Lecture 4

Numerical methods for combustion research

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

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

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

Outline

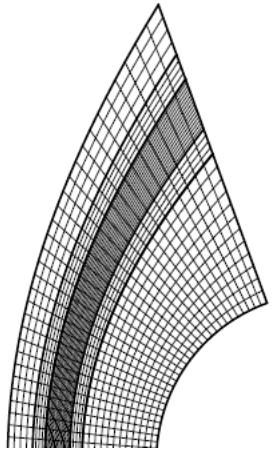
Outline

Outline

SAMR on boundary aligned meshes

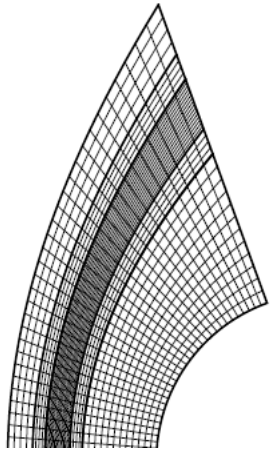
Analytic or stored geometric mapping of the coordinates (graphic from [Yamaleev and Carpenter, 2002])

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

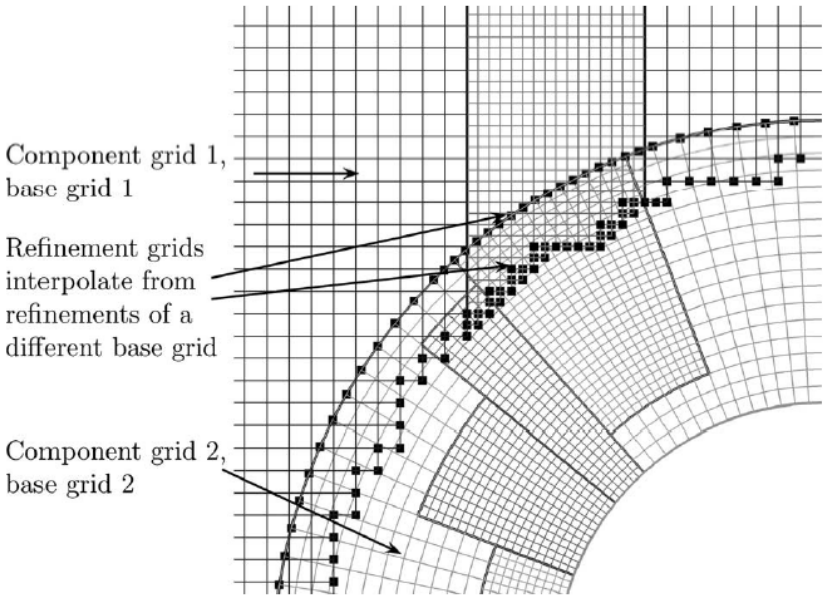


SAMR on boundary aligned meshes

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

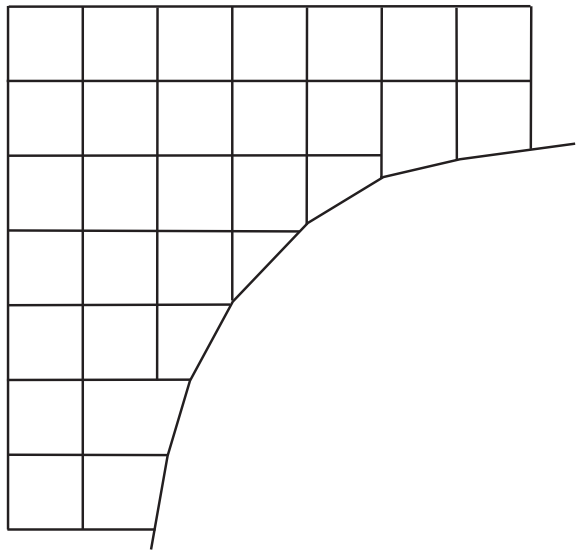
Cut-cell techniques

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:

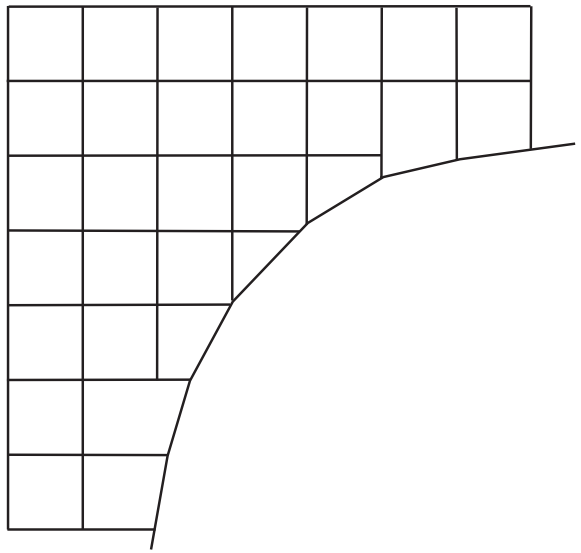
- Cut-cell approach constructs appropriate finite volumes
- Conservative by construction. Correct boundary flux



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

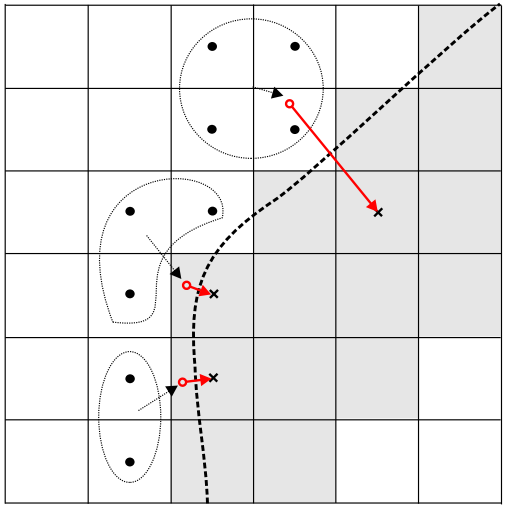
- Redistribution of boundary flux achieves conservation and bypasses time step restriction: [Pember et al., 1999], [Berger and Helzel, 2002]

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

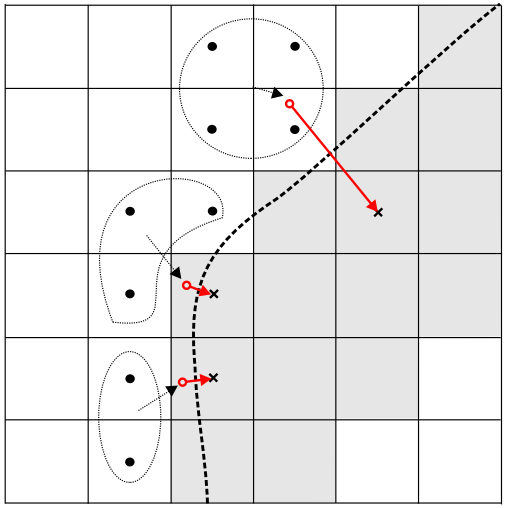
K. J. Richards et al., On the use of the immersed boundary method for engine modeling

Level-set method for boundary embedding



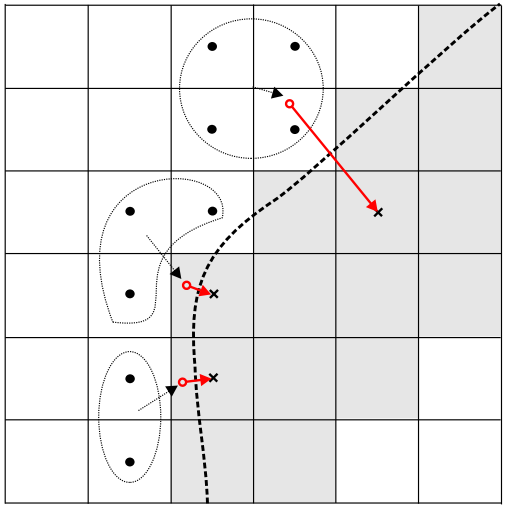
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$

Level-set method for boundary embedding



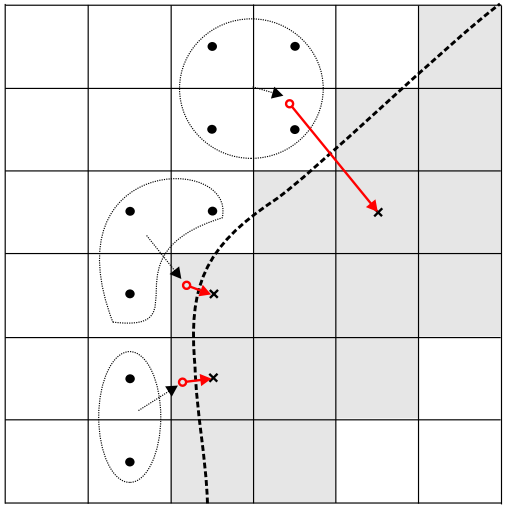
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$
- Complex boundary moving with local velocity $\mathbf{w}$, treat interface as moving rigid wall

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

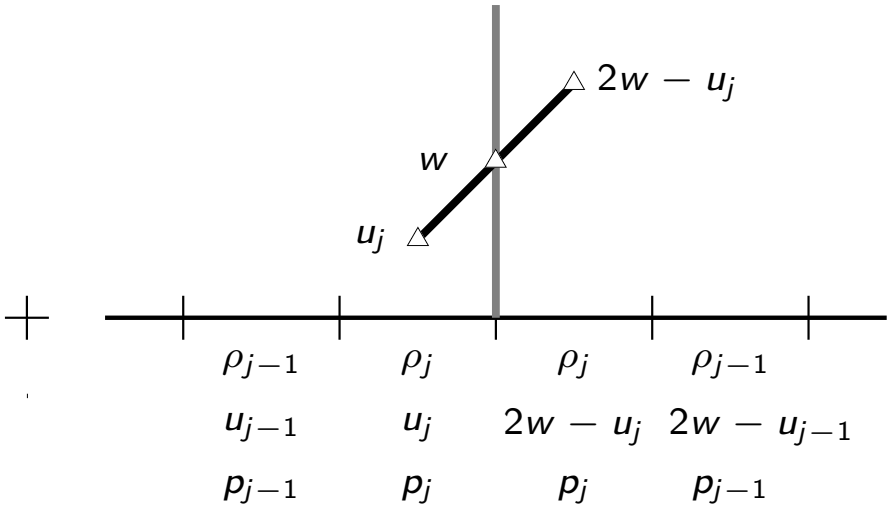
Level-set method for boundary embedding



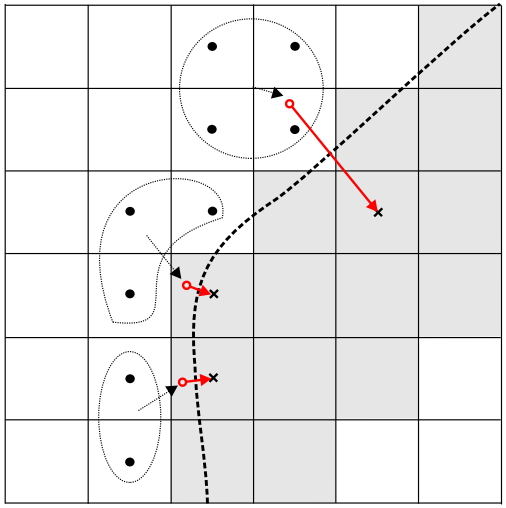
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$
- Complex boundary moving with local velocity $\mathbf{w}$, treat interface as moving rigid wall
- Construction of values in embedded boundary cells by interpolation / extrapolation

Interpolate / constant value extrapolate values at

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



Level-set method for boundary embedding



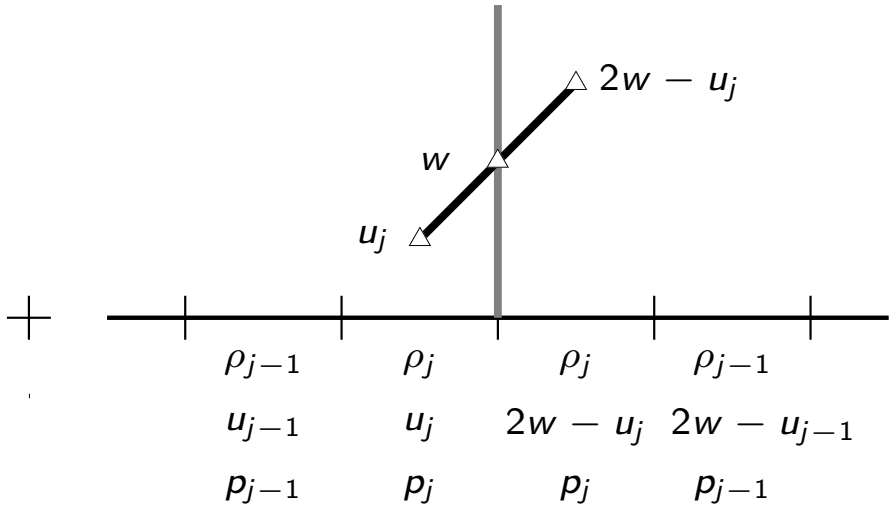
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$
- Complex boundary moving with local velocity $\mathbf{w}$, treat interface as moving rigid wall
- Construction of values in embedded boundary cells by interpolation / extrapolation

Interpolate / constant value extrapolate values at

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

Velocity in ghost cells

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



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

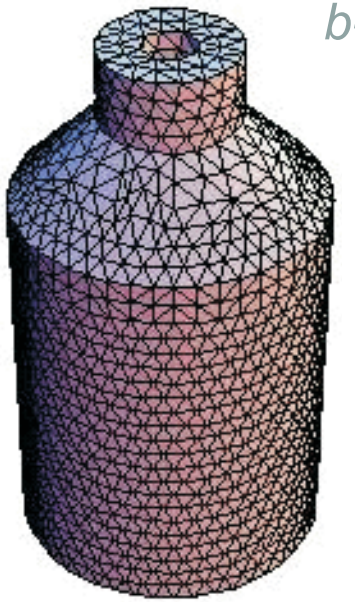
The signed distance φ to a surface $\mathcal{I}$ satisfies the eikonal equation [Sethian, 1999]

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

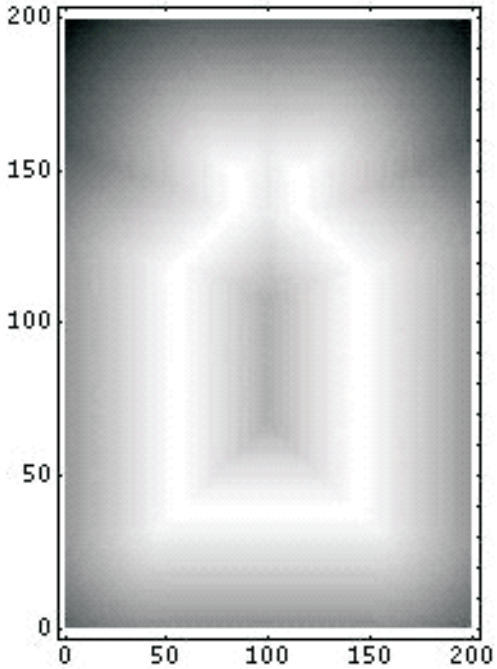
Solution smooth but non-diferentiable across characteristics.

