Extension of the MOOD Method to Eulerian Simulations of Multi-Material Compressible Flows on Unstructured Meshes

2D preliminary results with a 6-equation model and a fourth-order method

[a]**Steven Diot**, [a]Marianne François, [b]Edward Dendy.

[a]Fluid Dynamics and Solid Mechanics (T-3)
[b]Computational Physics and Methods (CCS-2)
Los Alamos National Laboratory, Los Alamos, NM, USA.

September 3, 2013.

LA-UR 13-26772

UNCLASSIFIED

Los Alamos NATIONAL LABORATORY EST.1943

NNSA

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

Outline

Overview of the MOOD method

A 6-equation model for multi-material flows

An Eulerian discretization on unstructured meshes

Adaptation of the MOOD detection criteria

Preliminary numerical results

Multi-dimensional Optimal Order Detection

To date

- Developed during my Ph.D. in Toulouse (FR) under S. Clain & R. Loubère
- Very-high-order Finite Volume method for single-material Euler equations
- Alternative to WENO limiting on multidimensional unstructured meshes
- Sucessfully tested up to 6^{th}-order of accuracy on 3D polyhedral meshes
- Papers: JCP 2011, CAF 2012, IJNMF 2013 $\Longrightarrow$ public.lanl.gov/diot

On-going

- Extension to multi-material compressible flows *(S.Diot, LANL.)*
- Efficient OpenMP parallelization of 3D code *(G.Moebs, Nantes, FR.)*
- MOOD concept for higher-order remapping *(R.Loubère, Toulouse, FR.)*
- Steady-state problems, shallow water, diffusion *(S.Clain, Guimaraes, PT.)*
- Extension with ADER one-step space-time discr. *(M.Dumbser, Trento, IT.)*

Multi-dimensional Optimal Order Detection

To date

- Developed during my Ph.D. in Toulouse (FR) under S. Clain & R. Loubère
- Very-high-order Finite Volume method for single-material Euler equations
- Alternative to WENO limiting on multidimensional unstructured meshes
- Sucessfully tested up to 6^{th}-order of accuracy on 3D polyhedral meshes
- Papers: JCP 2011, CAF 2012, IJNMF 2013 $\Longrightarrow$ public.lanl.gov/diot

On-going

- Extension to multi-material compressible flows *(S.Diot, LANL.)*
- Efficient OpenMP parallelization of 3D code *(G.Moebs, Nantes, FR.)*
- MOOD concept for higher-order remapping *(R.Loubère, Toulouse, FR.)*
- Steady-state problems, shallow water, diffusion *(S.Clain, Guimaraes, PT.)*
- Extension with ADER one-step space-time discr. *(M.Dumbser, Trento, IT.)*

Multi-dimensional Optimal Order Detection

Main ideas

- Use only one unlimited polynomial reconstruction per cell
- Check after time update (a posteriori) if the solution is acceptable
- If not, locally decrement the cell polynomial degree and recompute
- In the worst case, use the first-order scheme as parachute

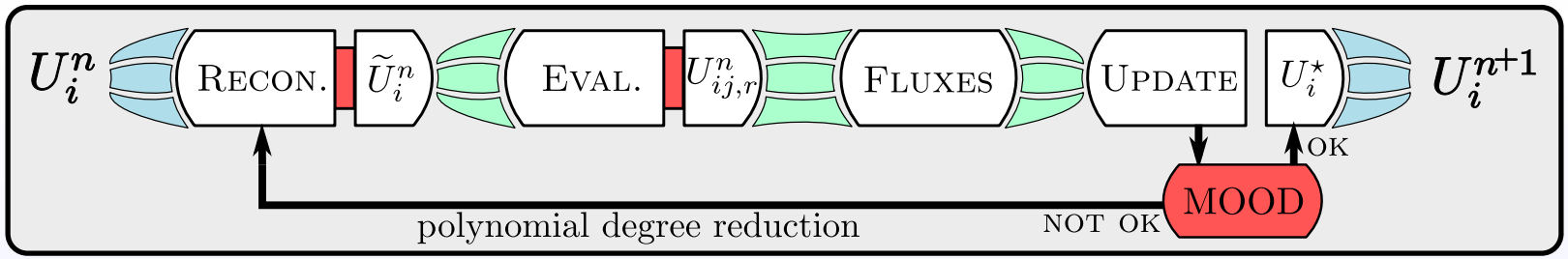
Tools

- Robust and effective framework developed in JCP 2011
- Need for Detection Criteria to define an *acceptable solution*!

Multi-dimensional Optimal Order Detection

Main ideas

- Use only one unlimited polynomial reconstruction per cell
- Check after time update (a posteriori) if the solution is acceptable
- If not, locally decrement the cell polynomial degree and recompute
- In the worst case, use the first-order scheme as parachute



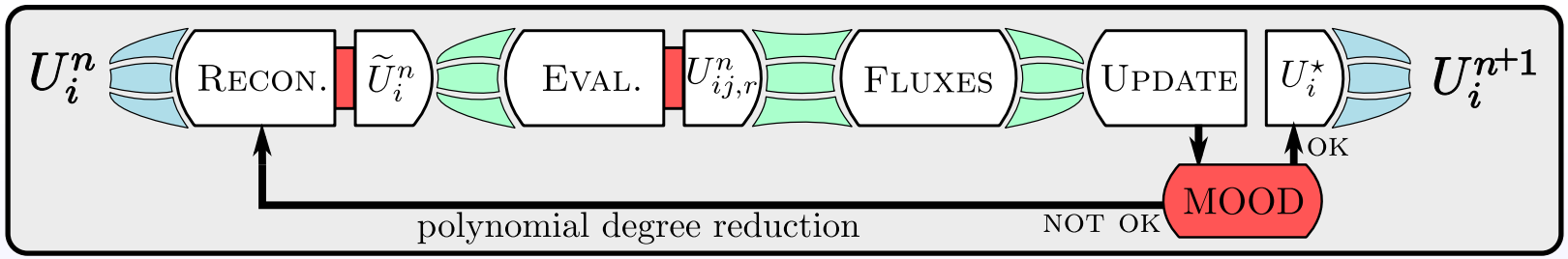
Tools

- Robust and effective framework developed in JCP 2011
- Need for Detection Criteria to define an *acceptable solution*!

Multi-dimensional Optimal Order Detection

Main ideas

- Use only one unlimited polynomial reconstruction per cell
- Check after time update (a posteriori) if the solution is acceptable
- If not, locally decrement the cell polynomial degree and recompute
- In the worst case, use the first-order scheme as parachute



Tools

- Robust and effective framework developed in JCP 2011
- Need for Detection Criteria to define an *acceptable solution*!

6-equation model of Saurel-Petitpas-Berry

Reference: *J. Comput. Phys. 228 (2009) 1678-1712.* (**SPB**)

The non-conservative hyperbolic system is given by

$$\begin{cases} \partial_t U & + & \nabla \cdot F(U) & = & 0 \\ \partial_t(\alpha_1) & + & \mathbf{V} \cdot \nabla(\alpha_1) & = & \mu(p_1 - p_2) \\ \partial_t(\alpha_1\rho_1 e_1) & + & \nabla \cdot (\alpha_1\rho_1 e_1 \mathbf{V}) + \alpha_1 p_1 \nabla \cdot \mathbf{V} & = & -\bar{p}_I \mu(p_1 - p_2) \\ \partial_t(\alpha_2\rho_2 e_2) & + & \nabla \cdot (\alpha_2\rho_2 e_2 \mathbf{V}) + \alpha_2 p_2 \nabla \cdot \mathbf{V} & = & \bar{p}_I \mu(p_1 - p_2) \end{cases}$$

- Conservation of material mass, momentum and total energy

$$U = \begin{pmatrix} \alpha_1\rho_1 \\ \alpha_2\rho_2 \\ \rho\mathbf{V} \\ \rho E \end{pmatrix} \quad \text{and} \quad F(U) = \begin{pmatrix} \alpha_1\rho_1\mathbf{V} \\ \alpha_2\rho_2\mathbf{V} \\ \rho\mathbf{V} \otimes \mathbf{V} + pI \\ (\rho E + p)\mathbf{V} \end{pmatrix}$$

6-equation model of Saurel-Petitpas-Berry

Reference: *J. Comput. Phys. 228 (2009) 1678-1712.* (**SPB**)

The non-conservative hyperbolic system is given by

$$\begin{cases} \partial_t U & + & \nabla \cdot F(U) & = & 0 \\ \partial_t(\alpha_1) & + & \mathbf{V} \cdot \nabla(\alpha_1) & = & \mu(p_1 - p_2) \\ \partial_t(\alpha_1\rho_1 e_1) & + & \nabla \cdot (\alpha_1\rho_1 e_1 \mathbf{V}) + \alpha_1 p_1 \nabla \cdot \mathbf{V} & = & -\bar{p}_I \mu(p_1 - p_2) \\ \partial_t(\alpha_2\rho_2 e_2) & + & \nabla \cdot (\alpha_2\rho_2 e_2 \mathbf{V}) + \alpha_2 p_2 \nabla \cdot \mathbf{V} & = & \bar{p}_I \mu(p_1 - p_2) \end{cases}$$

- Conservation of material mass, momentum and total energy
- Non-conservative volume fraction advection equation + relaxation terms
- Non-conservative material internal energies equations + relaxation terms

$$\sum_k \alpha_k = 1, \quad \rho = \sum_k \alpha_k \rho_k, \quad p = \sum_k \alpha_k p_k, \quad \rho c^2 = \sum_k \alpha_k \rho_k c_k^2 \quad \text{and} \quad \sum_k \alpha_k \rho_k e_k \overset{!}{=} \rho e$$

$\longrightarrow$ Solved in 3 steps: 1) Solve the system w/o relaxation terms
2) Perform the relaxation step
3) Correct the material internal energies

6-equation model of Saurel-Petitpas-Berry

Reference: *J. Comput. Phys. 228 (2009) 1678-1712.* (**SPB**)

The non-conservative hyperbolic system is given by

$$\begin{cases} \partial_t U & + & \nabla \cdot F(U) & = & 0 \\ \partial_t(\alpha_1) & + & \mathbf{V} \cdot \nabla(\alpha_1) & = & \mu(p_1 - p_2) \\ \partial_t(\alpha_1\rho_1 e_1) & + & \nabla \cdot (\alpha_1\rho_1 e_1 \mathbf{V}) + \alpha_1 p_1 \nabla \cdot \mathbf{V} & = & -\bar{p}_I \mu(p_1 - p_2) \\ \partial_t(\alpha_2\rho_2 e_2) & + & \nabla \cdot (\alpha_2\rho_2 e_2 \mathbf{V}) + \alpha_2 p_2 \nabla \cdot \mathbf{V} & = & \bar{p}_I \mu(p_1 - p_2) \end{cases}$$

- Conservation of material mass, momentum and total energy
- Non-conservative volume fraction advection equation + relaxation terms
- Non-conservative material internal energies equations + relaxation terms

$$\sum_k \alpha_k = 1, \quad \rho = \sum_k \alpha_k \rho_k, \quad p = \sum_k \alpha_k p_k, \quad \rho c^2 = \sum_k \alpha_k \rho_k c_k^2 \quad \text{and} \quad \sum_k \alpha_k \rho_k e_k \stackrel{!}{=} \rho e$$

$\longrightarrow$ Solved in 3 steps: 1) Solve the system w/o relaxation terms
2) Perform the relaxation step
3) Correct the material internal energies

6-equation model of Saurel-Petitpas-Berry

Reference: *J. Comput. Phys. 228 (2009) 1678-1712.* (**SPB**)

The non-conservative hyperbolic system is given by

$$\begin{cases} \partial_t U & + & \nabla \cdot F(U) & = & 0 \\ \partial_t(\alpha_1) & + & \mathbf{V} \cdot \nabla(\alpha_1) & = & \mu(p_1 - p_2) \\ \partial_t(\alpha_1\rho_1 e_1) & + & \nabla \cdot (\alpha_1\rho_1 e_1 \mathbf{V}) + \alpha_1 p_1 \nabla \cdot \mathbf{V} & = & -\bar{p}_I \mu(p_1 - p_2) \\ \partial_t(\alpha_2\rho_2 e_2) & + & \nabla \cdot (\alpha_2\rho_2 e_2 \mathbf{V}) + \alpha_2 p_2 \nabla \cdot \mathbf{V} & = & \bar{p}_I \mu(p_1 - p_2) \end{cases}$$

