Lecture 3

Complex hyperbolic applications

Course *Block-structured Adaptive Mesh Refinement Methods for Conservation Laws*
Theory, Implementation and Application

Ralf Deiterding
Computer Science and Mathematics Division
Oak Ridge National Laboratory
P.O. Box 2008 MS6367, Oak Ridge, TN 37831, USA

E-mail: deiterdingr@ornl.gov

Outline

Outline

Outline

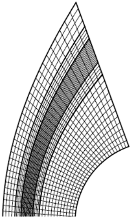
Outline

Outline

SAMR on boundary aligned meshes

Analytic or stored geometric mapping of the coordinates (graphic from [Yamaleev and Carpenter, 2002])

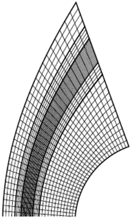
- Topology remains unchanged and thereby entire SAMR algorithm
- Patch solver and interpolation need to consider geometry transformation
- Handles boundary layers well

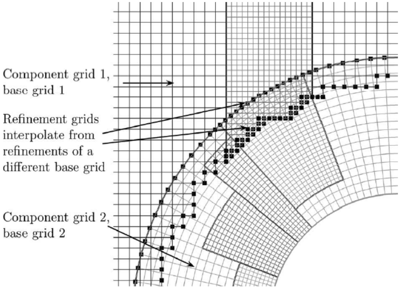


SAMR on boundary aligned meshes

Analytic or stored geometric mapping of the coordinates (graphic from [Yamaleev and Carpenter, 2002])

- Topology remains unchanged and thereby entire SAMR algorithm
- Patch solver and interpolation need to consider geometry transformation
- Handles boundary layers well





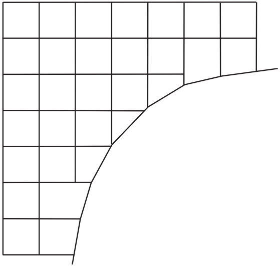
Overlapping adaptive meshes [Henshaw and Schwendeman, 2003], [Meakin, 1995]

- Idea is to use a non-Cartesian structured grids only near boundary
- Very suitable for moving objects with boundary layers
- Interpolation between meshes is usually non-conservative

Cut-cell techniques

Accurate embedded boundary method

$$V_j^{n+1}\mathbf{Q}_j^{n+1} = V_j^n\mathbf{Q}_j^n - \Delta t\left(A_{j+1/2}^{n+1/2}\,\mathbf{F}(\mathbf{Q},j) - A_{j-1/2}^{n+1/2}\,\mathbf{F}(\mathbf{Q},j-1)\right)$$



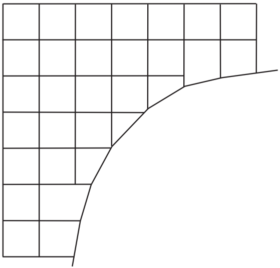
Methods that represent the boundary sharply:

- Cut-cell approach constructs appropriate finite volumes
- Conservative by construction. Correct boundary flux

Cut-cell techniques

Accurate embedded boundary method

$$V_j^{n+1}\mathbf{Q}_j^{n+1} = V_j^n\mathbf{Q}_j^n - \Delta t\left(A_{j+1/2}^{n+1/2}\,\mathbf{F}(\mathbf{Q},j) - A_{j-1/2}^{n+1/2}\,\mathbf{F}(\mathbf{Q},j-1)\right)$$



Methods that represent the boundary sharply:

- Cut-cell approach constructs appropriate finite volumes
- Conservative by construction. Correct boundary flux
- Key question: How to avoid small-cell time step restriction in explicit metyhods?
  - Cell merging: [Quirk, 1994a]
- Usually explicit geometry representation used [Aftosmis, 1997], but can also be implicit, cf. [Nourgaliev et al., 2003], [Murman et al., 2003]

Embedded boundary techniques

Volume of fluid methods that resemble a cut-cell technique on purely Cartesian mesh

- Redistribution of boundary flux achieves conservation and bypasses time step restriction: [Pember et al., 1999], [Berger and Helzel, 2002]

Embedded boundary techniques

Volume of fluid methods that resemble a cut-cell technique on purely Cartesian mesh

- Redistribution of boundary flux achieves conservation and bypasses time step restriction: [Pember et al., 1999], [Berger and Helzel, 2002]

Methods that diffuse the boundary in one cell (good overview in [Mittal and Iaccarino, 2005]):

- Related to the immersed boundary method by Peskin, cf. [Roma et al., 1999]
- Boundary prescription often by internal ghost cell values, cf. [Tseng and Ferziger, 2003]
- Not conservative by construction but conservative correction possible
- Usually combined with implicit geometry representation

Embedded boundary techniques

Volume of fluid methods that resemble a cut-cell technique on purely Cartesian mesh

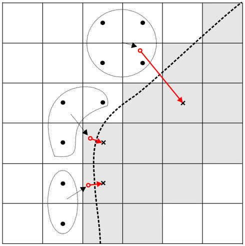
- Redistribution of boundary flux achieves conservation and bypasses time step restriction: [Pember et al., 1999], [Berger and Helzel, 2002]

Methods that diffuse the boundary in one cell (good overview in [Mittal and Iaccarino, 2005]):

- Related to the immersed boundary method by Peskin, cf. [Roma et al., 1999]
- Boundary prescription often by internal ghost cell values, cf. [Tseng and Ferziger, 2003]
- Not conservative by construction but conservative correction possible
- Usually combined with implicit geometry representation

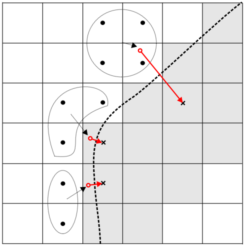
K. J. Richards et al., On the use of the immersed boundary method for engine modeling

Level-set method for boundary embedding



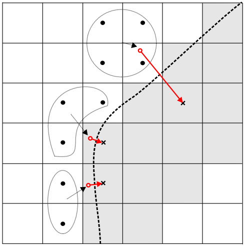
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi / |\nabla\varphi|$

Level-set method for boundary embedding



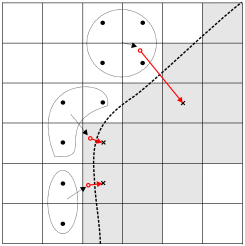
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi / |\nabla\varphi|$
- Complex boundary moving with local velocity $\mathbf{w}$, treat interface as moving rigid wall

Level-set method for boundary embedding



- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi / |\nabla\varphi|$
- Complex boundary moving with local velocity $\mathbf{w}$, treat interface as moving rigid wall
- Construction of values in embedded boundary cells by interpolation / extrapolation

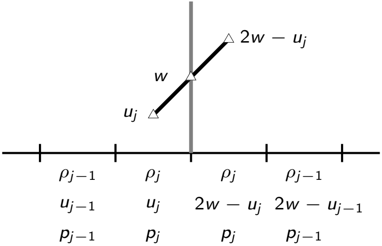
Level-set method for boundary embedding



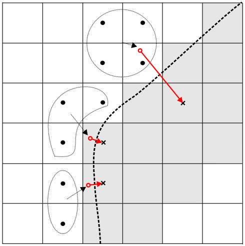
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi / |\nabla\varphi|$
- Complex boundary moving with local velocity $\mathbf{w}$, treat interface as moving rigid wall
- Construction of values in embedded boundary cells by interpolation / extrapolation

Interpolate / constant value extrapolate values at

$$\tilde{\mathbf{x}} = \mathbf{x} + 2\varphi\mathbf{n}$$



Level-set method for boundary embedding



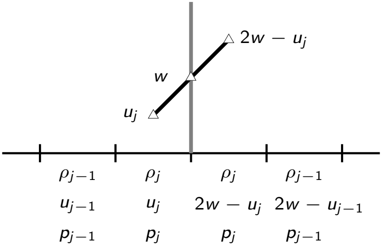
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi / |\nabla\varphi|$
- Complex boundary moving with local velocity $\mathbf{w}$, treat interface as moving rigid wall
- Construction of values in embedded boundary cells by interpolation / extrapolation

Interpolate / constant value extrapolate values at

$$\tilde{\mathbf{x}} = \mathbf{x} + 2\varphi\mathbf{n}$$

Velocity in ghost cells

$$\mathbf{u}' = (2\mathbf{w}\cdot\mathbf{n} - \mathbf{u}\cdot\mathbf{n})\mathbf{n} + (\mathbf{u}\cdot\mathbf{t})\mathbf{t}$$
$$= 2\left((\mathbf{w} - \mathbf{u})\cdot\mathbf{n}\right)\mathbf{n} + \mathbf{u}$$



Closest point transform algorithm

The signed distance φ to a surface $\mathcal{I}$ satisfies the eikonal equation [Sethian, 1999]

$$|\nabla\varphi| = 1 \quad \text{with} \quad \varphi\big|_{\mathcal{I}} = 0$$

Solution smooth but non-diferentiable across characteristics.

Closest point transform algorithm

The signed distance φ to a surface $\mathcal{I}$ satisfies the eikonal equation [Sethian, 1999]

$$|\nabla\varphi| = 1 \quad \text{with} \quad \varphi\big|_{\mathcal{I}} = 0$$

Solution smooth but non-diferentiable across characteristics.

Distance computation trivial for non-overlapping elementary shapes but difficult to do efficiently for triangulated surface meshes:

- Geometric solution approach with plosest-point-transform algorithm [Mauch, 2003]

Closest point transform algorithm

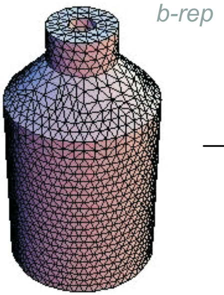
The signed distance φ to a surface $\mathcal{I}$ satisfies the eikonal equation [Sethian, 1999]

$$|\nabla\varphi| = 1 \quad \text{with} \quad \varphi\big|_{\mathcal{I}} = 0$$

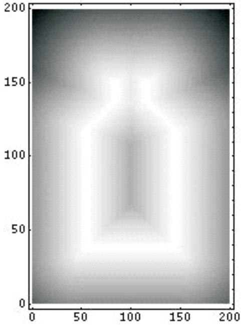
Solution smooth but non-diferentiable across characteristics.

Distance computation trivial for non-overlapping elementary shapes but difficult to do efficiently for triangulated surface meshes:

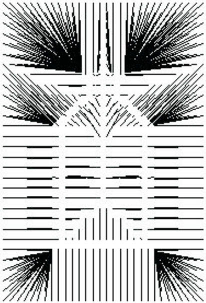
- Geometric solution approach with plosest-point-transform algorithm [Mauch, 2003]



Surface mesh $\mathcal{I}$



Distance φ

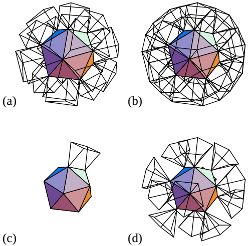


Normal to closest point

The characteristic / scan conversion algorithm

1. Build the characteristic polyhedrons for the surface mesh

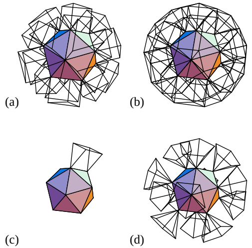
Characteristic polyhedra for faces, edges, and vertices



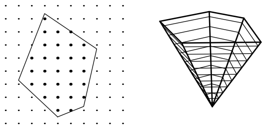
The characteristic / scan conversion algorithm

1. Build the characteristic polyhedrons for the surface mesh
2. For each face/edge/vertex
   2.1 Scan convert the polyhedron.

Characteristic polyhedra for faces, edges, and vertices



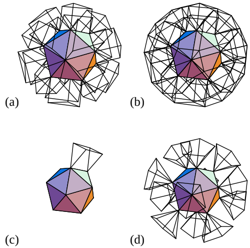
Slicing and scan conversion of apolygon



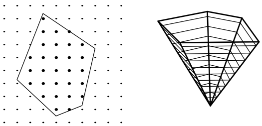
The characteristic / scan conversion algorithm

1. Build the characteristic polyhedrons for the surface mesh
2. For each face/edge/vertex
   2.1 Scan convert the polyhedron.
   2.2 Compute distance to that primitive for the scan converted points

Characteristic polyhedra for faces, edges, and vertices



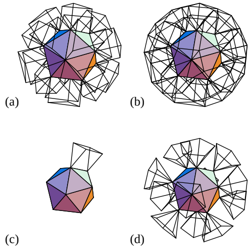
Slicing and scan conversion of apolygon



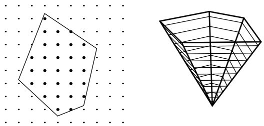
The characteristic / scan conversion algorithm

1. Build the characteristic polyhedrons for the surface mesh
2. For each face/edge/vertex
   2.1 Scan convert the polyhedron.
   2.2 Compute distance to that primitive for the scan converted points
3. Computational complexity.
   - $O(m)$ to build the b-rep and the polyhedra.
   - $O(n)$ to scan convert the polyhedra and compute the distance, etc.