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

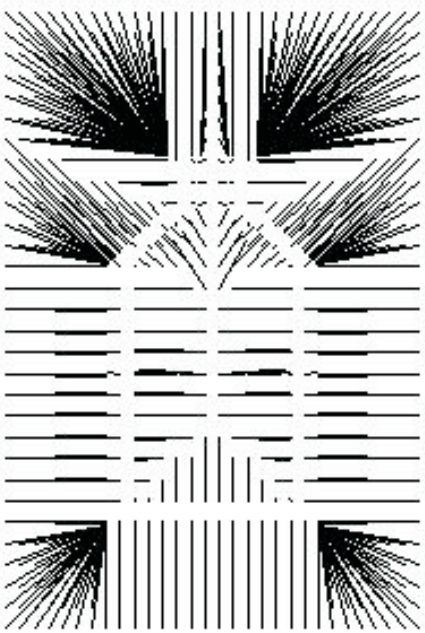
- Geometric solution approach with plosest-point-transform algorithm [Mauch, 2003]



Surface mesh $\mathcal{I}$



Distance φ

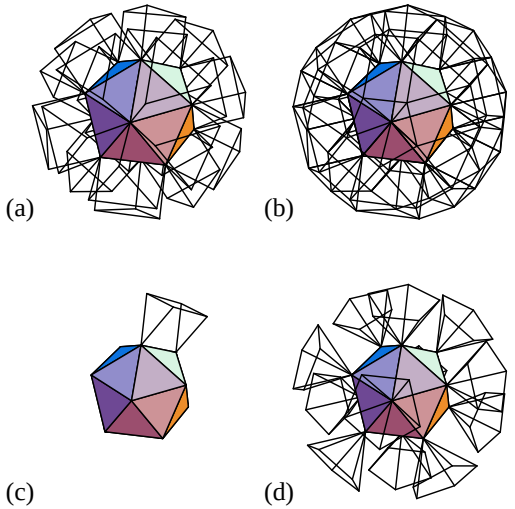


Normal to closest point

The characteristic / scan conversion algorithm

1. Build the characteristic polyhedrons for the surface mesh

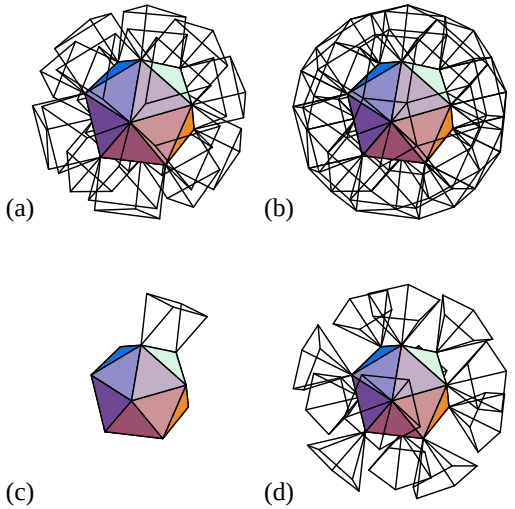
Characteristic polyhedra for faces, edges, and vertices



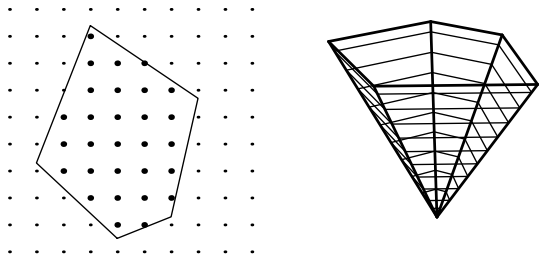
The characteristic / scan conversion algorithm

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



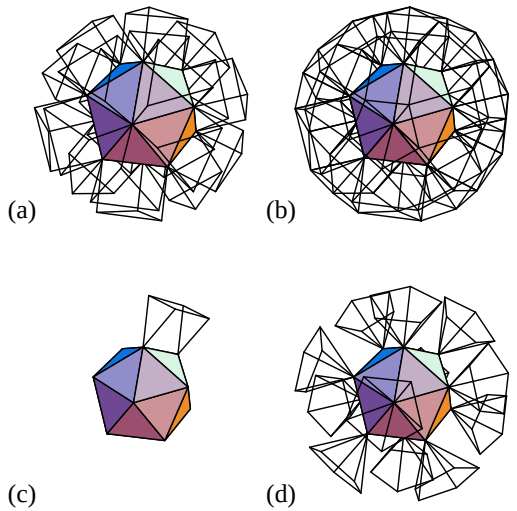
Slicing and scan conversion of apolygon



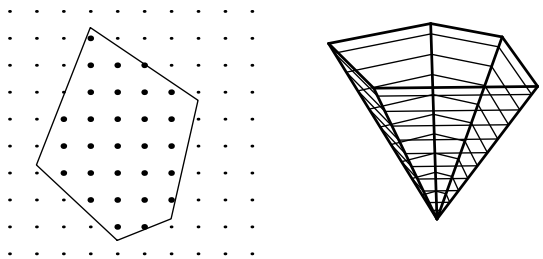
The characteristic / scan conversion algorithm

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



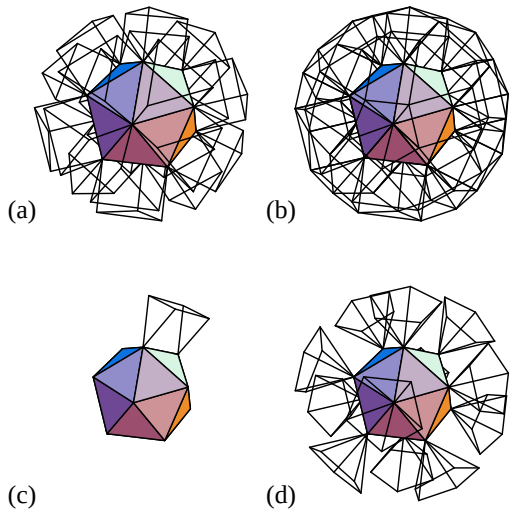
Slicing and scan conversion of apolygon



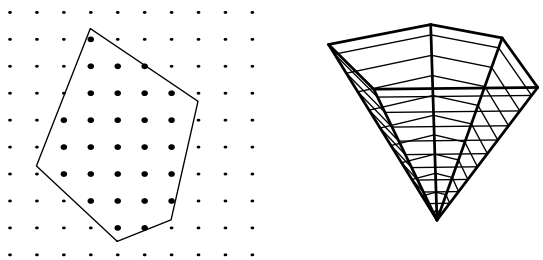
The characteristic / scan conversion algorithm

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

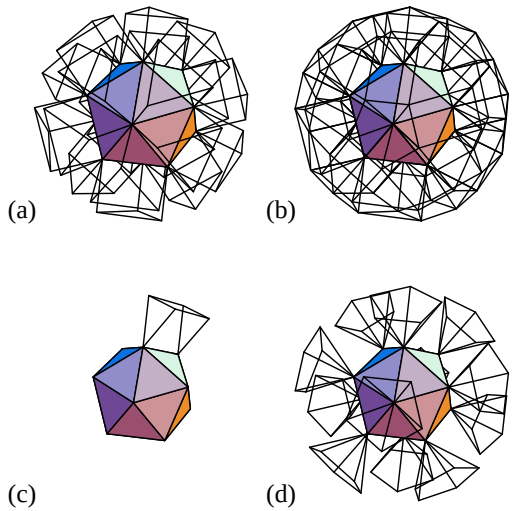


The characteristic / scan conversion algorithm

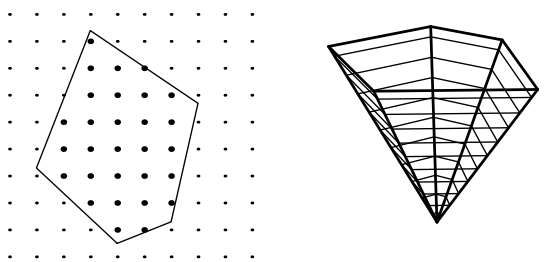
1. Build the characteristic polyhedrons for the surface mesh
2. For each face/edge/vertex
   - 2.1 Scan convert the polyhedron.
   - 2.2 Compute distance to that primitive for the scan converted points
3. Computational complexity.
   - $O(m)$ to build the b-rep and the polyhedra.
   - $O(n)$ to scan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



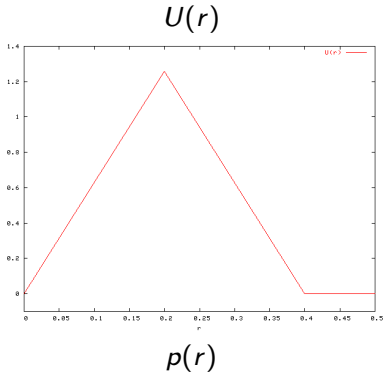
Accuracy test: stationary vortex

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

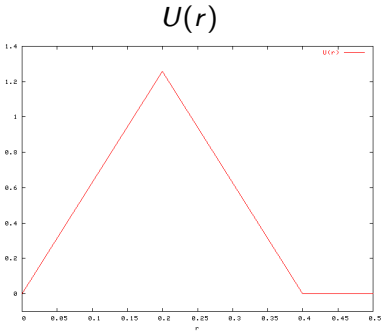
$$U(r) = \alpha \cdot \begin{cases} 2r/R & \text{if } 0 < r < R/2, \\ 2(1 - r/R) & \text{if } R/2 \leq r \leq R, \\ 0 & \text{if } r > R, \end{cases}$$



Accuracy test: stationary vortex

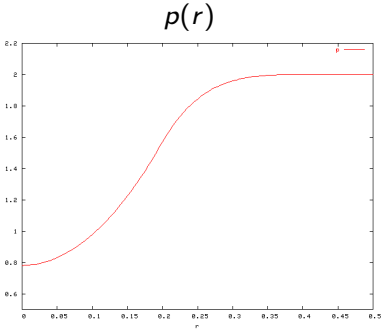
Construct non-trivial *radially symmetric* and *stationary* solution by balancing hydrodynamic pressure and centripetal force per volume element, i.e.

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



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

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



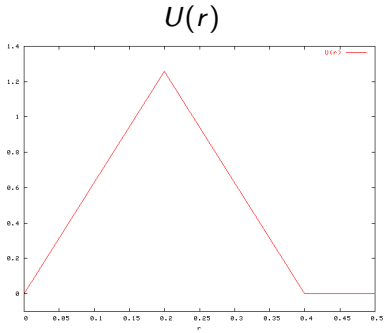
one gets with boundary condition $p(R) = p_0 \equiv 2$ the pressure distribution

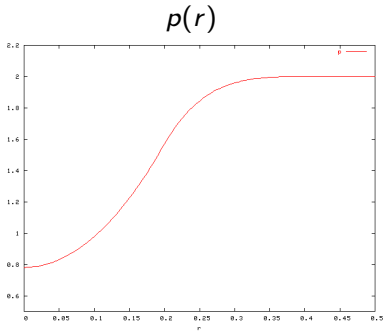
$$p(r) = p_0 + 2\rho_0\alpha^2 \cdot \begin{cases} r^2/R^2 + 1 - 2\log 2 & \text{if } 0 < r < R/2, \\ r^2/R^2 + 3 - 4r/R + 2\log(r/R) & \text{if } R/2 \le r \le R, \\ 0 & \text{if } r > R. \end{cases}$$

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

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

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 $\phi = \arctan\dfrac{x_2 - x_{2,c}}{x_1 - x_{1,c}}$

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, $\alpha = R\pi,\ R_G = R/2,\ U_W = 0$

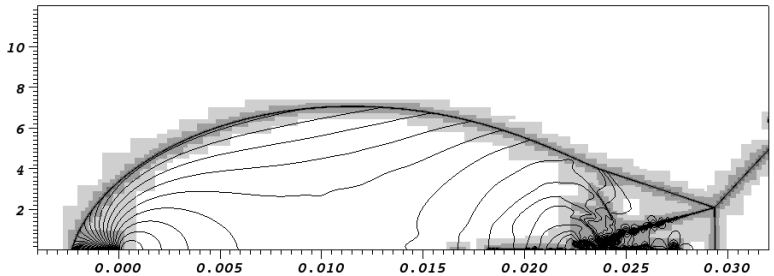
N	Wave Propagation			Godunov Splitting		
	Error	Order	Mass loss	Error	Order	Mass loss
20	0.0423925		0.0423925	0.0271446		0.0271446
40	0.0358735	0.24	0.0358735	0.0242260	0.16	0.0242260
80	0.0212340	0.76	0.0212340	0.0128638	0.91	0.0128638
160	0.0121089	0.81	0.0121089	0.0070906	0.86	0.0070906

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

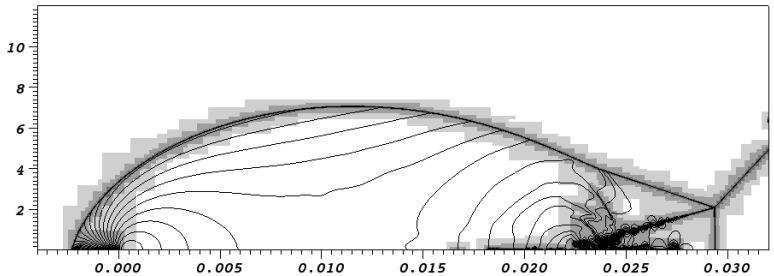
- Reflection of a Mach 2.38 shock in nitrogen at 43^o wedge
- 2nd order MUSCL scheme with Roe solver, 2nd order multidimensional wave propagation method



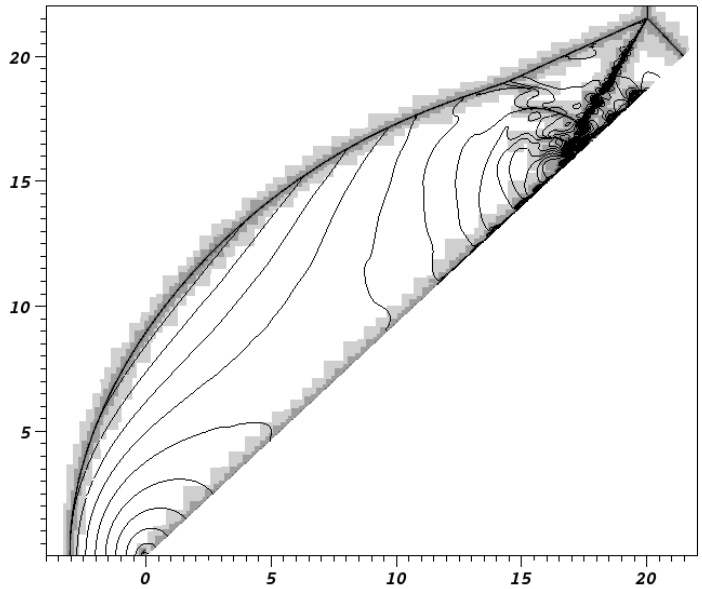
Cartesian base grid 360×160 cells on domain of $36\,\mathrm{mm} \times 16\,\mathrm{mm}$ with up to 3 refinement levels with $r_l = 2, 4, 4$ and $\Delta x_{1,2} = 3.125 \mu m$, $38\,\mathrm{h}$ CPU

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

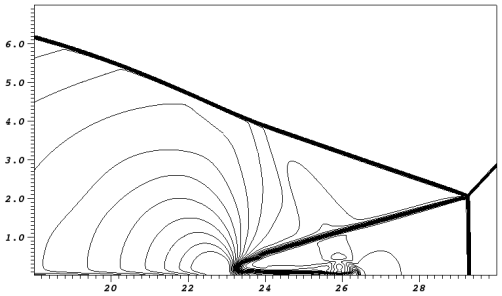


Cartesian base grid 360 $\times$ 160 cells on domain of $36\,\mathrm{mm} \times 16\,\mathrm{mm}$ with up to 3 refinement levels with $r_l = 2, 4, 4$ and $\Delta x_{1,2} = 3.125 \mu m$, 38 h CPU

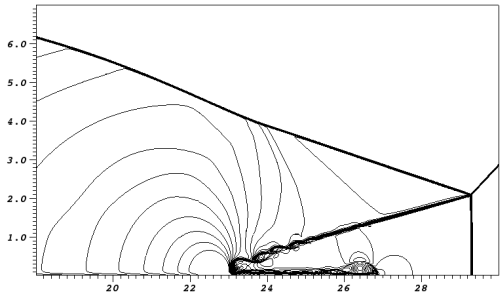


GFM base grid 390 $\times$ 330 cells on domain of $26\,\mathrm{mm} \times 22\,\mathrm{mm}$ with up to 3 refinement levels with $r_l = 2, 4, 4$ and $\Delta x_{e,1,2} = 2.849 \mu m$, 200 h CPU

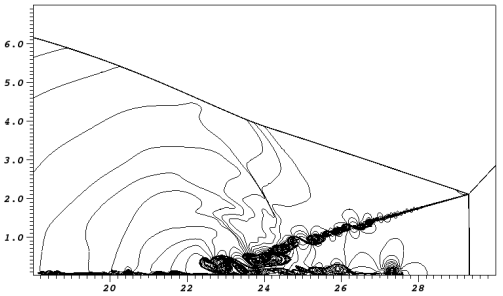
Shock reflection: SAMR solution for Euler equations



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

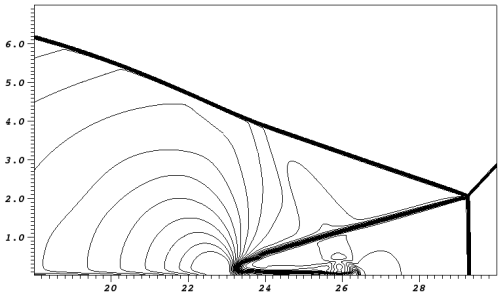


$\Delta x = 12.5\,\mathrm{mm}$

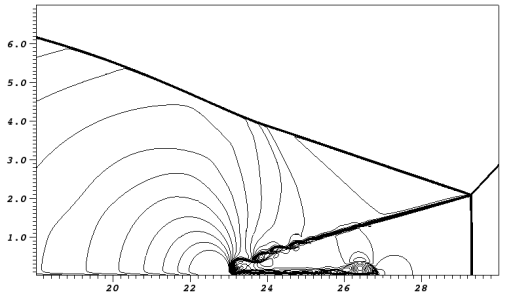


$\Delta x = 3.125\,\mathrm{mm}$

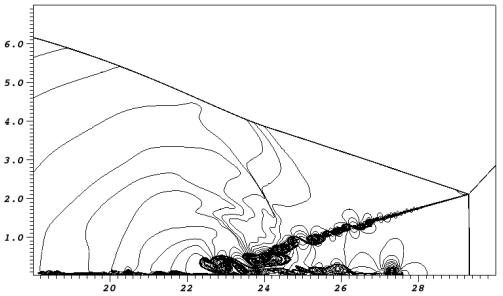
Shock reflection: SAMR solution for Euler equations



