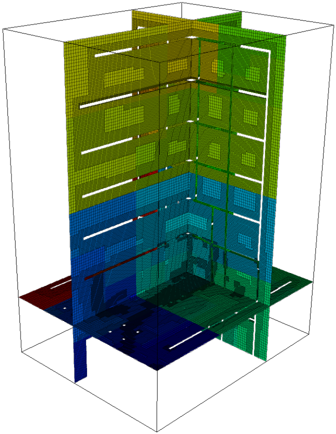
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

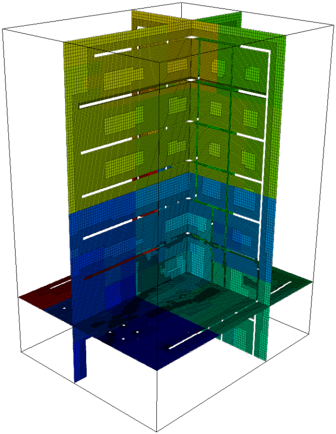
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

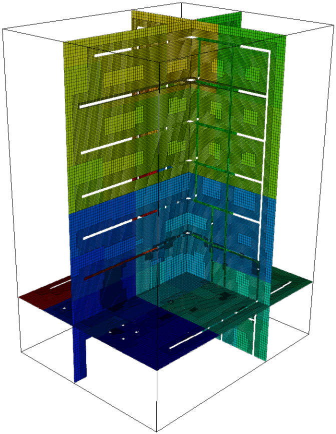
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

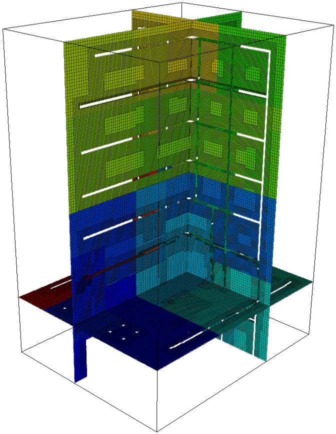
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

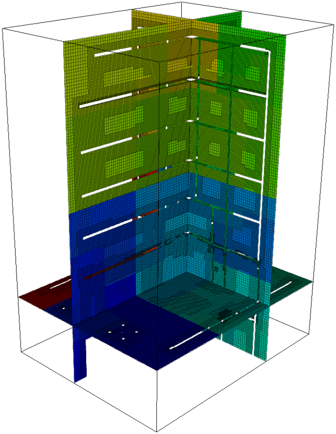
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

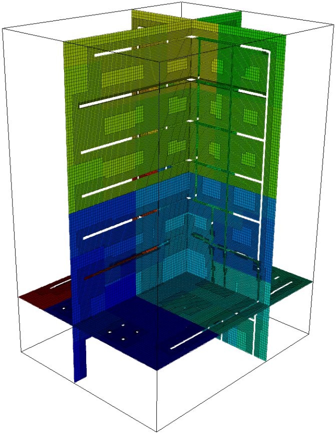
- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ seven-story building similar to [Luccioni et al., 2004]
- Spherical energy deposition $\equiv 400\,\mathrm{kg}$ TNT, $r = 0.5\,\mathrm{m}$ in lobby of building
- SAMR: $80 \times 120 \times 90$ base level, three additional levels $r_{1,2} = 2$, $l_{\mathrm{fsi}} = 1$, $k = 1$
- 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

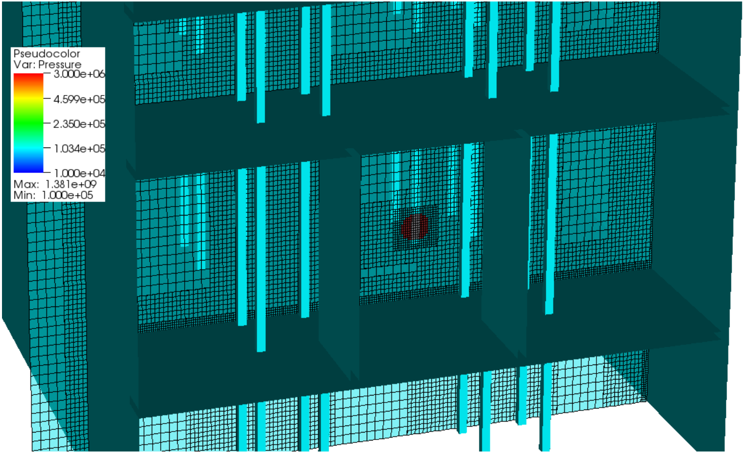
Blast explosion in a multistory building

- $20\,\mathrm{m} \times 40\,\mathrm{m} \times 25\,\mathrm{m}$ 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]



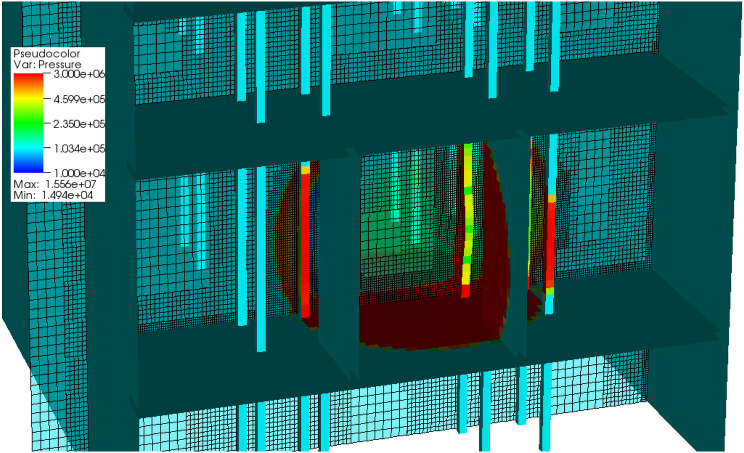
	ρ_s [kg/m^3]	σ_0 [MPa]	E_T [GPa]	β	K [GPa]	G [GPa]	$\bar{\epsilon}^p$	p_f [MPa]
Columns	2010	50	11.2	1.0	21.72	4.67	0.02	-30
Walls	2010	25	11.2	1.0	6.22	4.67	0.01	-15

Blast explosion in a multistory building – II

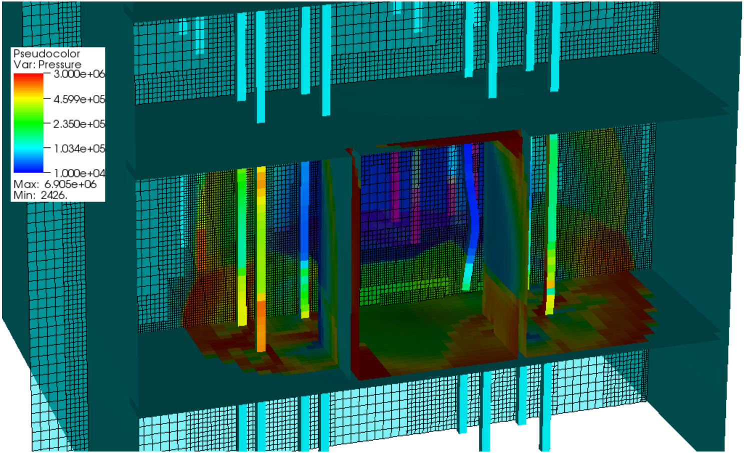


$t = 0$

Blast explosion in a multistory building – II

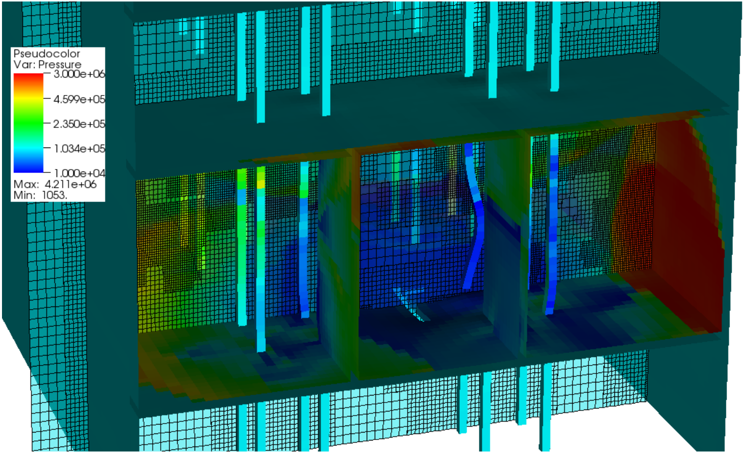


Blast explosion in a multistory building – II

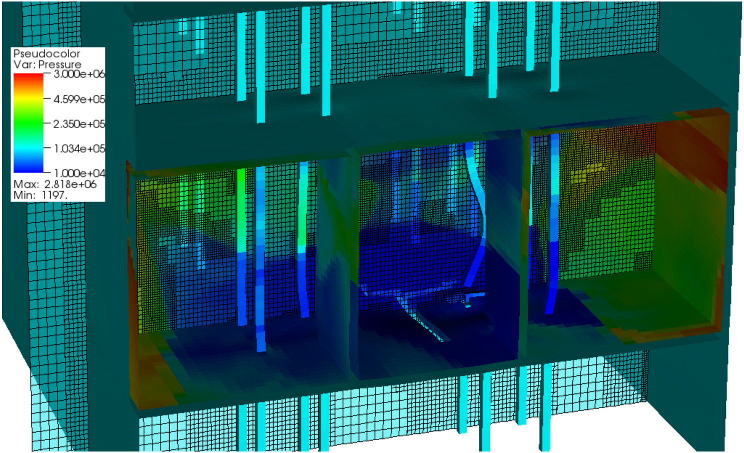


$t = 6.1\,\mathrm{ms}$

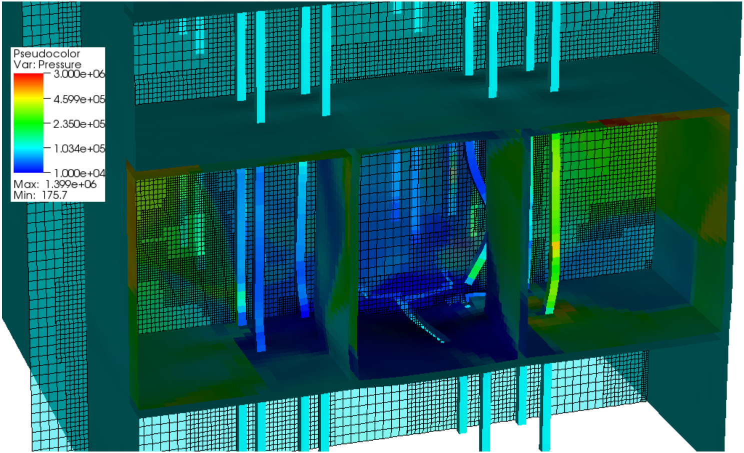
Blast explosion in a multistory building – II