Characteristic polyhedra for faces, edges, and vertices



Slicing and scan conversion of apolygon

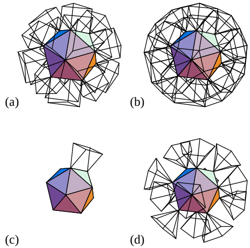


The characteristic / scan conversion algorithm

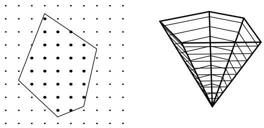
1. Build the characteristic polyhedrons for the surface mesh
2. For each face/edge/vertex
   2.1 Scan convert the polyhedron.
   2.2 Compute distance to that primitive for the scan converted points
3. Computational complexity.
   - $O(m)$ to build the b-rep and the polyhedra.
   - $O(n)$ to scan convert the polyhedra and compute the distance, etc.
4. Problem reduction by evaluation only within specified max. distance

[Mauch, 2003], see also [Deiterding et al., 2006]

Characteristic polyhedra for faces, edges, and vertices



Slicing and scan conversion of apolygon



Accuracy test: stationary vortex

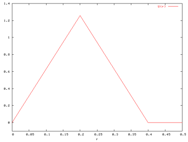
Construct non-trivial *radially symmetric* and *stationary* solution by balancing hydrodynamic pressure and centripetal force per volume element, i.e.

$$\frac{d}{dr}p(r) = \rho(r)\frac{U(r)^2}{r}$$

For $\rho_0 \equiv 1$ and the velocity field

$$U(r) = \alpha \cdot \begin{cases} 2r/R & \text{if } 0 < r < R/2, \\ 2(1 - r/R) & \text{if } R/2 \leq r \leq R, \\ 0 & \text{if } r > R, \end{cases}$$

$U(r)$



$p(r)$

Accuracy test: stationary vortex

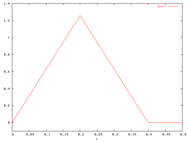
Construct non-trivial *radially symmetric* and *stationary* solution by balancing hydrodynamic pressure and centripetal force per volume element, i.e.

$$\frac{d}{dr}p(r) = \rho(r)\frac{U(r)^2}{r}$$

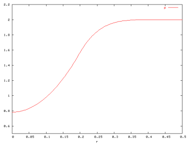
For $\rho_0 \equiv 1$ and the velocity field

$$U(r) = \alpha \cdot \begin{cases} 2r/R & \text{if } 0 < r < R/2, \\ 2(1 - r/R) & \text{if } R/2 \leq r \leq R, \\ 0 & \text{if } r > R, \end{cases}$$

$U(r)$



$p(r)$



one gets with boundary condition $p(R) = p_0 \equiv 2$ the pressure distribution

$$p(r) = p_0 + 2\rho_0\alpha^2 \cdot \begin{cases} r^2/R^2 + 1 - 2\log 2 & \text{if } 0 < r < R/2, \\ r^2/R^2 + 3 - 4r/R + 2\log(r/R) & \text{if } R/2 \leq r \leq R, \\ 0 & \text{if } r > R. \end{cases}$$

Accuracy test: stationary vortex

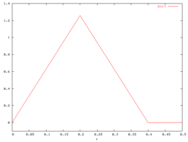
Construct non-trivial *radially symmetric* and *stationary* solution by balancing hydrodynamic pressure and centripetal force per volume element, i.e.

$$\frac{d}{dr}p(r) = \rho(r)\frac{U(r)^2}{r}$$

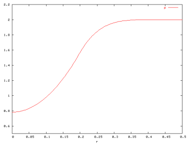
For $\rho_0 \equiv 1$ and the velocity field

$$U(r) = \alpha \cdot \begin{cases} 2r/R & \text{if } 0 < r < R/2, \\ 2(1 - r/R) & \text{if } R/2 \le r \le R, \\ 0 & \text{if } r > R, \end{cases}$$

$U(r)$



$p(r)$



one gets with boundary condition $p(R) = p_0 \equiv 2$ the pressure distribution

$$p(r) = p_0 + 2\rho_0\alpha^2 \cdot \begin{cases} r^2/R^2 + 1 - 2\log 2 & \text{if } 0 < r < R/2, \\ r^2/R^2 + 3 - 4r/R + 2\log(r/R) & \text{if } R/2 \le r \le R, \\ 0 & \text{if } r > R. \end{cases}$$

Entire solution for Euler equations reads

$$\rho(x_1, x_2, t) = \rho_0,\ u_1(x_1, x_2, t) = -U(r)\sin\phi,\ u_2(x_1, x_2, t) = U(r)\cos\phi,\ p(x_1, x_2, t) = p(r)$$

for all $t \ge 0$ with $r = \sqrt{(x_1 - x_{1,c})^2 + (x_2 - x_{2,c})^2}$ and $\phi = \arctan\frac{x_2 - x_{2,c}}{x_1 - x_{1,c}}$

Stationary vortex: results

Compute one full rotation, Roe solver, embedded slip wall boundary conditions
$x_{1,c} = 0.5,\ x_{2,c} = 0.5,\ R = 0.4,\ t_{end} = 1,\ \Delta h = \Delta x_1 = \Delta x_2 = 1/N,\ \alpha = R\pi$

No embedded boundary

N	Wave Propagation		Godunov Splitting	
	Error	Order	Error	Order
20	0.0111235		0.0182218	
40	0.0037996	1.55	0.0090662	1.01
80	0.0013388	1.50	0.0046392	0.97
160	0.0005005	1.42	0.0023142	1.00

Stationary vortex: results

Compute one full rotation, Roe solver, embedded slip wall boundary conditions
$x_{1,c} = 0.5,\ x_{2,c} = 0.5,\ R = 0.4,\ t_{end} = 1,\ \Delta h = \Delta x_1 = \Delta x_2 = 1/N,\ \alpha = R\pi$

No embedded boundary

N	Wave Propagation		Godunov Splitting	
	Error	Order	Error	Order
20	0.0111235		0.0182218	
40	0.0037996	1.55	0.0090662	1.01
80	0.0013388	1.50	0.0046392	0.97
160	0.0005005	1.42	0.0023142	1.00

Marginal shear flow along embedded boundary, $\alpha = R\pi,\ R_G = R,\ U_W = 0$

N	Wave Propagation			Godunov Splitting		
	Error	Order	Mass loss	Error	Order	Mass loss
20	0.0120056		0.0079236	0.0144203		0.0020241
40	0.0035074	1.78	0.0011898	0.0073070	0.98	0.0001300
80	0.0014193	1.31	0.0001588	0.0038401	0.93	-0.0001036
160	0.0005032	1.50	5.046e-05	0.0018988	1.02	-2.783e-06

Stationary vortex: results

Compute one full rotation, Roe solver, embedded slip wall boundary conditions
$x_{1,c} = 0.5,\ x_{2,c} = 0.5,\ R = 0.4,\ t_{end} = 1,\ \Delta h = \Delta x_1 = \Delta x_2 = 1/N,\ \alpha = R\pi$

No embedded boundary

N	Wave Propagation		Godunov Splitting	
	Error	Order	Error	Order
20	0.0111235		0.0182218	
40	0.0037996	1.55	0.0090662	1.01
80	0.0013388	1.50	0.0046392	0.97
160	0.0005005	1.42	0.0023142	1.00

Marginal shear flow along embedded boundary, $\alpha = R\pi,\ R_G = R,\ U_W = 0$

N	Wave Propagation			Godunov Splitting		
	Error	Order	Mass loss	Error	Order	Mass loss
20	0.0120056		0.0079236	0.0144203		0.0020241
40	0.0035074	1.78	0.0011898	0.0073070	0.98	0.0001300
80	0.0014193	1.31	0.0001588	0.0038401	0.93	-0.0001036
160	0.0005032	1.50	5.046e-05	0.0018988	1.02	-2.783e-06

Major shear flow along embedded boundary, $\alpha = R\pi,\ R_G = R/2,\ U_W = 0$

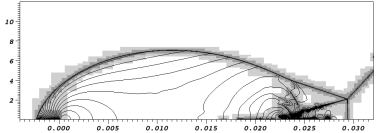
N	Wave Propagation			Godunov Splitting		
	Error	Order	Mass loss	Error	Order	Mass loss
20	0.0423925		0.0423925	0.0271446		0.0271446
40	0.0358735	0.24	0.0358735	0.0242260	0.16	0.0242260
80	0.0212340	0.76	0.0212340	0.0128638	0.91	0.0128638
160	0.0121089	0.81	0.0121089	0.0070906	0.86	0.0070906

Verification: shock reflection

- Reflection of a Mach 2.38 shock in nitrogen at 43^o wedge
- 2nd order MUSCL scheme with Roe solver, 2nd order multidimensional wave propagation method

Verification: shock reflection

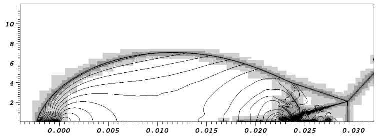
- Reflection of a Mach 2.38 shock in nitrogen at 43^o wedge
- 2nd order MUSCL scheme with Roe solver, 2nd order multidimensional wave propagation method



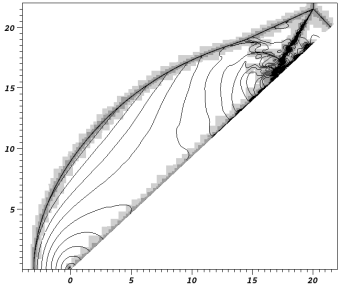
Cartesian base grid 360 $\times$ 160 cells on domain of 36 mm $\times$ 16 mm with up to 3 refinement levels with $r_l = 2, 4, 4$ and $\Delta x_{1,2} = 3.125\mu m$, 38 h CPU

Verification: shock reflection

- Reflection of a Mach 2.38 shock in nitrogen at 43^o wedge
- 2nd order MUSCL scheme with Roe solver, 2nd order multidimensional wave propagation method

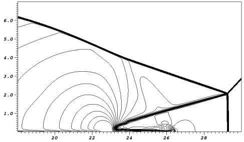


Cartesian base grid 360 $\times$ 160 cells on domain of $36\,\mathrm{mm} \times 16\,\mathrm{mm}$ with up to 3 refinement levels with $r_l = 2, 4, 4$ and $\Delta x_{1,2} = 3.125\mu m$, 38 h CPU

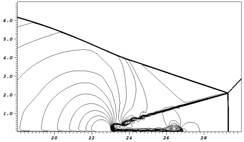


GFM base grid 390 $\times$ 330 cells on domain of $26\,\mathrm{mm} \times 22\,\mathrm{mm}$ with up to 3 refinement levels with $r_l = 2, 4, 4$ and $\Delta x_{e,1,2} = 2.849\mu m$, 200 h CPU

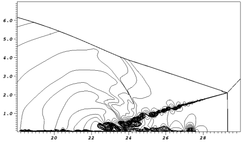
Shock reflection: SAMR solution for Euler equations



$\Delta x = 25\,\mathrm{mm}$

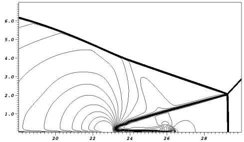


$\Delta x = 12.5\,\mathrm{mm}$

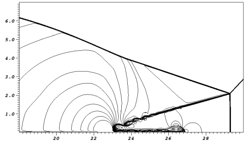


$\Delta x = 3.125\,\mathrm{mm}$

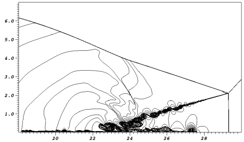
Shock reflection: SAMR solution for Euler equations



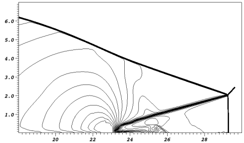
$\Delta x = 25\,\mathrm{mm}$



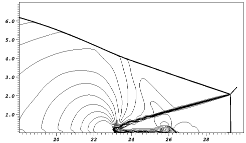
$\Delta x = 12.5\,\mathrm{mm}$



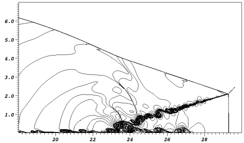
$\Delta x = 3.125\,\mathrm{mm}$



$\Delta x_e = 22.8\,\mathrm{mm}$

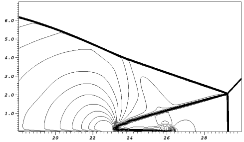


$\Delta x_e = 11.4\,\mathrm{mm}$

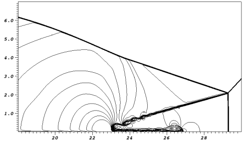


$\Delta x_e = 2.849\,\mathrm{mm}$

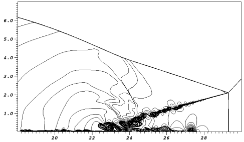
Shock reflection: SAMR solution for Euler equations



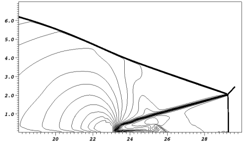
$\Delta x = 25\,\mathrm{mm}$



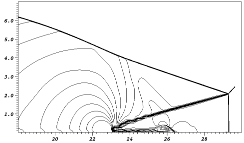
$\Delta x = 12.5\,\mathrm{mm}$



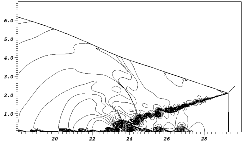
$\Delta x = 3.125\,\mathrm{mm}$



$\Delta x_e = 22.8\,\mathrm{mm}$

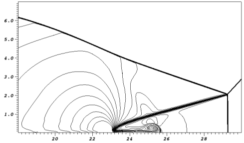


$\Delta x_e = 11.4\,\mathrm{mm}$

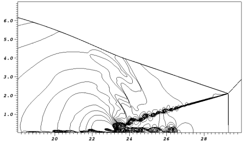


$\Delta x_e = 2.849\,\mathrm{mm}$

2nd order MUSCL scheme with Van Leer FVS, dimensional splitting



$\Delta x = 12.5\,\mathrm{mm}$



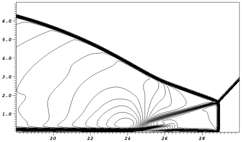
$\Delta x = 3.125\,\mathrm{mm}$

Shock reflection: solution for Navier-Stokes equations

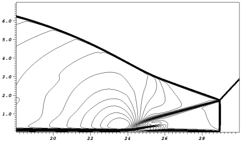
- No-slip boundary conditions enforced
- Conservative 2nd order centered differences to approximate stress tensor and heat flow

Shock reflection: solution for Navier-Stokes equations

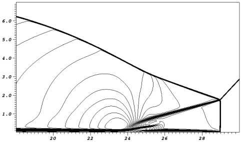
- No-slip boundary conditions enforced
- Conservative 2nd order centered differences to approximate stress tensor and heat flow



$\Delta x = 50\,\mathrm{mm}$



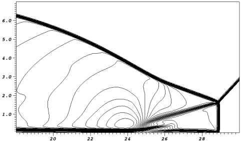
$\Delta x = 25\,\mathrm{mm}$



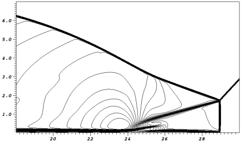
$\Delta x = 12.5\,\mathrm{mm}$, SAMR

Shock reflection: solution for Navier-Stokes equations

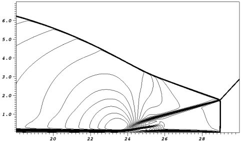
- No-slip boundary conditions enforced
- Conservative 2nd order centered differences to approximate stress tensor and heat flow