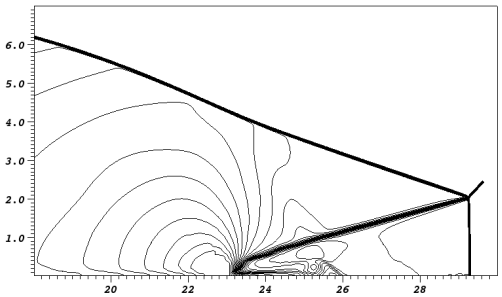
$\Delta x = 25\,\mathrm{mm}$



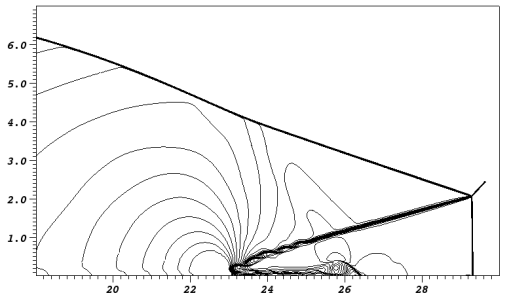
$\Delta x = 12.5\,\mathrm{mm}$



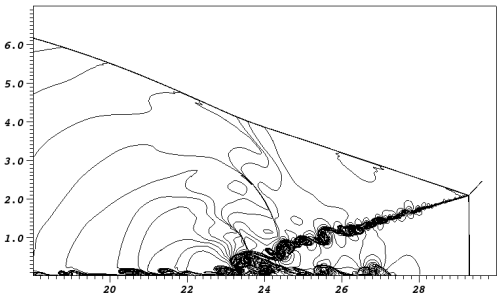
$\Delta x = 3.125\,\mathrm{mm}$



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

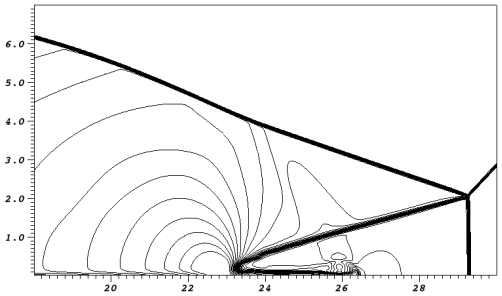


$\Delta x_e = 11.4\,\mathrm{mm}$

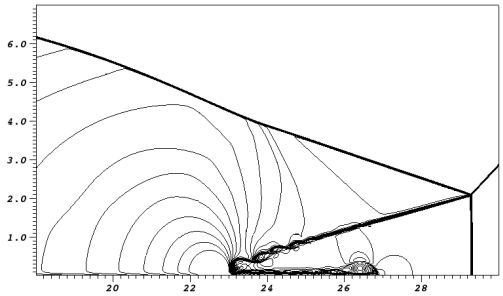


$\Delta x_e = 2.849\,\mathrm{mm}$

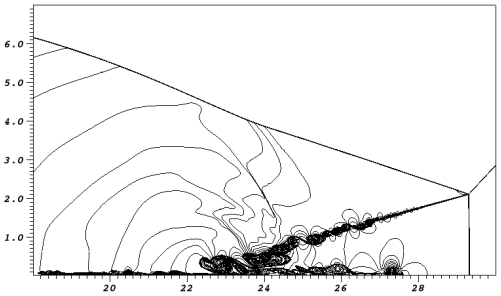
Shock reflection: SAMR solution for Euler equations



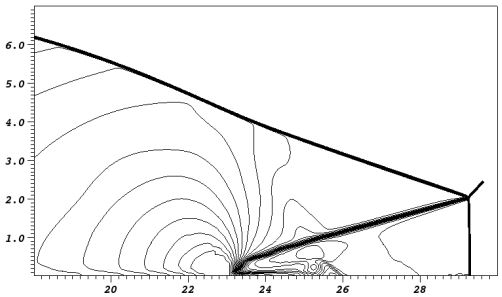
$\Delta x = 25\,\mathrm{mm}$



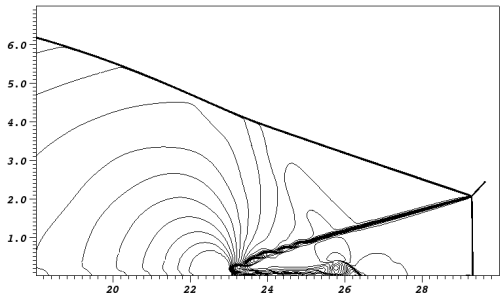
$\Delta x = 12.5\,\mathrm{mm}$



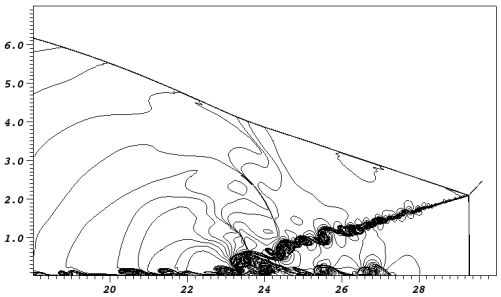
$\Delta x = 3.125\,\mathrm{mm}$



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

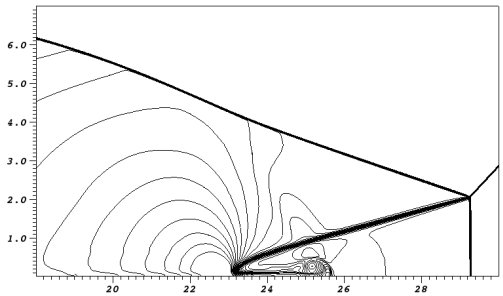


$\Delta x_e = 11.4\,\mathrm{mm}$

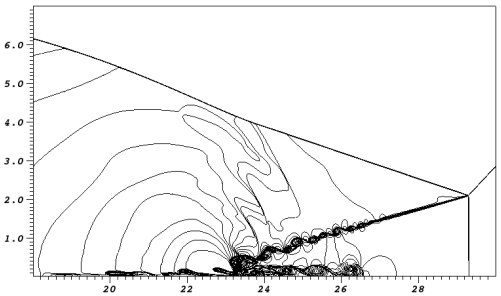


$\Delta x_e = 2.849\,\mathrm{mm}$

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



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



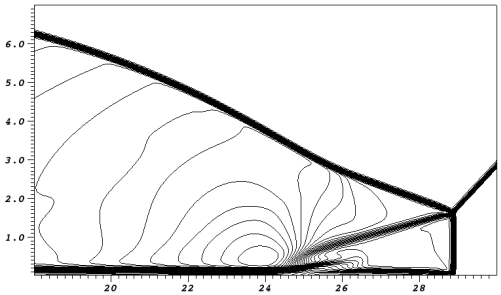
$\Delta x = 3.125\,\mathrm{mm}$

Shock reflection: solution for Navier-Stokes equations

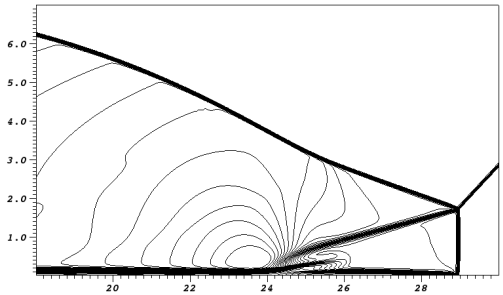
- No-slip boundary conditions enforced
- Conservative 2nd order centered differences to approximate stress tensor and heat flow

Shock reflection: solution for Navier-Stokes equations

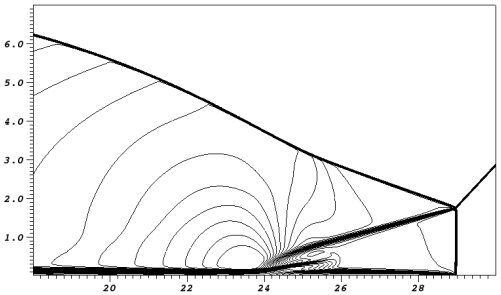
- No-slip boundary conditions enforced
- Conservative 2nd order centered differences to approximate stress tensor and heat flow



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



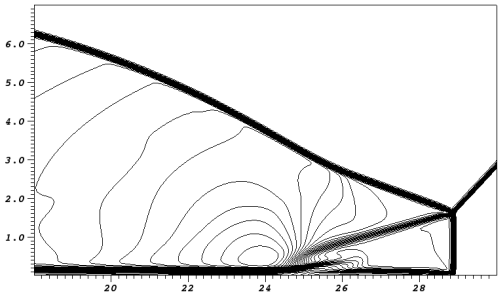
$\Delta x = 25\,\mathrm{mm}$



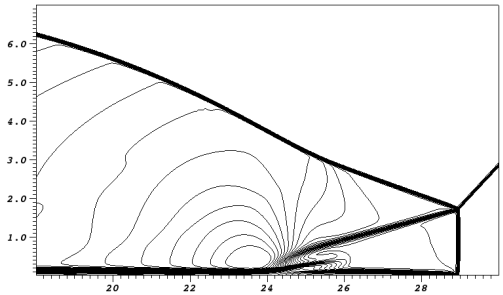
$\Delta x = 12.5\,\mathrm{mm}$, SAMR

Shock reflection: solution for Navier-Stokes equations

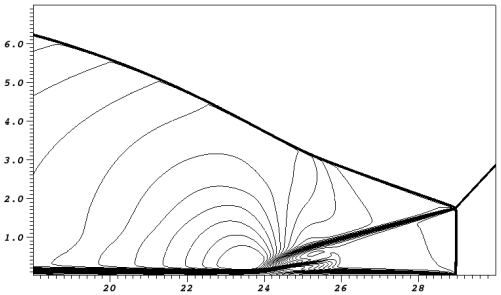
- No-slip boundary conditions enforced
- Conservative 2nd order centered differences to approximate stress tensor and heat flow



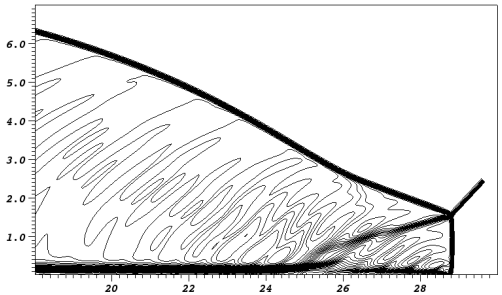
$\Delta x = 50\,\mathrm{mm}$



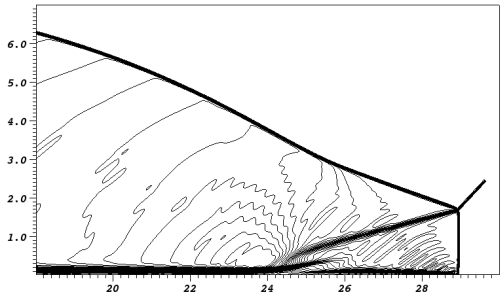
$\Delta x = 25\,\mathrm{mm}$



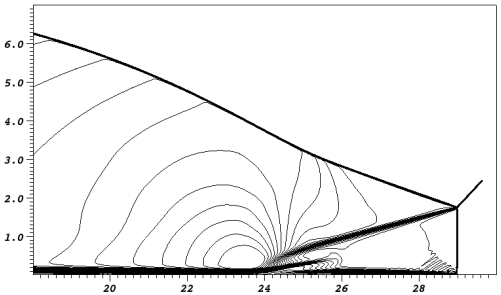
$\Delta x = 12.5\,\mathrm{mm}$, SAMR



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



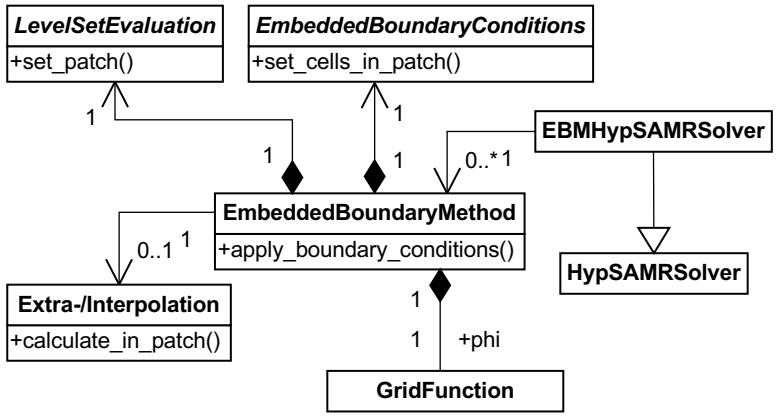
$\Delta x_e = 22.8\,\mathrm{mm}$



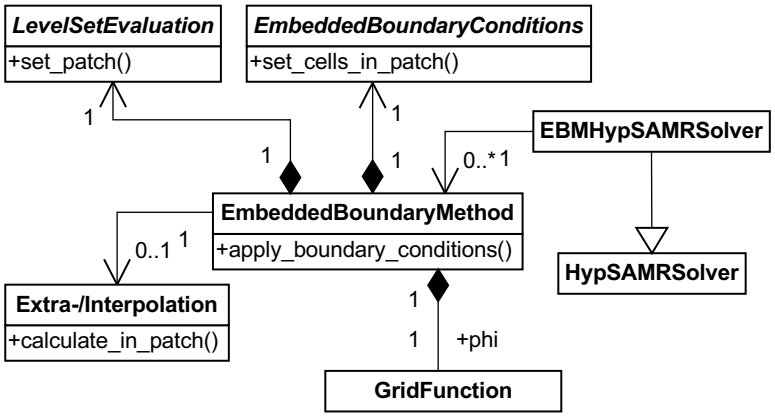
$\Delta x_e = 11.4\,\mathrm{mm}$, SAMR

Embedded boundary method

- Multiple independent EmbeddedBoundaryMethod objects possible
- Specialization of GFM boundary conditions



Embedded boundary method



- Multiple independent EmbeddedBoundaryMethod objects possible
- Specialization of GFM boundary conditions
- The generic embedded boundary method is implemented in **GhostFluidMethod<VectorType, dim >** and has a **GFMLevelSet<DataType, dim >** and **GFMBoundary<VectorType, dim >** object.

  `code/amroc/doc/html/amr/classGhostFluidMethod.html` `code/amroc/doc/html/amr/classGFMLevelSet.html`

  `code/amroc/doc/html/amr/classGFMBoundary_3_01VectorType_00_012_01_4.html`
- Multiple **GhostFluidMethod<VectorType, dim >** can be registered with **AMRGFMSolver<VectorType, FixupType, FlagType, dim >** and are called as part of the boundary condition setting routine.

  `code/amroc/doc/html/amr/classAMRGFMSolver.html`
- Interface classes to **GFMLevelSet<DataType, dim >** and **GFMBoundary<VectorType, dim >** are available amroc/amr/F77Interfaces and in amroc/amr/Interfaces

  `code/amroc/doc/html/amr/classF77GFMBoundary.html` `code/amroc/doc/html/amr/classSchemeGFMBoundary.html` make the approach available to all current solvers
- For instance `code/amroc/doc/html/clp/ClpStdGFMProblem_8h.html` or `code/amroc/doc/html/weno/WENOStdGFMProblem_8h.html` in Problem.h uses AMRGFMSolver<>

Outline

Governing equations for premixed combustion

Euler equations with reaction terms

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

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

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

Governing equations for premixed combustion

Euler equations with reaction terms

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

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

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

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

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

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.

Hyperbolicity of the homogeneous equations

Consider Jacobian $\mathbf{A}_1(\mathbf{q}) = \partial \mathbf{f}_1(\mathbf{q}) / \partial \mathbf{q}$

Hyperbolicity of the homogeneous equations

Consider Jacobian $\mathbf{A}_1(\mathbf{q}) = \partial \mathbf{f}_1(\mathbf{q})/\partial \mathbf{q}$

$$\mathbf{A}_1(\mathbf{q}) = \begin{bmatrix} u_1(1-Y_1) & -u_1 Y_1 & \dots & -u_1 Y_1 & Y_1 & 0 & 0 \\ -u_1 Y_2 & u_1(1-Y_2) & \dots & -u_1 Y_2 & & & \\ \vdots & & \ddots & \vdots & \vdots & \vdots & \vdots \\ -u_1 Y_{K-1} & \dots & u_1(1-Y_{K-1}) & -u_1 Y_{K-1} & & & \\ -u_1 Y_K & \dots & -u_1 Y_K & u_1(1-Y_K) & Y_K & 0 & 0 \\ \phi_1 - u_1^2 & & \dots & \phi_K - u_1^2 & (3-\gamma)u_1 & -\tilde{\gamma} u_2 & \tilde{\gamma} \\ -u_1 u_2 & & \dots & -u_1 u_2 & u_2 & u_1 & 0 \\ u_1(\phi_1 - H) & & \dots & u_1(\phi_K - H) & H - \tilde{\gamma} u_1^2 & -\tilde{\gamma} u_1 u_2 & \gamma u_1 \end{bmatrix}$$

with

$$\frac{\partial p}{\partial \rho_i} = \tilde{\gamma}\left(\frac{\mathbf{u}^2}{2} - h_i\right) + \gamma R_i T =: \phi_i \quad \text{und} \quad \frac{\partial p}{\partial \tilde{E}} = \gamma - 1 =: \tilde{\gamma}$$

Hyperbolicity of the homogeneous equations

Consider Jacobian $\mathbf{A}_1(\mathbf{q}) = \partial \mathbf{f}_1(\mathbf{q})/\partial \mathbf{q}$

$$\mathbf{A}_1(\mathbf{q}) = \begin{bmatrix} u_1(1-Y_1) & -u_1 Y_1 & & \dots & -u_1 Y_1 & Y_1 & 0 & 0 \\ -u_1 Y_2 & u_1(1-Y_2) & & \dots & -u_1 Y_2 & & & \\ \vdots & & \ddots & & \vdots & \vdots & \vdots & \vdots \\ -u_1 Y_{K-1} & \dots & & u_1(1-Y_{K-1}) & -u_1 Y_{K-1} & & & \\ -u_1 Y_K & \dots & & -u_1 Y_K & u_1(1-Y_K) & Y_K & 0 & 0 \\ \phi_1 - u_1^2 & & \dots & & \phi_K - u_1^2 & (3-\gamma)u_1 & -\tilde{\gamma} u_2 & \tilde{\gamma} \\ -u_1 u_2 & & \dots & & -u_1 u_2 & u_2 & u_1 & 0 \\ u_1(\phi_1 - H) & & \dots & & u_1(\phi_K - H) & H - \tilde{\gamma} u_1^2 & -\tilde{\gamma} u_1 u_2 & \gamma u_1 \end{bmatrix}$$

with

$$\frac{\partial p}{\partial \rho_i} = \tilde{\gamma}\left(\frac{\mathbf{u}^2}{2} - h_i\right) + \gamma R_i T =: \phi_i \quad \text{und} \quad \frac{\partial p}{\partial \tilde{E}} = \gamma - 1 =: \tilde{\gamma}$$

Eigenvalues of $\mathbf{A}_1(\mathbf{q})$ are $\{u_1 - a, u, \dots, u_1, u_1 + a\}$.
The system is hyperbolic, if the *frozen* speed of sound a is given by

$$a^2 = \left(\frac{\partial p}{\partial \rho}\right)_{s, Y_i} = \sum_{i=1}^{K} Y_i \phi_i - \tilde{\gamma}\mathbf{u}^2 + \tilde{\gamma} H = \gamma \frac{p}{\rho} > 0\,.$$

is real.