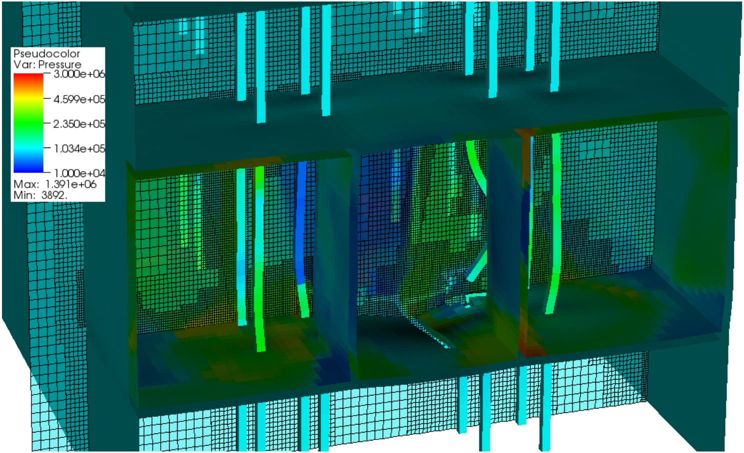
Blast explosion in a multistory building – II



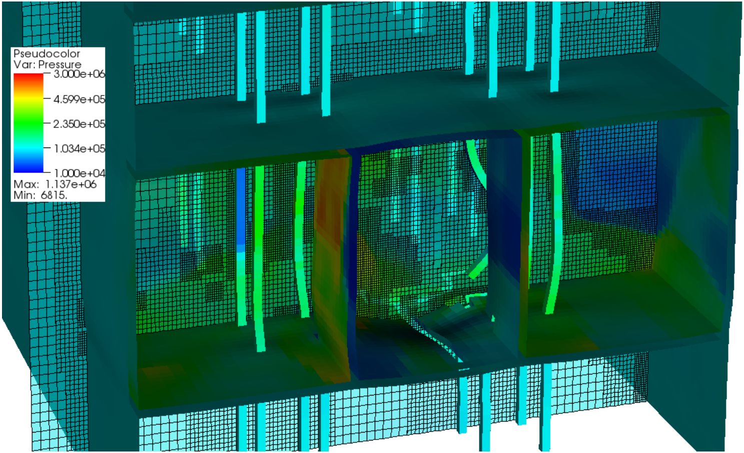
Blast explosion in a multistory building – II



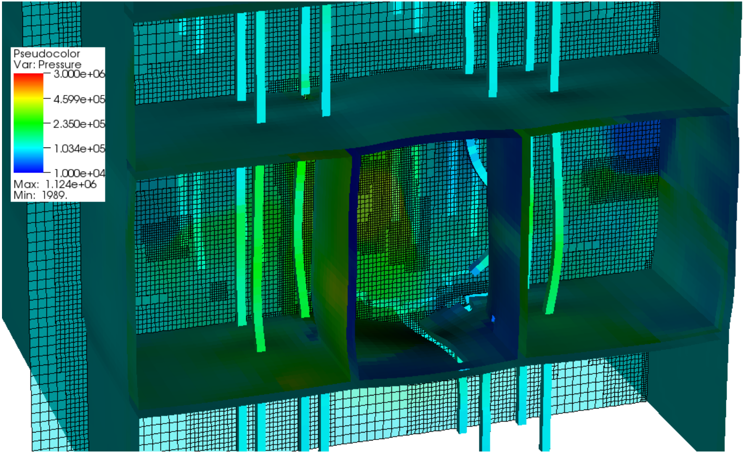
Blast explosion in a multistory building – II



Blast explosion in a multistory building – II

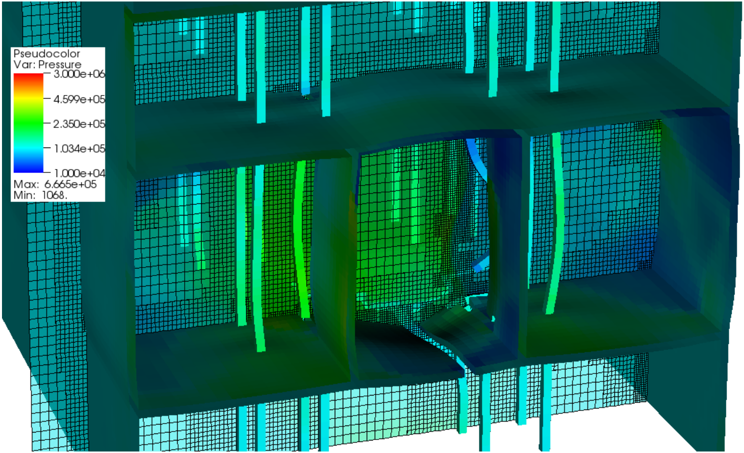


Blast explosion in a multistory building – II

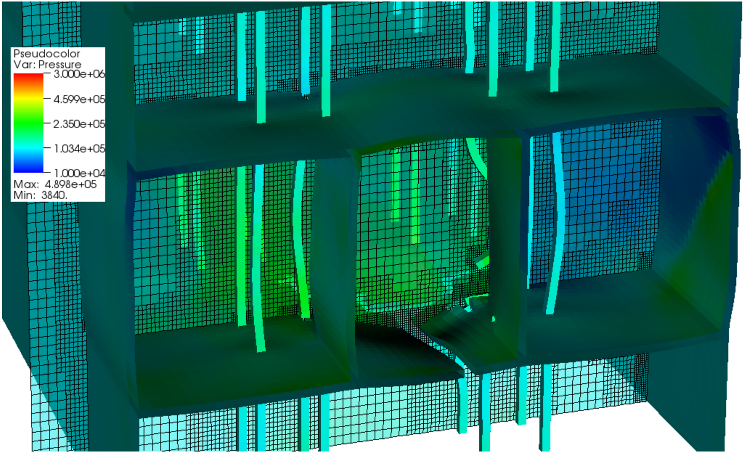


$t = 29.2\,\mathrm{ms}$

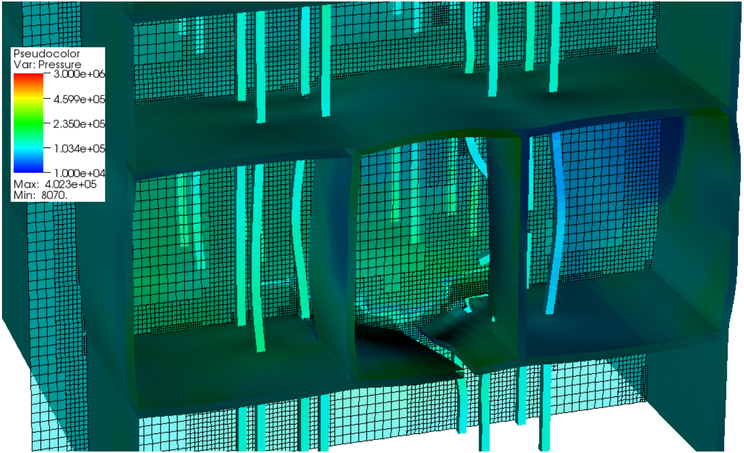
Blast explosion in a multistory building – II



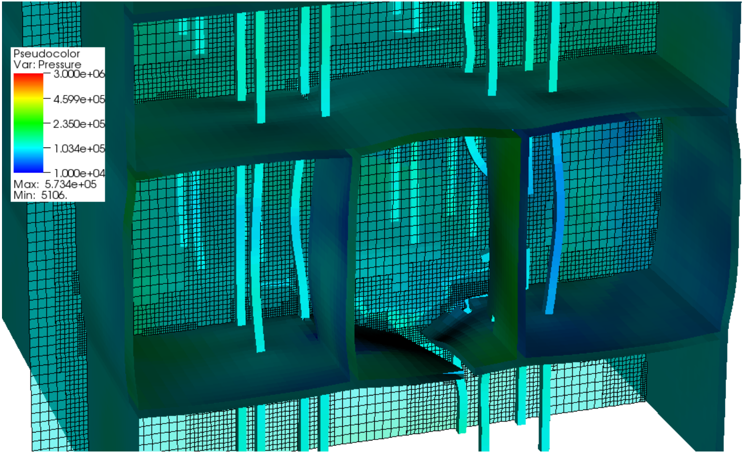
Blast explosion in a multistory building – II



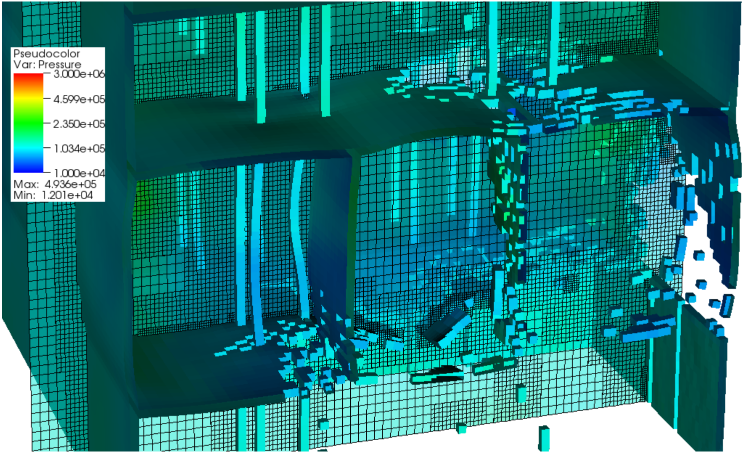
Blast explosion in a multistory building – II



Blast explosion in a multistory building – II



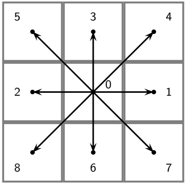
Blast explosion in a multistory building – II



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

Lattice Boltzmann method

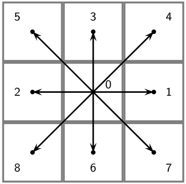
Boltzmann equation: $\partial_t f + \mathbf{u} \cdot \nabla f = \omega(f^{eq} - f)$
Two-dimensional LBM for weakly compressible flows
Formulated on FV grids! ($\rightarrow$ boundary conditions!)



$$\rho(\mathbf{x},t) = \sum_{\alpha=0}^{8} f_\alpha(\mathbf{x},t), \quad \rho(\mathbf{x},t)u_i(\mathbf{x},t) = \sum_{\alpha=0}^{8} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x},t)$$

Lattice Boltzmann method

Boltzmann equation: $\partial_t f + \mathbf{u} \cdot \nabla f = \omega(f^{eq} - f)$
Two-dimensional LBM for weakly compressible flows
Formulated on FV grids! ($\rightarrow$ boundary conditions!)