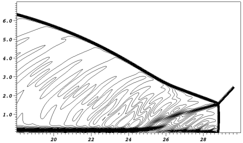
$\Delta x = 50\,\mathrm{mm}$



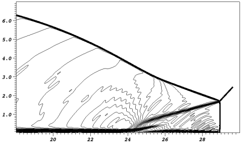
$\Delta x = 25\,\mathrm{mm}$



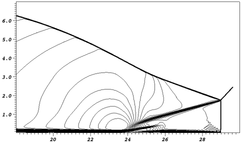
$\Delta x = 12.5\,\mathrm{mm}$, SAMR



$\Delta x_e = 45.6\,\mathrm{mm}$



$\Delta x_e = 22.8\,\mathrm{mm}$



$\Delta x_e = 11.4\,\mathrm{mm}$, SAMR

Outline

Governing equations for premixed combustion

Euler equations with reaction terms

$$\frac{\partial \rho_i}{\partial t} + \frac{\partial}{\partial x_n}(\rho_i u_n) = \dot{\omega}_i \ , \quad i = 1, \dots, K$$

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

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_n}(u_n(\rho E + p)) = 0$$

Governing equations for premixed combustion

Euler equations with reaction terms

$$\frac{\partial \rho_i}{\partial t} + \frac{\partial}{\partial x_n}(\rho_i u_n) = \dot{\omega}_i \,, \quad i = 1, \dots, K$$
$$\frac{\partial}{\partial t}(\rho u_k) + \frac{\partial}{\partial x_n}(\rho u_k u_n + \delta_{kn} p) = 0 \,, \quad k = 1, \dots, d$$
$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_n}(u_n(\rho E + p)) = 0$$

Ideal gas law and Dalton's law for gas-mixtures

$$p(\rho_1, \dots, \rho_K, T) = \sum_{i=1}^{K} p_i = \sum_{i=1}^{K} \rho_i \frac{\mathcal{R}}{W_i} T = \rho \frac{\mathcal{R}}{W} T \quad \text{with} \quad \sum_{i=1}^{K} \rho_i = \rho \,, Y_i = \frac{\rho_i}{\rho}$$

Governing equations for premixed combustion

Euler equations with reaction terms

$$\frac{\partial \rho_i}{\partial t} + \frac{\partial}{\partial x_n}(\rho_i u_n) = \dot{\omega}_i\,, \quad i = 1, \ldots, K$$

$$\frac{\partial}{\partial t}(\rho u_k) + \frac{\partial}{\partial x_n}(\rho u_k u_n + \delta_{kn} p) = 0\,, \quad k = 1, \ldots, d$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_n}(u_n(\rho E + p)) = 0$$

Ideal gas law and Dalton's law for gas-mixtures

$$p(\rho_1, \ldots, \rho_K, T) = \sum_{i=1}^{K} p_i = \sum_{i=1}^{K} \rho_i \frac{\mathcal{R}}{W_i} T = \rho \frac{\mathcal{R}}{W} T \quad \text{with} \quad \sum_{i=1}^{K} \rho_i = \rho\,, Y_i = \frac{\rho_i}{\rho}$$

Caloric equation

$$h(Y_1, \ldots, Y_K, T) = \sum_{i=1}^{K} Y_i h_i(T) \quad \text{with} \quad h_i(T) = h_i^0 + \int_0^T c_{pi}(s)ds$$

Governing equations for premixed combustion

Euler equations with reaction terms

$$\frac{\partial \rho_i}{\partial t} + \frac{\partial}{\partial x_n}(\rho_i u_n) = \dot{\omega}_i \;, \quad i = 1, \ldots, K$$
$$\frac{\partial}{\partial t}(\rho u_k) + \frac{\partial}{\partial x_n}(\rho u_k u_n + \delta_{kn} p) = 0 \;, \quad k = 1, \ldots, d$$
$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_n}(u_n(\rho E + p)) = 0$$

Ideal gas law and Dalton's law for gas-mixtures

$$p(\rho_1, \ldots, \rho_K, T) = \sum_{i=1}^{K} p_i = \sum_{i=1}^{K} \rho_i \frac{\mathcal{R}}{W_i} T = \rho \frac{\mathcal{R}}{W} T \quad \text{with} \quad \sum_{i=1}^{K} \rho_i = \rho \,, \, Y_i = \frac{\rho_i}{\rho}$$

Caloric equation

$$h(Y_1, \ldots, Y_K, T) = \sum_{i=1}^{K} Y_i h_i(T) \quad \text{with} \quad h_i(T) = h_i^0 + \int_0^T c_{pi}(s) ds$$

Computation of $T = T(\rho_1, \ldots, \rho_K, e)$ from implicit equation

$$\sum_{i=1}^{K} \rho_i \, h_i(T) - \mathcal{R} T \sum_{i=1}^{K} \frac{\rho_i}{W_i} - \rho e = 0$$

for *thermally perfect* gases with $\gamma_i(T) = c_{pi}(T)/c_{vi}(T)$

Chemistry

Arrhenius-Kinetics:

$$\dot{\omega}_i = \sum_{j=1}^{M} (\nu_{ji}^r - \nu_{ji}^f) \left[k_j^f \prod_{n=1}^{K} \left(\frac{\rho_n}{W_n} \right)^{\nu_{jn}^f} - k_j^r \prod_{n=1}^{K} \left(\frac{\rho_n}{W_n} \right)^{\nu_{jn}^r} \right] \quad i = 1, \ldots, K$$

Chemistry

Arrhenius-Kinetics:

$$\dot{\omega}_i = \sum_{j=1}^{M}(\nu_{ji}^r - \nu_{ji}^f)\left[k_j^f \prod_{n=1}^{K}\left(\frac{\rho_n}{W_n}\right)^{\nu_{jn}^f} - k_j^r \prod_{n=1}^{K}\left(\frac{\rho_n}{W_n}\right)^{\nu_{jn}^r}\right] \quad i = 1, \ldots, K$$

- Parsing of mechanisms with Chemkin-II
- Evalutation of $\dot{\omega}_i$ with automatically generated optimized Fortran-77 functions in the line of Chemkin-II

Chemistry

Arrhenius-Kinetics:

$$\dot{\omega}_i = \sum_{j=1}^{M} (\nu_{ji}^r - \nu_{ji}^f) \left[k_j^f \prod_{n=1}^{K} \left(\frac{\rho_n}{W_n} \right)^{\nu_{jn}^f} - k_j^r \prod_{n=1}^{K} \left(\frac{\rho_n}{W_n} \right)^{\nu_{jn}^r} \right] \quad i = 1, \ldots, K$$

- Parsing of mechanisms with Chemkin-II
- Evalutation of $\dot{\omega}_i$ with automatically generated optimized Fortran-77 functions in the line of Chemkin-II

Integration of reaction rates: ODE integration in $\mathcal{S}^{(\cdot)}$ for Euler equations with chemical reaction

- Standard implicit or semi-implicit ODE-solver subcycles within each cell
- ρ, e, u_k remain unchanged!

$$\partial_t \rho_i = W_i \dot{\omega}_i(\rho_1, \ldots, \rho_K, T) \qquad i = 1, \ldots, K$$

Use Newton or bisection method to compute T iteratively.

Non-equilibrium mechanism for hydrogen-oxgen combustion

				A [cm, mol, s]	β	E_{act} [cal mol^{-1}]
1.	$H + O_2$	$\longrightarrow$	$O + OH$	1.86×10^{14}	0.00	16790.
2.	$O + OH$	$\longrightarrow$	$H + O_2$	1.48×10^{13}	0.00	680.
3.	$H_2 + O$	$\longrightarrow$	$H + OH$	1.82×10^{10}	1.00	8900.
4.	$H + OH$	$\longrightarrow$	$H_2 + O$	8.32×10^{09}	1.00	6950.
5.	$H_2O + O$	$\longrightarrow$	$OH + OH$	3.39×10^{13}	0.00	18350.
6.	$OH + OH$	$\longrightarrow$	$H_2O + O$	3.16×10^{12}	0.00	1100.
7.	$H_2O + H$	$\longrightarrow$	$H_2 + OH$	9.55×10^{13}	0.00	20300.
8.	$H_2 + OH$	$\longrightarrow$	$H_2O + H$	2.19×10^{13}	0.00	5150.
9.	$H_2O_2 + OH$	$\longrightarrow$	$H_2O + HO_2$	1.00×10^{13}	0.00	1800.
10.	$H_2O + HO_2$	$\longrightarrow$	$H_2O_2 + OH$	2.82×10^{13}	0.00	32790.
...	...		...	...	...	...
30.	$OH + M$	$\longrightarrow$	$O + H + M$	7.94×10^{19}	-1.00	103720.
31.	$O_2 + M$	$\longrightarrow$	$O + O + M$	5.13×10^{15}	0.00	115000.
32.	$O + O + M$	$\longrightarrow$	$O_2 + M$	4.68×10^{15}	-0.28	0.
33.	$H_2 + M$	$\longrightarrow$	$H + H + M$	2.19×10^{14}	0.00	96000.
34.	$H + H + M$	$\longrightarrow$	$H_2 + M$	3.02×10^{15}	0.00	0.

Third body efficiencies: $f(O_2) = 0.40$, $f(H_2O) = 6.50$

C. K. Westbrook. Chemical kinetics of hydrocarbon oxidation in gaseous detonations. *J. Combustion and Flame*, 46:191–210, 1982.

Riemann solver for combustion

(S1) Calculate standard Roe-averages $\hat{\rho}$, $\hat{u}_n$, $\hat{H}$, $\hat{Y}_i$, $\hat{T}$.

(S2) Compute $\hat{\gamma} := \hat{c}_p/\hat{c}_v$ with $\hat{c}_{\{p/v\}i} = \dfrac{1}{T_R - T_L} \displaystyle\int_{T_L}^{T_R} c_{\{p,v\}i}(\tau)\, d\tau$.

(S3) Calculate $\hat{\phi}_i := (\hat{\gamma}-1)\left(\frac{\hat{\mathbf{u}}^2}{2} - \hat{h}_i\right) + \hat{\gamma}\, R_i\, \hat{T}$ with standard Roe-averages $\hat{e}_i$ or $\hat{h}_i$.

(S4) Calculate $\hat{c} := \left(\sum_{i=1}^K \hat{Y}_i\, \hat{\phi}_i - (\hat{\gamma}-1)\hat{\mathbf{u}}^2 + (\hat{\gamma}-1)\hat{H}\right)^{1/2}$.

(S5) Use $\Delta\mathbf{q} = \mathbf{q}_R - \mathbf{q}_L$ and Δp to compute the wave strengths a_m.

(S6) Calculate $\mathcal{W}_1 = a_1 \hat{\mathbf{r}}_1$, $\mathcal{W}_2 = \displaystyle\sum_{\iota=2}^{K+d} a_\iota \hat{\mathbf{r}}_\iota$, $\mathcal{W}_3 = a_{K+d+1} \hat{\mathbf{r}}_{K+d+1}$.

(S7) Evaluate $s_1 = \hat{u}_1 - \hat{c}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{c}$.

Riemann solver for combustion

(S1) Calculate standard Roe-averages $\hat{\rho}$, $\hat{u}_n$, $\hat{H}$, $\hat{Y}_i$, $\hat{T}$.

(S2) Compute $\hat{\gamma} := \hat{c}_p/\hat{c}_v$ with $\hat{c}_{\{p/v\}i} = \dfrac{1}{T_R - T_L} \displaystyle\int_{T_L}^{T_R} c_{\{p,v\}i}(\tau)\, d\tau$.

(S3) Calculate $\hat{\phi}_i := (\hat{\gamma} - 1)\left(\frac{\hat{\mathbf{u}}^2}{2} - \hat{h}_i\right) + \hat{\gamma}\, R_i\, \hat{T}$ with standard Roe-averages $\hat{e}_i$ or $\hat{h}_i$.

(S4) Calculate $\hat{c} := \left(\sum_{i=1}^{K} \hat{Y}_i\, \hat{\phi}_i - (\hat{\gamma} - 1)\hat{\mathbf{u}}^2 + (\hat{\gamma} - 1)\hat{H}\right)^{1/2}$.

(S5) Use $\Delta\mathbf{q} = \mathbf{q}_R - \mathbf{q}_L$ and Δp to compute the wave strengths a_m.

(S6) Calculate $\mathcal{W}_1 = a_1 \hat{\mathbf{r}}_1$, $\mathcal{W}_2 = \displaystyle\sum_{\iota=2}^{K+d} a_\iota \hat{\mathbf{r}}_\iota$, $\mathcal{W}_3 = a_{K+d+1} \hat{\mathbf{r}}_{K+d+1}$.

(S7) Evaluate $s_1 = \hat{u}_1 - \hat{c}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{c}$.

(S8) Evaluate $\rho^\star_{L/R}$, $u^\star_{1,L/R}$, $e^\star_{L/R}$, $c^\star_{1,L/R}$ from $\mathbf{q}^\star_L = \mathbf{q}_L + \mathcal{W}_1$ and $\mathbf{q}^\star_R = \mathbf{q}_R - \mathcal{W}_3$.

(S9) If $\rho^\star_{L/R} \le 0$ or $e^\star_{L/R} \le 0$ use $\mathbf{F}_{HLL}(\mathbf{q}_L, \mathbf{q}_R)$ and go to (S12).

Riemann solver for combustion

(S1) Calculate standard Roe-averages $\hat{\rho}$, $\hat{u}_n$, $\hat{H}$, $\hat{Y}_i$, $\hat{T}$.

(S2) Compute $\hat{\gamma} := \hat{c}_p/\hat{c}_v$ with $\hat{c}_{\{p/v\}i} = \dfrac{1}{T_R - T_L} \displaystyle\int_{T_L}^{T_R} c_{\{p,v\}i}(\tau)\, d\tau$.

(S3) Calculate $\hat{\phi}_i := (\hat{\gamma} - 1)\left(\frac{\hat{\mathbf{u}}^2}{2} - \hat{h}_i\right) + \hat{\gamma} R_i \hat{T}$ with standard Roe-averages $\hat{e}_i$ or $\hat{h}_i$.

(S4) Calculate $\hat{c} := \left(\sum_{i=1}^K \hat{Y}_i \hat{\phi}_i - (\hat{\gamma} - 1)\hat{\mathbf{u}}^2 + (\hat{\gamma} - 1)\hat{H}\right)^{1/2}$.

(S5) Use $\Delta \mathbf{q} = \mathbf{q}_R - \mathbf{q}_L$ and Δp to compute the wave strengths a_m.