Hyperbolicity of the homogeneous equations

Consider Jacobian $\mathbf{A}_1(\mathbf{q}) = \partial \mathbf{f}_1(\mathbf{q})/\partial \mathbf{q}$

$$\mathbf{A}_1(\mathbf{q}) = \begin{bmatrix} u_1(1-Y_1) & -u_1 Y_1 & & \dots & -u_1 Y_1 & Y_1 & 0 & 0 \\ -u_1 Y_2 & u_1(1-Y_2) & & \dots & -u_1 Y_2 & & & \\ \vdots & & \ddots & & \vdots & \vdots & \vdots & \vdots \\ -u_1 Y_{K-1} & \dots & & u_1(1-Y_{K-1}) & -u_1 Y_{K-1} & & & \\ -u_1 Y_K & \dots & & -u_1 Y_K & u_1(1-Y_K) & Y_K & 0 & 0 \\ \phi_1 - u_1^2 & & \dots & & \phi_K - u_1^2 & (3-\gamma)u_1 & -\tilde{\gamma} u_2 & \tilde{\gamma} \\ -u_1 u_2 & & \dots & & -u_1 u_2 & u_2 & u_1 & 0 \\ u_1(\phi_1 - H) & & \dots & & u_1(\phi_K - H) & H - \tilde{\gamma} u_1^2 & -\tilde{\gamma} u_1 u_2 & \gamma u_1 \end{bmatrix}$$

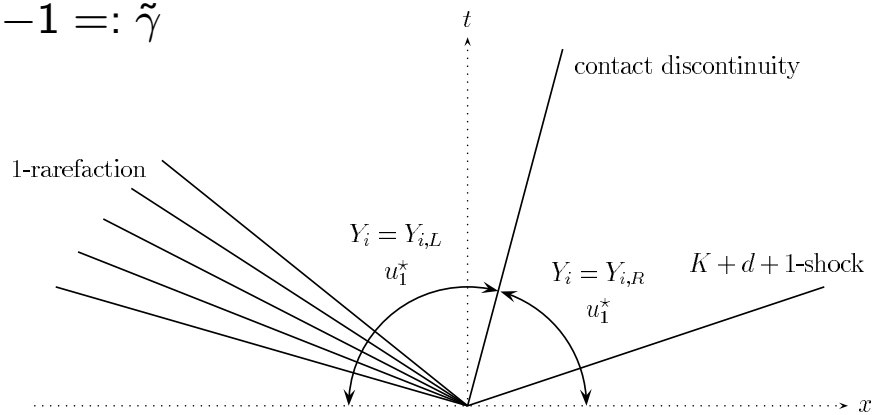
with

$$\frac{\partial p}{\partial \rho_i} = \tilde{\gamma}\left(\frac{\mathbf{u}^2}{2} - h_i\right) + \gamma R_i T =: \phi_i \quad \text{und} \quad \frac{\partial p}{\partial \tilde{E}} = \gamma - 1 =: \tilde{\gamma}$$

Eigenvalues of $\mathbf{A}_1(\mathbf{q})$ are $\{u_1 - a, u, \dots, u_1, u_1 + a\}$.
The system is hyperbolic, if the *frozen* speed of sound a is given by

$$a^2 = \left(\frac{\partial p}{\partial \rho}\right)_{s, Y_i} = \sum_{i=1}^{K} Y_i \phi_i - \tilde{\gamma}\mathbf{u}^2 + \tilde{\gamma} H = \gamma \frac{p}{\rho} > 0\,.$$

is real.



Hyperbolicity of the homogeneous equations

Consider Jacobian $\mathbf{A}_1(\mathbf{q}) = \partial \mathbf{f}_1(\mathbf{q})/\partial \mathbf{q}$

$$\mathbf{A}_1(\mathbf{q}) = \begin{bmatrix} u_1(1-Y_1) & -u_1 Y_1 & \dots & -u_1 Y_1 & Y_1 & 0 & 0 \\ -u_1 Y_2 & u_1(1-Y_2) & \dots & -u_1 Y_2 & & & \\ \vdots & & \ddots & \vdots & \vdots & \vdots & \vdots \\ -u_1 Y_{K-1} & \dots & u_1(1-Y_{K-1}) & -u_1 Y_{K-1} & & & \\ -u_1 Y_K & \dots & -u_1 Y_K & u_1(1-Y_K) & Y_K & 0 & 0 \\ \phi_1 - u_1^2 & & \dots & \phi_K - u_1^2 & (3-\gamma)u_1 & -\tilde{\gamma} u_2 & \tilde{\gamma} \\ -u_1 u_2 & & \dots & -u_1 u_2 & u_2 & u_1 & 0 \\ u_1(\phi_1 - H) & & \dots & u_1(\phi_K - H) & H - \tilde{\gamma} u_1^2 & -\tilde{\gamma} u_1 u_2 & \gamma u_1 \end{bmatrix}$$

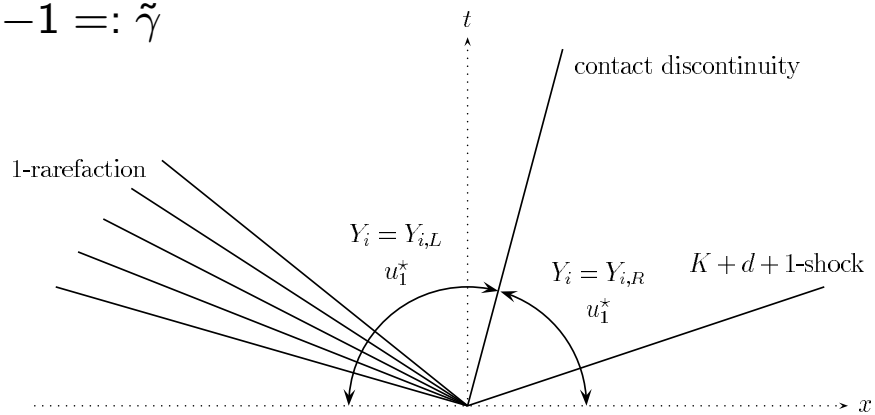
with

$$\frac{\partial p}{\partial \rho_i} = \tilde{\gamma}\left(\frac{\mathbf{u}^2}{2} - h_i\right) + \gamma R_i T =: \phi_i \quad \text{und} \quad \frac{\partial p}{\partial \tilde{E}} = \gamma - 1 =: \tilde{\gamma}$$

Eigenvalues of $\mathbf{A}_1(\mathbf{q})$ are $\{u_1 - a, u, \dots, u_1, u_1 + a\}$.
The system is hyperbolic, if the *frozen* speed of sound a is given by

$$a^2 = \left(\frac{\partial p}{\partial \rho}\right)_{s, Y_i} = \sum_{i=1}^{K} Y_i \phi_i - \tilde{\gamma}\mathbf{u}^2 + \tilde{\gamma} H = \gamma \frac{p}{\rho} > 0\,.$$

is real.



Empirical argument: $\partial_t Y_i + u_1\, \partial_x Y_i = 0$

Steger-Warming flux vector splitting

$$\mathbf{f}(\mathbf{q})^{+} = \mathbf{R}\mathbf{\Lambda}^{+}\mathbf{R}^{-1}\mathbf{q} \qquad \mathbf{f}(\mathbf{q})^{-} = \mathbf{R}\mathbf{\Lambda}^{-}\mathbf{R}^{-1}\mathbf{q}$$

Steger-Warming flux vector splitting

$$\mathbf{f}(\mathbf{q})^+ = \mathbf{R}\boldsymbol{\Lambda}^+\mathbf{R}^{-1}\mathbf{q} \qquad \mathbf{f}(\mathbf{q})^- = \mathbf{R}\boldsymbol{\Lambda}^-\mathbf{R}^{-1}\mathbf{q}$$

$$\mathbf{f}^{\pm}(\mathbf{q}) = \mathbf{A}^{\pm}(\mathbf{q})\,\mathbf{q} = \frac{\rho}{2\gamma}\begin{pmatrix} Y_1\tau^{\pm} \\ \vdots \\ Y_K\tau^{\pm} \\ u_1\tau^{\pm} - a\,\xi^{\pm} \\ u_2\tau^{\pm} \\ \vdots \\ u_d\tau^{\pm} \\ H\tau^{\pm} - u_1 a\,\xi^{\pm} - 2\,\delta^{\pm}a^2 \end{pmatrix} \quad \text{with} \quad \begin{aligned} \delta^{\pm} &= \lambda_2^{\pm} = \cdots = \lambda_{K+d}^{\pm}\,, \\ \tau^{\pm} &= \lambda_1^{\pm} + 2\,\delta^{\pm}(\gamma - 1) \\ &\quad + \lambda_{K+d+1}^{\pm}\,, \\ \xi^{\pm} &= \lambda_1^{\pm} - \lambda_{K+d+1}^{\pm}\,. \end{aligned}$$

Steger-Warming flux vector splitting

$$\mathbf{f}(\mathbf{q})^+ = \mathbf{R}\boldsymbol{\Lambda}^+\mathbf{R}^{-1}\mathbf{q} \qquad \mathbf{f}(\mathbf{q})^- = \mathbf{R}\boldsymbol{\Lambda}^-\mathbf{R}^{-1}\mathbf{q}$$

$$\mathbf{f}^{\pm}(\mathbf{q}) = \mathbf{A}^{\pm}(\mathbf{q})\,\mathbf{q} = \frac{\rho}{2\gamma}\begin{pmatrix} Y_1\tau^{\pm} \\ \vdots \\ Y_K\tau^{\pm} \\ u_1\tau^{\pm} - a\,\xi^{\pm} \\ u_2\tau^{\pm} \\ \vdots \\ u_d\tau^{\pm} \\ H\tau^{\pm} - u_1 a\,\xi^{\pm} - 2\,\delta^{\pm}a^2 \end{pmatrix} \quad \text{with} \quad \begin{aligned} \delta^{\pm} &= \lambda_2^{\pm} = \cdots = \lambda_{K+d}^{\pm}\,, \\ \tau^{\pm} &= \lambda_1^{\pm} + 2\,\delta^{\pm}(\gamma-1) \\ &\quad +\lambda_{K+d+1}^{\pm}\,, \\ \xi^{\pm} &= \lambda_1^{\pm} - \lambda_{K+d+1}^{\pm}\,. \end{aligned}$$

The corresponding stability condition is

$$C_{CFL}^{SW} := \max_{j\in\mathbb{Z}}(|u_{1,j}| + a_j) \leq 1$$

[Grossmann and Cinella, 1990, Larrouturou and Fezoui, 1989, Liu and Vinokur, 1989]

Steger-Warming flux vector splitting

$$\mathbf{f}(\mathbf{q})^+ = \mathbf{R}\mathbf{\Lambda}^+\mathbf{R}^{-1}\mathbf{q} \qquad \mathbf{f}(\mathbf{q})^- = \mathbf{R}\mathbf{\Lambda}^-\mathbf{R}^{-1}\mathbf{q}$$

$$\mathbf{f}^{\pm}(\mathbf{q}) = \mathbf{A}^{\pm}(\mathbf{q})\,\mathbf{q} = \frac{\rho}{2\gamma}\begin{pmatrix} Y_1\tau^{\pm} \\ \vdots \\ Y_K\tau^{\pm} \\ u_1\tau^{\pm} - a\,\xi^{\pm} \\ u_2\tau^{\pm} \\ \vdots \\ u_d\tau^{\pm} \\ H\tau^{\pm} - u_1 a\,\xi^{\pm} - 2\,\delta^{\pm}a^2 \end{pmatrix} \quad \text{with} \quad \begin{aligned} \delta^{\pm} &= \lambda_2^{\pm} = \cdots = \lambda_{K+d}^{\pm}\,, \\ \tau^{\pm} &= \lambda_1^{\pm} + 2\,\delta^{\pm}(\gamma-1) \\ &\quad +\lambda_{K+d+1}^{\pm}\,, \\ \xi^{\pm} &= \lambda_1^{\pm} - \lambda_{K+d+1}^{\pm}\,. \end{aligned}$$

The corresponding stability condition is

$$C_{CFL}^{SW} := \max_{j\in\mathbb{Z}}(|u_{1,j}| + a_j) \le 1$$

[Grossmann and Cinella, 1990, Larrouturou and Fezoui, 1989, Liu and Vinokur, 1989]

`code/amroc/doc/html/clp/rp1eurhokswg.f_fsource.html`

Van Leer flux vector splitting

$$\mathbf{f}^{\pm}(\mathbf{q}) = \pm\frac{\rho}{4a}(u_1 \pm a)^2 \begin{bmatrix} Y_1 \\ \vdots \\ Y_K \\ u_1 - (u_1 \mp 2a)/\gamma \\ u_2 \\ \vdots \\ u_d \\ H - \zeta(u_1 \mp a)^2 \end{bmatrix} \quad \text{with} \quad H = h + \frac{\mathbf{u}^2}{2}\,,$$

$$\zeta = \frac{h/a^2}{1 + 2h/a^2}\,.$$

Van Leer flux vector splitting

$$\mathbf{f}^{\pm}(\mathbf{q}) = \pm\frac{\rho}{4a}(u_1 \pm a)^2 \begin{bmatrix} Y_1 \\ \vdots \\ Y_K \\ u_1 - (u_1 \mp 2a)/\gamma \\ u_2 \\ \vdots \\ u_d \\ H - \zeta(u_1 \mp a)^2 \end{bmatrix} \quad \text{with} \quad H = h + \frac{\mathbf{u^2}}{2}\,, \quad \zeta = \frac{h/a^2}{1 + 2h/a^2}\,.$$

The splitting is explicitly constructed for $-a \le u_1 \le a$. Otherwise, we use

$$\mathbf{f}(\mathbf{q})^+ = \mathbf{f}(\mathbf{q}), \quad \mathbf{f}(\mathbf{q})^- = 0 \qquad \text{if } u_1 \ge a$$

$$\mathbf{f}(\mathbf{q})^- = \mathbf{f}(\mathbf{q}), \quad \mathbf{f}(\mathbf{q})^+ = 0 \qquad \text{if } u_1 \le -a$$

Van Leer flux vector splitting

$$\mathbf{f}^{\pm}(\mathbf{q}) = \pm\frac{\rho}{4a}(u_1 \pm a)^2 \begin{bmatrix} Y_1 \\ \vdots \\ Y_K \\ u_1 - (u_1 \mp 2a)/\gamma \\ u_2 \\ \vdots \\ u_d \\ H - \zeta(u_1 \mp a)^2 \end{bmatrix} \quad \text{with} \quad H = h + \frac{\mathbf{u}^2}{2}\ , \quad \zeta = \frac{h/a^2}{1 + 2h/a^2}\ .$$

The splitting is explicitly constructed for $-a \leq u_1 \leq a$. Otherwise, we use

$$\mathbf{f}(\mathbf{q})^+ = \mathbf{f}(\mathbf{q}), \quad \mathbf{f}(\mathbf{q})^- = 0 \qquad \text{if } u_1 \geq a$$
$$\mathbf{f}(\mathbf{q})^- = \mathbf{f}(\mathbf{q}), \quad \mathbf{f}(\mathbf{q})^+ = 0 \qquad \text{if } u_1 \leq -a$$

Stability condition

$$C_{CFL}^{VL} := \max_{j\in\mathbb{Z}} \left[(|u_{1,j}| + a_j)\,\Pi_j\right] \frac{\Delta t}{\Delta x} \leq 1$$

$$\text{with} \quad \Pi_j = \begin{cases} \dfrac{\gamma_j + 3}{2\gamma_j + u_{1,j}(3 - \gamma_j)/a_j} & \text{if } |u_{1,j}| < a_j\ , \\ 1 & \text{otherwise .} \end{cases}$$

[Shuen et al., 1990, Liu and Vinokur, 1989, Larrouturou and Fezoui, 1989, Grossmann and Cinella, 1990]

Van Leer flux vector splitting

$$\mathbf{f}^{\pm}(\mathbf{q}) = \pm\frac{\rho}{4a}(u_1 \pm a)^2 \begin{bmatrix} Y_1 \\ \vdots \\ Y_K \\ u_1 - (u_1 \mp 2a)/\gamma \\ u_2 \\ \vdots \\ u_d \\ H - \zeta(u_1 \mp a)^2 \end{bmatrix} \quad \text{with} \quad H = h + \frac{\mathbf{u}^2}{2}\ , \quad \zeta = \frac{h/a^2}{1 + 2h/a^2}\ .$$

The splitting is explicitly constructed for $-a \le u_1 \le a$. Otherwise, we use

$$\mathbf{f}(\mathbf{q})^+ = \mathbf{f}(\mathbf{q}), \quad \mathbf{f}(\mathbf{q})^- = 0 \qquad \text{if } u_1 \ge a$$

$$\mathbf{f}(\mathbf{q})^- = \mathbf{f}(\mathbf{q}), \quad \mathbf{f}(\mathbf{q})^+ = 0 \qquad \text{if } u_1 \le -a$$

Stability condition

$$C_{CFL}^{VL} := \max_{j\in\mathbb{Z}} \left[(|u_{1,j}| + a_j)\,\Pi_j\right] \frac{\Delta t}{\Delta x} \le 1$$

$$\text{with} \quad \Pi_j = \begin{cases} \dfrac{\gamma_j + 3}{2\gamma_j + u_{1,j}(3 - \gamma_j)/a_j} & \text{if } |u_{1,j}| < a_j\ , \\ 1 & \text{otherwise .} \end{cases}$$

[Shuen et al., 1990, Liu and Vinokur, 1989, Larrouturou and Fezoui, 1989, Grossmann and Cinella, 1990] `code/amroc/doc/html/clp/rp1eurhokvlg.f_fsource.html`

Roe solver

Roe averages $\hat{\rho}$, $\hat{u}$, $\hat{v}$, $\hat{H}$, $\hat{W}$, $\hat{T}$, $\hat{h}_i$, $\hat{e}_i$, $\hat{Y}_i$