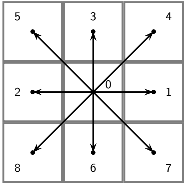


$$\rho(\mathbf{x}, t) = \sum_{\alpha=0}^{8} f_\alpha(\mathbf{x}, t), \quad \rho(\mathbf{x}, t) u_i(\mathbf{x}, t) = \sum_{\alpha=0}^{8} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x}, t)$$

1.) Transport step $\mathcal{T}$: $\tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t)$

Lattice Boltzmann method

Boltzmann equation: $\partial_t f + \mathbf{u} \cdot \nabla f = \omega(f^{eq} - f)$
Two-dimensional LBM for weakly compressible flows
Formulated on FV grids! ($\rightarrow$ boundary conditions!)



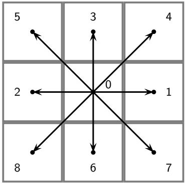
$$\rho(\mathbf{x},t) = \sum_{\alpha=0}^{8} f_\alpha(\mathbf{x},t), \quad \rho(\mathbf{x},t) u_i(\mathbf{x},t) = \sum_{\alpha=0}^{8} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x},t)$$

1.) Transport step $\mathcal{T}$: $\tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x},t)$
2.) Collision step $\mathcal{C}$:

$$f_\alpha(\cdot, t+\Delta t) = \tilde{f}_\alpha(\cdot, t+\Delta t) + \omega \Delta t \left(\tilde{f}_\alpha^{eq}(\cdot, t+\Delta t) - \tilde{f}_\alpha(\cdot, t+\Delta t) \right)$$

Lattice Boltzmann method

Boltzmann equation: $\partial_t f + \mathbf{u} \cdot \nabla f = \omega(f^{eq} - f)$
Two-dimensional LBM for weakly compressible flows
Formulated on FV grids! ($\rightarrow$ boundary conditions!)



$$\rho(\mathbf{x}, t) = \sum_{\alpha=0}^{8} f_\alpha(\mathbf{x}, t), \quad \rho(\mathbf{x}, t) u_i(\mathbf{x}, t) = \sum_{\alpha=0}^{8} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x}, t)$$

1.) Transport step $\mathcal{T}$: $\tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t)$
2.) Collision step $\mathcal{C}$:

$$f_\alpha(\cdot, t + \Delta t) = \tilde{f}_\alpha(\cdot, t + \Delta t) + \omega \Delta t \left(\tilde{f}^{eq}_\alpha(\cdot, t + \Delta t) - \tilde{f}_\alpha(\cdot, t + \Delta t) \right)$$

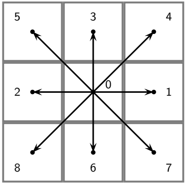
with equilibrium function

$$f^{eq}_\alpha(\rho, \mathbf{u}) = \rho t_\alpha \left[1 + \frac{\mathbf{e}_\alpha \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^4} \right]$$

mit $t_\alpha = \frac{1}{9}\left\{4, 1, 1, 1, \frac{1}{4}, \frac{1}{4}, 1, \frac{1}{4}, \frac{1}{4}\right\}$

Lattice Boltzmann method

Boltzmann equation: $\partial_t f + \mathbf{u} \cdot \nabla f = \omega(f^{eq} - f)$
Two-dimensional LBM for weakly compressible flows
Formulated on FV grids! ($\rightarrow$ boundary conditions!)



$$\rho(\mathbf{x}, t) = \sum_{\alpha=0}^{8} f_\alpha(\mathbf{x}, t), \quad \rho(\mathbf{x}, t) u_i(\mathbf{x}, t) = \sum_{\alpha=0}^{8} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x}, t)$$

1.) Transport step $\mathcal{T}$: $\tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t)$
2.) Collision step $\mathcal{C}$:

$$f_\alpha(\cdot, t + \Delta t) = \tilde{f}_\alpha(\cdot, t + \Delta t) + \omega \Delta t \left(\tilde{f}_\alpha^{eq}(\cdot, t + \Delta t) - \tilde{f}_\alpha(\cdot, t + \Delta t) \right)$$

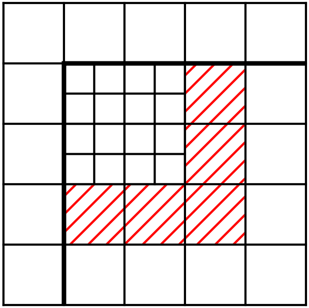
with equilibrium function

$$f_\alpha^{eq}(\rho, \mathbf{u}) = \rho t_\alpha \left[1 + \frac{\mathbf{e}_\alpha \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^4} \right]$$

mit $t_\alpha = \frac{1}{9}\left\{4, 1, 1, 1, \frac{1}{4}, \frac{1}{4}, 1, \frac{1}{4}, \frac{1}{4}\right\}$
Lattice speed of sound: $c_s = \frac{1}{\sqrt{3}} \frac{\Delta x}{\Delta t}$, pressure $p = \sum_\alpha f_\alpha^{eq} c_s^2 = \rho c_s^2 = \rho R T$
Collision frequency vs. kinematic viscosity: $\omega = \frac{c_s^2}{\nu + \Delta t c_s^2/2}$ cf. [Hähnel, 2004]

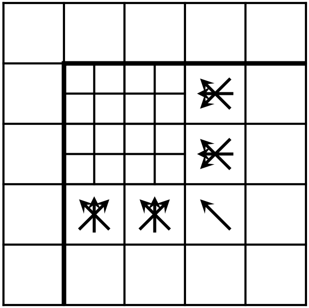
Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$



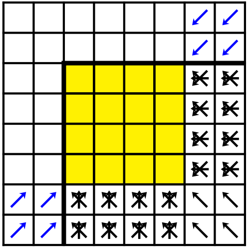
Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.



Adaptive LBM

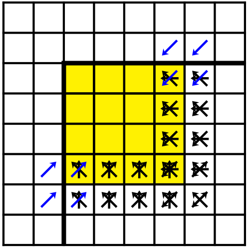
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.



$f_{\alpha,in}^{f,n}$

Adaptive LBM

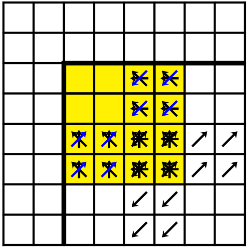
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.



$\tilde{f}_{\alpha,in}^{f,n}$

Adaptive LBM

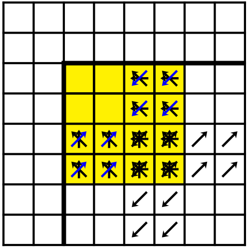
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



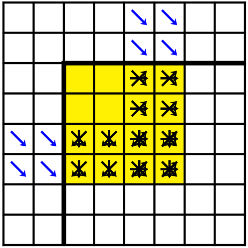
$\tilde{f}_{\alpha,in}^{f,n+1/2}$

Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



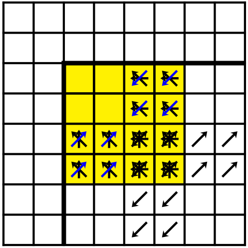
$\tilde{f}_{\alpha,in}^{f,n+1/2}$



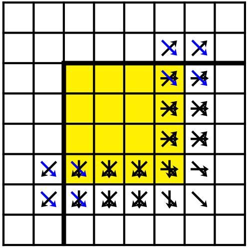
$f_{\alpha,out}^{f,n}$

Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



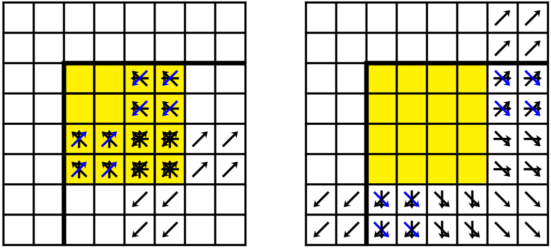
$\tilde{f}_{\alpha,in}^{f,n+1/2}$



$\tilde{f}_{\alpha,out}^{f,n}$

Adaptive LBM

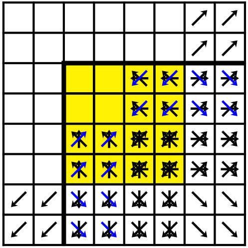
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



$\tilde{f}_{\alpha,in}^{f,n+1/2}$ $\quad$ $\tilde{f}_{\alpha,out}^{f,n+1/2}$

Adaptive LBM

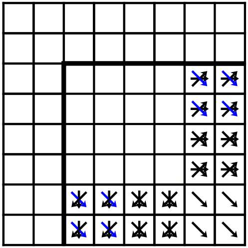
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



$\tilde{f}_{\alpha,out}^{f,n+1/2}$, $\tilde{f}_{\alpha,in}^{f,n+1/2}$

Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.