- Conservation of material mass, momentum and total energy
- Non-conservative volume fraction advection equation + relaxation terms
- Non-conservative material internal energies equations + relaxation terms

$$\sum_k \alpha_k = 1, \quad \rho = \sum_k \alpha_k \rho_k, \quad p = \sum_k \alpha_k p_k, \quad \rho c^2 = \sum_k \alpha_k \rho_k c_k^2 \quad \text{and} \quad \sum_k \alpha_k \rho_k e_k \stackrel{!}{=} \rho e$$

$\longrightarrow$ Solved in 3 steps: 1) Solve the system w/o relaxation terms
2) Perform the relaxation step
3) Correct the material internal energies

6-equation model of Saurel-Petitpas-Berry

Reference: *J. Comput. Phys. 228 (2009) 1678-1712.* (**SPB**)

The non-conservative hyperbolic system is given by

$$\begin{cases} \partial_t U & + & \nabla \cdot F(U) & = & 0 \\ \partial_t(\alpha_1) & + & \mathbf{V} \cdot \nabla(\alpha_1) & = & \mu(p_1 - p_2) \\ \partial_t(\alpha_1\rho_1 e_1) & + & \nabla \cdot (\alpha_1\rho_1 e_1 \mathbf{V}) + \alpha_1 p_1 \nabla \cdot \mathbf{V} & = & -\bar{p}_I \mu(p_1 - p_2) \\ \partial_t(\alpha_2\rho_2 e_2) & + & \nabla \cdot (\alpha_2\rho_2 e_2 \mathbf{V}) + \alpha_2 p_2 \nabla \cdot \mathbf{V} & = & \bar{p}_I \mu(p_1 - p_2) \end{cases}$$

- Conservation of material mass, momentum and total energy
- Non-conservative volume fraction advection equation + relaxation terms
- Non-conservative material internal energies equations + relaxation terms

$$\sum_k \alpha_k = 1, \quad \rho = \sum_k \alpha_k \rho_k, \quad p = \sum_k \alpha_k p_k, \quad \rho c^2 = \sum_k \alpha_k \rho_k c_k^2 \text{ and } \sum_k \alpha_k \rho_k e_k \stackrel{!}{=} \rho e$$

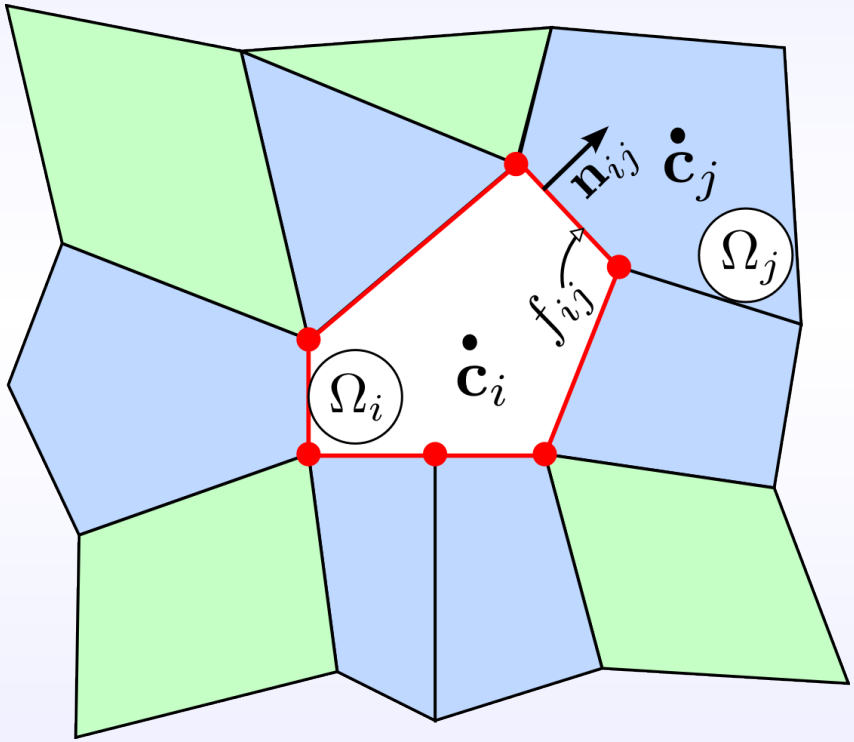
$\longrightarrow$ Solved in 3 steps: 1) Solve the system w/o relaxation terms
2) Perform the relaxation step
3) Correct the material internal energies

Notation

We consider a polygonal mesh of domain $\Omega \subset \mathbb{R}^2$.

Geometry

Ω_i — cell
f_{ij} — face $\Omega_i \cap \Omega_j$
$\mathbf{c}_i$ — centroid of Ω_i
$\mathbf{n}_{ij}$ — normal vector to f_{ij}



Index sets

$\underline{\nu}(i) = \{j \,|\, \Omega_j \cap \Omega_i = f_{ij}\}$
— blue cells

$\overline{\nu}(i) = \{j \,|\, \Omega_j \cap \Omega_i \neq \emptyset\}$
— all colored cells

First-order discretization

Conservative part of the system

$$\begin{aligned}\int_{\Omega_i} \partial_t U(\mathbf{x},t)\, d\mathbf{x} &= -\int_{\Omega_i} \nabla \cdot F(U(\mathbf{x},t))\, d\mathbf{x}, \\ &= -\sum_{j\in\underline{\nu}(i)} \int_{f_{ij}} F(U(\mathbf{x},t)) \cdot \mathbf{n}_{ij}\, dS,\end{aligned}$$

leads to

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{|\Omega_i|} \sum_{j\in\underline{\nu}(i)} |f_{ij}|\, \mathbb{F}(U_i, U_j, \mathbf{n}_{ij}),$$

where $\mathbb{F}$ is an approximate Riemann Solver in the $\mathbf{n}_{ij}$ direction.

$\Longrightarrow$ We use HLLC (see SPB)
(which also provides the normal velocity $u^\star_{\mathbf{n}_{ij}} = \mathbf{V}^\star \cdot \mathbf{n}_{ij}$)

First-order discretization

Volume fraction equations

Using that

$$\partial_t \alpha_1 + \mathbf{V}\cdot\nabla(\alpha_1) \;=\; \partial_t \alpha_1 + \nabla\cdot(\alpha_1\mathbf{V}) - \alpha_1\nabla\cdot\mathbf{V},$$

we have

$$\begin{aligned}\int_{\Omega_i} \partial_t \alpha_1 \, d\mathbf{x} \quad &= \quad -\int_{\Omega_i} \Big[\nabla\cdot(\alpha_1\mathbf{V}) - \alpha_1\nabla\cdot\mathbf{V}\Big]\, d\mathbf{x},\\ &\overset{approx}{\approx} \quad -\sum_{j\in\underline{\nu}(i)} \left[\int_{f_{ij}} (\alpha_1\mathbf{V})\cdot\mathbf{n}_{ij}\, dS - \alpha_1 \int_{f_{ij}} \mathbf{V}\cdot\mathbf{n}_{ij}\, dS\right],\end{aligned}$$

and

$$(\alpha_1)_i^{n+1} = (\alpha_1)_i^n - \frac{\Delta t}{|\Omega_i|}\sum_{j\in\underline{\nu}(i)} |f_{ij}| \left[\alpha_1^\star u^\star_{\mathbf{n}_{ij}} - (\alpha_1)_i^n (u^\star_{\mathbf{n}_{ij}})\right],$$

where $\begin{cases} u^\star_{\mathbf{n}_{ij}} \text{ is given by the Riemann Solver} \\ \alpha_1^\star \text{ is obtained as constant along fluid trajectories} \end{cases}$

First-order discretization

Energies equations

Following the same track, we get

$$\int_{\Omega_i} \partial_t(\alpha_k\rho_k e_k)\, d\mathbf{x} = -\int_{\Omega_i} \Big[\nabla\cdot\big((\alpha_k\rho_k e_k)\mathbf{V}\big) - (\alpha_k p_k)\nabla\cdot\mathbf{V}\Big]\, d\mathbf{x},$$

$$\overset{approx.}{\approx} -\sum_{j\in\underline{\nu}(i)}\left[\int_{f_{ij}}((\alpha_k\rho_k e_k)\mathbf{V})\cdot\mathbf{n}_{ij}\, dS - (\alpha_k p_k)\int_{f_{ij}}\mathbf{V}\cdot\mathbf{n}_{ij}\, dS\right],$$

and

$$(\alpha_k\rho_k e_k)_i^{n+1} = (\alpha_k\rho_k e_k)_i^n - \frac{\Delta t}{|\Omega_i|}\sum_{j\in\underline{\nu}(i)}|f_{ij}|\left[(\alpha_k\rho_k e_k)^\star u^\star_{\mathbf{n}_{ij}} - (\alpha_k p_k)_i^n(u^\star_{\mathbf{n}_{ij}})\right],$$

where $\begin{cases} u^\star_{\mathbf{n}_{ij}} \text{ is given by the Riemann Solver} \\ (\alpha_k\rho_k e_k)^\star \text{ is determined by Hugoniot relation proposed in SPB} \end{cases}$

Pressure Relaxation - Energies Correction

Stiffened gas EOS $p_k = \rho_k e_k(\gamma_k - 1) - \Pi_k \gamma_k$

Relaxation step (only affects α_k)

- After manipulations, the relaxation step consists in solving (for p)

$$\sum_k (\alpha_k \rho_k) \nu_k(p) = 1, \quad (\alpha_k \rho_k) \text{ being constant},$$

  where the specific volume is

$$\nu_k(p) = \nu_k^0 \frac{p_k^0 + \gamma_k \pi_k + (\gamma_k - 1)\hat{p}_I}{p + \gamma_k \pi_k + (\gamma_k - 1)\hat{p}_I}$$

- Use Newton's iterative method to find the relaxed pressure p
- Compute $\rho_k = \nu_k(p)^{-1}$ and deduce the corrected vol. fraction $\alpha_k = \dfrac{\alpha_k \rho_k}{\rho_k}$

Energies correction (only affects $\alpha_k \rho_k e_k$)

- Compute the mixture pressure $p_{mix} = \left(\rho e - \sum_k \dfrac{\alpha_k \gamma_k \pi_k}{\gamma_k - 1}\right) \Big/ \left(\sum_k \dfrac{\alpha_k}{\gamma_k - 1}\right)$
- Deduce the corrected material energies from EOS $e_k = e_k(\alpha_k, \alpha_k \rho_k, p_{mix})$

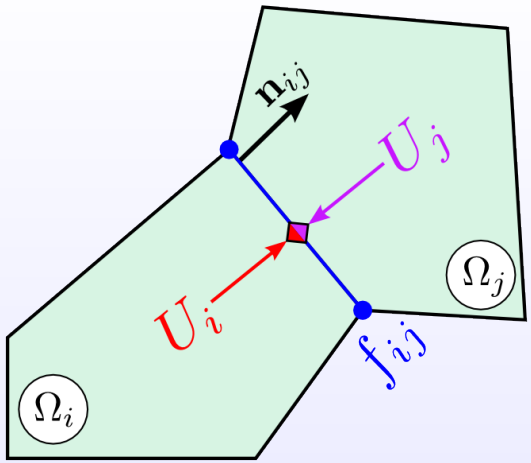
Higher-order discretization

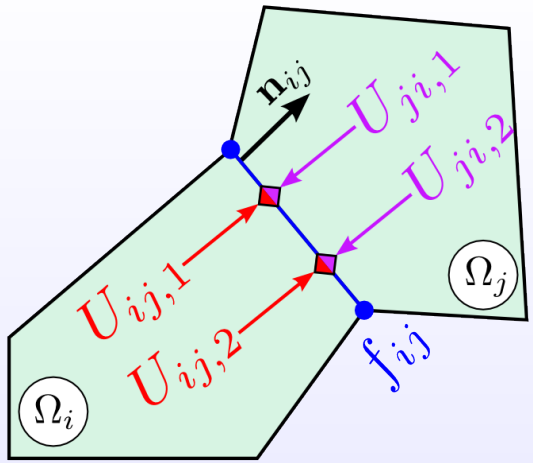
In space

- Integrals on face f_{ij} are approximated by quadrature rules ($r=1,\dots,R$)
  - ⋆ Points $q_{ij,r}$ on face f_{ij}
  - ⋆ Weights $\xi_r \geq 0$ such as $\sum_r \xi_r = 1$

$$\int_{f_{ij}} F(U)\cdot \mathbf{n}_{ij} dS \overset{LO}{\approx} \mathbb{F}(U_i, U_j, \mathbf{n}_{ij}) \overset{HO}{\approx} \sum_{r=1}^{R} \xi_r \mathbb{F}(U_{ij,r}, U_{ji,r}, \mathbf{n}_{ij})$$