Roe solver

Roe averages $\hat{\rho}$, $\hat{u}$, $\hat{v}$, $\hat{H}$, $\hat{W}$, $\hat{T}$, $\hat{h}_i$, $\hat{e}_i$, $\hat{Y}_i$

Define $\hat{c}_{pi} = \dfrac{1}{T_r - T_l} \displaystyle\int_{T_l}^{T_r} c_{pi}(\tau)\, d\tau$, $\hat{c}_{vi} = \dfrac{1}{T_r - T_l} \displaystyle\int_{T_l}^{T_r} c_{vi}(\tau)\, d\tau$

Roe solver

Roe averages $\hat{\rho}$, $\hat{u}$, $\hat{v}$, $\hat{H}$, $\hat{W}$, $\hat{T}$, $\hat{h}_i$, $\hat{e}_i$, $\hat{Y}_i$

Define $\hat{c}_{pi} = \dfrac{1}{T_r - T_l} \displaystyle\int_{T_l}^{T_r} c_{pi}(\tau)\, d\tau$, $\hat{c}_{vi} = \dfrac{1}{T_r - T_l} \displaystyle\int_{T_l}^{T_r} c_{vi}(\tau)\, d\tau$

$$\Delta \mathbf{f} := \mathbf{f}(\mathbf{q}_R) - \mathbf{f}(\mathbf{q}_L) = \sum_{m=1}^{M} a_m \lambda_m(\hat{\mathbf{q}}) \mathbf{r}_m(\hat{\mathbf{q}}) \quad \text{with} \quad \Delta \mathbf{q} := \mathbf{q}_R - \mathbf{q}_L = \sum_{m=1}^{M} a_m \mathbf{r}_m(\hat{\mathbf{q}}) \ .$$

Roe solver

Roe averages $\hat{\rho}$, $\hat{u}$, $\hat{v}$, $\hat{H}$, $\hat{W}$, $\hat{T}$, $\hat{h}_i$, $\hat{e}_i$, $\hat{Y}_i$

Define $\hat{c}_{pi} = \dfrac{1}{T_r - T_l} \displaystyle\int_{T_l}^{T_r} c_{pi}(\tau)\, d\tau$, $\hat{c}_{vi} = \dfrac{1}{T_r - T_l} \displaystyle\int_{T_l}^{T_r} c_{vi}(\tau)\, d\tau$

$$\Delta \mathbf{f} := \mathbf{f}(\mathbf{q}_R) - \mathbf{f}(\mathbf{q}_L) = \sum_{m=1}^{M} a_m \lambda_m(\hat{\mathbf{q}}) \mathbf{r}_m(\hat{\mathbf{q}}) \quad \text{with} \quad \Delta \mathbf{q} := \mathbf{q}_R - \mathbf{q}_L = \sum_{m=1}^{M} a_m \mathbf{r}_m(\hat{\mathbf{q}}) \ .$$

With matrix of right eigenvectors

$$\mathbf{R}_1(\mathbf{q}) = \begin{bmatrix} Y_1 & 1 & 0 & \dots & & 0 & 0 & 0 & Y_1 \\ & 0 & & & & & & & \\ \vdots & \vdots & & \ddots & & \vdots & \vdots & \vdots & \vdots \\ & & & & & 0 & & & \\ Y_K & 0 & & \dots & 0 & 1 & 0 & 0 & Y_K \\ u_1 - a & u_1 & & \dots & & u_1 & 0 & 0 & u_1 + a \\ u_2 & u_2 & & \dots & & u_2 & 1 & 0 & u_2 \\ u_3 & u_3 & & \dots & & u_3 & 0 & 1 & u_3 \\ H - u_1 a & \mathbf{u}^2 - \dfrac{\phi_1}{\bar{\gamma}} & & \dots & & \mathbf{u}^2 - \dfrac{\phi_K}{\bar{\gamma}} & u_2 & u_3 & H + u_1 a \end{bmatrix}$$

and evaluating $\mathbf{R}^{-1}(\hat{\mathbf{q}})\Delta\mathbf{q}$ one gets the characteristic wave strengths eventually as

$$a_1, a_{K+d+1} = \frac{\Delta p \mp \hat{\rho}\hat{a}\Delta u_1}{2\hat{a}^2} \ , \quad a_{1+i} = \Delta\rho_i - \hat{Y}_i \frac{\Delta p}{\hat{a}^2} \ , \quad a_{K+n} = \hat{\rho}\Delta u_n \ .$$

Roe solver

Roe averages $\hat{\rho}$, $\hat{u}$, $\hat{v}$, $\hat{H}$, $\hat{W}$, $\hat{T}$, $\hat{h}_i$, $\hat{e}_i$, $\hat{Y}_i$

Define $\hat{c}_{pi} = \dfrac{1}{T_r - T_l} \displaystyle\int_{T_l}^{T_r} c_{pi}(\tau)\, d\tau$, $\hat{c}_{vi} = \dfrac{1}{T_r - T_l} \displaystyle\int_{T_l}^{T_r} c_{vi}(\tau)\, d\tau$

$$\Delta \mathbf{f} := \mathbf{f}(\mathbf{q}_R) - \mathbf{f}(\mathbf{q}_L) = \sum_{m=1}^{M} a_m \lambda_m(\hat{\mathbf{q}}) \mathbf{r}_m(\hat{\mathbf{q}}) \quad \text{with} \quad \Delta \mathbf{q} := \mathbf{q}_R - \mathbf{q}_L = \sum_{m=1}^{M} a_m \mathbf{r}_m(\hat{\mathbf{q}}) \ .$$

With matrix of right eigenvectors

$$\mathbf{R}_1(\mathbf{q}) = \begin{bmatrix} Y_1 & 1 & 0 & \dots & & 0 & 0 & 0 & Y_1 \\ & 0 & & & & & & & \\ \vdots & \vdots & & \ddots & & \vdots & \vdots & \vdots & \vdots \\ & & & & & 0 & & & \\ Y_K & 0 & & \dots & 0 & 1 & 0 & 0 & Y_K \\ u_1 - a & u_1 & & \dots & & u_1 & 0 & 0 & u_1 + a \\ u_2 & u_2 & & \dots & & u_2 & 1 & 0 & u_2 \\ u_3 & u_3 & & \dots & & u_3 & 0 & 1 & u_3 \\ H - u_1 a & \mathbf{u}^2 - \dfrac{\phi_1}{\bar{\gamma}} & & \dots & & \mathbf{u}^2 - \dfrac{\phi_K}{\bar{\gamma}} & u_2 & u_3 & H + u_1 a \end{bmatrix}$$

and evaluating $\mathbf{R}^{-1}(\hat{\mathbf{q}})\Delta\mathbf{q}$ one gets the characteristic wave strengths eventually as

$$a_1, a_{K+d+1} = \frac{\Delta p \mp \hat{\rho}\hat{a}\Delta u_1}{2\hat{a}^2} \ , \quad a_{1+i} = \Delta\rho_i - \hat{Y}_i \frac{\Delta p}{\hat{a}^2} \ , \quad a_{K+n} = \hat{\rho}\Delta u_n \ .$$

`code/amroc/doc/html/clp/rp1eurhok.f_fsource.html`

Roe solver - fixes

Mass fraction positivity: Calculate numerical fluxes of partial densities by [Larrouturou, 1991]

$$\mathbf{F}_i^\star(\mathbf{q}_L, \mathbf{q}_R) = \mathbf{F}_\rho(\mathbf{q}_L, \mathbf{q}_R) \cdot \begin{cases} Y_{i,L}\,, & \mathbf{F}_\rho(\mathbf{q}_L, \mathbf{q}_R) > 0\,, \\ Y_{i,R}\,, & \mathbf{F}_\rho(\mathbf{q}_L, \mathbf{q}_R) < 0\,. \end{cases}$$

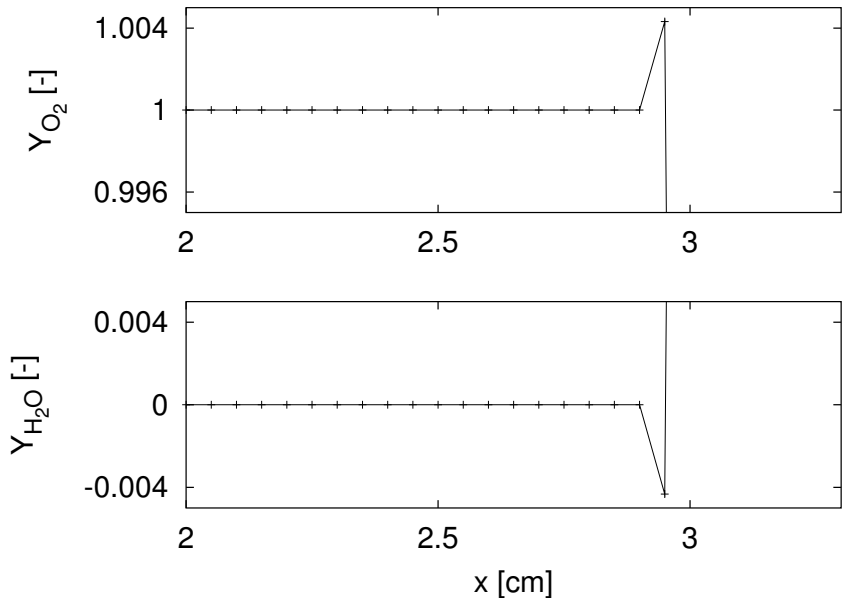
to ensure positivity of Y_i.

Roe solver - fixes

Mass fraction positivity: Calculate numerical fluxes of partial densities by [Larrouturou, 1991]

$$\mathbf{F}_i^\star(\mathbf{q}_L, \mathbf{q}_R) = \mathbf{F}_\rho(\mathbf{q}_L, \mathbf{q}_R) \cdot \begin{cases} Y_{i,L}\,, & \mathbf{F}_\rho(\mathbf{q}_L, \mathbf{q}_R) > 0\,, \\ Y_{i,R}\,, & \mathbf{F}_\rho(\mathbf{q}_L, \mathbf{q}_R) < 0\,. \end{cases}$$

to ensure positivity of Y_i.
Example: Mass fraction for a typical Riemann problem after 1 time step with the Roe solver

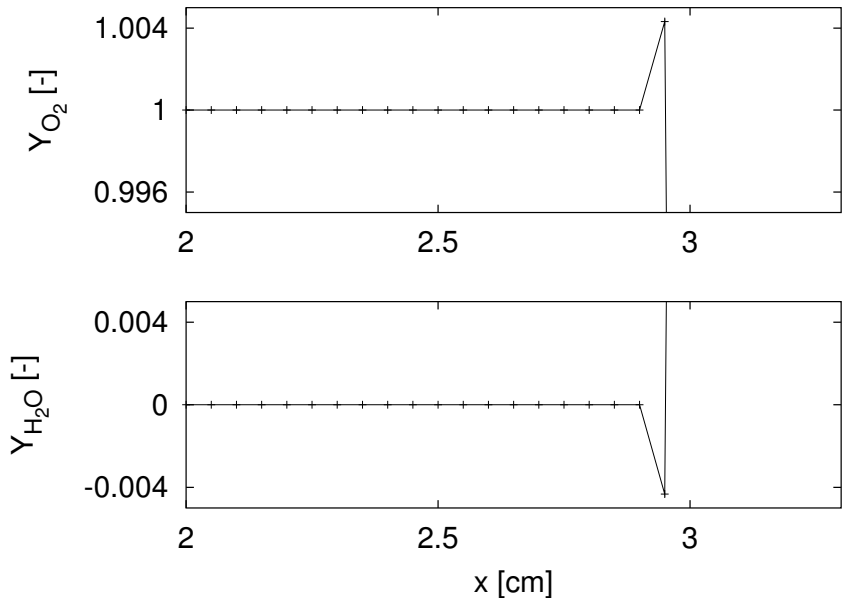


Roe solver - fixes

Mass fraction positivity: Calculate numerical fluxes of partial densities by [Larrouturou, 1991]

$$\mathbf{F}_i^\star(\mathbf{q}_L, \mathbf{q}_R) = \mathbf{F}_\rho(\mathbf{q}_L, \mathbf{q}_R) \cdot \begin{cases} Y_{i,L}\,, & \mathbf{F}_\rho(\mathbf{q}_L, \mathbf{q}_R) > 0\,, \\ Y_{i,R}\,, & \mathbf{F}_\rho(\mathbf{q}_L, \mathbf{q}_R) < 0\,. \end{cases}$$

to ensure positivity of Y_i.
Example: Mass fraction for a typical Riemann problem after 1 time step with the Roe solver



Further: Switch from Roe to HLL scheme near vacuum state to avoid unphysical values.

Roe solver - entropy and carbuncle fix

Entropy corrections

Roe solver - entropy and carbuncle fix

Entropy corrections [Harten, 1983]

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

$\eta = \frac{1}{2} \max_\iota \left\{ |s_\iota(\mathbf{q}_R) - s_\iota(\mathbf{q}_L)| \right\}$

Roe solver - entropy and carbuncle fix

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

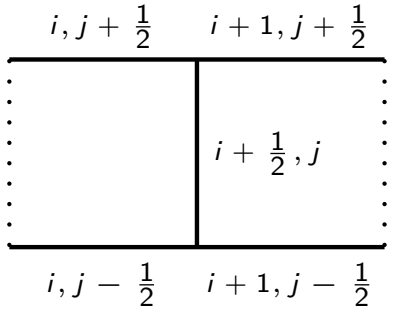
1. $|\tilde{s}_\iota| = \begin{cases} |s_\iota| & \text{if} |s_\iota| \geq 2\eta \\ \frac{|s_\iota^2|}{4\eta} + \eta & \text{otherwise} \end{cases}$
   $\eta = \frac{1}{2} \max_\iota \left\{ |s_\iota(\mathbf{q}_R) - s_\iota(\mathbf{q}_L)| \right\}$
2. Replace $|s_\iota|$ by $|\tilde{s}_\iota|$ only if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

Roe solver - entropy and 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 (here named EF 3) [Sanders et al., 1998]:



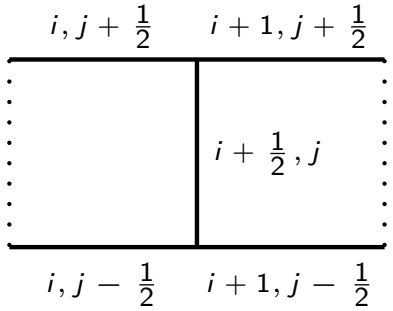
$$\tilde{\eta}_{i+1/2,j} = \max \left\{ \eta_{i+1/2,j}, \eta_{i,j-1/2}, \ \eta_{i,j+1/2}, \eta_{i+1,j-1/2}, \eta_{i+1,j+1/2} \right\}$$

Roe solver - entropy and 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 (here named EF 3) [Sanders et al., 1998]:



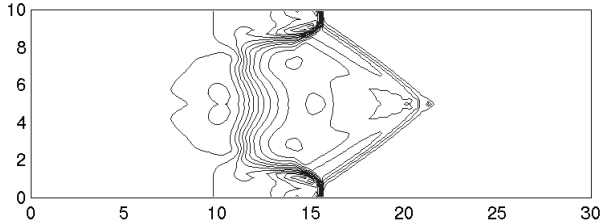
$$\tilde{\eta}_{i+1/2,j} = \max \left\{ \eta_{i+1/2,j}, \eta_{i,j-1/2}, \ \eta_{i,j+1/2}, \eta_{i+1,j-1/2}, \eta_{i+1,j+1/2} \right\}$$

Carbuncle phenomenon

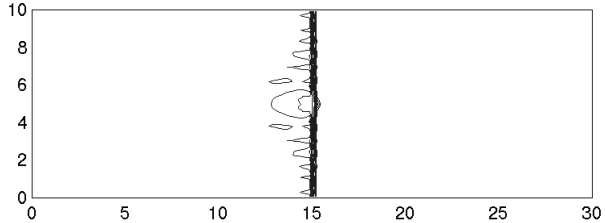
- [Quirk, 1994b]
- Test from [Deiterding, 2003]

`code/amroc/doc/html/apps/clawpack_2applications_2euler_znd_22d_2Carbuncle_2src_2Problem_8h_source.html`

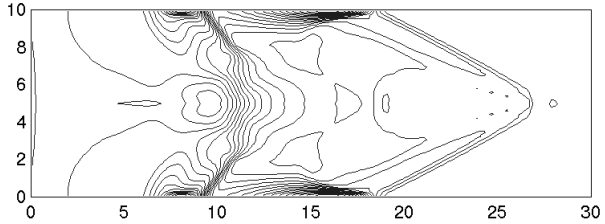
Roe + EF 1.



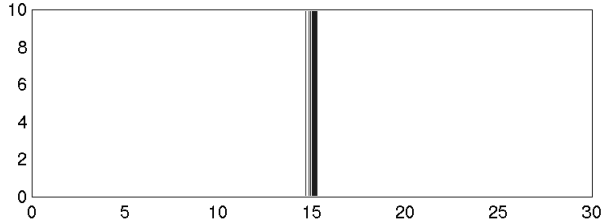
Exact Riemann solver



Roe + EF 2.



SW FVS, VL FVS, HLL, Roe + EF 3



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

(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 $s_1 = \hat{u}_1 - \hat{a}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{a}$.

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

(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 $s_1 = \hat{u}_1 - \hat{a}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{a}$.

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

(S9) If $\rho^\star_{L/R} \leq 0$ or $e^\star_{L/R} \leq 0$ 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 $\hat{a} := \left(\sum_{i=1}^{K} \hat{Y}_i\, \hat{\phi}_i - (\hat{\gamma}-1)\hat{\mathbf{u}}^2 + (\hat{\gamma}-1)\hat{H}\right)^{1/2}$.

(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 $s_1 = \hat{u}_1 - \hat{a}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{a}$.