(S6) Calculate $\mathcal{W}_1 = a_1 \hat{\mathbf{r}}_1$, $\mathcal{W}_2 = \displaystyle\sum_{\iota=2}^{K+d} a_\iota \hat{\mathbf{r}}_\iota$, $\mathcal{W}_3 = a_{K+d+1} \hat{\mathbf{r}}_{K+d+1}$.

(S7) Evaluate $s_1 = \hat{u}_1 - \hat{c}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{c}$.

(S8) Evaluate $\rho^\star_{L/R}$, $u^\star_{1,L/R}$, $e^\star_{L/R}$, $c^\star_{1,L/R}$ from $\mathbf{q}^\star_L = \mathbf{q}_L + \mathcal{W}_1$ and $\mathbf{q}^\star_R = \mathbf{q}_R - \mathcal{W}_3$.

(S9) If $\rho^\star_{L/R} \le 0$ or $e^\star_{L/R} \le 0$ use $\mathbf{F}_{HLL}(\mathbf{q}_L, \mathbf{q}_R)$ and go to (S12).

(S10) Entropy correction: Evaluate $|\tilde{s}_\iota|$.
$\mathbf{F}_{Roe}(\mathbf{q}_L, \mathbf{q}_R) = \frac{1}{2}\left(\mathbf{f}(\mathbf{q}_L) + \mathbf{f}(\mathbf{q}_R) - \sum_{\iota=1}^3 |\tilde{s}_\iota| \mathcal{W}_\iota\right)$

Riemann solver for combustion

(S1) Calculate standard Roe-averages $\hat\rho$, $\hat u_n$, $\hat H$, $\hat Y_i$, $\hat T$.

(S2) Compute $\hat\gamma := \hat c_p/\hat c_v$ with $\hat c_{\{p/v\}i} = \dfrac{1}{T_R - T_L}\displaystyle\int_{T_L}^{T_R} c_{\{p,v\}i}(\tau)\,d\tau$.

(S3) Calculate $\hat\phi_i := (\hat\gamma - 1)\left(\frac{\hat{\mathbf{u}}^2}{2} - \hat h_i\right) + \hat\gamma\, R_i\, \hat T$ with standard Roe-averages $\hat e_i$ or $\hat h_i$.

(S4) Calculate $\hat c := \left(\sum_{i=1}^K \hat Y_i\, \hat\phi_i - (\hat\gamma - 1)\hat{\mathbf{u}}^2 + (\hat\gamma - 1)\hat H\right)^{1/2}$.

(S5) Use $\Delta\mathbf{q} = \mathbf{q}_R - \mathbf{q}_L$ and Δp to compute the wave strengths a_m.

(S6) Calculate $\mathcal{W}_1 = a_1\hat{\mathbf{r}}_1$, $\mathcal{W}_2 = \displaystyle\sum_{\iota=2}^{K+d} a_\iota \hat{\mathbf{r}}_\iota$, $\mathcal{W}_3 = a_{K+d+1}\hat{\mathbf{r}}_{K+d+1}$.

(S7) Evaluate $s_1 = \hat u_1 - \hat c$, $s_2 = \hat u_1$, $s_3 = \hat u_1 + \hat c$.

(S8) Evaluate $\rho^\star_{L/R}$, $u^\star_{1,L/R}$, $e^\star_{L/R}$, $c^\star_{1,L/R}$ from $\mathbf{q}^\star_L = \mathbf{q}_L + \mathcal{W}_1$ and $\mathbf{q}^\star_R = \mathbf{q}_R - \mathcal{W}_3$.

(S9) If $\rho^\star_{L/R} \le 0$ or $e^\star_{L/R} \le 0$ use $\mathbf{F}_{HLL}(\mathbf{q}_L, \mathbf{q}_R)$ and go to (S12).

(S10) Entropy correction: Evaluate $|\tilde s_\iota|$.

$$\mathbf{F}_{Roe}(\mathbf{q}_L, \mathbf{q}_R) = \tfrac{1}{2}\left(\mathbf{f}(\mathbf{q}_L) + \mathbf{f}(\mathbf{q}_R) - \textstyle\sum_{\iota=1}^3 |\tilde s_\iota| \mathcal{W}_\iota\right)$$

(S11) Positivity correction: Replace $\mathbf{F}_i$ by

$$\mathbf{F}_i^\star = \mathbf{F}_\rho \cdot \begin{cases} Y_i^l\,, & \mathbf{F}_\rho \ge 0\,, \\ Y_i^r\,, & \mathbf{F}_\rho < 0\,. \end{cases}$$

(S12) Evaluate maximal signal speed by $S = \max(|s_1|, |s_3|)$.

Riemann solver for combustion: carbuncle fix

Entropy corrections

Riemann solver for combustion: carbuncle fix

Entropy corrections [Harten, 1983]

1. $$|\tilde{s}_\iota| = \begin{cases} |s_\iota| & \text{if} |s_\iota| \geq 2\eta \\ \frac{|s_\iota^2|}{4\eta} + \eta & \text{otherwise} \end{cases}$$
   $$\eta = \frac{1}{2} \max_\iota \left\{ |s_\iota(\mathbf{q}_R) - s_\iota(\mathbf{q}_L)| \right\}$$

Riemann solver for combustion: carbuncle fix

Entropy corrections [Harten, 1983] [Harten and Hyman, 1983]

1. $|\tilde{s}_\iota| = \begin{cases} |s_\iota| & \text{if} |s_\iota| \geq 2\eta \\ \frac{|s_\iota^2|}{4\eta} + \eta & \text{otherwise} \end{cases}$
   $\eta = \frac{1}{2} \max_\iota \left\{ |s_\iota(\mathbf{q}_R) - s_\iota(\mathbf{q}_L)| \right\}$
2. Replace $|s_\iota|$ by $|\tilde{s}_\iota|$ only if $s_\iota(\mathbf{q}_L) < 0 < s_\iota(\mathbf{q}_R)$

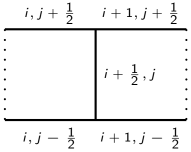
Riemann solver for combustion: carbuncle fix

Entropy corrections [Harten, 1983] [Harten and Hyman, 1983]

1. $|\tilde{s}_\iota| = \begin{cases} |s_\iota| & \text{if} |s_\iota| \geq 2\eta \\ \frac{|s_\iota^2|}{4\eta} + \eta & \text{otherwise} \end{cases}$

   $\eta = \frac{1}{2} \max_\iota \left\{ |s_\iota(\mathbf{q}_R) - s_\iota(\mathbf{q}_L)| \right\}$
2. Replace $|s_\iota|$ by $|\tilde{s}_\iota|$ only if $s_\iota(\mathbf{q}_L) < 0 < s_\iota(\mathbf{q}_R)$

2D modification of entropy correction [Sanders et al., 1998]:



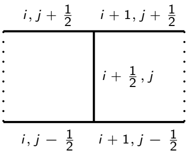
$$\tilde{\eta}_{i+1/2,j} = \max \left\{ \eta_{i+1/2,j}, \eta_{i,j-1/2}, \ \eta_{i,j+1/2}, \eta_{i+1,j-1/2}, \eta_{i+1,j+1/2} \right\}$$

Riemann solver for combustion: carbuncle fix

Entropy corrections [Harten, 1983] [Harten and Hyman, 1983]

1. $|\tilde{s}_\iota| = \begin{cases} |s_\iota| & \text{if} |s_\iota| \geq 2\eta \\ \frac{|s_\iota^2|}{4\eta} + \eta & \text{otherwise} \end{cases}$
   $\eta = \frac{1}{2} \max_\iota \left\{ |s_\iota(\mathbf{q}_R) - s_\iota(\mathbf{q}_L)| \right\}$
2. Replace $|s_\iota|$ by $|\tilde{s}_\iota|$ only if $s_\iota(\mathbf{q}_L) < 0 < s_\iota(\mathbf{q}_R)$

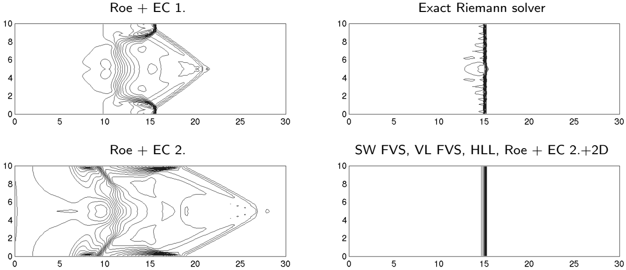
2D modification of entropy correction [Sanders et al., 1998]:



$$\tilde{\eta}_{i+1/2,j} = \max \left\{ \eta_{i+1/2,j}, \eta_{i,j-1/2}, \ \eta_{i,j+1/2}, \eta_{i+1,j-1/2}, \eta_{i+1,j+1/2} \right\}$$

Carbuncle phenomenon

- [Quirk, 1994b]
- Test from [Deiterding, 2003]

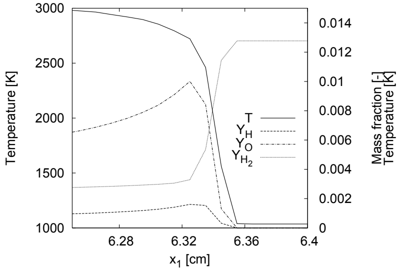


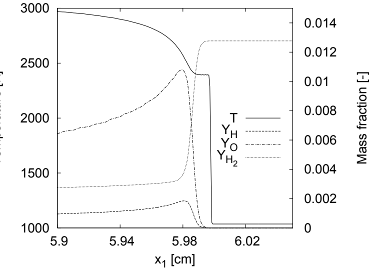
Detonations - motivation for SAMR

- Extremly high spatial resolution in reaction zone necessary.

Detonations - motivation for SAMR

- Extremly high spatial resolution in reaction zone necessary.
- Minimal spatial resolution: $7 - 8\,\mathrm{Pts}/l_{ig} \longrightarrow \Delta x_1 \approx 0.2 - 0.175\,\mathrm{mm}$

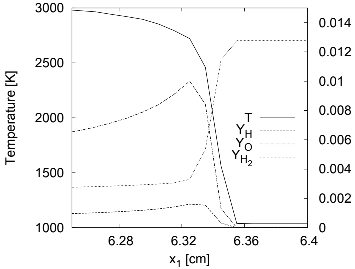


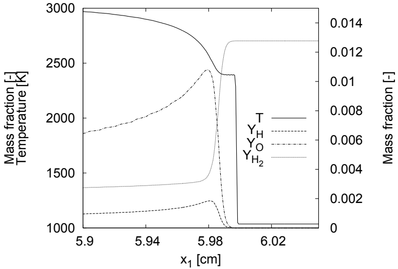


Approximation of $H_2 : O_2$ detonation at $\sim 1.5\,\mathrm{Pts}/l_{ig}$ (left) and $\sim 24\,\mathrm{Pts}/l_{ig}$ (right)

Detonations - motivation for SAMR

- Extremly high spatial resolution in reaction zone necessary.
- Minimal spatial resolution: $7-8\,\mathrm{Pts}/l_{ig} \longrightarrow \Delta x_1 \approx 0.2-0.175\,\mathrm{mm}$
- Uniform grids for typical geometries: $>10^7\,\mathrm{Pts}$ in 2D, $>10^9\,\mathrm{Pts}$ in 3D $\longrightarrow$ Self-adaptive finite volume method (AMR)





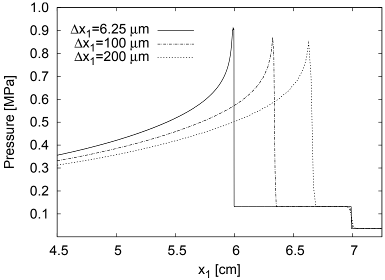
Approximation of $\mathrm{H_2} : \mathrm{O_2}$ detonation at $\sim 1.5\,\mathrm{Pts}/l_{ig}$ (left) and $\sim 24\,\mathrm{Pts}/l_{ig}$ (right)

Detonation ignition in a shock tube

- Shock-induced detonation ignition of $\mathrm{H_2 : O_2 : Ar}$ mixture at molar ratios 2:1:7 in closed 1d shock tube
- Insufficient resolution leads to inaccurate results

Detonation ignition in a shock tube

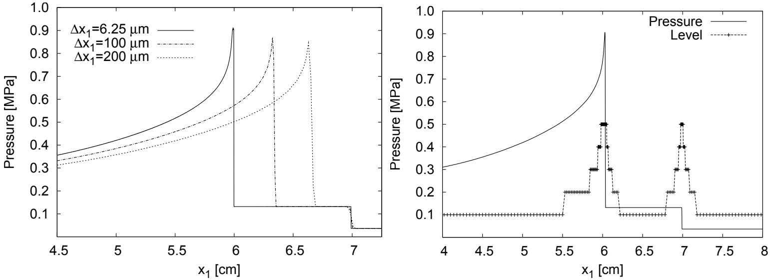
- Shock-induced detonation ignition of $\mathrm{H_2 : O_2 : Ar}$ mixture at molar ratios 2:1:7 in closed 1d shock tube
- Insufficient resolution leads to inaccurate results
- Reflected shock is captured correctly by FV scheme, detonation is resolution dependent



Left: Comparison of pressure distribution $t = 170\,\mu\mathrm{s}$ after shock reflection.