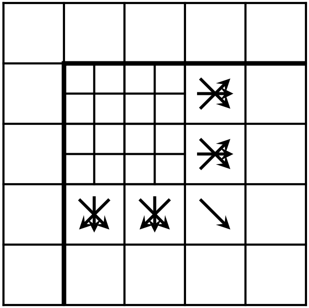


$\tilde{f}_{\alpha,out}^{f,n+1/2}$, $\tilde{f}_{\alpha,out}^{f,n}$

5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.

Adaptive LBM

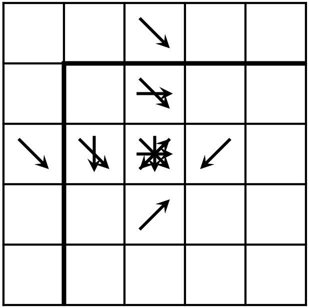
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.

Adaptive LBM

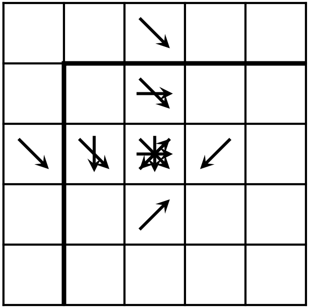
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{C}\mathcal{T}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.
6. Revert transport into halos: $\bar{f}_{\alpha,out}^{C,n} := \mathcal{T}^{-1}(\tilde{f}_{\alpha,out}^{C,n})$

Adaptive LBM

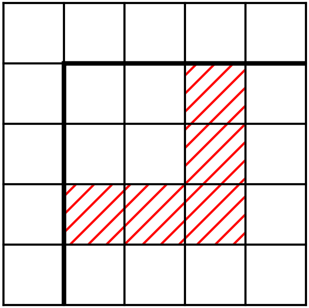
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{C}\mathcal{T}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.
6. Revert transport into halos: $\bar{f}_{\alpha,out}^{C,n} := \mathcal{T}^{-1}(\tilde{f}_{\alpha,out}^{C,n})$
7. Parallel synchronization of $f_\alpha^{C,n}, \bar{f}_{\alpha,out}^{C,n}$

Adaptive LBM

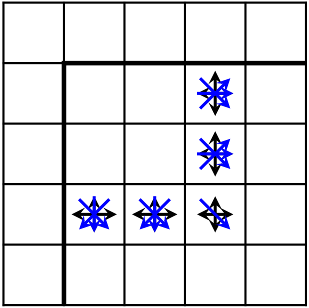
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.
6. Revert transport into halos: $\bar{f}_{\alpha,out}^{C,n} := \mathcal{T}^{-1}(\tilde{f}_{\alpha,out}^{C,n})$
7. Parallel synchronization of $f_\alpha^{C,n}$, $\bar{f}_{\alpha,out}^{C,n}$
8. Cell-wise update where correction is needed: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n}, \bar{f}_{\alpha,out}^{C,n})$

Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.

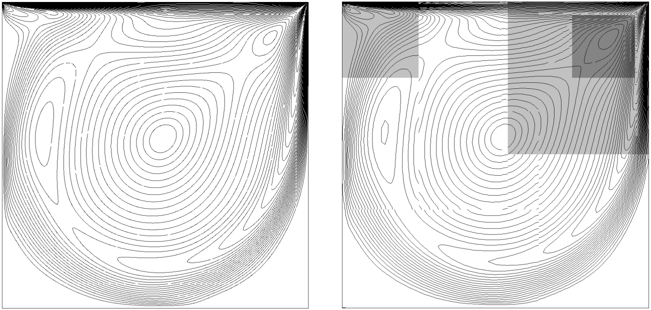


5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.
6. Revert transport into halos: $\bar{f}_{\alpha,out}^{C,n} := \mathcal{T}^{-1}(\tilde{f}_{\alpha,out}^{C,n})$
7. Parallel synchronization of $f_\alpha^{C,n}$, $\bar{f}_{\alpha,out}^{C,n}$
8. Cell-wise update where correction is needed: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n}, \bar{f}_{\alpha,out}^{C,n})$

Algorithm equivalent to [Chen et al., 2006].

Verification - driven cavity

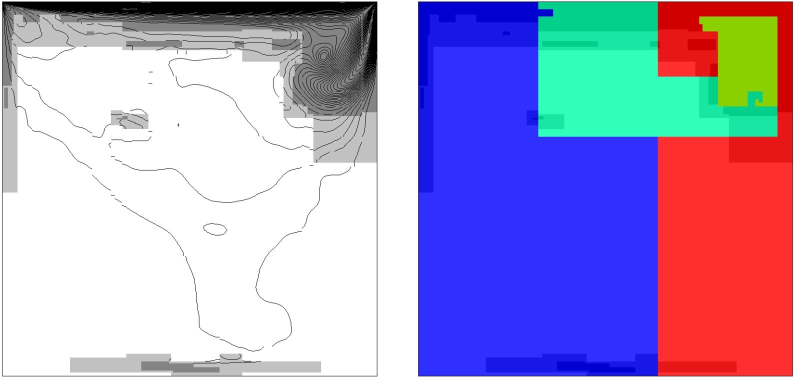
- $\mathrm{Re} = 1500$ in air, $\nu = 1.5 \cdot 10^{-5}\,\mathrm{m}^2/\mathrm{s}$, $u = 22.5\,\mathrm{m/s}$.
- Domain size $1\,\mathrm{mm} \times 1\,\mathrm{mm}$.
- Reference computation uses 800×800 lattice.
- 588,898 time steps to $t_e = 5 \cdot 10^{-3}\,\mathrm{s}$, $\sim 35\,\mathrm{h}$ CPU.
- Statically adaptive computation uses 100×100 lattice with $r_{1,2} = 2$.
- 294,452 time steps to $t_e = 5 \cdot 10^{-3}\,\mathrm{s}$ on finest level.



Isolines of density. Left: reference, right on refinement at t_e.

Driven cavity - dynamic refinement

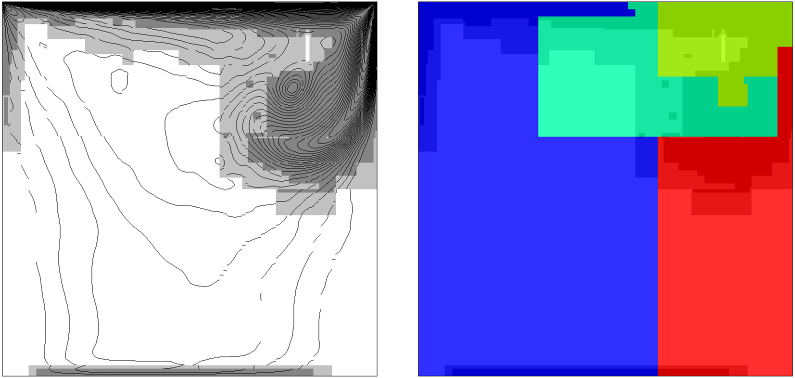
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity



Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

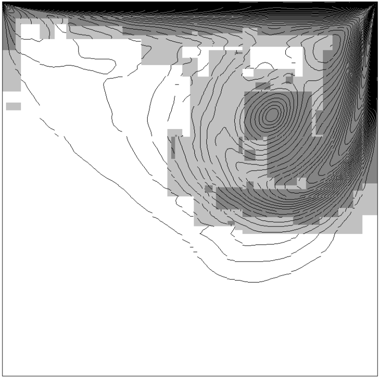
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity



Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

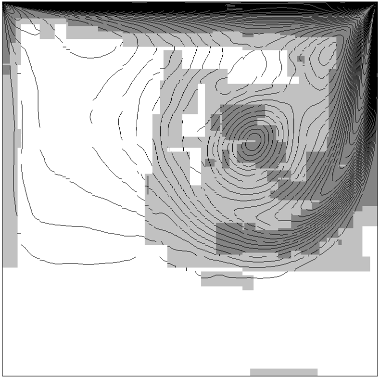
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity



Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

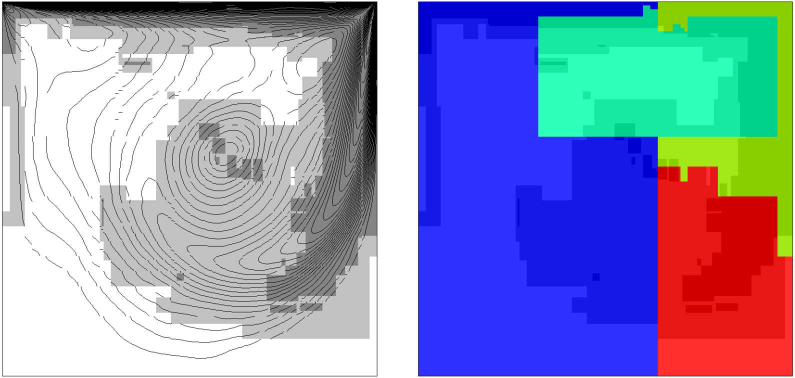
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity



Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

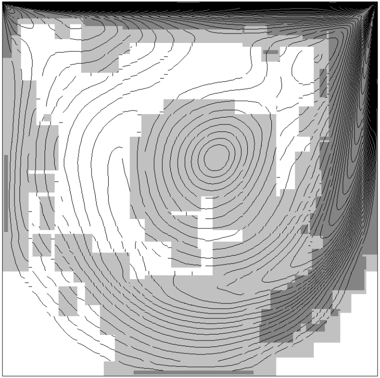
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity

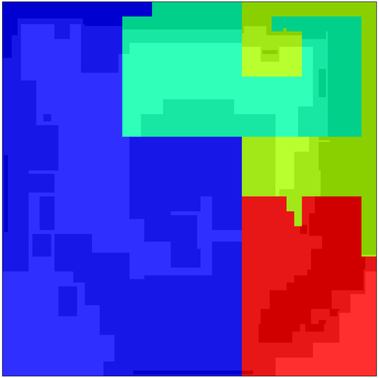


Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity

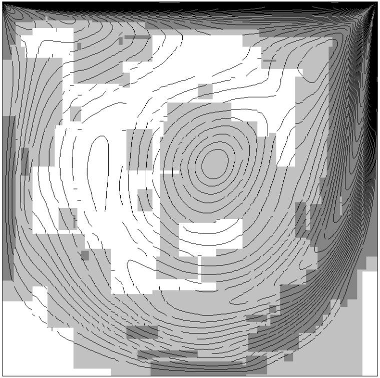


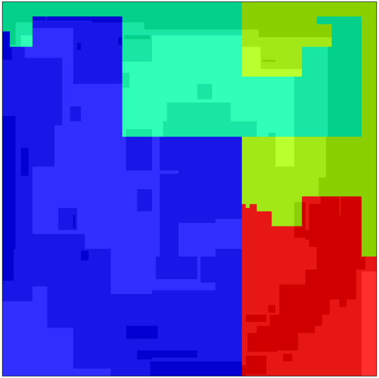


Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity

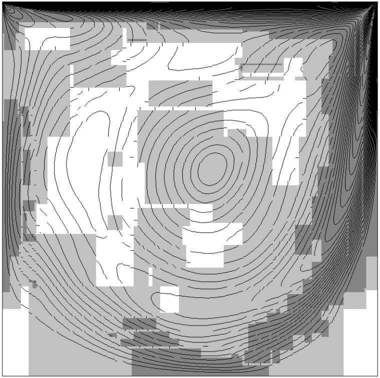




Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

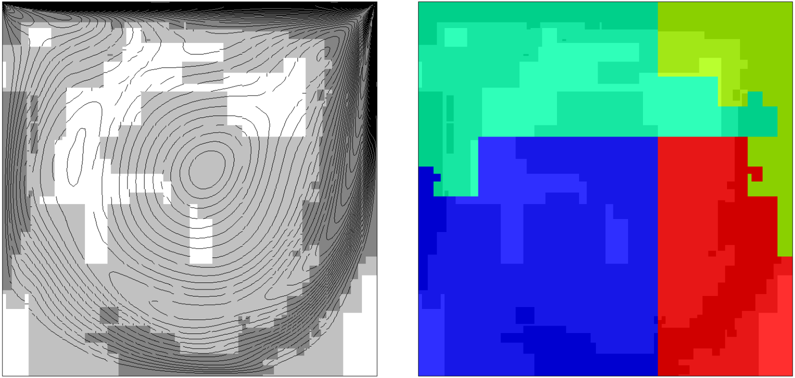
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity



Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

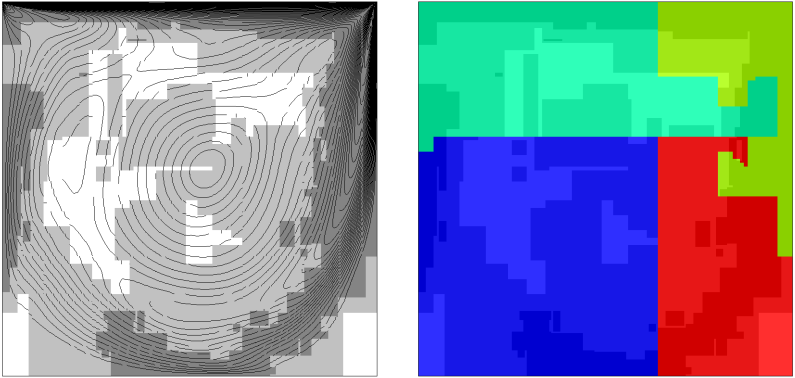
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity



Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

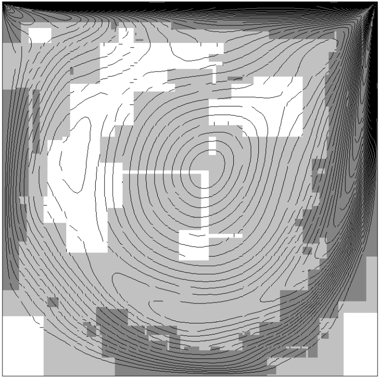
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity



Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

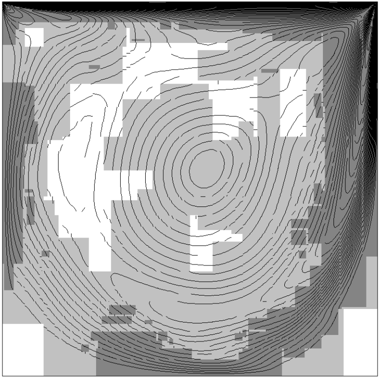
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity




Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

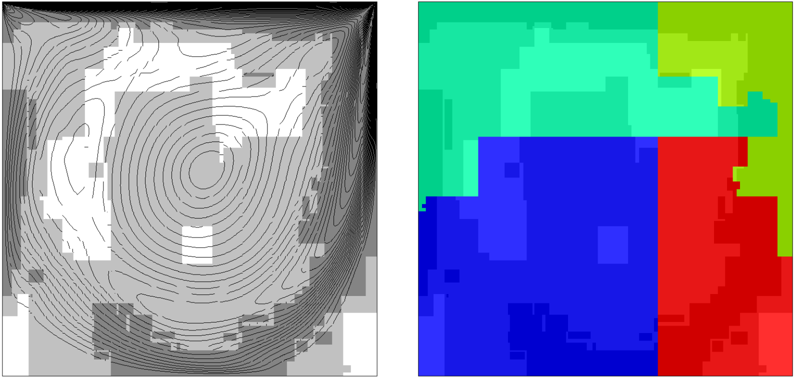
- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity



Isolines of density on refinement (left), distribution to 4 processors (right).

Driven cavity - dynamic refinement

- Dynamic refinement based on heuristic error estimation of $|\mathbf{u}|$
- Threshold intentionally chosen to show refinement evolution
- vtf/amroc/lbm/applications/Navier-Stokes/2d/Cavity



Isolines of density on refinement (left), distribution to 4 processors (right).

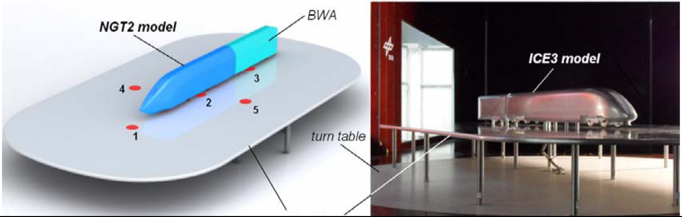
Side-wind investigation for a train model

Complex boundary consideration with level set method

- Construction of macro-values in embedded cells by inter- / extrapolation.
- Then use $f_\alpha^{eq}(\rho', \mathbf{u}')$ to construct distributions in embedded ghost cells.
- 2nd order improvements possible, cf. [Peng and Luo, 2008].

Typical DLR problem

- 1:25 train model represented with 74,670 triangles (41,226 front body, 12,398 back body, 21,006 blade)



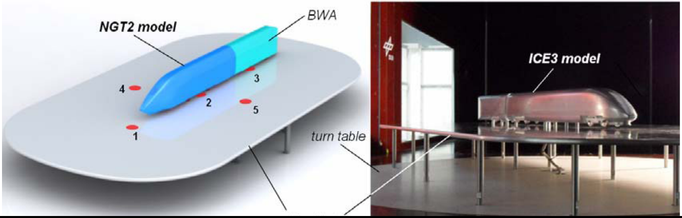
Side-wind investigation for a train model

Complex boundary consideration with level set method

- Construction of macro-values in embedded cells by inter- / extrapolation.
- Then use $f_\alpha^{eq}(\rho', \mathbf{u}')$ to construct distributions in embedded ghost cells.
- 2nd order improvements possible, cf. [Peng and Luo, 2008].

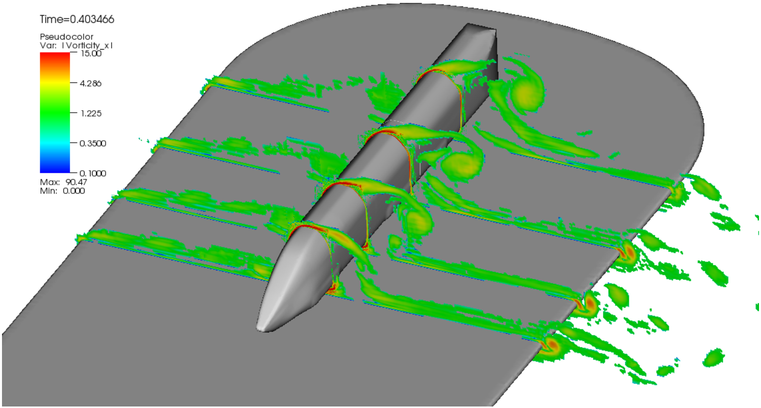
Typical DLR problem

- 1:25 train model represented with 74,670 triangles (41,226 front body, 12,398 back body, 21,006 blade)
- Wind tunnel conditions: air at room temperature with 60.25 $\mathrm{m/s}$ ($M = 0.18$), $\mathrm{Re} = 450,000$
- Systematic side wind investigation with $0 \geq \beta \geq 30^o$ to obtain lift, drag and roll moment coefficients
- vtf/amroc/lbm/applications/Navier-Stokes/3d/NGT2



Flow prediction, $\mathrm{Re} = 450,000$, $\beta = 30^o$

- Domain $10\,\mathrm{m} \times 2.4\,\mathrm{m} \times 1.6\,\mathrm{m}$
- Computation started in 3 steps. Full resolution after 5889 coarsest level steps or $t \geq 0.4\,\mathrm{s}$
- $\sim$ 1140 coarsest level steps in 24 h on 96 cores shown above. Overall cost $\sim$ 4600 h CPU.

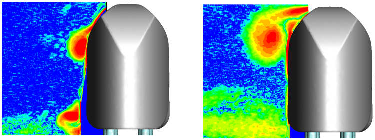


Vorticity vector component perpendicular to middle axis.

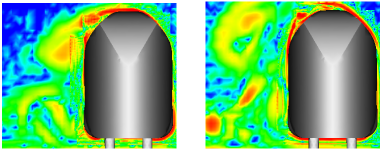
Flow prediction, $\mathrm{Re} = 450,000$, $\beta = 30^o$

- Domain $10\,\mathrm{m} \times 2.4\,\mathrm{m} \times 1.6\,\mathrm{m}$
- Computation started in 3 steps. Full resolution after 5889 coarsest level steps or $t \geq 0.4\,\mathrm{s}$
- $\sim$ 1140 coarsest level steps in 24 h on 96 cores shown above. Overall cost $\sim$ 4600 h CPU.