(S8) Evaluate $\rho^\star_{L/R}$, $u^\star_{1,L/R}$, $e^\star_{L/R}$, $a^\star_{L/R}$ 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 $\hat{a} := \left(\sum_{i=1}^{K} \hat{Y}_i\, \hat{\phi}_i - (\hat{\gamma} - 1)\hat{\mathbf{u}}^2 + (\hat{\gamma} - 1)\hat{H}\right)^{1/2}$.

(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 $s_1 = \hat{u}_1 - \hat{a}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{a}$.

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

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

(S10) Entropy correction: Evaluate $|\tilde{s}_\iota|$.

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

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

$$\mathbf{F}^\star_i = \mathbf{F}_\rho \cdot \begin{cases} Y^l_i\,, & \mathbf{F}_\rho \geq 0\,, \\ Y^r_i\,, & \mathbf{F}_\rho < 0\,. \end{cases}$$

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

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

(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 $s_1 = \hat{u}_1 - \hat{a}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{a}$.

(S8) Evaluate $\rho^\star_{L/R}$, $u^\star_{1,L/R}$, $e^\star_{L/R}$, $a^\star_{L/R}$ 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 $|\tilde{s}_\iota|$.

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

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

$$\mathbf{F}^\star_i = \mathbf{F}_\rho \cdot \begin{cases} Y^l_i\,, & \mathbf{F}_\rho \ge 0\,, \\ Y^r_i\,, & \mathbf{F}_\rho < 0\,. \end{cases}$$

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

`code/amroc/doc/html/clp/rp1eurhokefix.f_fsource.html`
`code/amroc/doc/html/clp/rp1eurhokefixg.f_fsource.html`

ZND Detonation Model with Simplified Chemistry

Assume a stationary 1D detonation with irreversible reaction

$$A \longrightarrow B$$

with energy release $q_0 = -\Delta h^0$ and $k^f(T) = K \exp(-E_A/\mathcal{R}T)$.

ZND Detonation Model with Simplified Chemistry

Assume a stationary 1D detonation with irreversible reaction

$$A \longrightarrow B$$

with energy release $q_0 = -\Delta h^0$ and $k^f(T) = K \exp(-E_A/\mathcal{R}T)$.

Simplified Kinetics $\dot{W}_A \dot{\omega}_A = -K \rho_A \exp(-E_A/\mathcal{R}T)\,, \quad W_B \dot{\omega}_B = -W_A \dot{\omega}_A$

ZND Detonation Model with Simplified Chemistry

Assume a stationary 1D detonation with irreversible reaction

$$A \longrightarrow B$$

with energy release $q_0 = -\Delta h^0$ and $k^f(T) = K \exp(-E_A/\mathcal{R}T)$.

Simplified Kinetics $\dot{W}_A \dot{\omega}_A = -K \rho_A \exp(-E_A/\mathcal{R}T)\,, \quad W_B \dot{\omega}_B = -W_A \dot{\omega}_A$

With $\gamma_A = \gamma_B$ the equation of state is $p(\rho_A, \rho_B, e) = (\gamma - 1)\,(\rho e - \rho_A\, q_0)$

ZND Detonation Model with Simplified Chemistry

Assume a stationary 1D detonation with irreversible reaction

$$A \longrightarrow B$$

with energy release $q_0 = -\Delta h^0$ and $k^f(T) = K \exp(-E_A/\mathcal{R}T)$.

Simplified Kinetics $\dot{W}_A \dot{\omega}_A = -K \rho_A \exp(-E_A/\mathcal{R}T)\,, \quad W_B \dot{\omega}_B = -W_A \dot{\omega}_A$

With $\gamma_A = \gamma_B$ the equation of state is $p(\rho_A, \rho_B, e) = (\gamma - 1)\,(\rho e - \rho_A\, q_0)$

Integration of the stationary 1D Euler equations

$$\partial_{x'}(\rho u') = 0, \qquad \partial_{x'}(\rho u'^2 + p) = 0, \qquad \partial_{x'}\left[u'(\rho E + p)\right] = 0$$

$$\partial_{x'}\lambda = \frac{K\rho(1-\lambda)\exp(-E_A^\star\ \rho/p)}{\rho u'}$$

ZND Detonation Model with Simplified Chemistry

Assume a stationary 1D detonation with irreversible reaction

$$A \longrightarrow B$$

with energy release $q_0 = -\Delta h^0$ and $k^f(T) = K \exp(-E_A/\mathcal{R}T)$.

Simplified Kinetics $\dot{W}_A \dot{\omega}_A = -K \rho_A \exp(-E_A/\mathcal{R}T)\,, \quad W_B \dot{\omega}_B = -W_A \dot{\omega}_A$

With $\gamma_A = \gamma_B$ the equation of state is $p(\rho_A, \rho_B, e) = (\gamma - 1)\,(\rho e - \rho_A\, q_0)$

Integration of the stationary 1D Euler equations

$$\partial_{x'}(\rho u') = 0, \qquad \partial_{x'}(\rho u'^2 + p) = 0, \qquad \partial_{x'}\left[u'(\rho E + p)\right] = 0$$

$$\partial_{x'}\lambda = \frac{K\rho(1-\lambda)\exp(-E_A^\star\ \rho/p)}{\rho u'}$$

gives for λ, the mass fraction of the product B,

$$\frac{d\lambda}{dx'} = K(1-\lambda)\exp\left(\frac{-E_A^\star\ \rho(\lambda)/p(\lambda)}{u'(\lambda) + D}\right) = f(\lambda)\,,$$

i.e. $$\int_0^{x'} f(\lambda(\xi))\,d\xi = \lambda(x') \quad .$$

ZND Detonation Model with Simplified Chemistry

Assume a stationary 1D detonation with irreversible reaction

$$A \longrightarrow B$$

with energy release $q_0 = -\Delta h^0$ and $k^f(T) = K \exp(-E_A/\mathcal{R}T)$.
Simplified Kinetics $\dot{W}_A \dot{\omega}_A = -K \rho_A \exp(-E_A/\mathcal{R}T)\,, \quad W_B \dot{\omega}_B = -W_A \dot{\omega}_A$
With $\gamma_A = \gamma_B$ the equation of state is $p(\rho_A, \rho_B, e) = (\gamma - 1)\,(\rho e - \rho_A\, q_0)$
Integration of the stationary 1D Euler equations

$$\partial_{x'}(\rho u') = 0, \qquad \partial_{x'}(\rho u'^2 + p) = 0, \qquad \partial_{x'}\left[u'(\rho E + p)\right] = 0$$

$$\partial_{x'}\lambda = \frac{K\rho(1-\lambda)\exp(-E_A^\star\ \rho/p)}{\rho u'}$$

gives for λ, the mass fraction of the product B,

$$\frac{d\lambda}{dx'} = K(1-\lambda)\exp\left(\frac{-E_A^\star\ \rho(\lambda)/p(\lambda)}{u'(\lambda) + D}\right) = f(\lambda)\,,$$

i.e. $$\int_0^{x'} f(\lambda(\xi))\,d\xi = \lambda(x') \quad .$$

D_{CJ} is the minimal detonation velocity. The overdrive-parameter $f = (D/D_{CJ})^2 \geq 1$ determines stability.
Normalization:

$$L_{1/2} = \int_0^{\frac{1}{2}} \frac{d\lambda}{f(\lambda)}$$

ZND Detonation Model with Simplified Chemistry

Assume a stationary 1D detonation with irreversible reaction

$$A \longrightarrow B$$

with energy release $q_0 = -\Delta h^0$ and $k^f(T) = K \exp(-E_A/\mathcal{R}T)$.

Simplified Kinetics $\dot{W}_A \dot{\omega}_A = -K \rho_A \exp(-E_A/\mathcal{R}T)\,, \quad W_B \dot{\omega}_B = -W_A \dot{\omega}_A$

With $\gamma_A = \gamma_B$ the equation of state is $p(\rho_A, \rho_B, e) = (\gamma - 1)\,(\rho e - \rho_A\, q_0)$

Integration of the stationary 1D Euler equations

$$\partial_{x'}(\rho u') = 0, \qquad \partial_{x'}(\rho u'^2 + p) = 0, \qquad \partial_{x'}\left[u'(\rho E + p)\right] = 0$$

$$\partial_{x'}\lambda = \frac{K\rho(1-\lambda)\exp(-E_A^\star\ \rho/p)}{\rho u'}$$

gives for λ, the mass fraction of the product B,

$$\frac{d\lambda}{dx'} = K(1-\lambda)\exp\left(\frac{-E_A^\star\ \rho(\lambda)/p(\lambda)}{u'(\lambda) + D}\right) = f(\lambda)\,,$$

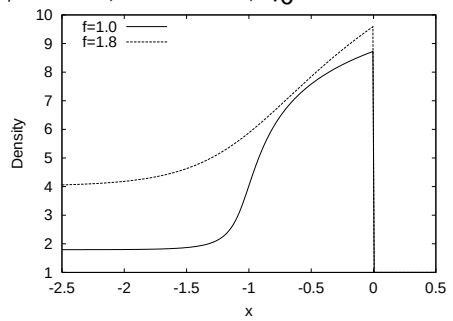
i.e. $$\int_0^{x'} f(\lambda(\xi))\,d\xi = \lambda(x') \quad .$$

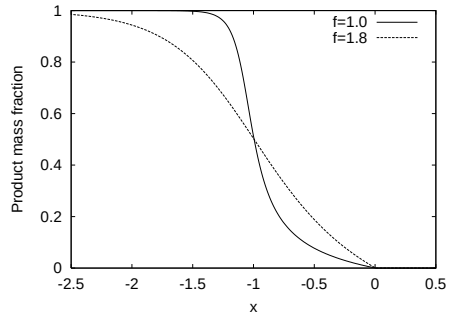
D_{CJ} is the minimal detonation velocity. The overdrive-parameter $f = (D/D_{CJ})^2 \geq 1$ determines stability.

Normalization:

$$L_{1/2} = \int_0^{\frac{1}{2}} \frac{d\lambda}{f(\lambda)}$$

$\gamma = 1.2$, $E^+ = 50$, $q_0 = 50$

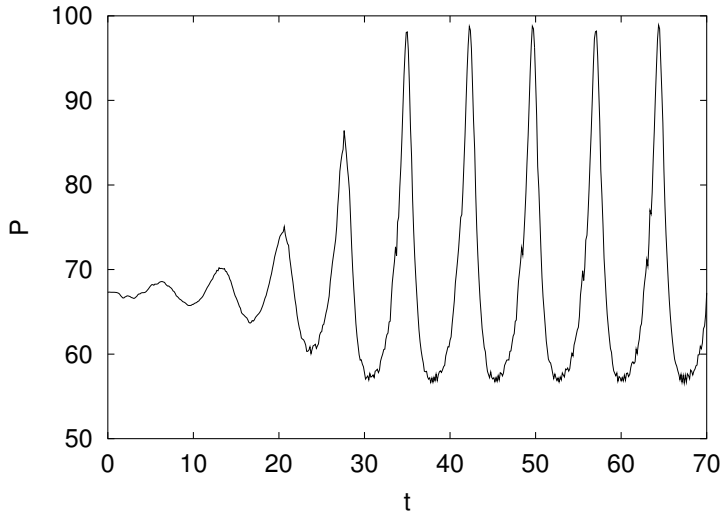




Unstable ZND Detonation

$\gamma = 1.2,\ E^+ = 50,\ q_0 = 50,\ f = 1.6,\ CFL = 0.9$

quasi-stationary

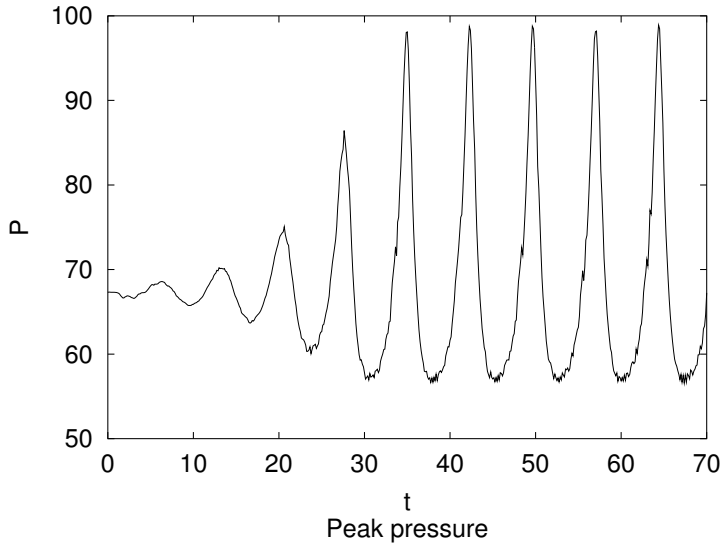


moving

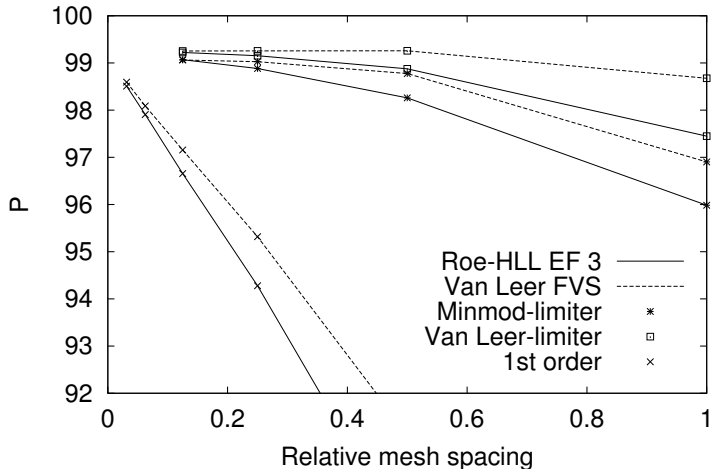
Unstable ZND Detonation

$\gamma = 1.2,\ E^+ = 50,\ q_0 = 50,\ f = 1.6,\ CFL = 0.9$

quasi-stationary moving



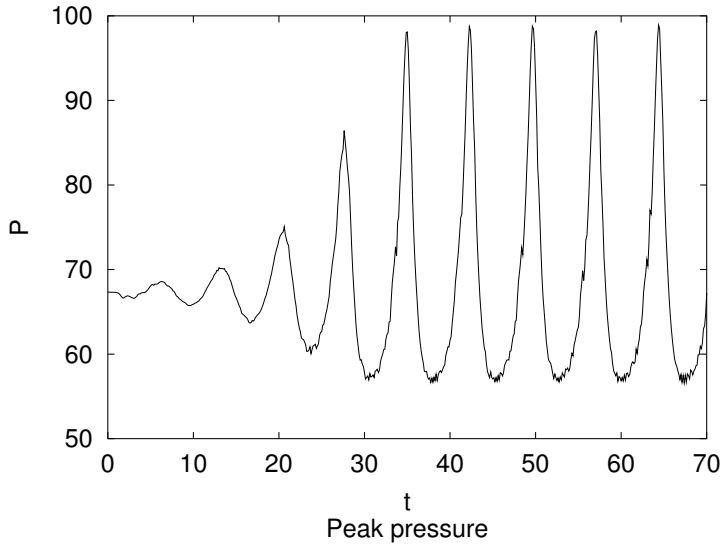
Peak pressure



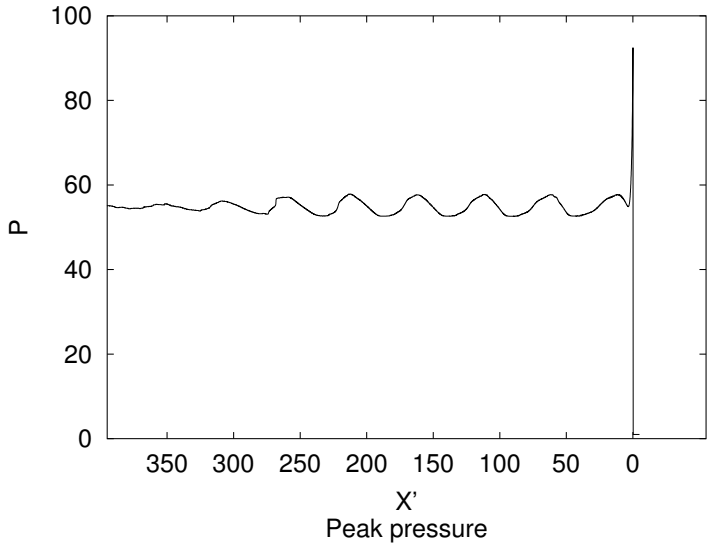
Unstable ZND Detonation

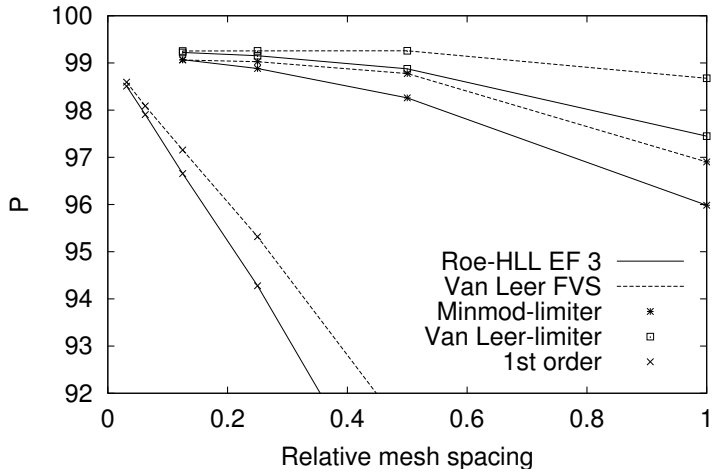
$\gamma = 1.2,\ E^+ = 50,\ q_0 = 50,\ f = 1.6,\ CFL = 0.9$