Detonation ignition in a shock tube

- Shock-induced detonation ignition of $\mathrm{H_2 : O_2 : Ar}$ mixture at molar ratios 2:1:7 in closed 1d shock tube
- Insufficient resolution leads to inaccurate results
- Reflected shock is captured correctly by FV scheme, detonation is resolution dependent
- Fine mesh necessary in the induction zone at the head of the detonation



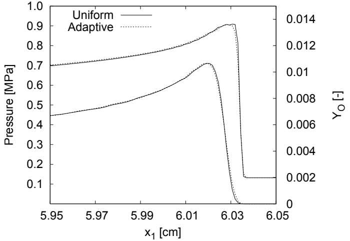
Left: Comparison of pressure distribution $t = 170\,\mu\mathrm{s}$ after shock reflection.Right: Domains of refinement levels

Detonation ignition in 1d - adaptive vs. uniform

Uniformly refined vs. dynamic adaptive simulations (Intel Xeon 3.4 GHz CPU)

	Uniform			Adaptive			
$\Delta x_1[\mu\mathrm{m}]$	Cells	$t_m[\mu\mathrm{s}]$	Time [s]	l_{max}	r_l	$t_m[\mu\mathrm{s}]$	Time [s]
400	300	166.1	31				
200	600	172.6	90	2	2	172.6	99
100	1200	175.5	277	3	2,2	175.8	167
50	2400	176.9	858	4	2,2,2	177.3	287
25	4800	177.8	2713	4	2,2,4	177.9	393
12.5	9600	178.3	9472	5	2,2,2,4	178.3	696
6.25	19200	178.6	35712	5	2,2,4,4	178.6	1370

$\sim 12\,\mathrm{Pts}/l_{ig}$

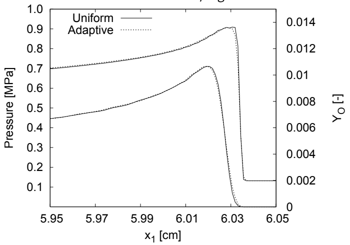


Detonation ignition in 1d - adaptive vs. uniform

Uniformly refined vs. dynamic adaptive simulations (Intel Xeon 3.4 GHz CPU)

	Uniform			Adaptive			
$\Delta x_1[\mu\mathrm{m}]$	Cells	$t_m[\mu\mathrm{s}]$	Time [s]	l_{max}	r_l	$t_m[\mu\mathrm{s}]$	Time [s]
400	300	166.1	31				
200	600	172.6	90	2	2	172.6	99
100	1200	175.5	277	3	2,2	175.8	167
50	2400	176.9	858	4	2,2,2	177.3	287
25	4800	177.8	2713	4	2,2,4	177.9	393
12.5	9600	178.3	9472	5	2,2,2,4	178.3	696
6.25	19200	178.6	35712	5	2,2,4,4	178.6	1370

$\sim 12\,\mathrm{Pts}/l_{ig}$



Refinement criteria:

Y_i	$S_{Y_i} \cdot 10^{-4}$	$\eta^r_{Y_i} \cdot 10^{-3}$
O_2	10.0	2.0
H_2O	7.8	8.0
H	0.16	5.0
O	1.0	5.0
OH	1.8	5.0
H_2	1.3	2.0

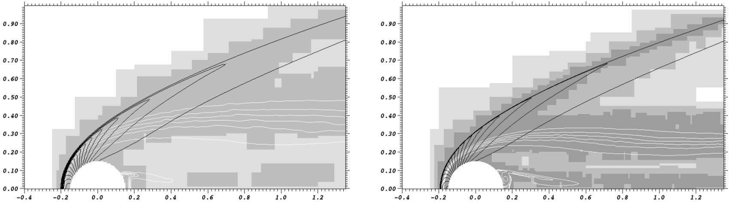
$\epsilon_\rho = 0.07\,\mathrm{kg\,m^{-3}}$, $\epsilon_p = 50\,\mathrm{kPa}$

Shock-induced combustion around a sphere

- Spherical projectile of radius 1.5 mm travels with constant velocity $v_I = 2170.6\,\mathrm{m/s}$ through $\mathrm{H_2 : O_2 : Ar}$ mixture (molar ratios 2:1:7) at 6.67 kPa and $T = 298\,\mathrm{K}$
- Cylindrical symmetric simulation on AMR base mesh of 70×40 cells

Shock-induced combustion around a sphere

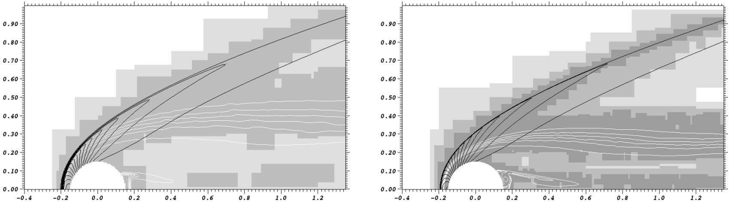
- Spherical projectile of radius 1.5 mm travels with constant velocity $v_I = 2170.6\,\mathrm{m/s}$ through $\mathrm{H_2 : O_2 : Ar}$ mixture (molar ratios 2:1:7) at 6.67 kPa and $T = 298\,\mathrm{K}$
- Cylindrical symmetric simulation on AMR base mesh of 70×40 cells
- Comparison of 3-level computation with refinement factors 2,2 ($\sim 5\,\mathrm{Pts}/l_{ig}$) and a 4-level computation with refinement factors 2,2,4 ($\sim 19\,\mathrm{Pts}/l_{ig}$) at $t = 350\,\mu\mathrm{s}$



Iso-contours of p (black) and Y_{H_2} (white) on refinement domains for 3-level (left) and 4-level computation (right)

Shock-induced combustion around a sphere

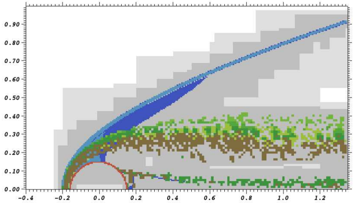
- Spherical projectile of radius 1.5 mm travels with constant velocity $v_I = 2170.6\,\mathrm{m/s}$ through $\mathrm{H_2 : O_2 : Ar}$ mixture (molar ratios 2:1:7) at 6.67 kPa and $T = 298\,\mathrm{K}$
- Cylindrical symmetric simulation on AMR base mesh of 70×40 cells
- Comparison of 3-level computation with refinement factors 2,2 ($\sim 5\,\mathrm{Pts}/l_{ig}$) and a 4-level computation with refinement factors 2,2,4 ($\sim 19\,\mathrm{Pts}/l_{ig}$) at $t = 350\,\mu\mathrm{s}$
- Higher resolved computation captures combustion zone visibly better and at slightly different position (see below)



Iso-contours of p (black) and Y_{H_2} (white) on refinement domains for 3-level (left) and 4-level computation (right)

Combustion around a sphere - adaptation

Refinement indicators on $l = 2$ at $t = 350\,\mu\mathrm{s}$.
Blue: ϵ_ρ, light blue: ϵ_p, green shades: $\eta^r_{Y_i}$,
red: embedded boundary



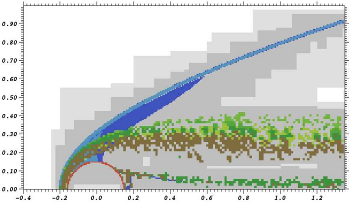
Refinement criteria:

Y_i	$S_{Y_i} \cdot 10^{-4}$	$\eta^r_{Y_i} \cdot 10^{-4}$
O_2	10.0	4.0
H_2O	5.8	3.0
H	0.2	10.0
O	1.4	10.0
OH	2.3	10.0
H_2	1.3	4.0

$\epsilon_\rho = 0.02\,\mathrm{kg\,m^{-3}}$, $\epsilon_p = 16\,\mathrm{kPa}$

Combustion around a sphere - adaptation

Refinement indicators on $l = 2$ at $t = 350\,\mu$s. Blue: ϵ_ρ, light blue: ϵ_p, green shades: $\eta^r_{Y_i}$, red: embedded boundary

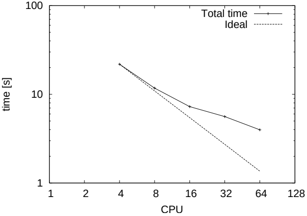


Refinement criteria:

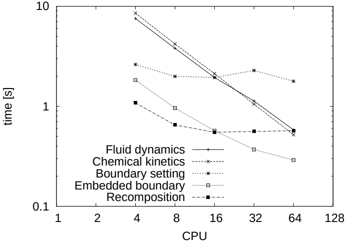
Y_i	$S_{Y_i} \cdot 10^{-4}$	$\eta^r_{Y_i} \cdot 10^{-4}$
O_2	10.0	4.0
H_2O	5.8	3.0
H	0.2	10.0
O	1.4	10.0
OH	2.3	10.0
H_2	1.3	4.0

$\epsilon_\rho = 0.02\,\mathrm{kg\,m^{-3}}$, $\epsilon_p = 16\,\mathrm{kPa}$

Parallel performance

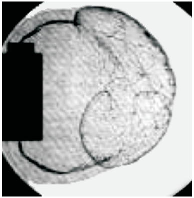


Scaling of different code portions

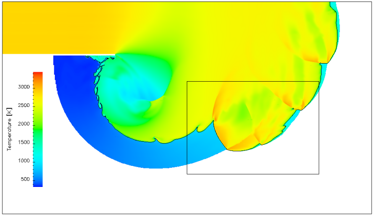


Detonation diffraction

- CJ detonation for $\mathrm{H_2 : O_2 : Ar}/2:1:7$ at $T_0 = 298\,\mathrm{K}$ and $p_0 = 10\,\mathrm{kPa}$. Cell width $\lambda_c = 1.6\,\mathrm{cm}$
- Adaption criteria (similar as before):
  1. Scaled gradients of ρ and p
  2. Error estimation in Y_i by Richardson extrapolation
- $25\,\mathrm{Pts}/l_{ig}$. 5 refinement levels (2,2,2,4).
- Adaptive computations use up to $\sim 2.2\,\mathrm{M}$ instead of $\sim 150\,\mathrm{M}$ cells (uniform grid)
- $\sim 3850\,\mathrm{h}$ CPU ($\sim 80\,\mathrm{h}$ real time) on 48 nodes Athlon 1.4GHz

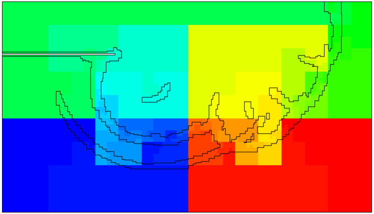


E. Schultz. *Detonation diffraction through an abrupt area expansion*. PhD thesis, California Institute of Technology, Pasadena, California, April 2000.



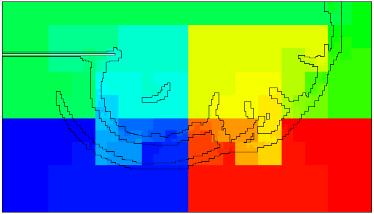
Detonation diffraction - adaptation

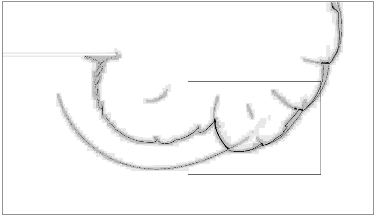
Final distribution to 48 nodes and density distribution on four refinement levels



Detonation diffraction - adaptation

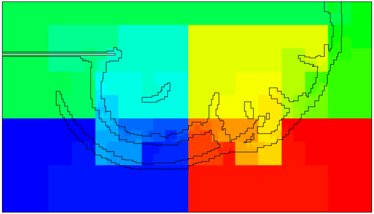
Final distribution to 48 nodes and density distribution on four refinement levels

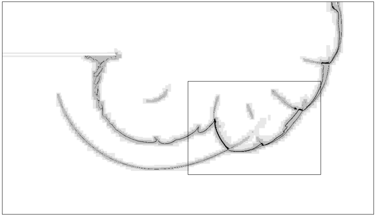


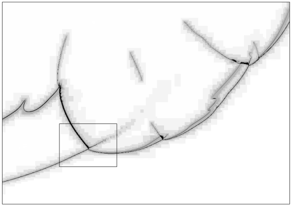


Detonation diffraction - adaptation

Final distribution to 48 nodes and density distribution on four refinement levels

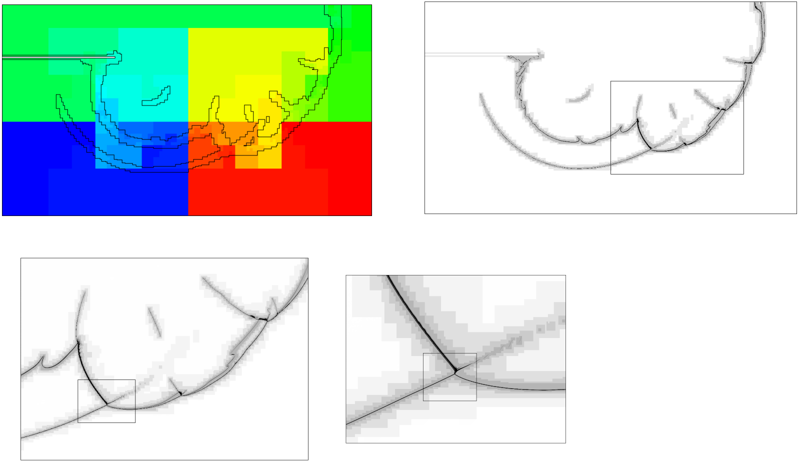






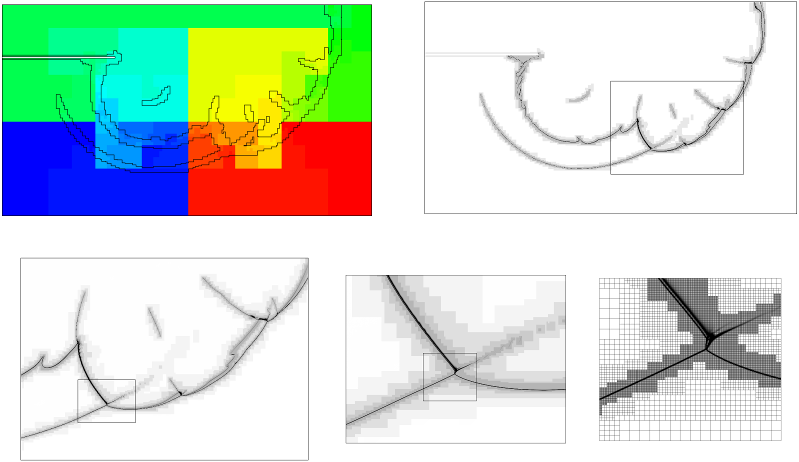
Detonation diffraction - adaptation

Final distribution to 48 nodes and density distribution on four refinement levels



Detonation diffraction - adaptation

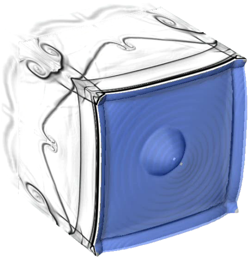
Final distribution to 48 nodes and density distribution on four refinement levels



Detonation cell structure in 3D

- Simulation of only one quadrant
- 44.8 Pts/l_{ig} for $\mathrm{H}_2 : \mathrm{O}_2 : \mathrm{Ar}$ CJ detonation
- SAMR base grid 400x24x24, 2 additional refinement levels (2, 4)
- Simulation uses $\sim$ 18 M cells instead of $\sim$ 118 M (unigrid)
- $\sim$ 51, 000 h CPU on 128 CPU Compaq Alpha. $\mathcal{H}$: 37.6 %, $\mathcal{S}$: 25.1 %