Experiment (time-averaged)



AMROC-LBM Simulation (instantaneous snapshots)

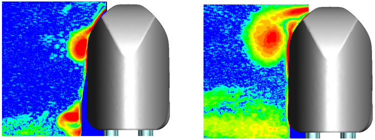


Vorticity component (seen from behind) in axial direction 80 mm and 290 mm away from model tip.

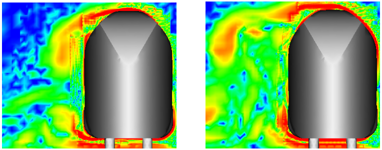
Flow prediction, $\mathrm{Re} = 450,000$, $\beta = 30^o$

- Domain $10\,\mathrm{m} \times 2.4\,\mathrm{m} \times 1.6\,\mathrm{m}$
- Computation started in 3 steps. Full resolution after 5889 coarsest level steps or $t \geq 0.4\,\mathrm{s}$
- $\sim$ 1140 coarsest level steps in $24\,\mathrm{h}$ on 96 cores shown above. Overall cost $\sim$ $4600\,\mathrm{h}$ CPU.

Experiment (time-averaged)



AMROC-LBM Simulation (instantaneous snapshots)

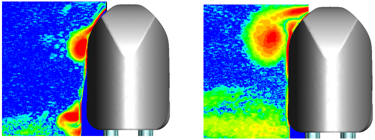


Vorticity component (seen from behind) in axial direction $80\,\mathrm{mm}$ and $290\,\mathrm{mm}$ away from model tip.

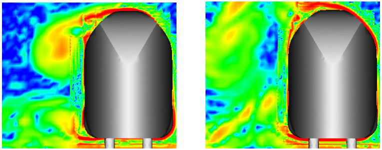
Flow prediction, $\mathrm{Re} = 450,000$, $\beta = 30^o$

- Domain $10\,\mathrm{m} \times 2.4\,\mathrm{m} \times 1.6\,\mathrm{m}$
- Computation started in 3 steps. Full resolution after 5889 coarsest level steps or $t \geq 0.4\,\mathrm{s}$
- $\sim$ 1140 coarsest level steps in $24\,\mathrm{h}$ on 96 cores shown above. Overall cost $\sim$ $4600\,\mathrm{h}$ CPU.

Experiment (time-averaged)



AMROC-LBM Simulation (instantaneous snapshots)

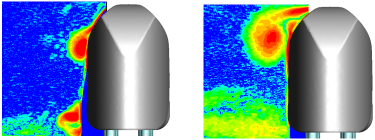


Vorticity component (seen from behind) in axial direction $80\,\mathrm{mm}$ and $290\,\mathrm{mm}$ away from model tip.

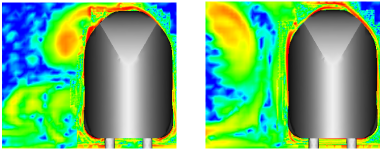
Flow prediction, $\mathrm{Re} = 450,000$, $\beta = 30^o$

- Domain $10\,\mathrm{m} \times 2.4\,\mathrm{m} \times 1.6\,\mathrm{m}$
- Computation started in 3 steps. Full resolution after 5889 coarsest level steps or $t \geq 0.4\,\mathrm{s}$
- $\sim$ 1140 coarsest level steps in $24\,\mathrm{h}$ on 96 cores shown above. Overall cost $\sim 4600\,\mathrm{h}$ CPU.

Experiment (time-averaged)



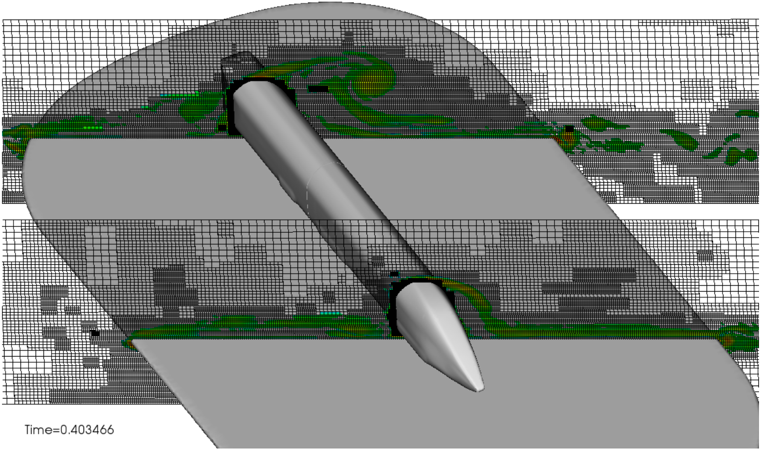
AMROC-LBM Simulation (instantaneous snapshots)



Vorticity component (seen from behind) in axial direction $80\,\mathrm{mm}$ and $290\,\mathrm{mm}$ away from model tip.

Dynamic mesh adaptation

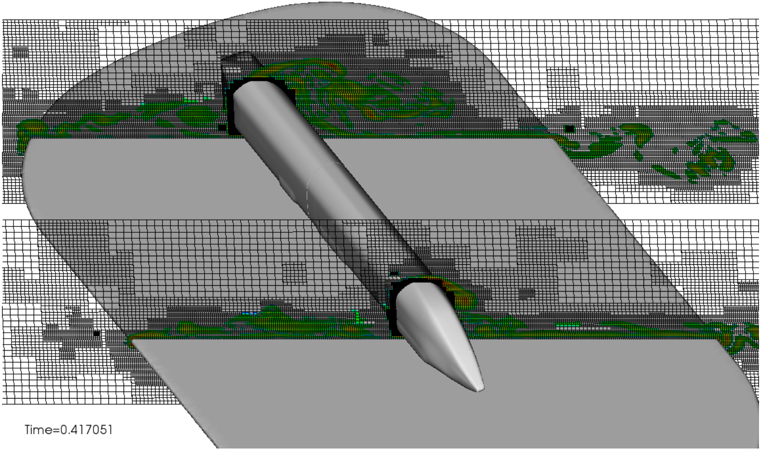
- Base mesh $500 \times 120 \times 80$ cells, refinement factors 2,2,4.
- Refinement based on error estimation of $|\mathbf{u}|$ up to second highest level.
- Highest level reserved to geometry refinement with $\Delta x = 1.25\,\mathrm{mm}$.



Dynamically adapting mesh. View in wind direction.

Dynamic mesh adaptation

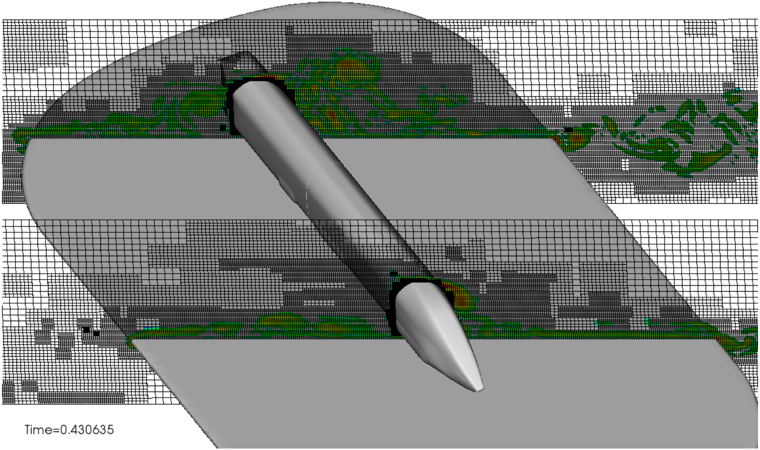
- Base mesh $500 \times 120 \times 80$ cells, refinement factors 2,2,4.
- Refinement based on error estimation of $|\mathbf{u}|$ up to second highest level.
- Highest level reserved to geometry refinement with $\Delta x = 1.25\,\mathrm{mm}$.



Dynamically adapting mesh. View in wind direction.

Dynamic mesh adaptation

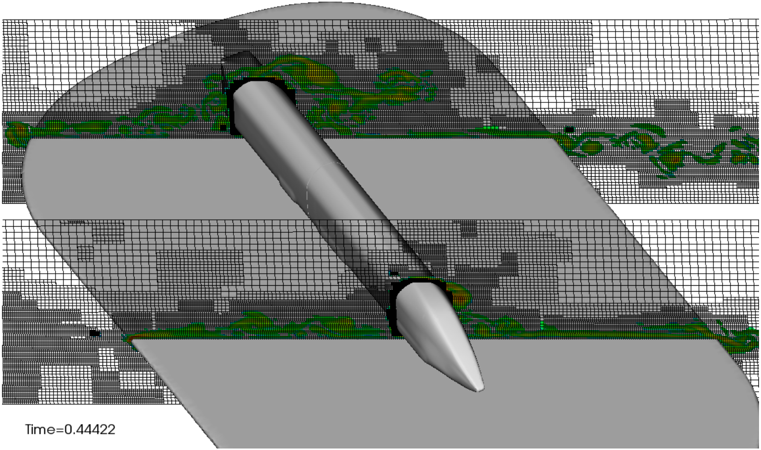
- Base mesh $500 \times 120 \times 80$ cells, refinement factors 2,2,4.
- Refinement based on error estimation of $|\mathbf{u}|$ up to second highest level.
- Highest level reserved to geometry refinement with $\Delta x = 1.25\,\mathrm{mm}$.



Dynamically adapting mesh. View in wind direction.