Illustration:





Higher-order discretization

In space

- Integrals on face f_{ij} are approximated by quadrature rules ($r=1,...,R$)
  - ⋆ Points $q_{ij,r}$ on face f_{ij}
  - ⋆ Weights $\xi_r \geq 0$ such as $\sum_r \xi_r = 1$

$$\int_{f_{ij}} F(U)\cdot \mathbf{n}_{ij} dS \overset{LO}{\approx} \mathbb{F}(U_i, U_j, \mathbf{n}_{ij}) \overset{HO}{\approx} \sum_{r=1}^{R} \xi_r \mathbb{F}(U_{ij,r}, U_{ji,r}, \mathbf{n}_{ij})$$

- Higher-order approximations
  - ⋆ Compute (LS) polynomial reconstructions $\widetilde{U}_i(\mathbf{x})$ on every Ω_i
  - ⋆ Evaluate reconstructions at $q_{ij,r}$: $U_{ij,r} = \widetilde{U}_i(q_{ij,r})$ / $U_{ji,r} = \widetilde{U}_j(q_{ij,r})$

$\Longrightarrow$ Same method for face integrals in non-conservative equations

In time

- We use the RK3 TVD method (3 sub-steps)
- The MOOD method is performed over each substep

Higher-order discretization

In space

- Integrals on face f_{ij} are approximated by quadrature rules ($r=1,...,R$)
  - ⋆ Points $q_{ij,r}$ on face f_{ij}
  - ⋆ Weights $\xi_r \geq 0$ such as $\sum_r \xi_r = 1$

$$\int_{f_{ij}} F(U)\cdot \mathbf{n}_{ij} dS \overset{LO}{\approx} \mathbb{F}(U_i, U_j, \mathbf{n}_{ij}) \overset{HO}{\approx} \sum_{r=1}^{R} \xi_r \mathbb{F}(U_{ij,r}, U_{ji,r}, \mathbf{n}_{ij})$$

- Higher-order approximations
  - ⋆ Compute (LS) polynomial reconstructions $\widetilde{U}_i(\mathbf{x})$ on every Ω_i
  - ⋆ Evaluate reconstructions at $q_{ij,r}$: $U_{ij,r} = \widetilde{U}_i(q_{ij,r})$ / $U_{ji,r} = \widetilde{U}_j(q_{ij,r})$

$\Longrightarrow$ Same method for face integrals in non-conservative equations

In time

- We use the RK3 TVD method (3 sub-steps)
- The MOOD method is performed over each substep

Higher-order discretization

In space

- Integrals on face f_{ij} are approximated by quadrature rules ($r=1,...,R$)
  - ⋆ Points $q_{ij,r}$ on face f_{ij}
  - ⋆ Weights $\xi_r \geq 0$ such as $\sum_r \xi_r = 1$

$$\int_{f_{ij}} F(U)\cdot \mathbf{n}_{ij} dS \overset{LO}{\approx} \mathbb{F}(U_i, U_j, \mathbf{n}_{ij}) \overset{HO}{\approx} \sum_{r=1}^{R} \xi_r \mathbb{F}(U_{ij,r}, U_{ji,r}, \mathbf{n}_{ij})$$

- Higher-order approximations
  - ⋆ Compute (LS) polynomial reconstructions $\widetilde{U}_i(\mathbf{x})$ on every Ω_i
  - ⋆ Evaluate reconstructions at $q_{ij,r}$: $U_{ij,r} = \widetilde{U}_i(q_{ij,r})$ / $U_{ji,r} = \widetilde{U}_j(q_{ij,r})$

  $\Longrightarrow$ Same method for face integrals in non-conservative equations

In time

- We use the RK3 TVD method (3 sub-steps)
- The MOOD method is performed over each substep

MOOD criteria - From the single-material case

Detection Criteria

- **PAD**: Physical Admissibility Detection
  - ⋆ Minimal conditions to ensure physicality of the solution
  - ⋆ For the single-material case: $\rho > 0$, $p > 0$, $e > 0$, etc.
- **DMP**: Discrete Maximum Principle
  - ⋆ Designed to prevent spurious oscillations from appearing

Check if $u_i^\star$ fulfills the Discrete Maximum Principle:

$$\min_j(u_i^n, u_j^n) \leq u_i^\star \leq \max_j(u_i^n, u_j^n)$$

  - ⋆ DMP violation must be allowed at smooth extrema to reach VHO
- **u2**: Smoothness Detection
  - ⋆ Only checked if the DMP is violated, *i.e.* 2^{nd} filter
  - ⋆ Distinguishes a discontinuity from a smooth extremum
  - ⋆ Uses approximations of *curvatures* over a local neigborhood

$\Longrightarrow$ For the single-material case: DMP+u2 applied on density!

MOOD criteria - From the single-material case

Detection Criteria

- **PAD**: Physical Admissibility Detection
  - ⋆ Minimal conditions to ensure physicality of the solution
  - ⋆ For the single-material case: $\rho > 0$, $p > 0$, $e > 0$, etc.
- **DMP**: Discrete Maximum Principle
  - ⋆ Designed to prevent spurious oscillations from appearing

Check if $u_i^\star$ fulfills the Discrete Maximum Principle:

$$\min_j(u_i^n, u_j^n) \leq u_i^\star \leq \max_j(u_i^n, u_j^n)$$

  - ⋆ DMP violation must be allowed at smooth extrema to reach VHO
- **u2**: Smoothness Detection
  - ⋆ Only checked if the DMP is violated, *i.e.* 2^{nd} filter
  - ⋆ Distinguishes a discontinuity from a smooth extremum
  - ⋆ Uses approximations of *curvatures* over a local neigborhood

$\Longrightarrow$ For the single-material case: DMP+u2 applied on density!

MOOD criteria - From the single-material case

Detection Criteria

- **PAD**: Physical Admissibility Detection
  - ⋆ Minimal conditions to ensure physicality of the solution
  - ⋆ For the single-material case: $\rho > 0$, $p > 0$, $e > 0$, etc.
- **DMP**: Discrete Maximum Principle
  - ⋆ Designed to prevent spurious oscillations from appearing

Check if $u_i^\star$ fulfills the Discrete Maximum Principle:

$$\min_j(u_i^n, u_j^n) \leq u_i^\star \leq \max_j(u_i^n, u_j^n)$$

  - ⋆ DMP violation must be allowed at smooth extrema to reach VHO
- **u2**: Smoothness Detection
  - ⋆ Only checked if the DMP is violated, *i.e.* 2^{nd} filter
  - ⋆ Distinguishes a discontinuity from a smooth extremum
  - ⋆ Uses approximations of *curvatures* over a local neigborhood

$\Longrightarrow$ For the single-material case: DMP+u2 applied on density!

MOOD criteria - From the single-material case

Detection Criteria

- **PAD**: Physical Admissibility Detection
  - ⋆ Minimal conditions to ensure physicality of the solution
  - ⋆ For the single-material case: $\rho > 0$, $p > 0$, $e > 0$, etc.
- **DMP**: Discrete Maximum Principle
  - ⋆ Designed to prevent spurious oscillations from appearing

Check if $u_i^\star$ fulfills the Discrete Maximum Principle:

$$\min_j(u_i^n, u_j^n) \leq u_i^\star \leq \max_j(u_i^n, u_j^n)$$

  - ⋆ DMP violation must be allowed at smooth extrema to reach VHO
- **u2**: Smoothness Detection
  - ⋆ Only checked if the DMP is violated, *i.e.* 2^{nd} filter
  - ⋆ Distinguishes a discontinuity from a smooth extremum
  - ⋆ Uses approximations of *curvatures* over a local neigborhood

$\Longrightarrow$ For the single-material case: DMP+u2 applied on density!

MOOD criteria - To the multi-material one

Detection Criteria

- **PAD**: Physical Admissibility Detection
  - ⋆ Positivity of volume fractions: $\alpha_k > 0,\ \forall k$,
  - ⋆ Positivity of material densities: $\alpha_k \rho_k > 0,\ \forall k$,
  - ⋆ Positivity of material energies: $\alpha_k \rho_k e_k > 0,\ \forall k$,
  - ⋆ Positivity of mixture pressure: $p = \sum_k \alpha_k p_k > 0$.
- **DMP+u2** applied on mixture density $\rho = \sum_k \alpha_k \rho_k$

Remarks

- PAD seems to ensure robustness, but still to be proven
- DMP+u2 on ρ reduces the appearance of spurious oscillations
- Preliminary choice $\longrightarrow$ May be improved in the future

MOOD criteria - To the multi-material one

Detection Criteria

- **PAD**: Physical Admissibility Detection
  - ⋆ Positivity of volume fractions: $\alpha_k > 0,\ \forall k$,
  - ⋆ Positivity of material densities: $\alpha_k \rho_k > 0,\ \forall k$,
  - ⋆ Positivity of material energies: $\alpha_k \rho_k e_k > 0,\ \forall k$,
  - ⋆ Positivity of mixture pressure: $p = \sum_k \alpha_k p_k > 0$.
- **DMP+u2** applied on mixture density $\rho = \sum_k \alpha_k \rho_k$

Remarks

- PAD seems to ensure robustness, but still to be proven
- DMP+u2 on ρ reduces the appearance of spurious oscillations
- Preliminary choice $\longrightarrow$ May be improved in the future

MOOD criteria - To the multi-material one

Detection Criteria

- **PAD**: Physical Admissibility Detection
  - ⋆ Positivity of volume fractions: $\alpha_k > 0,\ \forall k$,
  - ⋆ Positivity of material densities: $\alpha_k \rho_k > 0,\ \forall k$,
  - ⋆ Positivity of material energies: $\alpha_k \rho_k e_k > 0,\ \forall k$,
  - ⋆ Positivity of mixture pressure: $p = \sum_k \alpha_k p_k > 0$.
- **DMP+u2** applied on mixture density $\rho = \sum_k \alpha_k \rho_k$

Remarks

- PAD seems to ensure robustness, but still to be proven
- DMP+u2 on ρ reduces the appearance of spurious oscillations
- Preliminary choice $\longrightarrow$ May be improved in the future

Numerical results - General comments

About the MUSCL-like implementation

- 1D results are computed with my research code using SPB model
- 2D results are computed with LANL xRage code (dim. splitting)
  - ⋆ Modified 6-eq. model of Miller-Puckett (Francois *et al*, CaF 2012)
  - ⋆ Interface Preserver is always enabled to reduce interface diffusion
  - ⋆ Only cartesian meshes without AMR technique