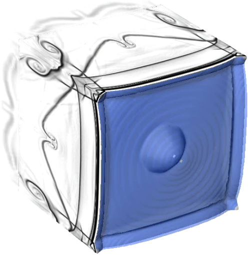
Schlieren and isosurface of Y_{OH}



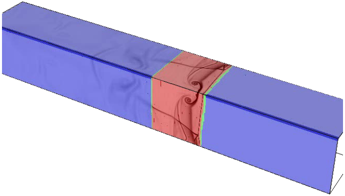
Detonation cell structure in 3D

- Simulation of only one quadrant
- $44.8\,\mathrm{Pts}/l_{ig}$ for $\mathrm{H_2}:\mathrm{O_2}:\mathrm{Ar}$ CJ detonation
- SAMR base grid 400x24x24, 2 additional refinement levels (2, 4)
- Simulation uses $\sim 18\,\mathrm{M}$ cells instead of $\sim 118\,\mathrm{M}$ (unigrid)
- $\sim 51,000\,\mathrm{h}$ CPU on 128 CPU Compaq Alpha. $\mathcal{H}$: 37.6 %, $\mathcal{S}$: 25.1 %

Schlieren and isosurface of Y_{OH}



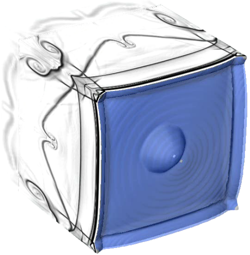
Schlieren on refinement levels



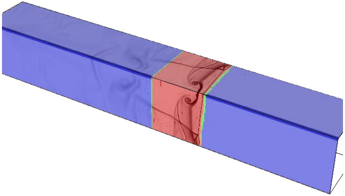
Detonation cell structure in 3D

- Simulation of only one quadrant
- 44.8 Pts/l_{ig} for $\mathrm{H_2} : \mathrm{O_2} : \mathrm{Ar}$ CJ detonation
- SAMR base grid 400x24x24, 2 additional refinement levels (2, 4)
- Simulation uses $\sim$ 18 M cells instead of $\sim$ 118 M (unigrid)
- $\sim$ 51,000 h CPU on 128 CPU Compaq Alpha. $\mathcal{H}$: 37.6 %, $\mathcal{S}$: 25.1 %

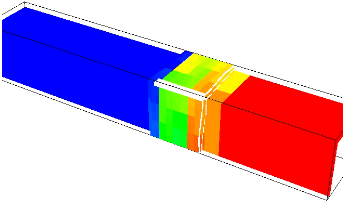
Schlieren and isosurface of Y_{OH}



Schlieren on refinement levels



Distribution to 128 processors



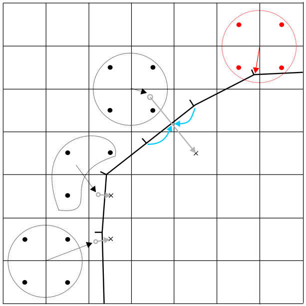
Outline

Construction of coupling data

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

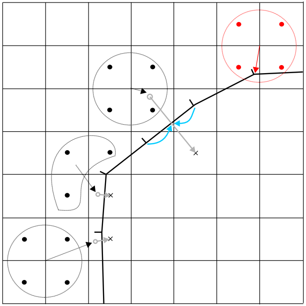
Construction of coupling data

- 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



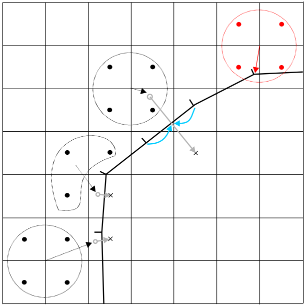
Construction of coupling data

- 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



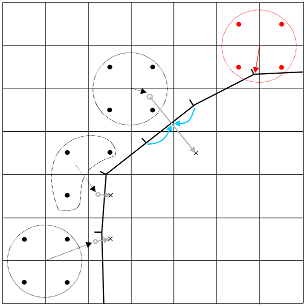
Coupling conditions on interface

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

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 $l_c \leq l_{\max}$

Usage of SAMR

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

Usage of SAMR

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

Usage of SAMR

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

Usage of SAMR

- Eulerian SAMR + non-adaptive Lagrangian FEM scheme
- Exploit SAMR time step refinement for effective coupling to solid solver
  - Lagrangian simulation is called only at level $l_c \leq l_{\max}$
  - 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 $\mathbf{Q}^l(t)$
    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 $\mathbf{Q}^l(t+\Delta t_l)$

    $t := t + \Delta t_l$

- 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

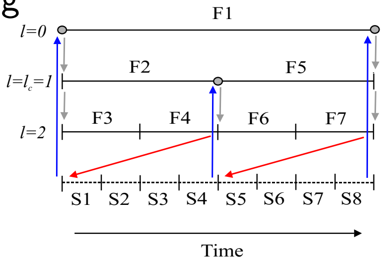
SAMR algorithm for FSI coupling

```
AdvanceLevel(l)

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

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

SAMR algorithm for FSI coupling

```
AdvanceLevel(l)