Dynamic mesh adaptation

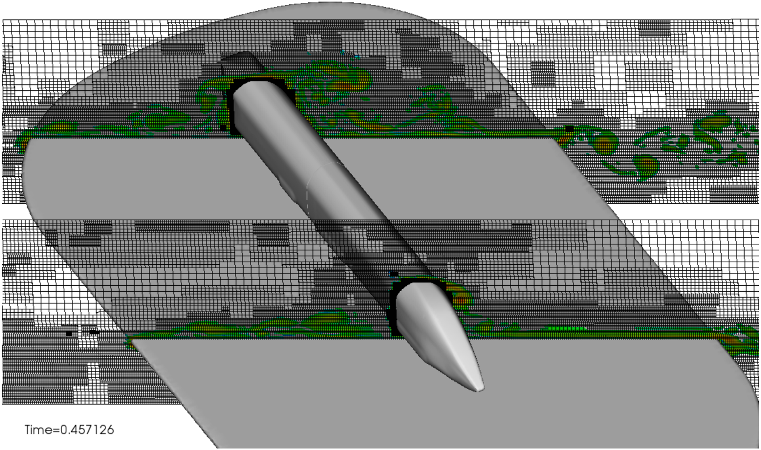
- Base mesh $500 \times 120 \times 80$ cells, refinement factors 2,2,4.
- Refinement based on error estimation of $|\mathbf{u}|$ up to second highest level.
- Highest level reserved to geometry refinement with $\Delta x = 1.25\,\mathrm{mm}$.



Dynamically adapting mesh. View in wind direction.

Dynamic mesh adaptation

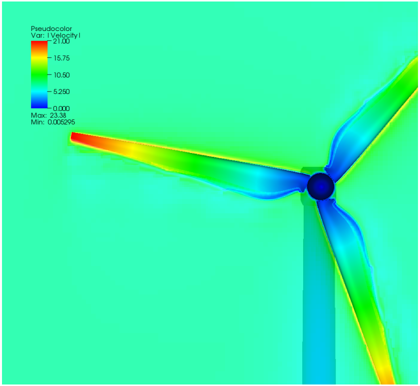
- Base mesh $500 \times 120 \times 80$ cells, refinement factors 2,2,4.
- Refinement based on error estimation of $|\mathbf{u}|$ up to second highest level.
- Highest level reserved to geometry refinement with $\Delta x = 1.25\,\mathrm{mm}$.



Dynamically adapting mesh. View in wind direction.

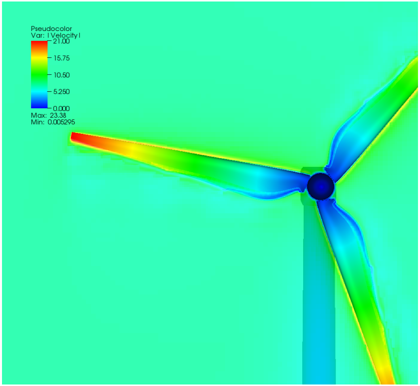
Simulation of a single turbine

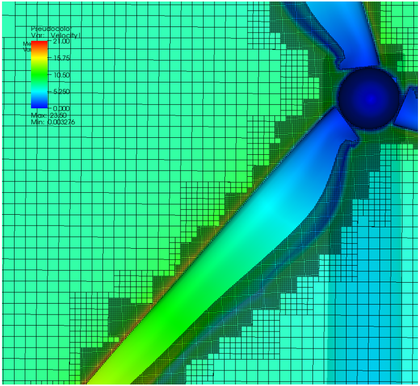
- Geometry from realistic Vestas V27 turbine. Rotor diameter $27\,\mathrm{m}$, tower height $\sim 35\,\mathrm{m}$. Ground considered.
- Prescribed motion of rotor with $15\,\mathrm{rpm}$. Inflow velocity $7\,\mathrm{m/s}$.
- Simulation domain $200\,\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$.
- Base mesh $400 \times 200 \times 200$ cells with refinement factors 2,2,4. Resolution of rotor and tower $\Delta x = 3.125\,\mathrm{cm}$.
- 141,344 highest level iterations to $t_e = 30\,\mathrm{s}$ computed.



Simulation of a single turbine

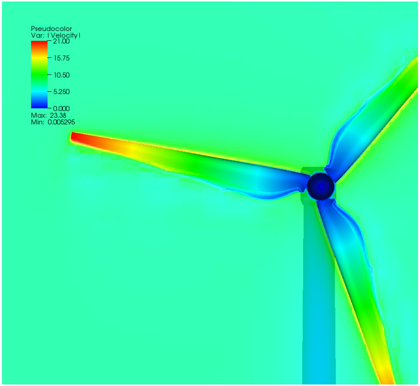
- Geometry from realistic Vestas V27 turbine. Rotor diameter 27 m, tower height $\sim$ 35 m. Ground considered.
- Prescribed motion of rotor with 15 rpm. Inflow velocity 7 $\mathrm{m/s}$.
- Simulation domain $200\,\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$.
- Base mesh 400 $\times$ 200 $\times$ 200 cells with refinement factors 2,2,4. Resolution of rotor and tower $\Delta x = 3.125\,\mathrm{cm}$.
- 141,344 highest level iterations to $t_e = 30\,\mathrm{s}$ computed.

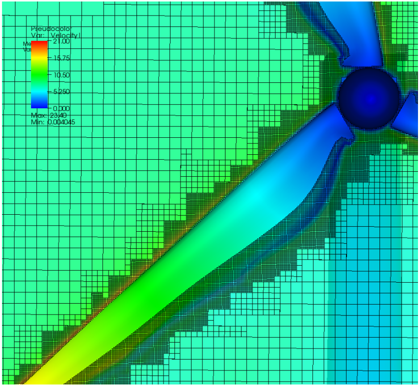




Simulation of a single turbine

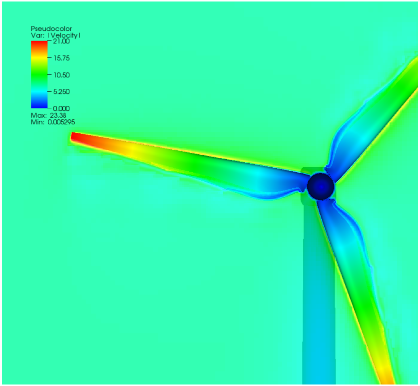
- Geometry from realistic Vestas V27 turbine. Rotor diameter $27\,\mathrm{m}$, tower height $\sim 35\,\mathrm{m}$. Ground considered.
- Prescribed motion of rotor with $15\,\mathrm{rpm}$. Inflow velocity $7\,\mathrm{m/s}$.
- Simulation domain $200\,\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$.
- Base mesh $400 \times 200 \times 200$ cells with refinement factors 2,2,4. Resolution of rotor and tower $\Delta x = 3.125\,\mathrm{cm}$.
- 141,344 highest level iterations to $t_e = 30\,\mathrm{s}$ computed.

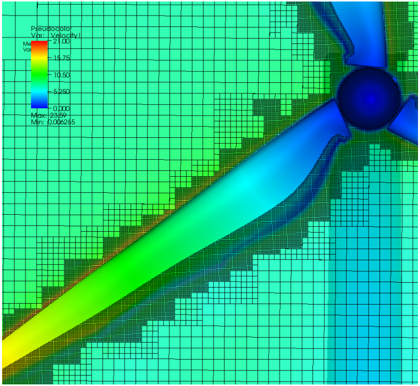




Simulation of a single turbine

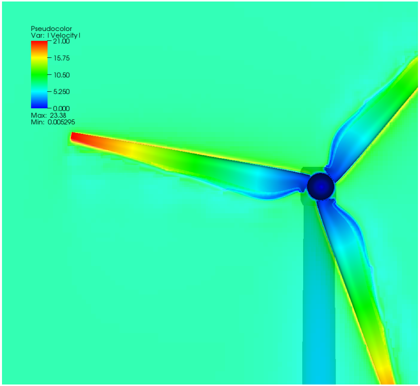
- Geometry from realistic Vestas V27 turbine. Rotor diameter 27 m, tower height $\sim$ 35 m. Ground considered.
- Prescribed motion of rotor with 15 rpm. Inflow velocity 7 $\mathrm{m/s}$.
- Simulation domain 200 $\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$.
- Base mesh 400 $\times$ 200 $\times$ 200 cells with refinement factors 2,2,4. Resolution of rotor and tower $\Delta x = 3.125\,\mathrm{cm}$.
- 141,344 highest level iterations to $t_e = 30\,\mathrm{s}$ computed.

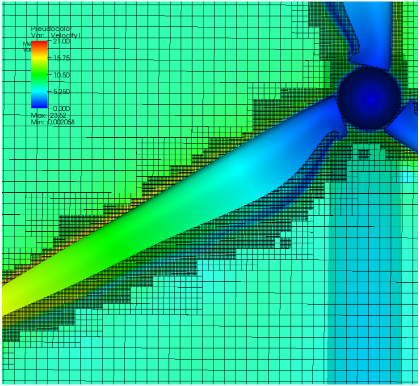




Simulation of a single turbine

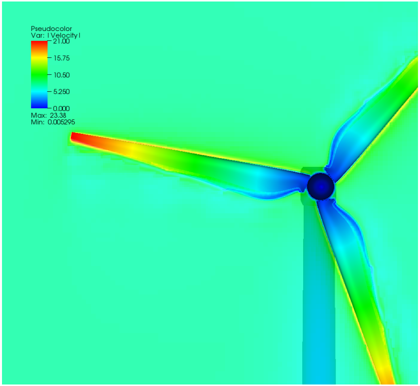
- Geometry from realistic Vestas V27 turbine. Rotor diameter 27 m, tower height $\sim$ 35 m. Ground considered.
- Prescribed motion of rotor with 15 rpm. Inflow velocity 7 $\mathrm{m/s}$.
- Simulation domain $200\,\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$.
- Base mesh $400 \times 200 \times 200$ cells with refinement factors 2,2,4. Resolution of rotor and tower $\Delta x = 3.125\,\mathrm{cm}$.
- 141,344 highest level iterations to $t_e = 30\,\mathrm{s}$ computed.