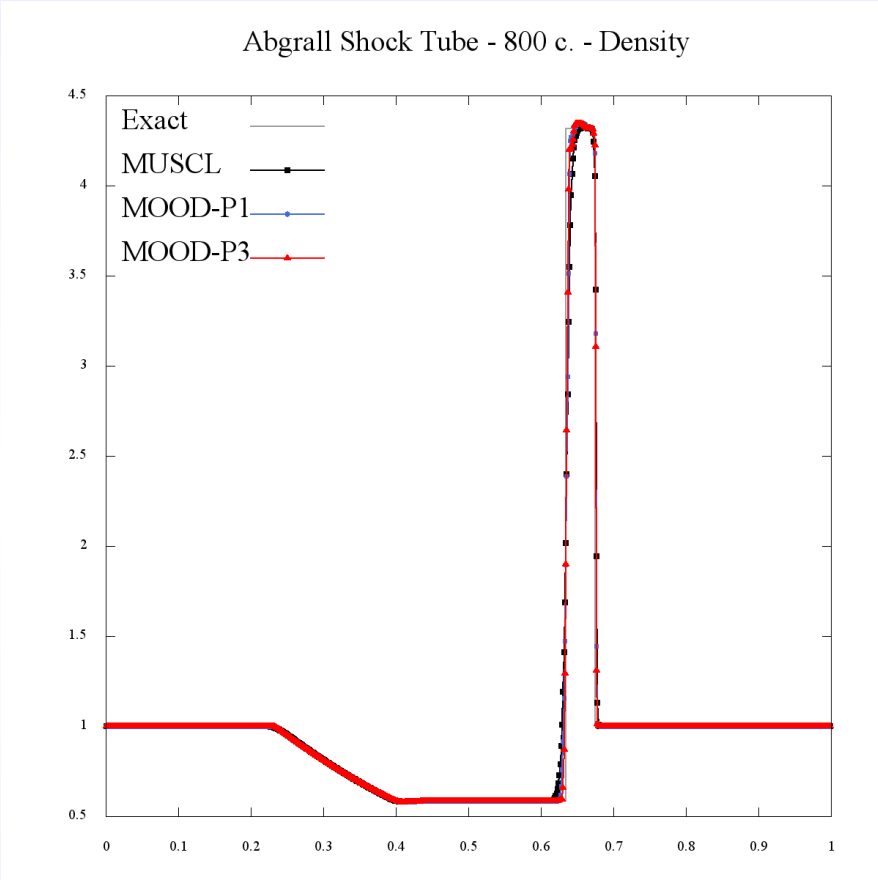
About the MOOD implementation

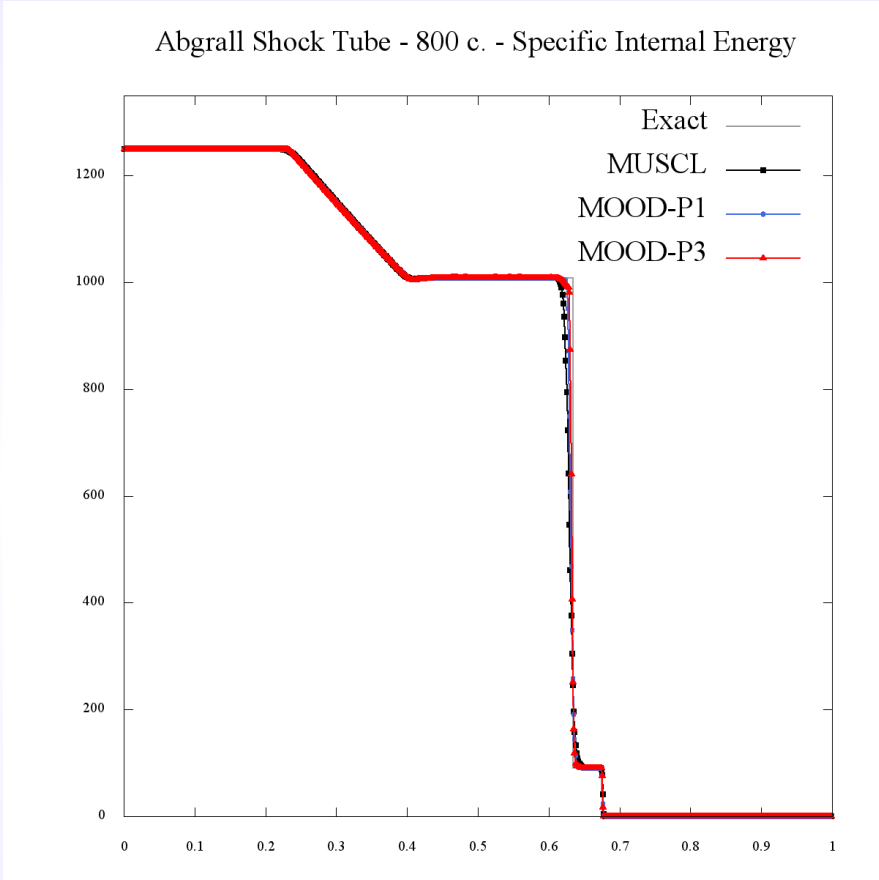
- 2D unstructured (polygonal) multi-material research code
- Polynomial reconstructions of arbitrary degree, here $\mathbb{P}_1$ & $\mathbb{P}_3$
- MOOD *parameters* are set independently of cases/meshes

Numerical parameters

- The CFL coefficient is set to 0.6
- The smallest volume fraction is always 10^{-8}

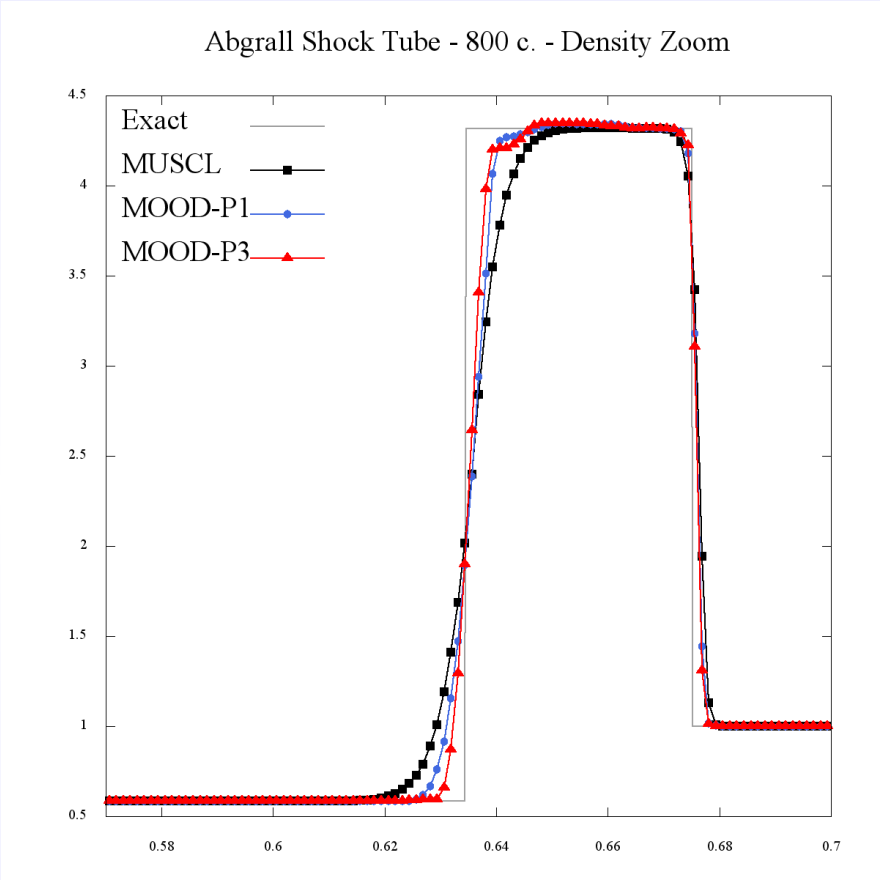
Abgrall shock tube - 800 cells - Solution

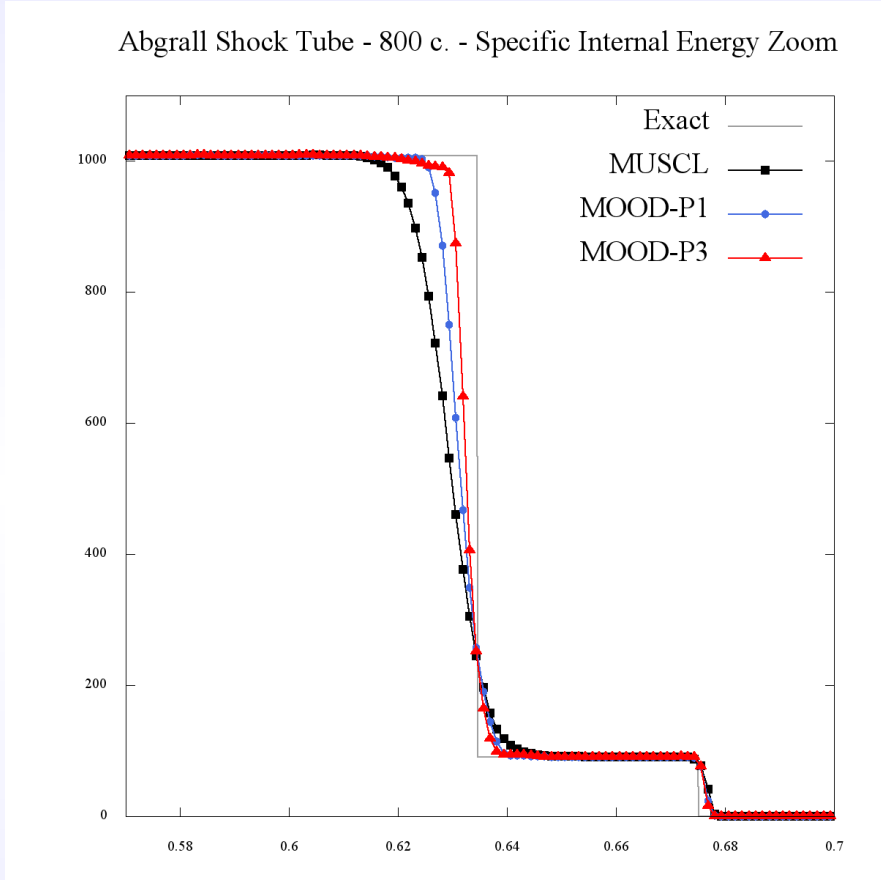




$t_{end} = 0.01s$	ρ	u	p	γ	π	Boundary Cond.
Left State on $[0.0; 0.5]$	1	0	500	1.4	0	Reflective at x=0.0
Right State on $[0.5; 1.0]$	1	0	0.2	1.6	0	Reflective at x=1.0

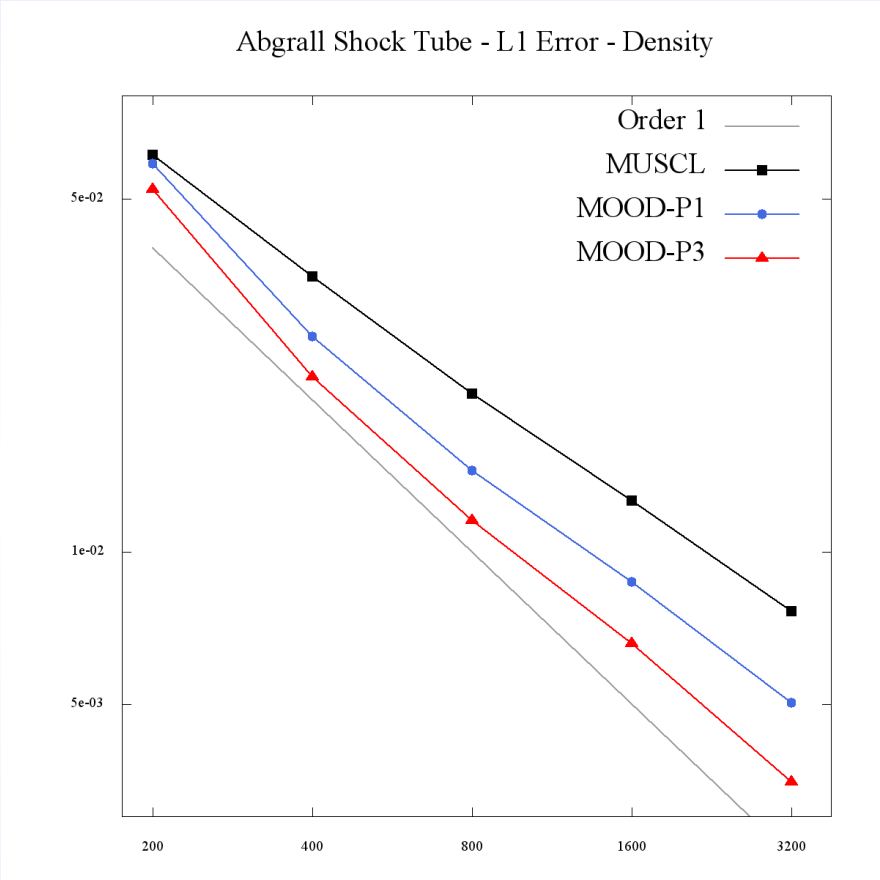
Abgrall shock tube - 800 cells - Solution