 Repeat r_l times
   Set ghost cells of 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( )
```

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

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

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

$\texttt{AdvanceLevel(0)}$

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

Fluid and solid update / exchange of time steps

`FluidStep( )`

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

`SolidStep( )`

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

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

Fluid and solid update / exchange of time steps

`FluidStep( )`

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

`SolidStep( )`

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

- 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( )`

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

`SolidStep( )`

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

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

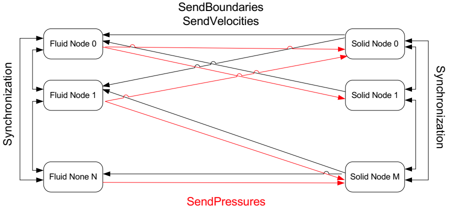
Parallelization strategy for coupled simulations

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:

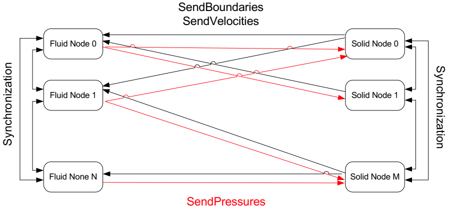
- Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes
- Setting of ghost cell values becomes strictly local operation



Parallelization strategy for coupled simulations

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

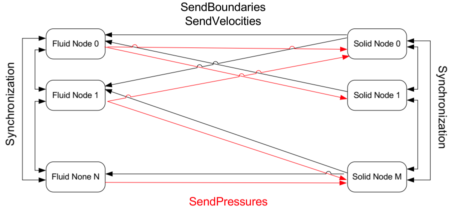
- Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes
- Setting of ghost cell values becomes strictly local operation
- Construct new nodal values strictly local on fluid nodes and transfer them back to solid nodes
- Only surface data is transfered



Parallelization strategy for coupled simulations

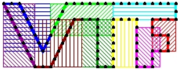
Coupling of an Eulerian FV fluid Solver and a Lagrangian FEM Solver:

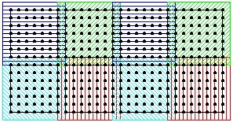
- Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes
- Setting of ghost cell values becomes strictly local operation
- Construct new nodal values strictly local on fluid nodes and transfer them back to solid nodes
- Only surface data is transfered
- Asynchronous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



Eulerian/Lagrangian communication module

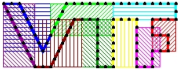
1. Put bounding boxes around each solid processors piece of the boundary and around each fluid processors grid

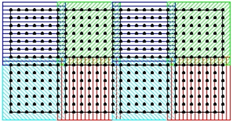


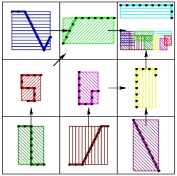


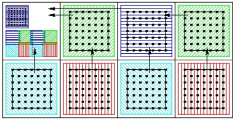
Eulerian/Lagrangian communication module

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



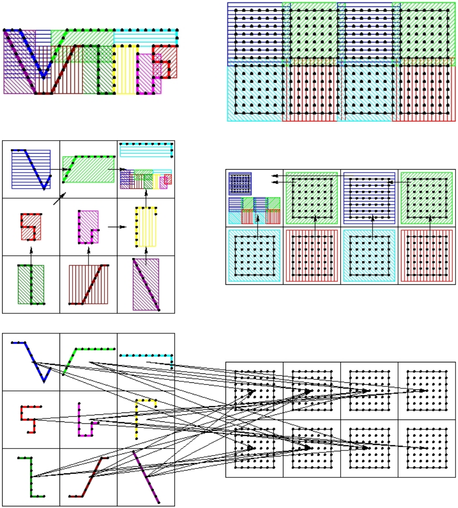






Eulerian/Lagrangian communication module

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

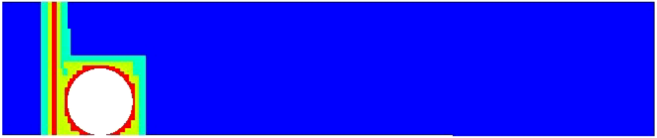


Lift-up of a spherical body

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

Schlieren plot of density

Refinement levels



Treatment of thin structures

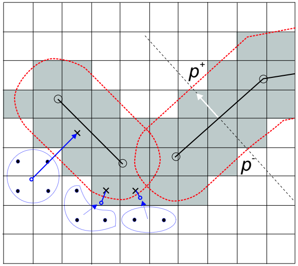
- Thin boundary structures or lower-dimensional shells require “thickening” to apply embedded boundary method

Treatment of thin structures

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

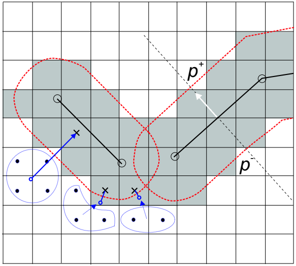
Treatment of thin structures

- 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

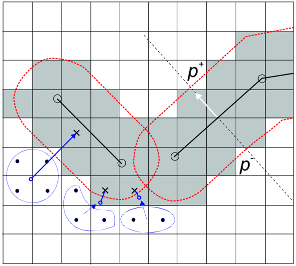
- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method
- Unsigned distance level set function φ
- Treat cells with $0 < \varphi < d$ as ghost fluid cells



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

Treatment of thin structures

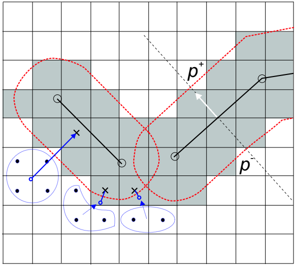
- Thin boundary structures or lower-dimensional shells require "thickening" to apply embedded boundary method
- Unsigned distance level set function φ
- Treat cells with $0 < \varphi < d$ as ghost fluid cells



- 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

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

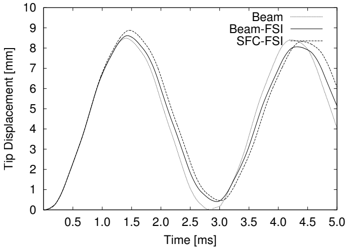
to verify FSI algorithms

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

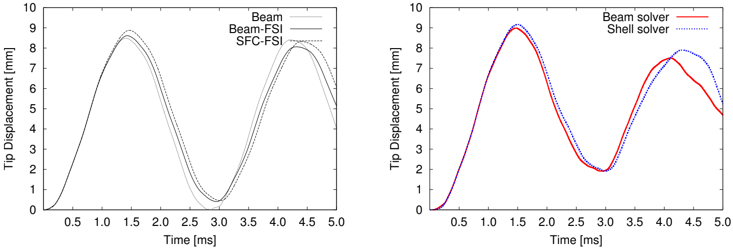
FSI verification by elastic vibration

- 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

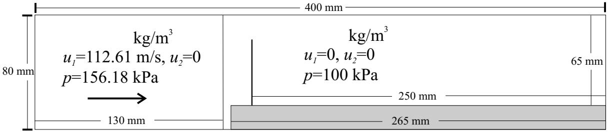
- Thin steel plate (thickness $h = 1\,\mathrm{mm}$, length $50\,\mathrm{mm}$), clamped at lower end
- $\rho_s = 7600\,\mathrm{kg/m^3}$, $E = 220\,\mathrm{GPa}$, $I = h^3/12$, $\nu = 0.3$
- Modeled with beam solver (101 points) and thin-shell FEM solver (325 triangles) by F. Cirak
- Left: Coupling verification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

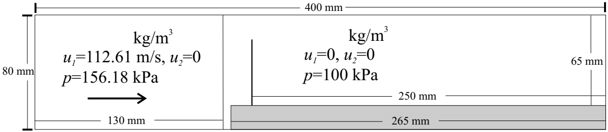


- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$

Shock-driven elastic panel motion

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

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

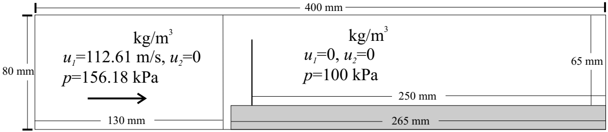


- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$
- Intel 3.4GHz Xeon dual processors, GB Ethernet interconnect
  - Beam-FSI: $12.25\,\mathrm{h}$ CPU on 3 fluid CPU + 1 solid CPU
  - FEM-FSI: $322\,\mathrm{h}$ CPU on 14 fluid CPU + 2 solid CPU

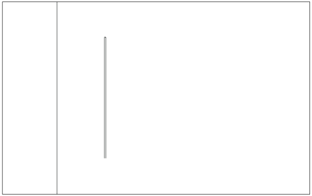
Shock-driven elastic panel motion

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

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



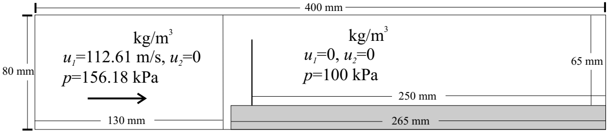
- SAMR base mesh $320 \times 64(\times 2)$, $r_{1,2} = 2$
- Intel 3.4GHz Xeon dual processors, GB Ethernet interconnect
  - Beam-FSI: 12.25 h CPU on 3 fluid CPU + 1 solid CPU
  - FEM-FSI: 322 h CPU on 14 fluid CPU + 2 solid CPU



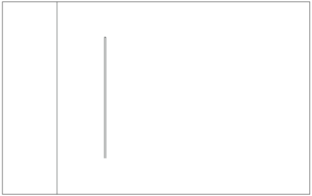
Shock-driven elastic panel motion

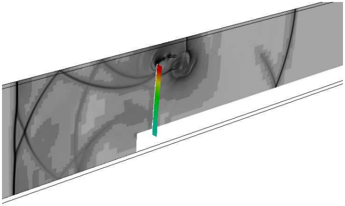
Test case suggested by [Giordano et al., 2005]

- 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
  - FEM-FSI: $322\,\mathrm{h}$ CPU on 14 fluid CPU + 2 solid CPU



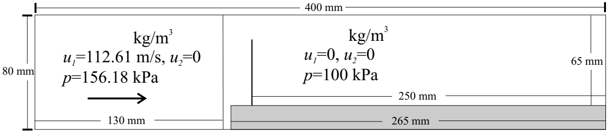


$t = 0.43\,\mathrm{ms}$ after impact

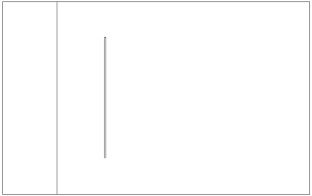
Shock-driven elastic panel motion

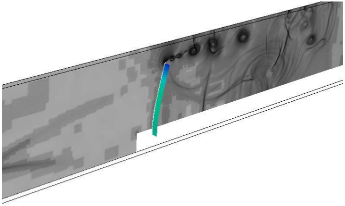
Test case suggested by [Giordano et al., 2005]

- 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
  - FEM-FSI: 322 h CPU on 14 fluid CPU + 2 solid CPU





$t = 1.56$ 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 $A \longrightarrow B$

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

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

with

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

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` $p' := p$
  $\rho_{\mathrm{A}} := Y'\rho$
  $E := p'/(\rho(\gamma - 1)) + Y' q_0 + \frac{1}{2} u_n u_n$

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 $A \longrightarrow B$

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

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

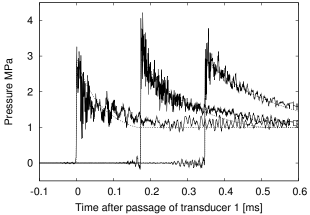
with

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

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

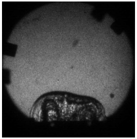
Tube with flaps

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

Tube with flaps

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

0.032 ms

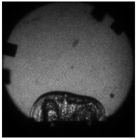


0.030 ms

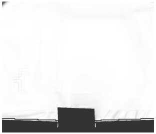
Tube with flaps

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

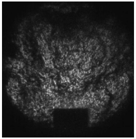
0.032 ms



0.030 ms

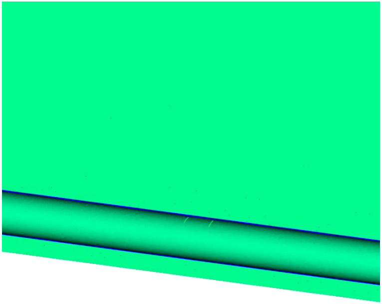


0.212 ms



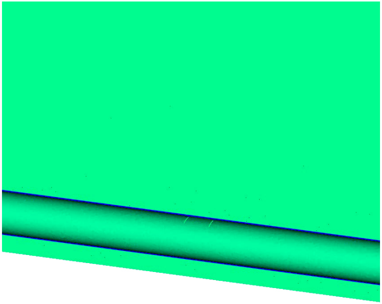
0.210 ms

Tube with flaps: results

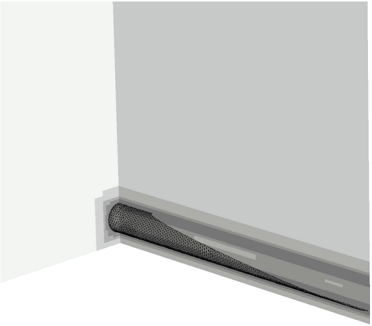


Fluid density and diplacement in y-direction in solid

Tube with flaps: results



Fluid density and diplacement in y-direction in solid



Schlieren plot of fluid density on refinement levels

[Cirak et al., 2007]

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]

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

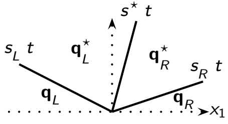
$$\frac{\partial}{\partial t}\left(\frac{1}{\gamma - 1}\right) + u_n \frac{\partial}{\partial x_n}\left(\frac{1}{\gamma - 1}\right) = 0 \ , \quad \frac{\partial}{\partial t}\left(\frac{\gamma p_\infty}{\gamma - 1}\right) + u_n \frac{\partial}{\partial x_n}\left(\frac{\gamma p_\infty}{\gamma - 1}\right)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

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

Approximate Riemann solver

Use HLLC approach because of robustness and positivity preservation

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

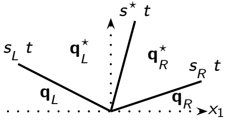


Wave speed estimates [Davis, 1988] $s_L = \min\{u_{1,L} - c_L, u_{1,R} - c_R\}$, $s_R = \max\{u_{1,L} + c_L, u_{1,R} + c_R\}$

Approximate Riemann solver

Use HLLC approach because of robustness and positivity preservation

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



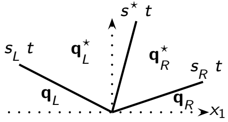
Wave speed estimates [Davis, 1988] $s_L = \min\{u_{1,L} - c_L, u_{1,R} - c_R\}$, $s_R = \max\{u_{1,L} + c_L, u_{1,R} + c_R\}$
Unkown state [Toro et al., 1994]

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

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]

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

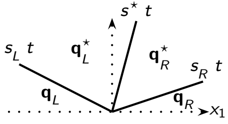
$$\mathbf{q}_\tau^\star = \left[\eta, \eta s^\star, \eta u_2, \eta\left[\frac{(\rho E)_\tau}{\rho_\tau} + (s^\star - u_{1,\tau})\left(s_\tau + \frac{p_\tau}{\rho_\tau(s_\tau - u_{1,\tau})}\right)\right], \frac{1}{\gamma_\tau - 1}, \frac{\gamma_\tau p_{\infty,\tau}}{\gamma_\tau - 1}\right]^T$$

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

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

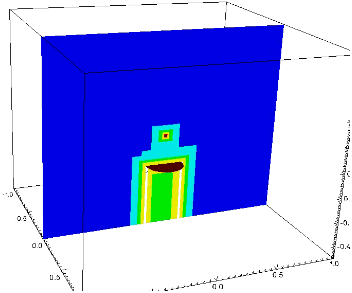
- Air: $\gamma^A = 1.4$, $p_\infty^A = 0$, $\rho^A = 1.29\,\mathrm{kg/m^3}$
- Water: $\gamma^W = 7.415$, $p_\infty^W = 296.2\,\mathrm{MPa}$, $\rho^W = 1027\,\mathrm{kg/m^3}$
- Cavitation modeling with pressure cut-off model at $p = -1\,\mathrm{MPa}$
- 3D simulation of deformation of air backed aluminum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

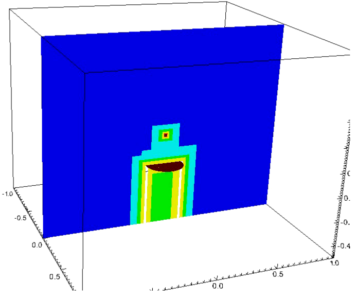


Underwater explosion simulation

- 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 [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



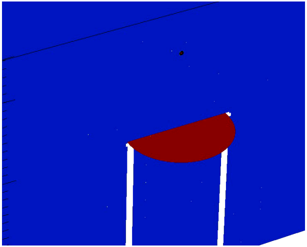
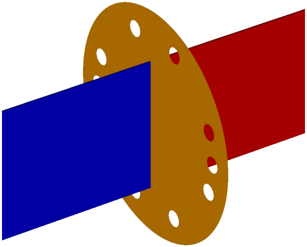


Plate in underwater shocktube

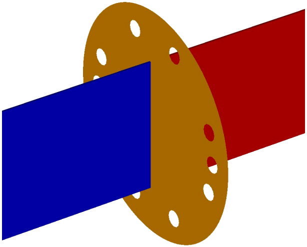
- AMR base mesh $374 \times 20 \times 20$, $r_{1,2} = 2$, $l_c = 2$, solid mesh: 8896 triangles
- $\sim$ 1250 coupled time steps to $t_{end} = 1\,\mathrm{ms}$
- 6+6 nodes 3.4 GHz Intel Xeon dual processor, $\sim 800\,\mathrm{h}$ CPU

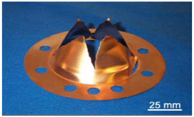


$p_0 = 64\,\mathrm{MPa}$

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

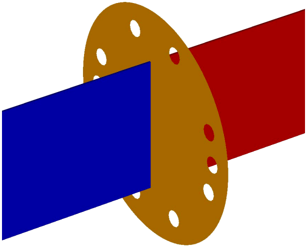


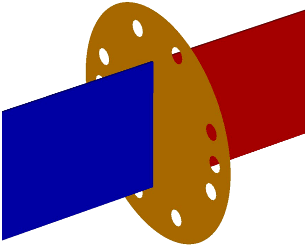


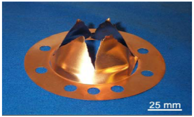
$p_0 = 64\,\mathrm{MPa}$

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





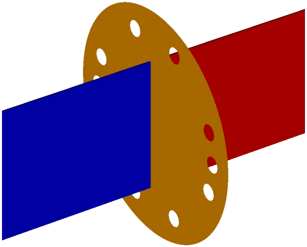


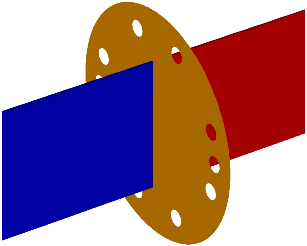
$p_0 = 64\,\mathrm{MPa}$

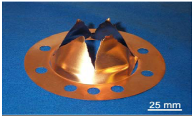
$p_0 = 173\,\mathrm{MPa}$

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







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

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

Outline

Favre-averaged Navier-Stokes equations

$$\frac{\partial\bar\rho}{\partial t}+\frac{\partial}{\partial x_n}\left(\bar\rho\tilde u_n\right)=0$$

$$\frac{\partial}{\partial t}\left(\bar\rho\tilde u_k\right)+\frac{\partial}{\partial x_n}\left(\bar\rho\tilde u_k\tilde u_n+\delta_{kn}\bar p-\tilde\tau_{kn}+\sigma_{kn}\right)=0$$

$$\frac{\partial\bar\rho\bar E}{\partial t}+\frac{\partial}{\partial x_n}\left(\tilde u_n(\bar\rho\bar E+\bar p)+\tilde q_n-\tilde\tau_{nj}\tilde u_j+\sigma_n^e\right)=0$$

$$\frac{\partial}{\partial t}\left(\bar\rho\tilde Y_i\right)+\frac{\partial}{\partial x_n}\left(\bar\rho\tilde Y_i\tilde u_n+\tilde J_n^i+\sigma_n^i\right)=0$$

with stress tensor

$$\tilde\tau_{kn}=\tilde\mu\left(\frac{\partial\tilde u_n}{\partial x_k}+\frac{\partial\tilde u_k}{\partial x_n}\right)-\frac{2}{3}\tilde\mu\frac{\partial\tilde u_j}{\partial x_j}\delta_{in}\ ,$$

heat conduction

$$\tilde q_n=-\tilde\lambda\frac{\partial\tilde T}{\partial x_n}\ ,$$

Favre-averaged Navier-Stokes equations

$$\frac{\partial\bar\rho}{\partial t} + \frac{\partial}{\partial x_n}\left(\bar\rho\tilde u_n\right) = 0$$

$$\frac{\partial}{\partial t}\left(\bar\rho\tilde u_k\right) + \frac{\partial}{\partial x_n}\left(\bar\rho\tilde u_k\tilde u_n + \delta_{kn}\bar p - \tilde\tau_{kn} + \sigma_{kn}\right) = 0$$

$$\frac{\partial\bar\rho\bar E}{\partial t} + \frac{\partial}{\partial x_n}\left(\tilde u_n(\bar\rho\bar E + \bar p) + \tilde q_n - \tilde\tau_{nj}\tilde u_j + \sigma_n^e\right) = 0$$