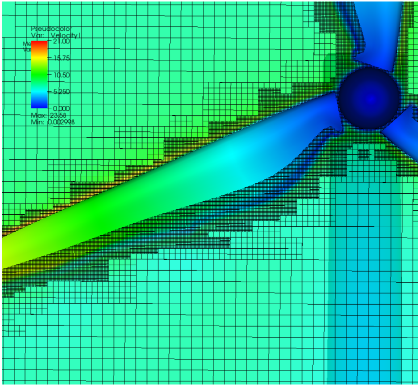




Simulation of a single turbine

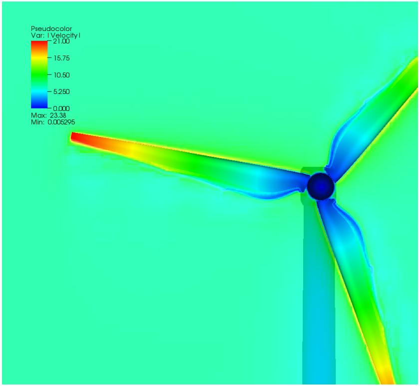
- Geometry from realistic Vestas V27 turbine. Rotor diameter 27 m, tower height $\sim$ 35 m. Ground considered.
- Prescribed motion of rotor with 15 rpm. Inflow velocity 7 $\mathrm{m/s}$.
- Simulation domain 200 $\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$.
- Base mesh 400 $\times$ 200 $\times$ 200 cells with refinement factors 2,2,4. Resolution of rotor and tower $\Delta x = 3.125\,\mathrm{cm}$.
- 141,344 highest level iterations to $t_e = 30\,\mathrm{s}$ computed.

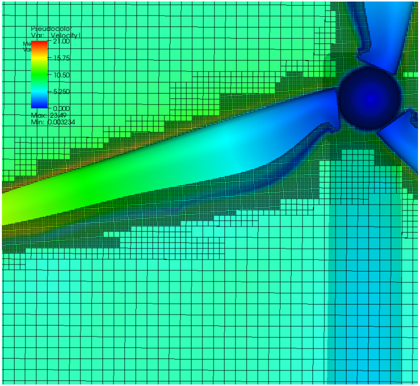




Simulation of a single turbine

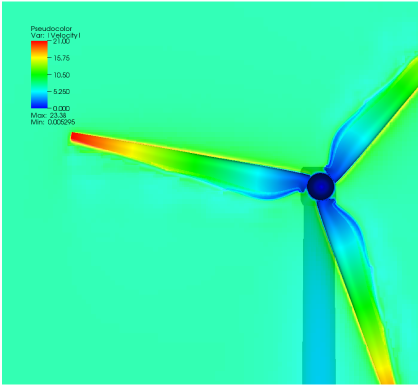
- Geometry from realistic Vestas V27 turbine. Rotor diameter $27\,\mathrm{m}$, tower height $\sim 35\,\mathrm{m}$. Ground considered.
- Prescribed motion of rotor with $15\,\mathrm{rpm}$. Inflow velocity $7\,\mathrm{m/s}$.
- Simulation domain $200\,\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$.
- Base mesh $400 \times 200 \times 200$ cells with refinement factors 2,2,4. Resolution of rotor and tower $\Delta x = 3.125\,\mathrm{cm}$.
- 141,344 highest level iterations to $t_e = 30\,\mathrm{s}$ computed.

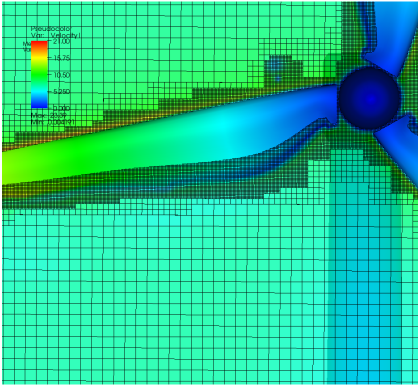




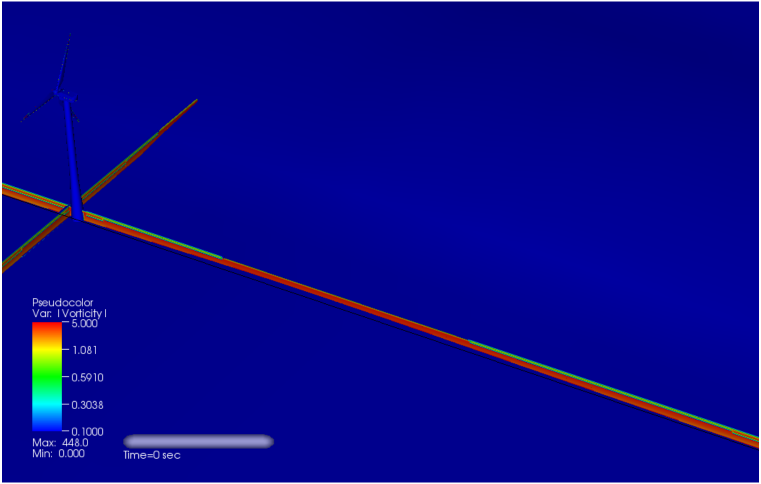
Simulation of a single turbine

- Geometry from realistic Vestas V27 turbine. Rotor diameter 27 m, tower height $\sim 35\,\mathrm{m}$. Ground considered.
- Prescribed motion of rotor with 15 rpm. Inflow velocity 7 $\mathrm{m/s}$.
- Simulation domain $200\,\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$.
- Base mesh $400 \times 200 \times 200$ cells with refinement factors 2,2,4. Resolution of rotor and tower $\Delta x = 3.125\,\mathrm{cm}$.
- 141,344 highest level iterations to $t_e = 30\,\mathrm{s}$ computed.



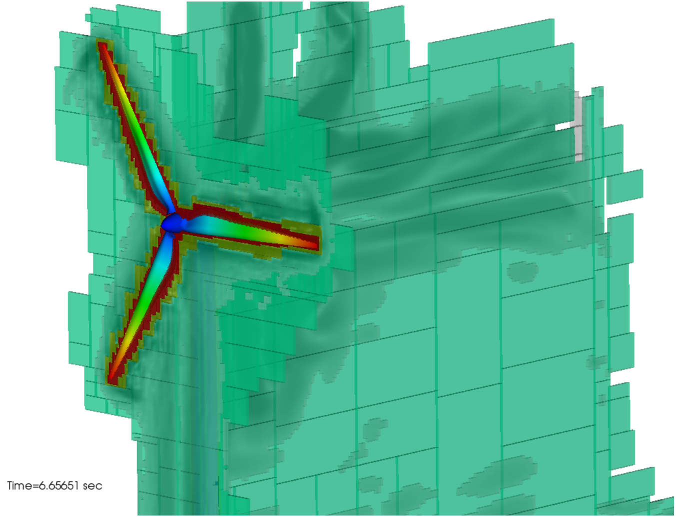


Wake field behind turbine



- Simulation on 96 cores Intel Xeon-Westmere. $\sim 10,400\,\mathrm{h}$ CPU.
- Error estimation in $|\mathbf{u}|$ refines wake up to level 1 ($\Delta x = 25\,\mathrm{cm}$).
- Rotation starts at $t = 4\,\mathrm{s}$.
- vtf/fsi/motion-amroc/WindTurbine_Terrain - Fluid, Solid

Adaptive refinement

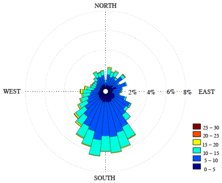


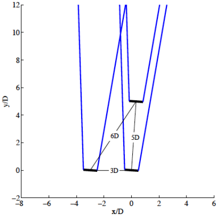
Dynamic evolution of refinement blocks (indicated by color).

Preliminary simulation of the SWIFT array

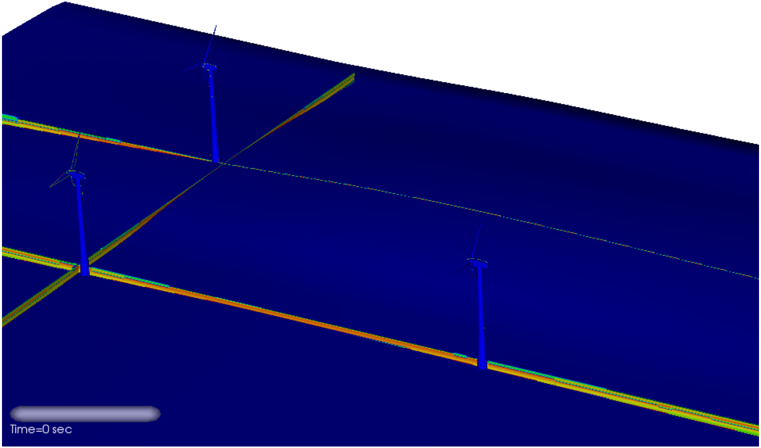
- Three Vestas V27 turbines. 225 kW power generation at wind speeds 14 to 25 m/s (then cut-off).
- Prescribed motion of rotor with 15 rpm. Inflow velocity 7 m/s (power generation 52.5 kW).

- Simulation domain $488\,\mathrm{m} \times 240\,\mathrm{m} \times 100\,\mathrm{m}$.
- Base mesh $448 \times 240 \times 100$ cells with refinement factors 2,2,2. Resolution of rotor and tower $\Delta x = 12.5\,\mathrm{cm}$.
- 47,120 highest level iterations to $t_e = 40\,\mathrm{s}$ computed.





Wakes in SWIFT array (preliminary)



- Simulation on 288 cores Intel Xeon-Westmere. $\sim 28,000\,\mathrm{h}$ CPU.
- Refinement of wake up to level 2 ($\Delta x = 25\,\mathrm{cm}$).
- Rotation starts at $t = 4\,\mathrm{s}$, full refinement at $t = 8\,\mathrm{s}$ to avoid refining initial acoustic waves.

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.

[Chen et al., 2006] Chen, H., Filippova, O., Hoch, J., Molvig, K., Shock, R., Teixeira, C., and Zhang, R. (2006). Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Physica A*, 362:158–167.

[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 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.

[Hähnel, 2004] Hähnel, D., editor (2004). *Molekulare Gasdynamik*. Springer.

[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.

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

References IV

[Peng and Luo, 2008] Peng, Y. and Luo, L.-S. (2008). A comparative study of immersed-boundary and interpolated bounce-back methods in lbe. *Prog. Comp. Fluid Dynamics*, 8(1-4):156–167.

[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.