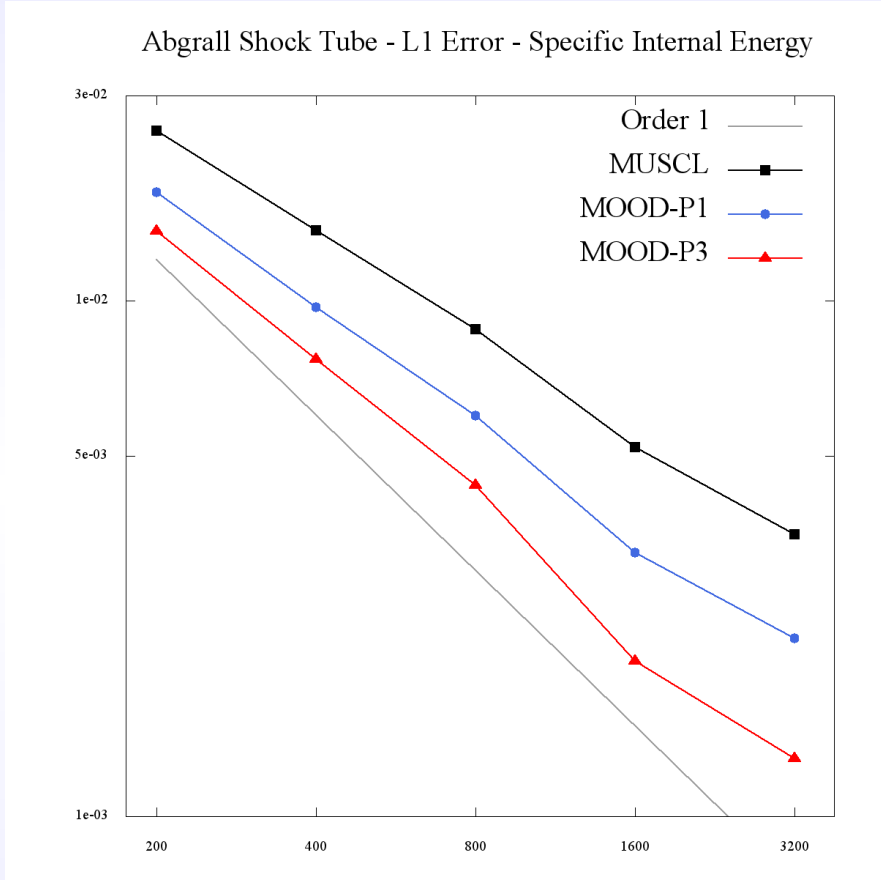




$t_{end} = 0.01s$	ρ	u	p	γ	π	Boundary Cond.
Left State on $[0.0; 0.5]$	1	0	500	1.4	0	Reflective at x=0.0
Right State on $[0.5; 1.0]$	1	0	0.2	1.6	0	Reflective at x=1.0

Abgrall shock tube - 800 cells - Convergence

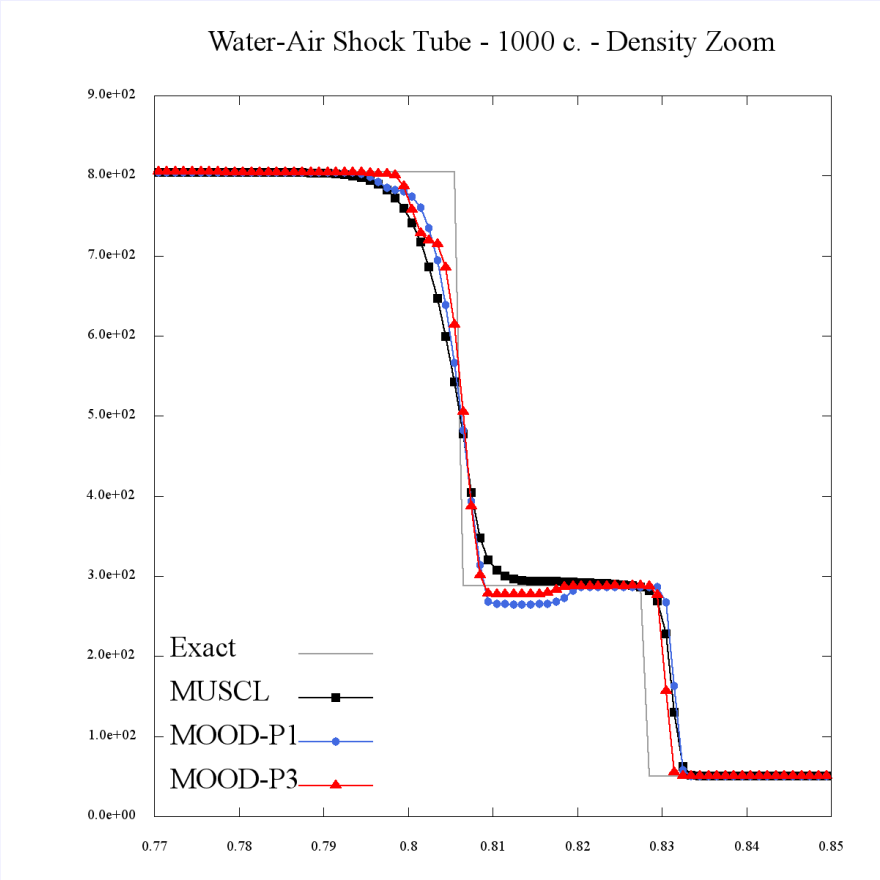


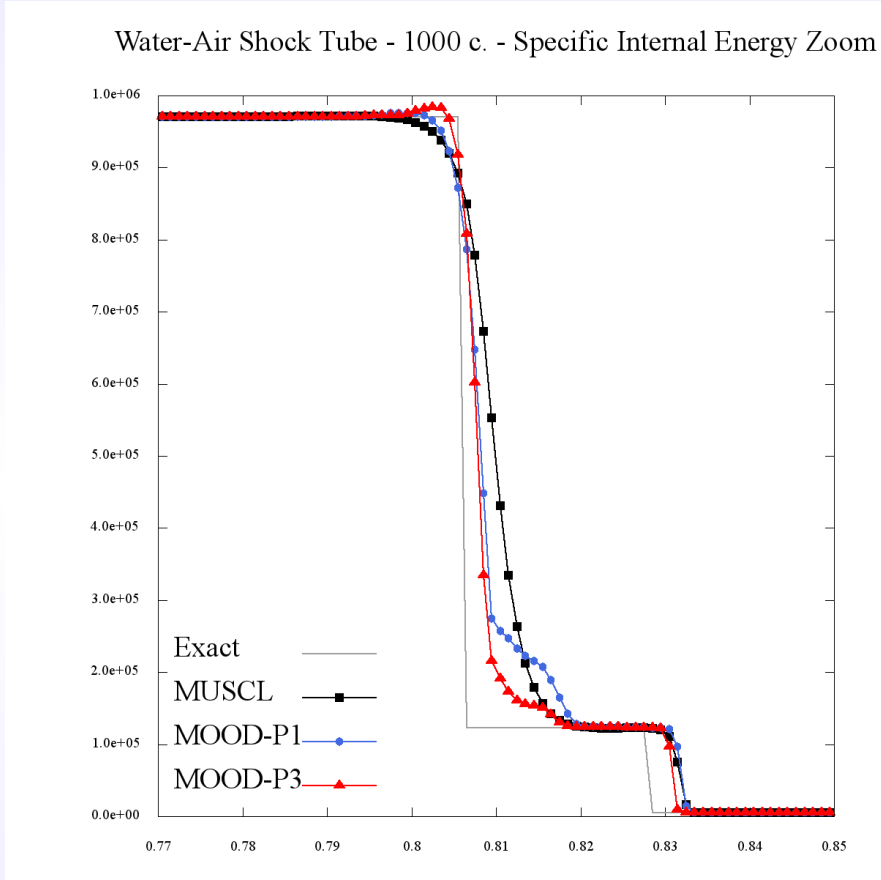


3200 cells	Relative Error (Ratio) MUSCL	MOOD-ℙ1	MOOD-ℙ3	Relative Time Units MUSCL	MOOD-ℙ1	MOOD-ℙ3
Density	**1.0**	**0.66** (1.5)	**0.45** (2.2)	**1.0**	**1.4** (**2.2**)	**2.5** (**4.8**)
Sp.Int.E.	**1.0**	**0.63** (1.6)	**0.37** (2.7)	**1.0**	**1.4** (**2.5**)	**2.5** (**7.2**)

In red: Estimated R.T.U. for MUSCL to reach same error assuming order 1 (on a finer mesh)

Water-Air shock tube - 1000 cells - Solution





$t_{end} = 220\mu s$	ρ	u	p	γ	π	Boundary Cond.
Left State on $[0.0; 0.7]$	1000	0	1.10^9	4.4	6.10^8	Reflective at x=0.0
Right State on $[0.7; 1.0]$	50	0	1.10^5	1.4	0	Reflective at x=1.0

Water-Air shock tube - 1000 cells - Solution

Effect of the energy correction:

- SPB: Energy correction from mixture pressure as presented before
- RED: Simple REDistribution algorithm
  - ⋆ Compute internal energy delta: $\delta_{\rho e} = \rho e - \sum_k (\alpha_k \rho_k e_k)^{old}$
  - ⋆ Distribute it according to $\frac{(\alpha_k \rho_k)}{\rho}$: $(\alpha_k \rho_k e_k)^{new} = (\alpha_k \rho_k e_k)^{old} + \frac{(\alpha_k \rho_k)}{\rho} \delta_{\rho e}$

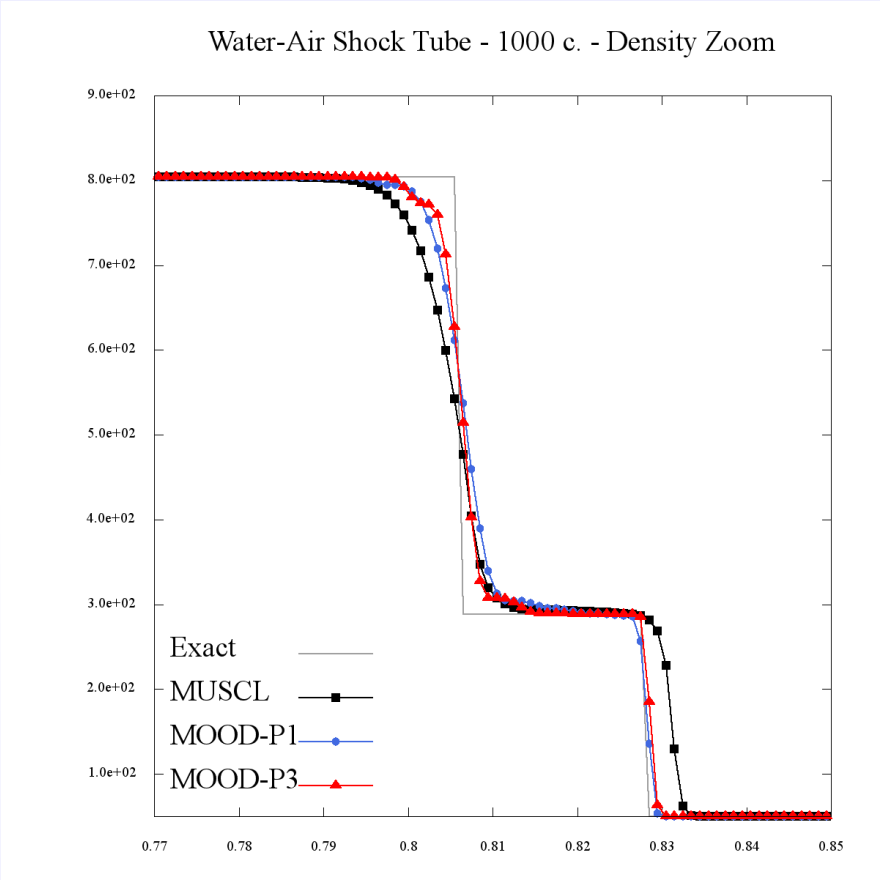
L1 Errors:

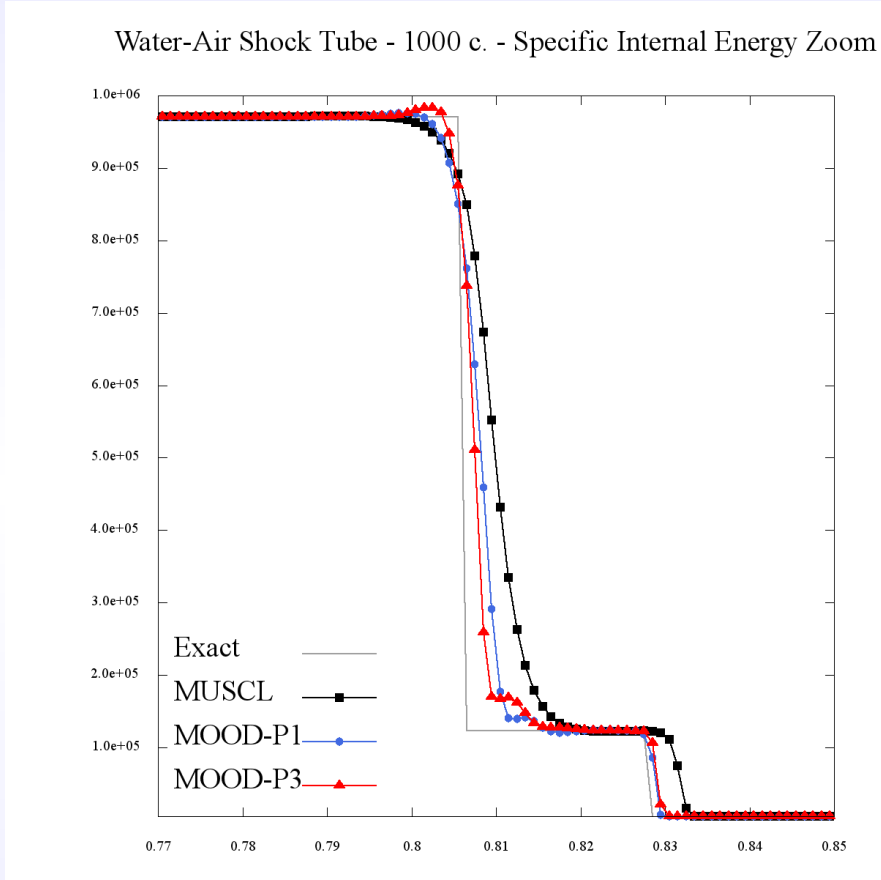
	MUSCL	MOOD-P1		MOOD-P3	
	SPB	SPB	RED	SPB	RED
Density	3.466E-03	3.271E-03	2.246E-03	2.595E-03	1.716E-03
Velocity	8.028E-03	8.499E-03	3.276E-03	6.625E-03	3.233E-03
Pressure	4.156E-03	3.971E-03	3.937E-03	3.734E-03	3.236E-03
Sp.Int.En.	4.932E-03	3.947E-03	2.720E-03	2.795E-03	2.131E-03

$\longrightarrow$ Energy correction significantly affects the solution

$\Longrightarrow$ Need more work

Water-Air shock tube - 1000 cells - Solution

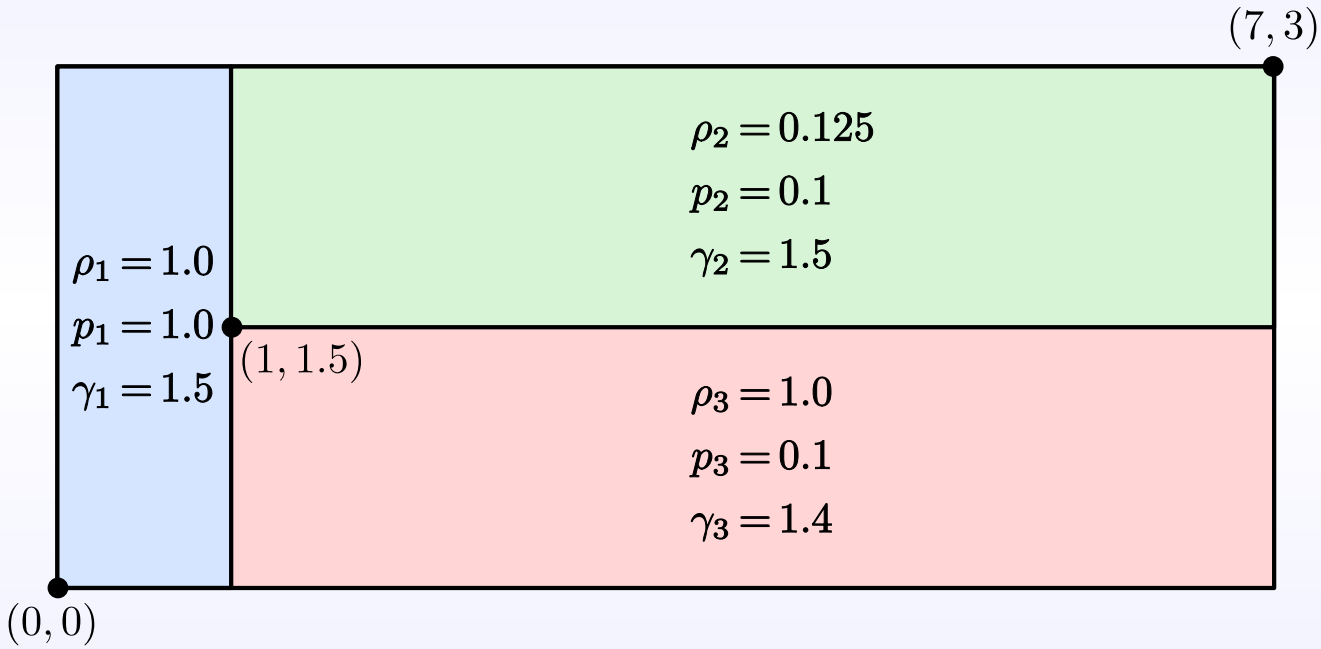




$t_{end} = 220\mu s$	ρ	u	p	γ	π	Boundary Cond.
Left State on $[0.0; 0.7]$	1000	0	1.10^9	4.4	6.10^8	Reflective at x=0.0
Right State on $[0.7; 1.0]$	50	0	1.10^5	1.4	0	Reflective at x=1.0

Triple Point Problem - Initialization

- Galera *et al.* "A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction"

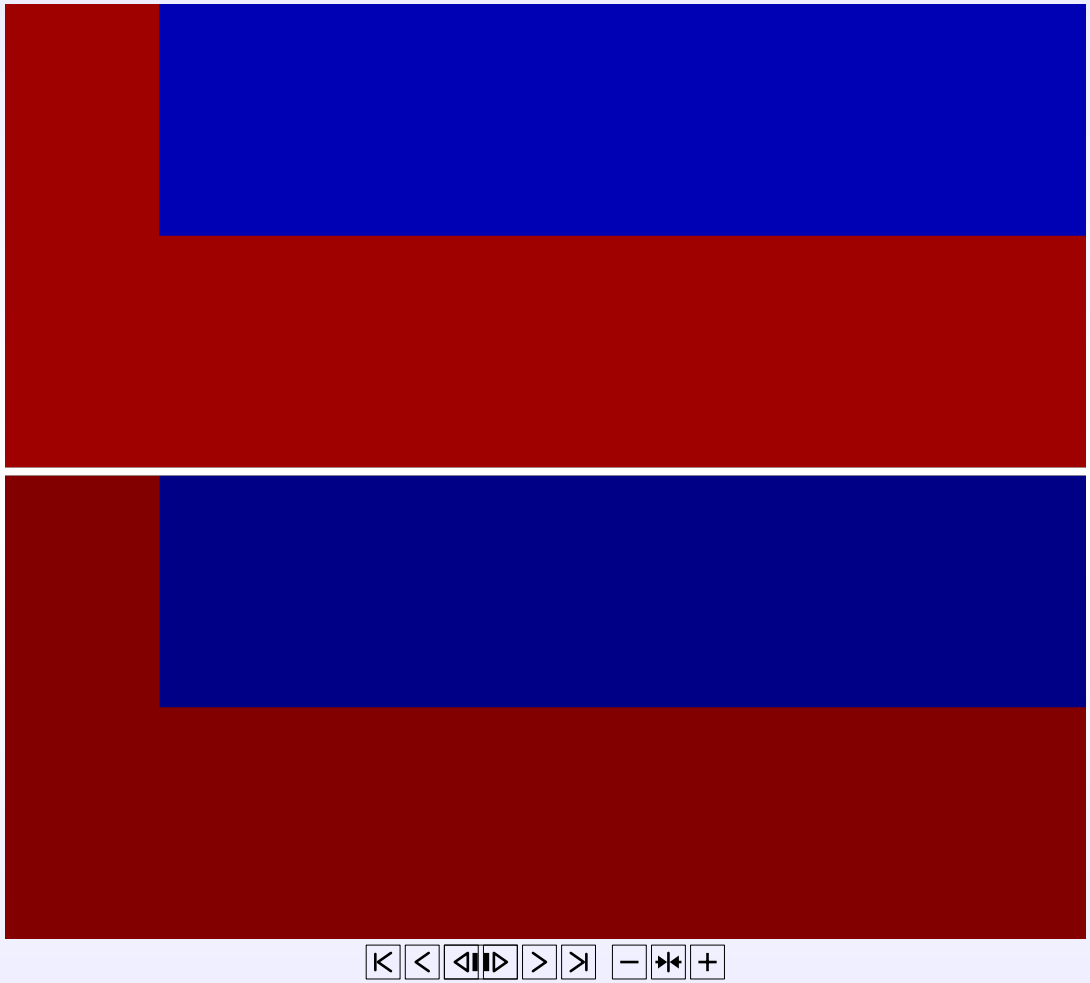


- Final time = 5.0
- Wall boundary conditions

Triple Point - 2^{nd}-order comparison

MOOD-P1
1120x480
Density

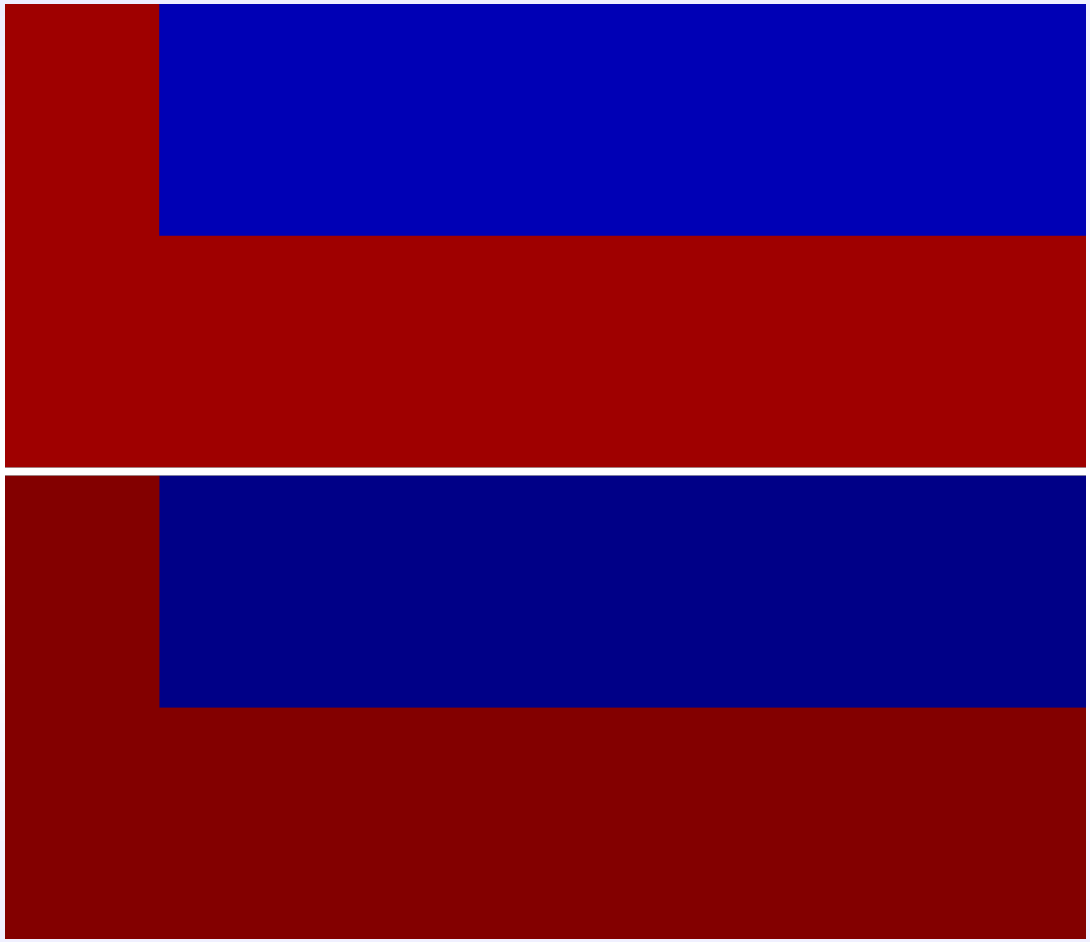
xRAGE
1120x480
Density



$\longrightarrow$ 2^{nd}-order results are qualitatively equivalent (validation)