$$\frac{\partial}{\partial t}\left(\bar\rho\tilde Y_i\right) + \frac{\partial}{\partial x_n}\left(\bar\rho\tilde Y_i\tilde u_n + \tilde J_n^i + \sigma_n^i\right) = 0$$

with stress tensor

$$\tilde\tau_{kn} = \tilde\mu\left(\frac{\partial\tilde u_n}{\partial x_k} + \frac{\partial\tilde u_k}{\partial x_n}\right) - \frac{2}{3}\tilde\mu\frac{\partial\tilde u_j}{\partial x_j}\delta_{in} \ ,$$

heat conduction

$$\tilde q_n = -\tilde\lambda\frac{\partial\tilde T}{\partial x_n} \ ,$$

and inter-species diffusion

$$\tilde J_n^i = -\bar\rho\tilde D_i\frac{\partial\tilde Y_i}{\partial x_n}$$

Favre-averaged Navier-Stokes equations

$$\frac{\partial \bar\rho}{\partial t} + \frac{\partial}{\partial x_n}\left(\bar\rho \tilde u_n\right) = 0$$
$$\frac{\partial}{\partial t}\left(\bar\rho \tilde u_k\right) + \frac{\partial}{\partial x_n}\left(\bar\rho \tilde u_k \tilde u_n + \delta_{kn}\bar p - \tilde\tau_{kn} + \sigma_{kn}\right) = 0$$
$$\frac{\partial \bar\rho \bar E}{\partial t} + \frac{\partial}{\partial x_n}\left(\tilde u_n(\bar\rho \bar E + \bar p) + \tilde q_n - \tilde\tau_{nj}\tilde u_j + \sigma_n^e\right) = 0$$
$$\frac{\partial}{\partial t}\left(\bar\rho \tilde Y_i\right) + \frac{\partial}{\partial x_n}\left(\bar\rho \tilde Y_i \tilde u_n + \tilde J_n^i + \sigma_n^i\right) = 0$$

with stress tensor

$$\tilde\tau_{kn} = \tilde\mu\left(\frac{\partial \tilde u_n}{\partial x_k} + \frac{\partial \tilde u_k}{\partial x_n}\right) - \frac{2}{3}\tilde\mu \frac{\partial \tilde u_j}{\partial x_j}\delta_{in} ,$$

heat conduction

$$\tilde q_n = -\tilde\lambda \frac{\partial \tilde T}{\partial x_n} ,$$

and inter-species diffusion

$$\tilde J_n^i = -\bar\rho \tilde D_i \frac{\partial \tilde Y_i}{\partial x_n}$$

Favre-filtering

$$\tilde\phi = \frac{\overline{\rho\phi}}{\bar\rho} \quad \text{with} \quad \bar\phi(\mathbf{x}, t; \Delta_c) = \int_\Omega G(\mathbf{x} - \mathbf{x}'; \Delta_c)\phi(\mathbf{x}', t)d\mathbf{x}'$$

Numerical solution approach

- Subgrid terms σ_{kn}, σ_n^e, σ_n^i are computed by Pullin's stretched-vortex model

Numerical solution approach

- Subgrid terms σ_{kn}, σ_n^e, σ_n^i are computed by Pullin's stretched-vortex model
- Cutoff Δ_c is set to local SAMR resolution Δx_l

Numerical solution approach

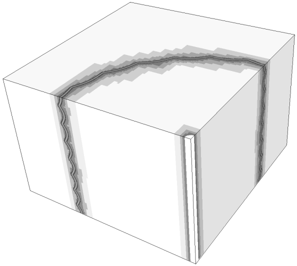
- Subgrid terms σ_{kn}, σ_n^e, σ_n^i are computed by Pullin's stretched-vortex model
- Cutoff Δ_c is set to local SAMR resolution Δx_l
- It remains to solve the Navier-Stokes equations in the hyperbolic regime
  - 3rd order WENO method (hybridized with a tuned centered difference stencil) for convection
  - 2nd order conservative centered differences for diffusion

Numerical solution approach

- Subgrid terms σ_{kn}, σ_n^e, σ_n^i are computed by Pullin's stretched-vortex model
- Cutoff Δ_c is set to local SAMR resolution Δx_l
- It remains to solve the Navier-Stokes equations in the hyperbolic regime
  - 3rd order WENO method (hybridized with a tuned centered difference stencil) for convection
  - 2nd order conservative centered differences for diffusion

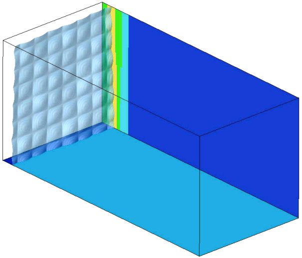
Example: Cylindrical Richtmyer-Meshkov instability

- Sinusoidal interface between two gases hit by shock wave
- Objective is correctly predict turbulent mixing
- Embedded boundary method used to regularize apex
- AMR base grid $95 \times 95 \times 64$ cells, $r_{1,2,3} = 2$
- $\sim 70,000\,\mathrm{h}$ CPU on 32 AMD 2.5GHZ-quad-core nodes



Planar Richtmyer-Meshkov instability

- Perturbed Air-SF6 interface shocked and re-shocked by Mach 1.5 shock
- Containment of turbulence in refined zones
- 96 CPUs IBM SP2-Power3
- WENO-TCD scheme with LES model
- AMR base grid $172 \times 56 \times 56$, $r_{1,2} = 2$, $10\,\mathrm{M}$ cells in average instead of $3\,\mathrm{M}$ (uniform)

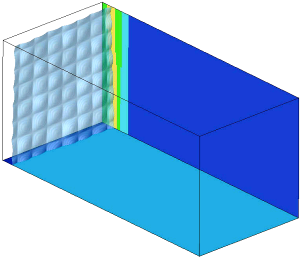


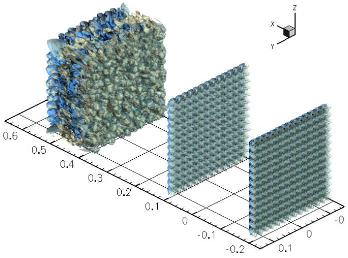
Task	2ms (%)	5ms (%)	10ms (%)
Integration	45.3	65.9	52.0
Boundary setting	44.3	28.6	41.9
Flux correction	7.2	3.4	4.1
Interpolation	0.9	0.4	0.3
Reorganization	1.6	1.2	1.2
Misc.	0.6	0.5	0.5
Max. imbalance	1.25	1.23	1.30

Planar Richtmyer-Meshkov instability

- Perturbed Air-SF6 interface shocked and re-shocked by Mach 1.5 shock
- Containment of turbulence in refined zones
- 96 CPUs IBM SP2-Power3
- WENO-TCD scheme with LES model
- AMR base grid $172 \times 56 \times 56$, $r_{1,2} = 2$, $10\,\mathrm{M}$ cells in average instead of $3\,\mathrm{M}$ (uniform)

Task	2ms (%)	5ms (%)	10ms (%)
Integration	45.3	65.9	52.0
Boundary setting	44.3	28.6	41.9
Flux correction	7.2	3.4	4.1
Interpolation	0.9	0.4	0.3
Reorganization	1.6	1.2	1.2
Misc.	0.6	0.5	0.5
Max. imbalance	1.25	1.23	1.30





Flux correction for Runge-Kutta method

Recall Runge-Kutta temporal update

$$\tilde{\mathbf{Q}}_j^{\upsilon} = \alpha_\upsilon \mathbf{Q}_j^n + \beta_\upsilon \tilde{\mathbf{Q}}_j^{\upsilon-1} + \gamma_\upsilon \frac{\Delta t}{\Delta x_k} \Delta \mathbf{F}^k(\tilde{\mathbf{Q}}^{\upsilon-1})$$

Flux correction for Runge-Kutta method

Recall Runge-Kutta temporal update

$$\tilde{\mathbf{Q}}_j^{\upsilon} = \alpha_{\upsilon} \mathbf{Q}_j^n + \beta_{\upsilon} \tilde{\mathbf{Q}}_j^{\upsilon-1} + \gamma_{\upsilon} \frac{\Delta t}{\Delta x_k} \Delta \mathbf{F}^k(\tilde{\mathbf{Q}}^{\upsilon-1})$$

rewrite scheme as

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n - \sum_{\upsilon=1}^{\Upsilon} \varphi_{\upsilon} \frac{\Delta t}{\Delta x_k} \Delta \mathbf{F}^k(\tilde{\mathbf{Q}}^{\upsilon-1}) \quad \text{with} \quad \varphi_{\upsilon} = \gamma_{\upsilon} \prod_{\nu=\upsilon+1}^{\Upsilon} \beta_{\nu}$$

Flux correction for Runge-Kutta method

Recall Runge-Kutta temporal update

$$\tilde{\mathbf{Q}}_j^{\upsilon} = \alpha_\upsilon \mathbf{Q}_j^n + \beta_\upsilon \tilde{\mathbf{Q}}_j^{\upsilon-1} + \gamma_\upsilon \frac{\Delta t}{\Delta x_k} \Delta \mathbf{F}^k(\tilde{\mathbf{Q}}^{\upsilon-1})$$

rewrite scheme as

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n - \sum_{\upsilon=1}^{\Upsilon} \varphi_\upsilon \frac{\Delta t}{\Delta x_k} \Delta \mathbf{F}^k(\tilde{\mathbf{Q}}^{\upsilon-1}) \quad \text{with} \quad \varphi_\upsilon = \gamma_\upsilon \prod_{\nu=\upsilon+1}^{\Upsilon} \beta_\nu$$

Flux correction to be used [Pantano et al., 2007]

1. $\delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} := -\varphi_1 \mathbf{F}^{1,l}_{i-\frac{1}{2},j}(\tilde{\mathbf{Q}}^0)\,, \qquad \delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} := \delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} - \sum_{\upsilon=2}^{\Upsilon} \varphi_\upsilon \mathbf{F}^{1,l}_{i-\frac{1}{2},j}(\tilde{\mathbf{Q}}^{\upsilon-1})$
2. $\delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} := \delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} + \frac{1}{r_{l+1}^2} \sum_{m=0}^{r_{l+1}-1} \sum_{\upsilon=1}^{\Upsilon} \varphi_\upsilon \mathbf{F}^{1,l+1}_{v+\frac{1}{2},w+m} \left(\tilde{\mathbf{Q}}^{\upsilon-1}(t+\kappa\Delta t_{l+1})\right)$

Flux correction for Runge-Kutta method

Recall Runge-Kutta temporal update

$$\tilde{\mathbf{Q}}_j^{\upsilon} = \alpha_\upsilon \mathbf{Q}_j^n + \beta_\upsilon \tilde{\mathbf{Q}}_j^{\upsilon-1} + \gamma_\upsilon \frac{\Delta t}{\Delta x_k} \Delta \mathbf{F}^k(\tilde{\mathbf{Q}}^{\upsilon-1})$$

rewrite scheme as

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n - \sum_{\upsilon=1}^{\Upsilon} \varphi_\upsilon \frac{\Delta t}{\Delta x_k} \Delta \mathbf{F}^k(\tilde{\mathbf{Q}}^{\upsilon-1}) \quad \text{with} \quad \varphi_\upsilon = \gamma_\upsilon \prod_{\nu=\upsilon+1}^{\Upsilon} \beta_\nu$$

Flux correction to be used [Pantano et al., 2007]

1. $\delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} := -\varphi_1 \mathbf{F}^{1,l}_{i-\frac{1}{2},j}(\tilde{\mathbf{Q}}^0)\,, \qquad \delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} := \delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} - \sum_{\upsilon=2}^{\Upsilon} \varphi_\upsilon \mathbf{F}^{1,l}_{i-\frac{1}{2},j}(\tilde{\mathbf{Q}}^{\upsilon-1})$
2. $\delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} := \delta \mathbf{F}^{1,l+1}_{i-\frac{1}{2},j} + \frac{1}{r_{l+1}^2} \sum_{m=0}^{r_{l+1}-1} \sum_{\upsilon=1}^{\Upsilon} \varphi_\upsilon \mathbf{F}^{1,l+1}_{v+\frac{1}{2},w+m} \left(\tilde{\mathbf{Q}}^{\upsilon-1}(t+\kappa\Delta t_{l+1})\right)$

Storage-efficient SSPRK(3,3):

υ	α_υ	β_υ	γ_υ	φ_υ
1	1	0	1	$\frac{1}{6}$
2	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{6}$
3	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$

References I

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

[Aftosmis, 1997] Aftosmis, M. J. (1997). Solution adaptive Ccartesian grid methods for aerodynamic flows with complex geometries. Technical Report Lecture Series 1997-2, von Karman Institute for Fluid Dynamics.

[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.

[Berger and Helzel, 2002] Berger, M. J. and Helzel, C. (2002). Grid aligned h-box methods for conservation laws in complex geometries. In *Proc. 3rd Intl. Symp. Finite Volumes for Complex Applications*, Porquerolles.

[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.

References II

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

[Deiterding, 2003] Deiterding, R. (2003). *Parallel adaptive simulation of multi-dimensional detonation structures*. PhD thesis, Brandenburgische Technische Universität Cottbus.

[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 et al., 2006] Deiterding, R., Radovitzky, R., Mauch, S. P., Noels, L., Cummings, J. C., and Meiron, D. I. (2006). A virtual test facility for the efficient simulation of solid materials under high energy shock-wave loading. *Engineering with Computers*, 22(3-4):325–347.

[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.

References III

[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.

[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.

[Harten, 1983] Harten, A. (1983). High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 49:357–393.

[Harten and Hyman, 1983] Harten, A. and Hyman, J. M. (1983). Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.*, 50:235–269.

[Henshaw and Schwendeman, 2003] Henshaw, W. D. and Schwendeman, D. W. (2003). An adaptive numerical scheme for high-speed reactive flow on overlapping grids. *J. Comput. Phys.*, 191:420–447.

References IV

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

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

[Meakin, 1995] Meakin, R. L. (1995). An efficient means of adaptive refinement within systems of overset grids. In *12th AIAA Computational Fluid Dynamics Conference, San Diego*, AIAA-95-1722-CP.

[Mittal and Iaccarino, 2005] Mittal, R. and Iaccarino, G. (2005). Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261.

[Murman et al., 2003] Murman, S. M., Aftosmis, M. J., and Berger, M. J. (2003). Implicit approaches for moving boundaries in a 3-d Cartesian method. In *41st AIAA Aerospace Science Meeting*, AIAA 2003-1119.

[Nourgaliev et al., 2003] Nourgaliev, R. R., Dinh, T. N., and Theofanus, T. G. (2003). On capturing of interfaces in multimaterial compressible flows using a level-set-based Cartesian grid method. Technical Report 05/03-1, Center for Risk Studies and Safety, UC Santa Barbara.

References V

[Pantano et al., 2007] Pantano, C., Deiterding, R., Hill, D. J., and Pullin, D. I. (2007). A low-numerical dissipation patch-based adaptive mesh refinement method for large-eddy simulation of compressible flows. *J. Comput. Phys.*, 221(1):63–87.

[Pember et al., 1999] Pember, R. B., Bell, J. B., Colella, P., Crutchfield, W. Y., and Welcome, M. L. (1999). An adaptive Cartesian grid method for unsteady compressible flows in irregular regions. *J. Comput. Phys.*, 120:287–304.

[Quirk, 1994a] Quirk, J. J. (1994a). An alternative to unstructured grids for computing gas dynamics flows around arbitrarily complex two-dimensional bodies. *Computers Fluids*, 23:125–142.

[Quirk, 1994b] Quirk, J. J. (1994b). A contribution to the great Riemann solver debate. *Int. J. Numer. Meth. Fluids*, 18:555–574.

[Roma et al., 1999] Roma, A. M., Perskin, C. S., and Berger, M. J. (1999). An adaptive version of the immersed boundary method. *J. Comput. Phys.*, 153:509–534.

[Sanders et al., 1998] Sanders, R., Morano, E., and Druguett, M.-C. (1998). Multidimensional dissipation for upwind schemes: Stability and applications to gas dynamics. *J. Comput. Phys.*, 145:511–537.

References VI

[Sethian, 1999] Sethian, J. A. (1999). *Level set methods and fast marching methods*. Cambridge University Press, Cambridge, New York.

[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.

[Tseng and Ferziger, 2003] Tseng, Y.-H. and Ferziger, J. H. (2003). A ghost-cell immersed boundary method for flow in complex geometry. *J. Comput. Phys.*, 192:593–623.

[Yamaleev and Carpenter, 2002] Yamaleev, N. K. and Carpenter, M. H. (2002). On accuracy of adaptive grid methods for captured shocks. *J. Comput. Phys.*, 181:280–316.