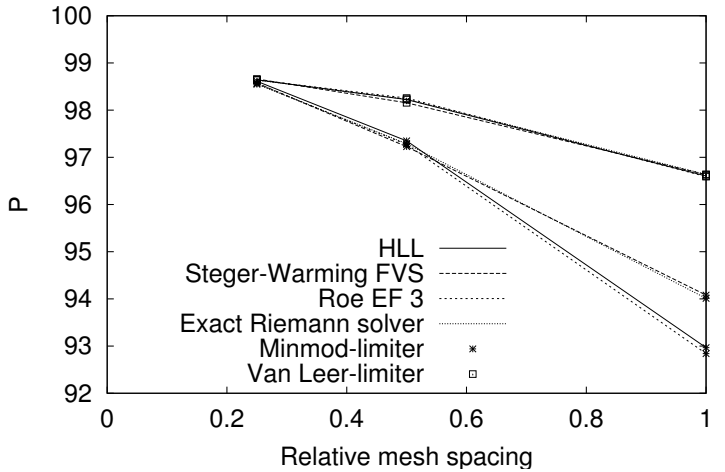
quasi-stationary



moving



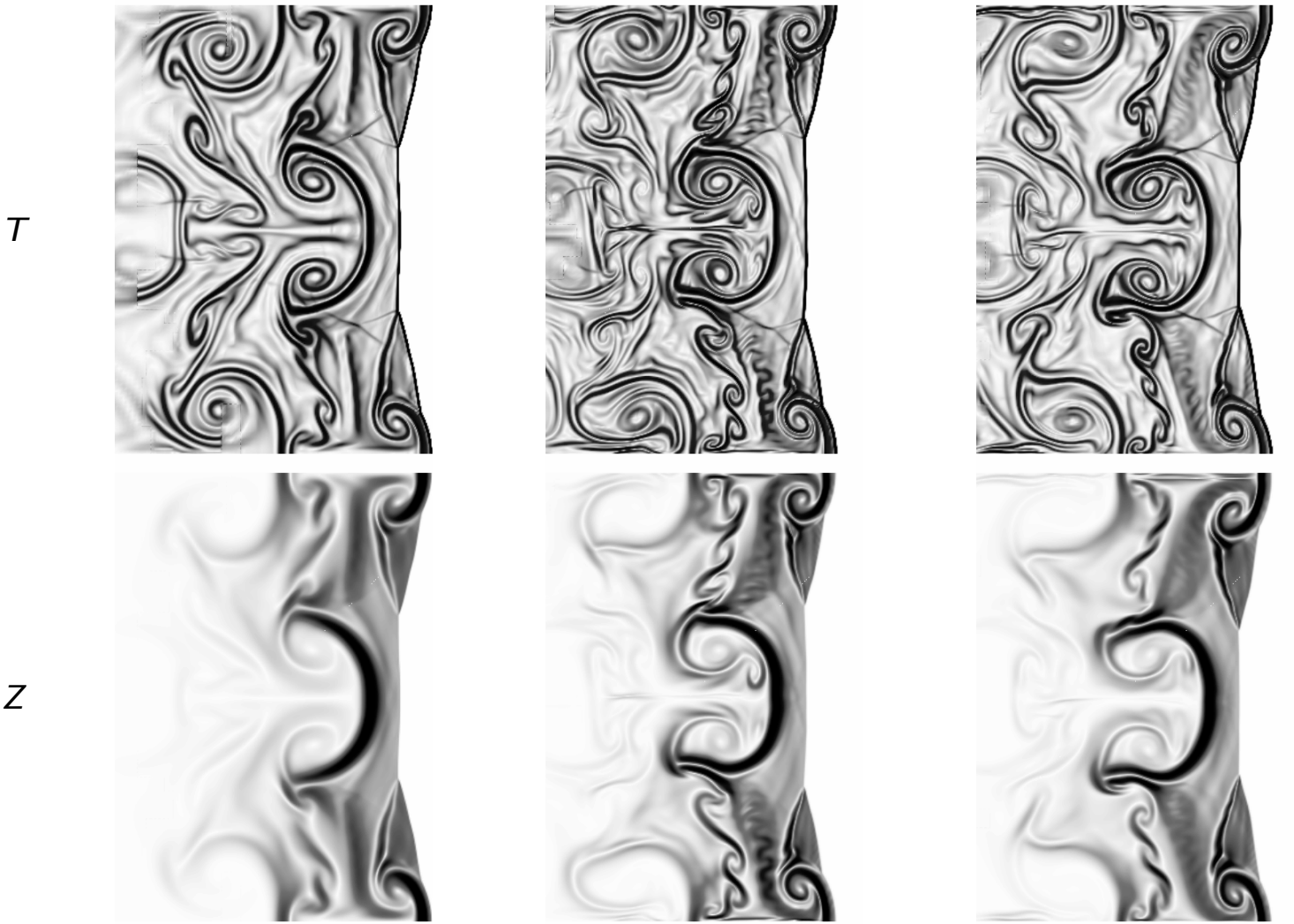




code/amroc/doc/html/apps/clawpack_2applications_2euler__znd_21d_2StatDet_2src_2Problem_8h_source.html

Comparison of FV Schemes: MUSCL, Van Albada-limiter

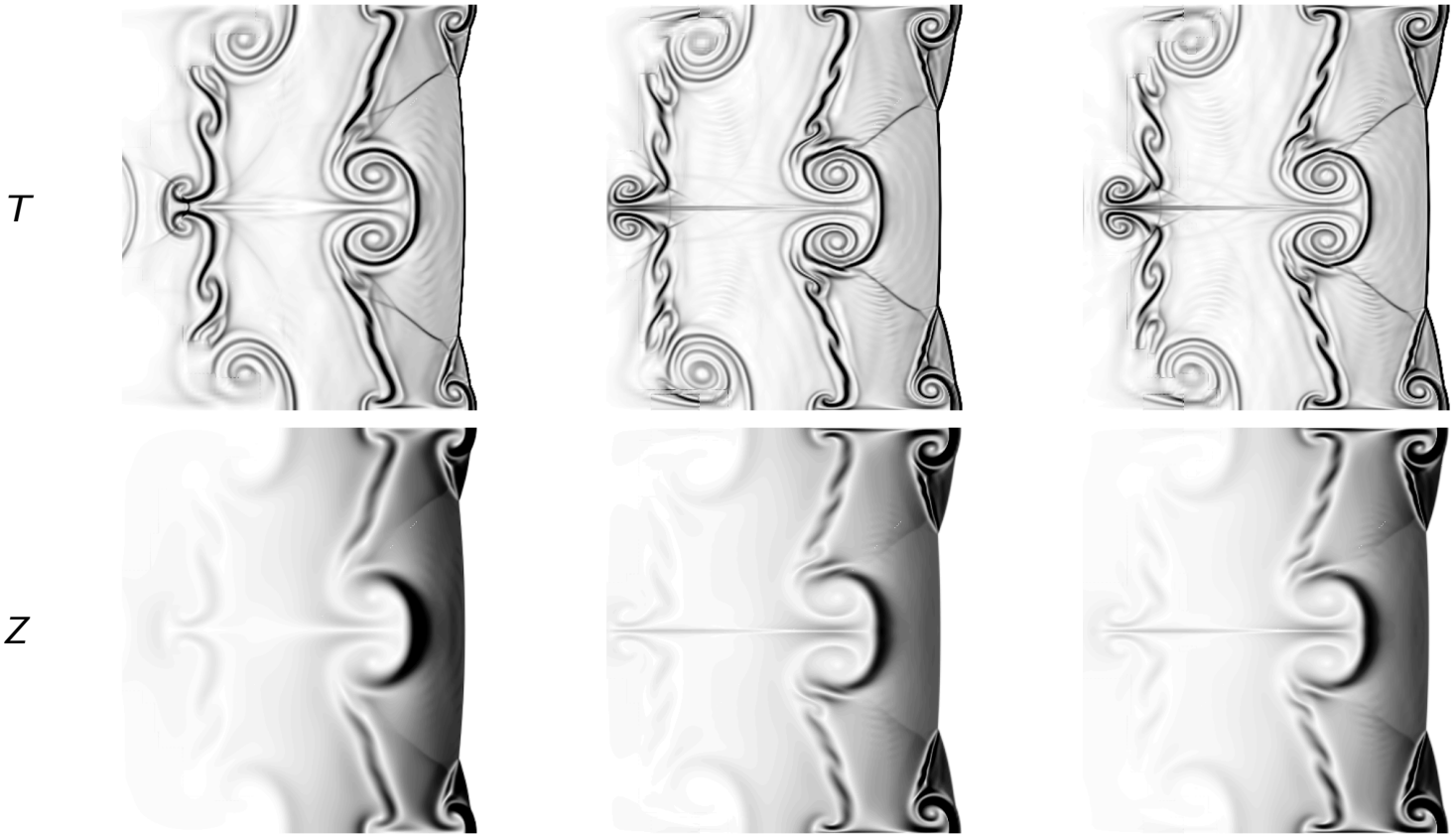
$\gamma = 1.2$, $E = 50$, $q_0 = 50$, $f = 3.0$, $40\,\mathrm{Pts}/L_{1/2}$



Van Leer FVS, DimSplit | Roe-HLL EF 3, DimSplit | Roe-HLL EF 3, WaveProp

Comparison of FV Schemes - II

$\gamma = 1.2,\ E = 10,\ q_0 = 50,\ f = 1.2,\ 40\,\mathrm{Pts}/L_{1/2}$



Van Leer FVS, DimSplit | Roe-HLL EF 3*-H, DimSplit | Roe-HLL EF 3*-H, WaveProp

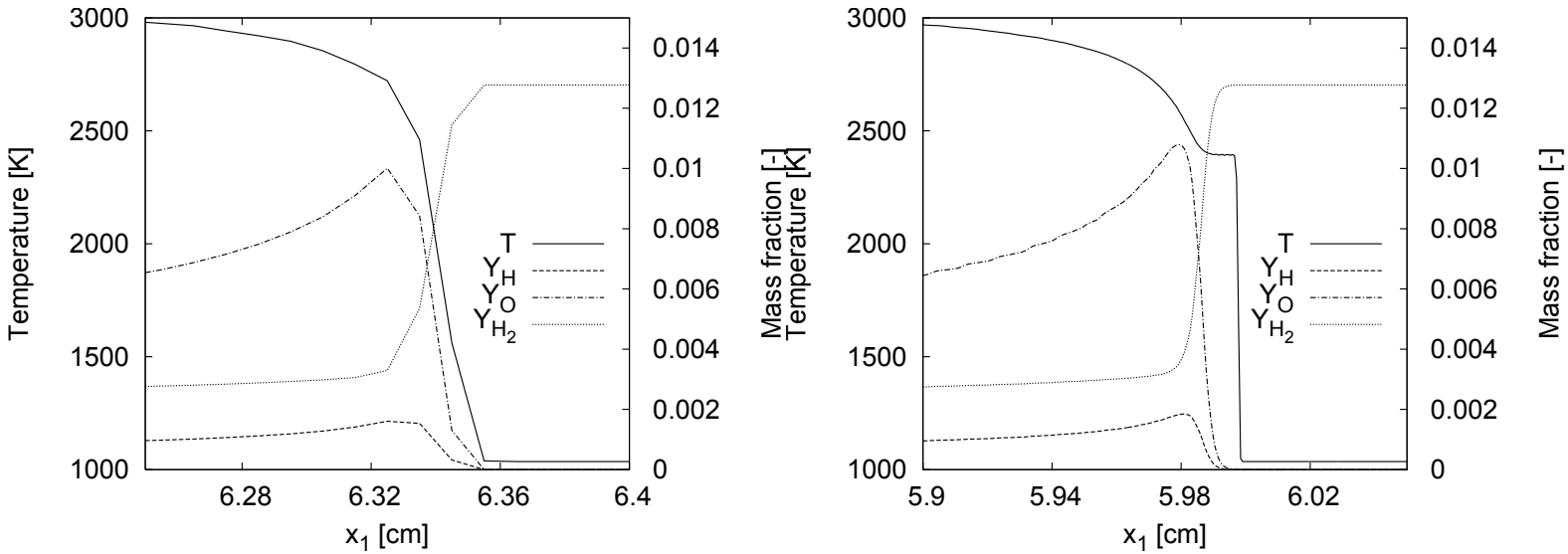
code/amroc/doc/html/apps/clawpack_2applications_2euler__znd_22d_2StatDet_2src_2Problem_8h_source.html

Detonations - motivation for SAMR

- Extremly high spatial resolution in reaction zone necessary.

Detonations - motivation for SAMR

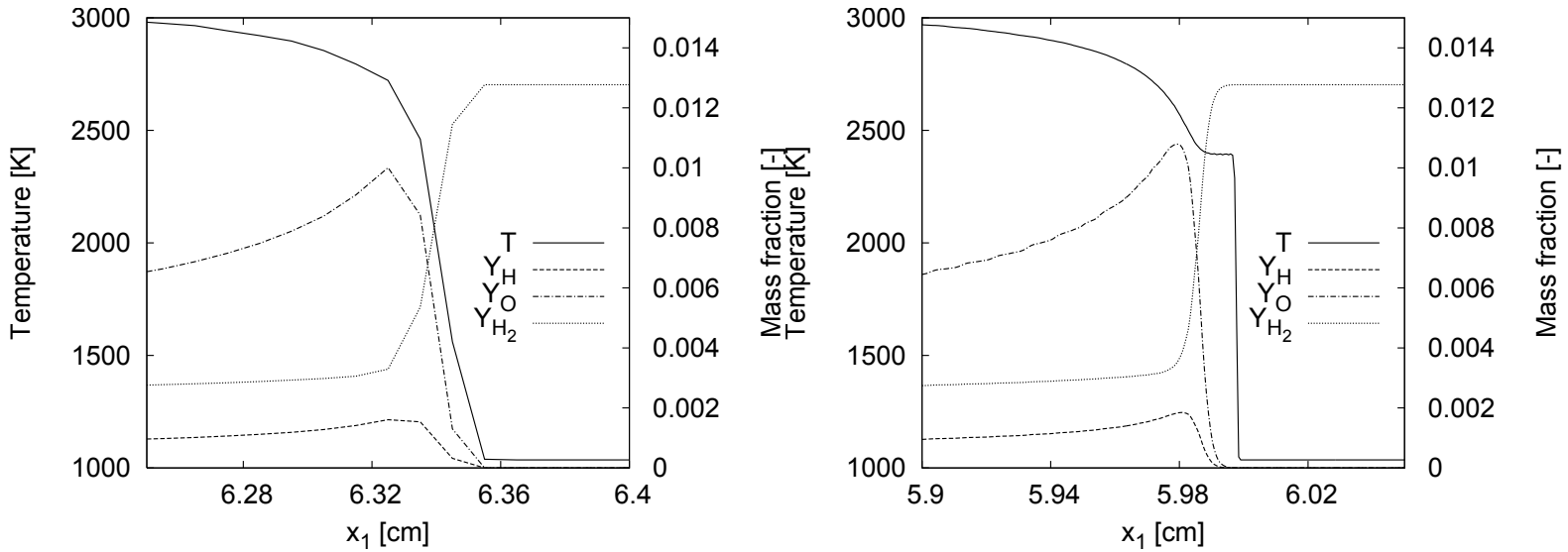
- Extremly high spatial resolution in reaction zone necessary.
- Minimal spatial resolution: $7 - 8\,\mathrm{Pts}/l_{ig} \longrightarrow \Delta x_1 \approx 0.2 - 0.175\,\mathrm{mm}$



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)



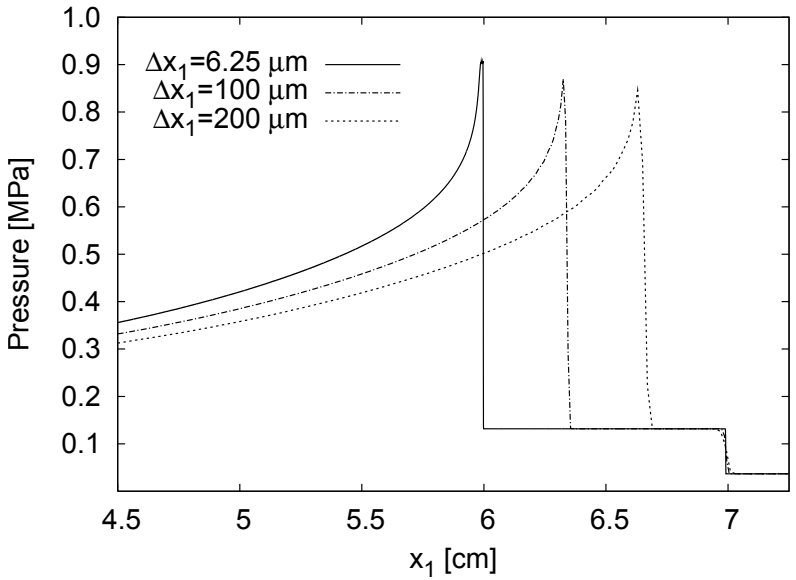
Approximation of $\mathrm{H_2} : \mathrm{O_2}$ detonation at $\sim 1.5\,\mathrm{Pts}/l_{ig}$ (left) and $\sim 24\,\mathrm{Pts}/l_{ig}$ (right)