Triple Point - 4^{th}-order *vs* 2^{nd}-order

MOOD-P3
560x240
Density



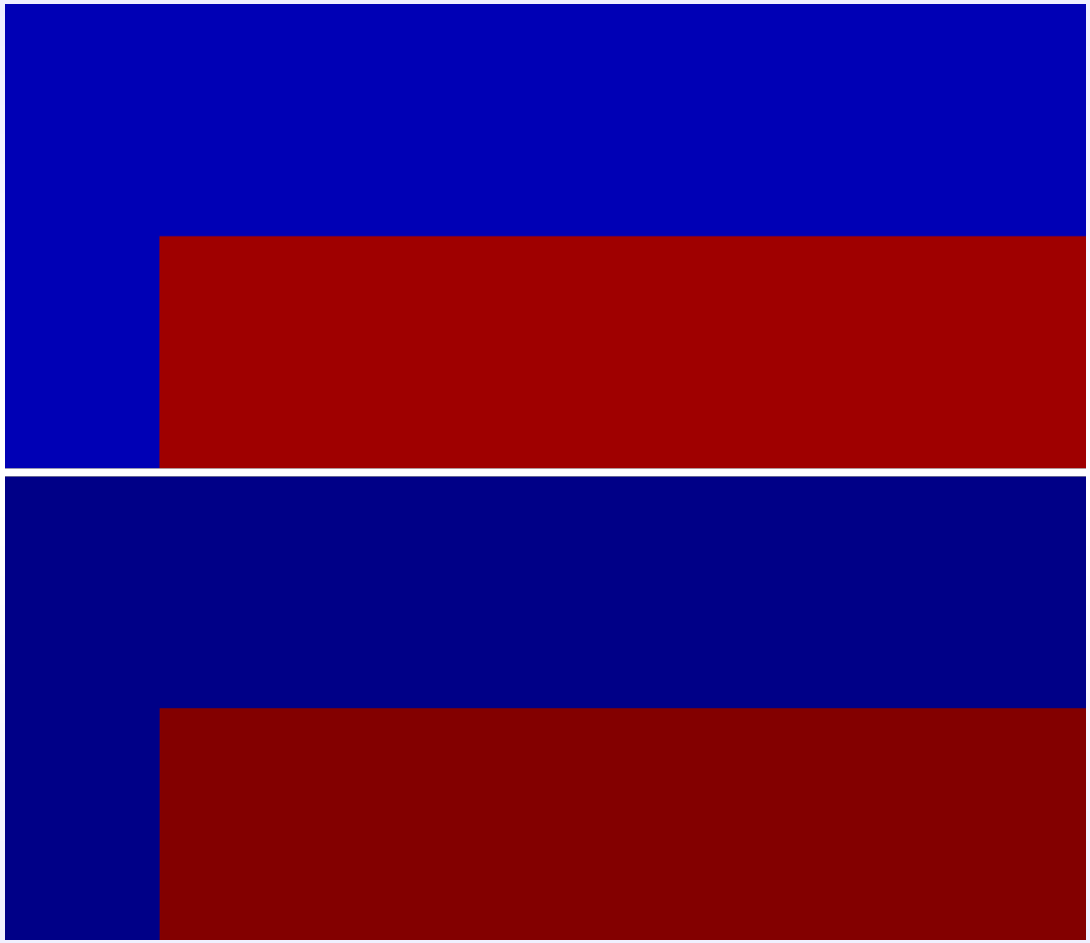
xRAGE
1680x720
Density

$\longrightarrow$ 4^{th}-order method reveals small structures on a 9$\times$ coarser mesh!
(~13.5$\times$ faster)

Triple Point - Interface diffusion comparison

MOOD-P3
1120x480
Vol. Fraction

xRAGE
1120x480
Vol. Fraction



$\longrightarrow$ Limiting interface diffusion in a 2^{nd}-order method does not have the same effect than increasing the method order of accuracy.

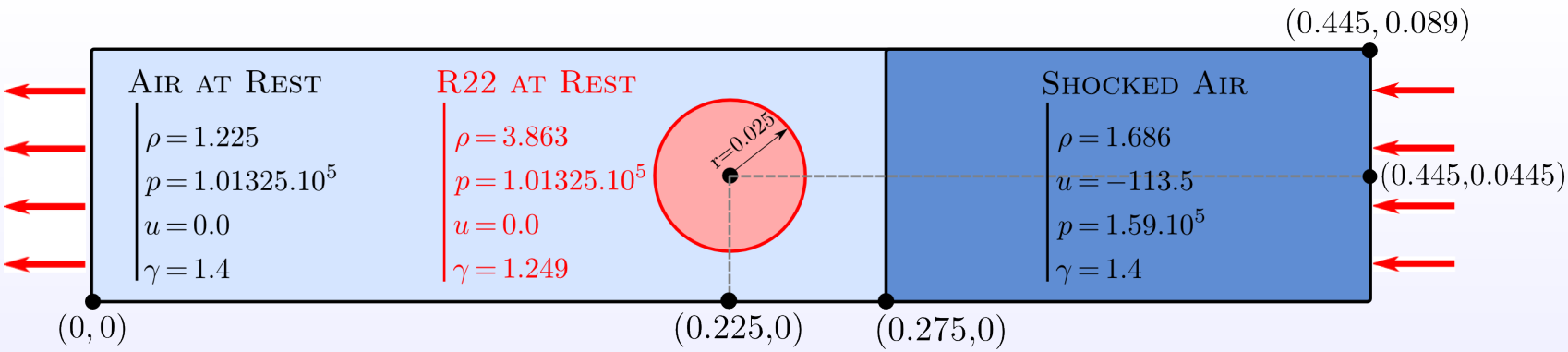
R22 Bubble - Initialization

Experiment

- J.F.Haas, B.Sturtevant. "Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities"

Simulations

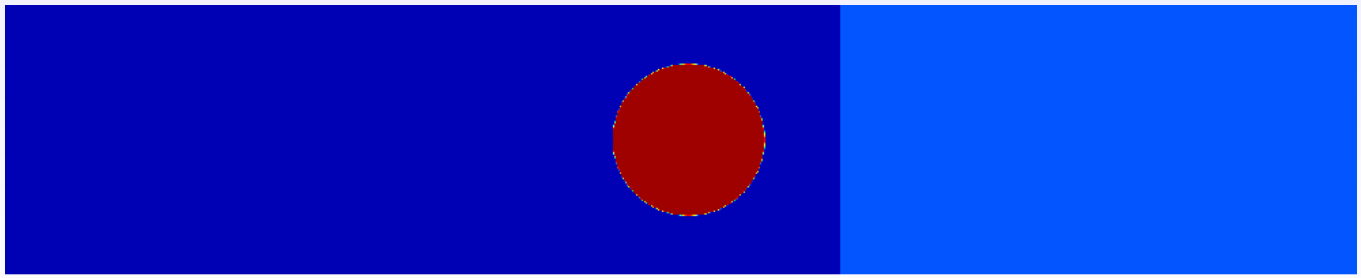
- J.J.Quirk, S.Karni. "On the dynamics of a shock-bubble interaction"
- S.Kokh, F.Lagoutière. "An anti-diffusive numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model"



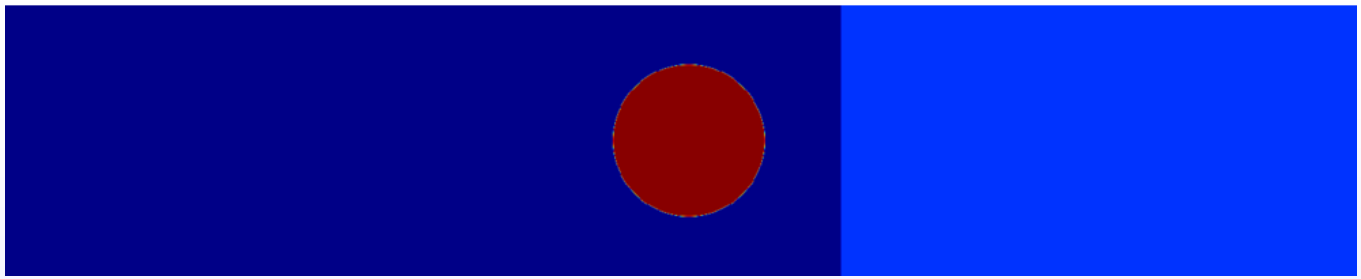
- Final time = 1020.0 μs
- Wall boundary conditions on top/bottom

R22 Bubble - 2^{nd}-order comparison

MOOD-P1
1600x320
Density



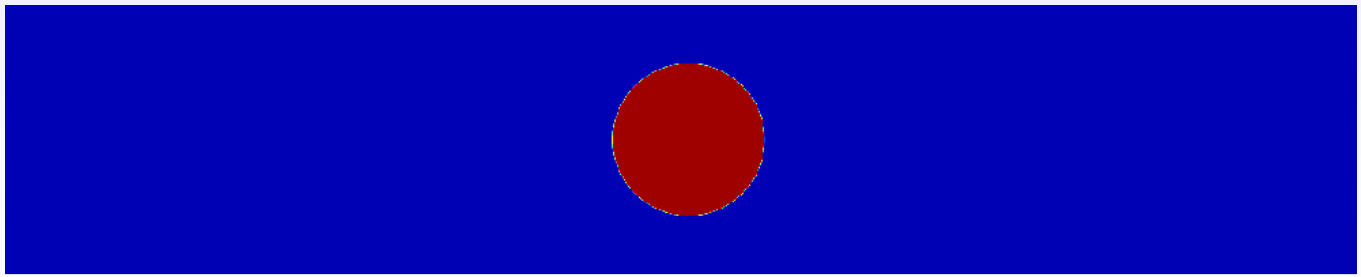
xRAGE
1600x320
Density



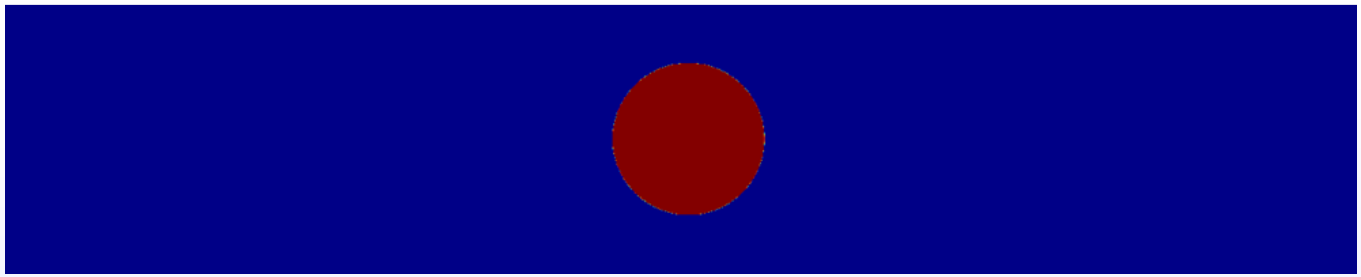
$\longrightarrow$ 2^{nd}-order results are qualitatively equivalent (validation)