Detonation ignition in a shock tube

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

Detonation ignition in a shock tube

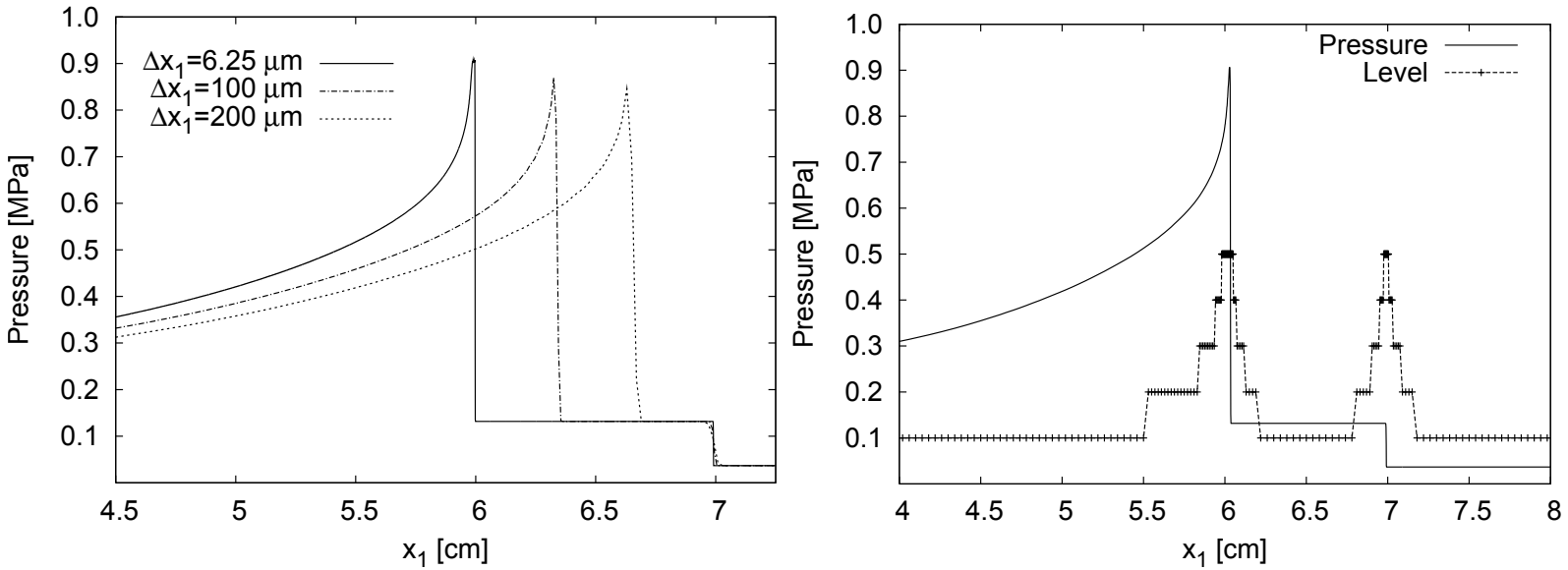
- Shock-induced detonation ignition of $H_2 : O_2 : Ar$ mixture at molar ratios 2:1:7 in closed 1d shock tube
- Insufficient resolution leads to inaccurate results
- Reflected shock is captured correctly by FV scheme, detonation is resolution dependent



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

Detonation ignition in a shock tube

- Shock-induced detonation ignition of $H_2 : O_2 : Ar$ 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 $t = 170\,\mu s$ after shock reflection.Right: Domains of refinement levels

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: $f(O_2) = 0.40$, $f(H_2O) = 6.50$

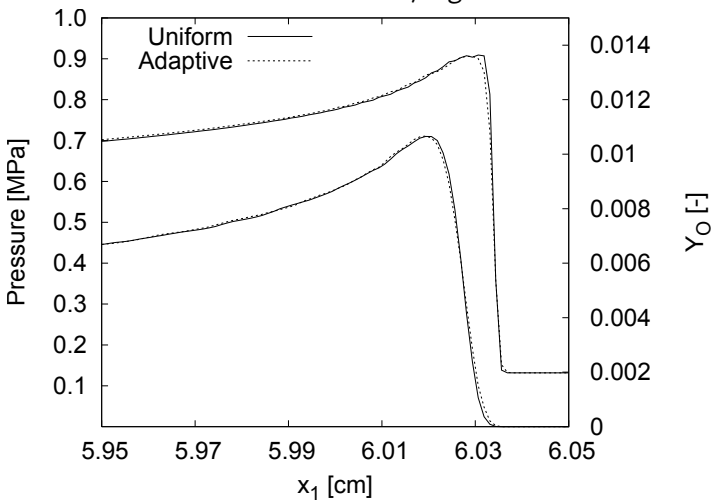
[Westbrook, 1982]

Detonation ignition in 1d - adaptive vs. uniform

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

	Uniform			Adaptive			
$\Delta x_1[\mu m]$	Cells	$t_m[\mu s]$	Time [s]	l_{max}	r_l	$t_m[\mu s]$	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

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

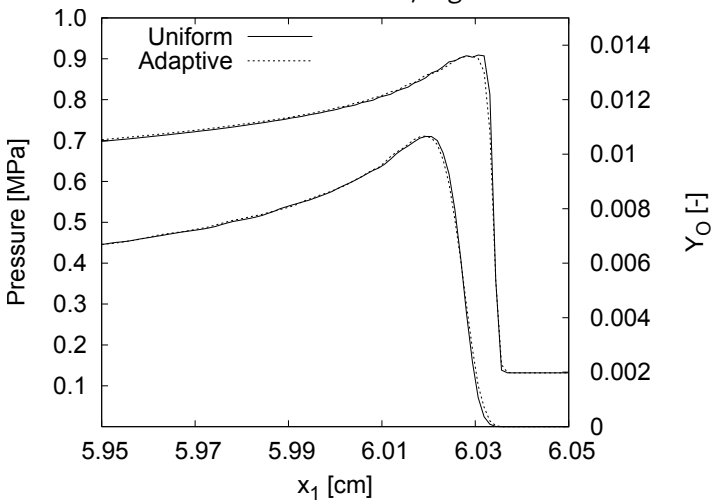


Detonation ignition in 1d - adaptive vs. uniform

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

	Uniform			Adaptive			
$\Delta x_1[\mu m]$	Cells	$t_m[\mu s]$	Time [s]	l_{max}	r_l	$t_m[\mu s]$	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

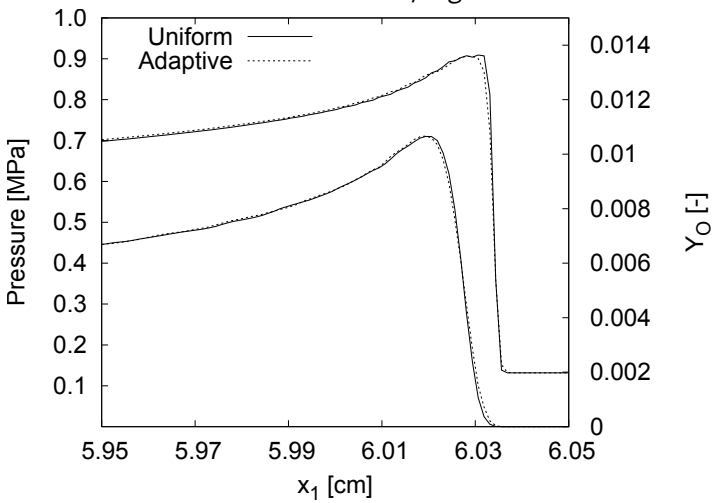
$\epsilon_\rho = 0.07\,\mathrm{kg\,m^{-3}}$, $\epsilon_p = 50\,\mathrm{kPa}$

Detonation ignition in 1d - adaptive vs. uniform

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

	Uniform			Adaptive			
$\Delta x_1[\mu m]$	Cells	$t_m[\mu s]$	Time [s]	l_{max}	r_l	$t_m[\mu s]$	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

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

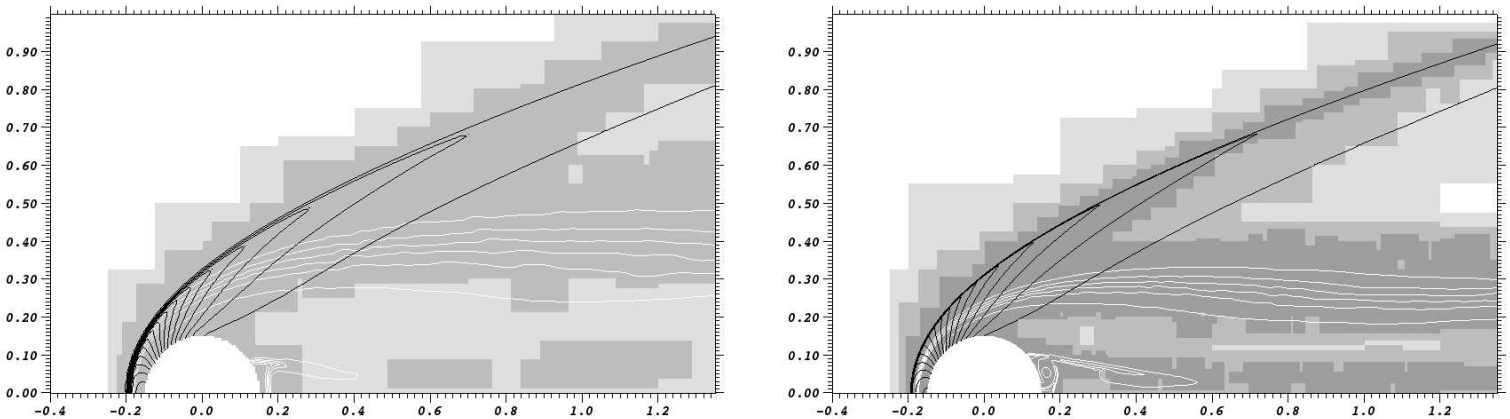
`code/amroc/doc/html/apps/clawpack_2applications_2euler__chem_21d_2ReflecDet_2src_2Problem_8h_source.html`

Shock-induced combustion around a sphere

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

Shock-induced combustion around a sphere

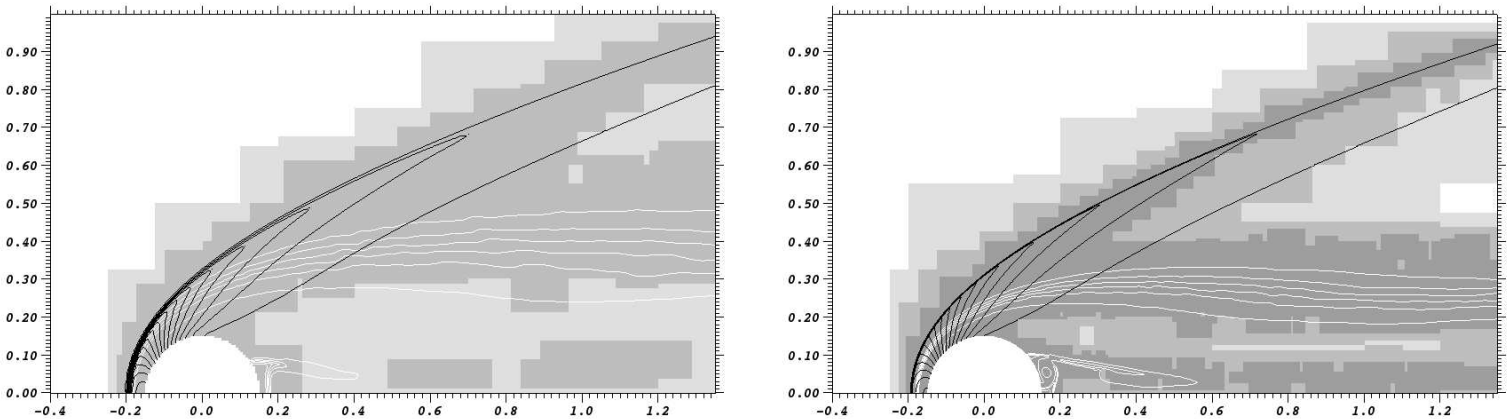
- Spherical projectile of radius $1.5\,\mathrm{mm}$ travels with constant velocity $v_I = 2170.6\,\mathrm{m/s}$ through $\mathrm{H_2} : \mathrm{O_2} : \mathrm{Ar}$ mixture (molar ratios 2:1:7) at $6.67\,\mathrm{kPa}$ and $T = 298\,\mathrm{K}$
- Cylindrical symmetric simulation on AMR base mesh of 70×40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

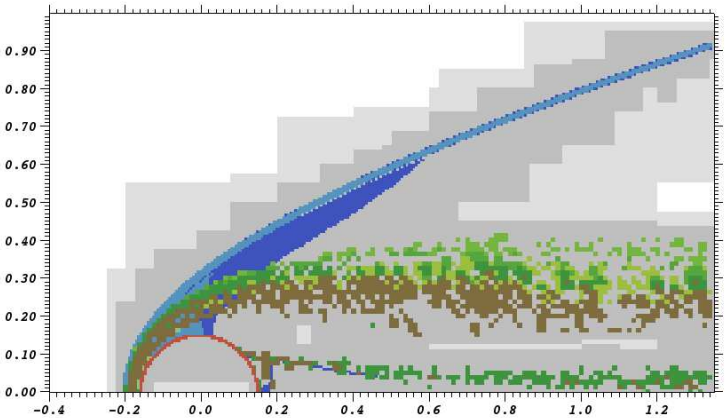


Iso-contours of p (black) and Y_{H_2} (white) on refinement domains for 3-level (left) and 4-level computation (right)

`code/amroc/doc/html/apps/clawpack_2applications_2euler__chem_22d_2Sphere_2src_2Problem_8h_source.html`

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



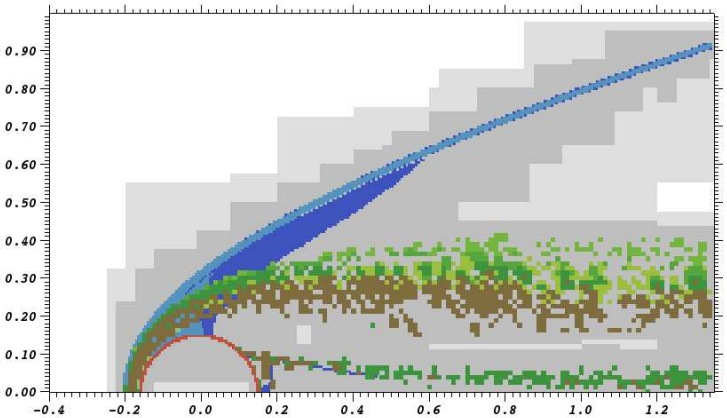
Refinement criteria:

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

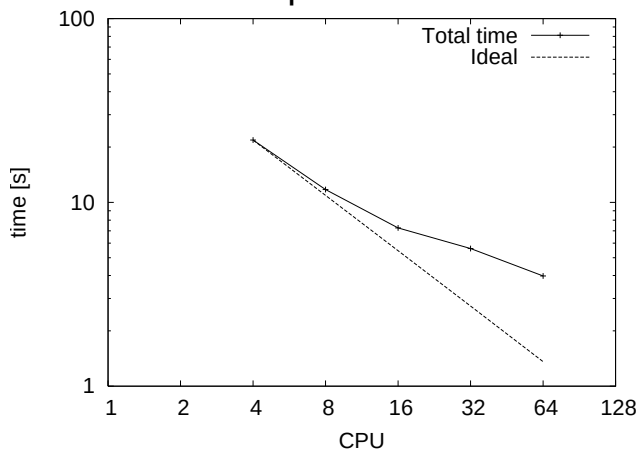


Refinement criteria:

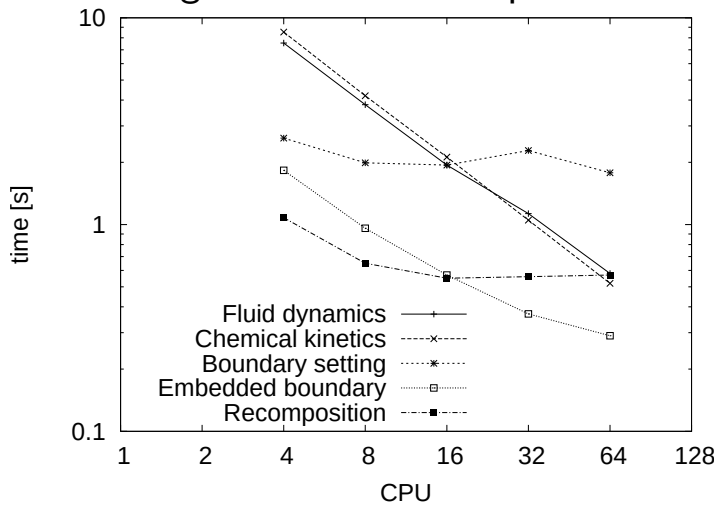
Y_i	$S_{Y_i} \cdot 10^{-4}$	$\eta^r_{Y_i} \cdot 10^{-4}$
O_2	10.0	4.0
H_2O	5.8	3.0
H	0.2	10.0
O	1.4	10.0
OH	2.3	10.0
H_2	1.3	4.0

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

Parallel performance



Scaling of different code portions



References I

[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.

[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.

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

[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.

[Grossmann and Cinella, 1990] Grossmann, B. and Cinella, P. (1990). Flux-split algorithms for flows with non-equilibrium chemistry and vibrational relaxation. *J. Comput. Phys.*, 88:131–168.

References II

[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.

[Larrouturou, 1991] Larrouturou, B. (1991). How to preserve the mass fractions positivity when computing compressible multi-component flows. *J. Comput. Phys.*, 95:59–84.

[Larrouturou and Fezoui, 1989] Larrouturou, B. and Fezoui, L. (1989). On the equations of multi-component perfect or real gas inviscid flow. In Carasso et al., editor, *Proc. of Second Int. Conf. on Nonlinear Hyperbolic Equations - Theory, Numerical Methods, and Applications, Aachen 1988*, Lecture Notes in Mathematics 1402, pages 69–98. Springer-Verlag Berlin.

References III

[Liu and Vinokur, 1989] Liu, Y. and Vinokur, M. (1989). Nonequilibrium flow computations. I. An analysis of numerical formulations of conservation laws. *J. Comput. Phys.*, 83:373–397.

[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 IV

[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.

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

References V

[Shuen et al., 1990] Shuen, J.-S., Liou, M.-S., and van Leer, B. (1990). Inviscid flux-splitting algorithms for real gases with non-equilibrium chemistry. *J. Comput. Phys.*, 90:371–395.

[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.

[Westbrook, 1982] Westbrook, C. K. (1982). Chemical kinetics of hydrocarbon oxidation in gaseous detonations. *Combust. Flame*, 46:191–210.

[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.