R22 Bubble - 4^{th}-order *vs* 2^{nd}-order

MOOD-P3
1200x240
Vol. Fraction



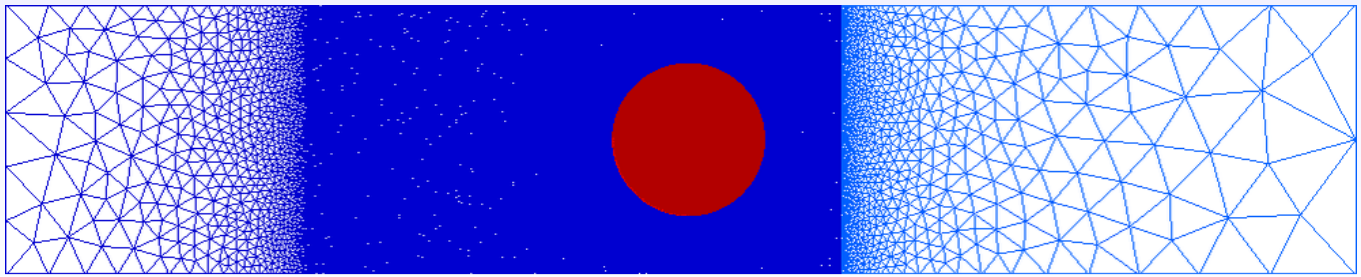
xRAGE
3200x640
Vol. Fraction



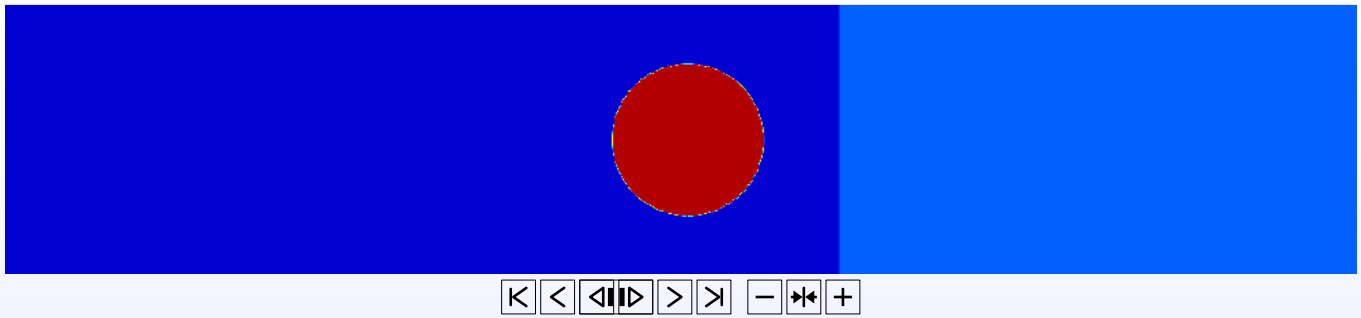
$\longrightarrow$ 4^{th}-order method reveals small structures on a 7$\times$ coarser mesh!
(~9.5$\times$ faster)

R22 Bubble - Structured *vs* Unstructured

MOOD-P3
Unstructured
101923 cells
$h > 5.0 \times 10^{-4}$
Density



MOOD-P3
Structured
288000 cells
$h = 3.7 \times 10^{-4}$
Density



$\longrightarrow$ The ability to reveal structures is *not affected* by unstructured meshes

Conclusion & Future work

Conclusion

- First extension of the MOOD method to a 6-eq. multi-material model
- Developed for unstructured meshes with diffuse interface treatment
- Better efficiency of MOOD-$\mathbb{P}_3$ (4^{th}-order) *vs* MUSCL (2^{nd}-order)
- MOOD-$\mathbb{P}_3$ reveals the small structures *earlier* (space/time)

Future work

- Develop a 2D axisymmetric version of the research code
- Quantitatively assess the relevancy of the 4^{th}-order method
- Demonstrate the positivity-preserving property of the MOOD method
- Develop a MOOD direct Eulerian VOF method on unstructured meshes

Thanks for your attention!

[a]**Steven Diot**, [a]Marianne François, [b]Edward Dendy.

[a]Fluid Dynamics and Solid Mechanics (T-3)
[b]Computational Physics and Methods (CCS-2)
Los Alamos National Laboratory, Los Alamos, NM, USA.

http://public.lanl.gov/diot

Extra Slides

Polynomial reconstruction

Form of polynomial reconstruction of degree d

$$\widetilde{U}_i(\mathbf{x}) = U_i^n + \sum_{1 \leq |\boldsymbol{\alpha}| \leq d} \mathcal{R}_{\boldsymbol{\alpha}} \left((\mathbf{x} - \mathbf{c_i})^{\boldsymbol{\alpha}} - \frac{1}{|K_i|} \int_{K_i} (\mathbf{x} - \mathbf{c_i})^{\boldsymbol{\alpha}} d\mathbf{x} \right)$$

- $\boldsymbol{\alpha} \in \mathbb{R}^m$ multiindex $\Longrightarrow$ covers all monomials
- $\mathcal{R}_{\boldsymbol{\alpha}} \in \mathbb{R}$ unknown poly. coefficients s.t. $\#(\mathcal{R}) = \frac{\prod_{i=1}^{m}(d+i)}{m!} - 1$

Computation of $\mathcal{R}_{\boldsymbol{\alpha}}$

- Approximations of mean values on a local stencil
- Local stencil contains more than $\#(\mathcal{R})$ cells
- Resolution of an overdetermined linear system $\equiv$ *Least-Squares*

Polynomial reconstruction

Form of polynomial reconstruction of degree d

$$\widetilde{U}_i(\mathbf{x}) = U_i^n + \sum_{1 \leq |\boldsymbol{\alpha}| \leq d} \mathcal{R}_{\boldsymbol{\alpha}} \left((\mathbf{x} - \mathbf{c_i})^{\boldsymbol{\alpha}} - \frac{1}{|K_i|} \int_{K_i} (\mathbf{x} - \mathbf{c_i})^{\boldsymbol{\alpha}} d\mathbf{x} \right)$$

- $\boldsymbol{\alpha} \in \mathbb{R}^m$ multiindex $\Longrightarrow$ covers all monomials
- $\mathcal{R}_{\boldsymbol{\alpha}} \in \mathbb{R}$ unknown poly. coefficients s.t. $\#(\mathcal{R}) = \frac{\prod_{i=1}^{m}(d+i)}{m!} - 1$

Computation of $\mathcal{R}_{\boldsymbol{\alpha}}$

- Approximations of mean values on a local stencil
- Local stencil contains more than $\#(\mathcal{R})$ cells
- Resolution of an overdetermined linear system $\equiv$ *Least-Squares*

Polynomial reconstruction

Mean value of $\widetilde{U}_i^n$ *on a* K_j

$$U_i^n + \sum_{1 \leq |\boldsymbol{\alpha}| \leq d} \mathcal{R}_{\boldsymbol{\alpha}} \left(\frac{1}{|K_j|} \int_{K_j} (\mathbf{x} - \mathbf{c_i})^{\boldsymbol{\alpha}} d\mathbf{x} - \frac{1}{|K_i|} \int_{K_i} (\mathbf{x} - \mathbf{c_i})^{\boldsymbol{\alpha}} d\mathbf{x} \right)$$

Packing in a Matrix-vector form

$$\begin{pmatrix} \bar{\mathbf{x}}_1^{(1,0,0)} & \bar{\mathbf{x}}_1^{(0,1,0)} & \bar{\mathbf{x}}_1^{(0,0,1)} & \cdots & \bar{\mathbf{x}}_1^{(0,0,d)} \\ \bar{\mathbf{x}}_2^{(1,0,0)} & \bar{\mathbf{x}}_2^{(0,1,0)} & \bar{\mathbf{x}}_2^{(0,0,1)} & \cdots & \bar{\mathbf{x}}_2^{(0,0,d)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{\mathbf{x}}_N^{(1,0,0)} & \bar{\mathbf{x}}_N^{(0,1,0)} & \bar{\mathbf{x}}_N^{(0,0,1)} & \cdots & \bar{\mathbf{x}}_N^{(0,0,d)} \end{pmatrix} \begin{pmatrix} \mathcal{R}_{(1,0,0)} \\ \mathcal{R}_{(0,1,0)} \\ \mathcal{R}_{(0,0,1)} \\ \vdots \\ \mathcal{R}_{(0,0,d)} \end{pmatrix} = \begin{pmatrix} U_1^n - U_i^n \\ U_2^n - U_i^n \\ \vdots \\ U_N^n - U_i^n \end{pmatrix}$$

$$A\mathcal{R} = \bar{U}$$

Polynomial reconstruction

Mean value of $\widetilde{U}_i^n$ *on a* K_j

$$U_i^n + \sum_{1 \leq |\boldsymbol{\alpha}| \leq d} \mathcal{R}_{\boldsymbol{\alpha}} \left(\qquad\qquad \bar{\mathbf{x}}_j^{\boldsymbol{\alpha}} \qquad\qquad \right)$$

Packing in a Matrix-vector form

$$\begin{pmatrix} \bar{\mathbf{x}}_1^{(1,0,0)} & \bar{\mathbf{x}}_1^{(0,1,0)} & \bar{\mathbf{x}}_1^{(0,0,1)} & \cdots & \bar{\mathbf{x}}_1^{(0,0,d)} \\ \bar{\mathbf{x}}_2^{(1,0,0)} & \bar{\mathbf{x}}_2^{(0,1,0)} & \bar{\mathbf{x}}_2^{(0,0,1)} & \cdots & \bar{\mathbf{x}}_2^{(0,0,d)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{\mathbf{x}}_N^{(1,0,0)} & \bar{\mathbf{x}}_N^{(0,1,0)} & \bar{\mathbf{x}}_N^{(0,0,1)} & \cdots & \bar{\mathbf{x}}_N^{(0,0,d)} \end{pmatrix} \begin{pmatrix} \mathcal{R}_{(1,0,0)} \\ \mathcal{R}_{(0,1,0)} \\ \mathcal{R}_{(0,0,1)} \\ \vdots \\ \mathcal{R}_{(0,0,d)} \end{pmatrix} = \begin{pmatrix} U_1^n - U_i^n \\ U_2^n - U_i^n \\ \vdots \\ U_N^n - U_i^n \end{pmatrix}$$

$$A\mathcal{R} = \bar{U}$$

Polynomial reconstruction

Idea for resolution

$$A\mathcal{R} = \bar{U} \Leftrightarrow A^t A\mathcal{R} = A^t \bar{U} \Leftrightarrow \mathcal{R} = (A^t A)^{-1} A^t \bar{U} \Leftrightarrow \mathcal{R} = A^\dagger \bar{U}$$

How to get $A^\dagger$

- Using a QR decomposition of A:
  - $A = QR$, with $Q^t = Q^{-1}$ and R triangular superior

$$A^\dagger = ((QR)^t (QR))^{-1} A^t \Rightarrow A^\dagger = (R^t R)^{-1} A^t$$

- Other possibilities: SVD, etc.
- Pre-processing: pseudo-inverse matrices are stored

Polynomial reconstruction

Idea for resolution

$$A\mathcal{R} = \bar{U} \Leftrightarrow A^t A\mathcal{R} = A^t\bar{U} \Leftrightarrow \mathcal{R} = (A^tA)^{-1}A^t\bar{U} \Leftrightarrow \mathcal{R} = A^\dagger\bar{U}$$

How to get $A^\dagger$

- Using a QR decomposition of A:
  - $A = QR$, with $Q^t = Q^{-1}$ and R triangular superior

$$A^\dagger = ((QR)^t(QR))^{-1}A^t \Rightarrow A^\dagger = (R^tR)^{-1}A^t$$

- Other possibilities: SVD, etc.
- Pre-processing: pseudo-inverse matrices are stored

Polynomial reconstruction

Idea for resolution

$$A\mathcal{R} = \bar{U} \Leftrightarrow A^t A\mathcal{R} = A^t \bar{U} \Leftrightarrow \mathcal{R} = (A^t A)^{-1} A^t \bar{U} \Leftrightarrow \mathcal{R} = A^\dagger \bar{U}$$

How to get $A^\dagger$

- Using a QR decomposition of A:
  - $A = QR$, with $Q^t = Q^{-1}$ and R triangular superior

$$A^\dagger = ((QR)^t (QR))^{-1} A^t \Rightarrow A^\dagger = (R^t R)^{-1} A^t$$

- Other possibilities: SVD, etc.
- Pre-processing: pseudo-inverse matrices are stored

Polynomial reconstruction

Idea for resolution

$$A\mathcal{R} = \bar{U} \Leftrightarrow A^t A\mathcal{R} = A^t \bar{U} \Leftrightarrow \mathcal{R} = (A^t A)^{-1} A^t \bar{U} \Leftrightarrow \mathcal{R} = A^\dagger \bar{U}$$

How to get $A^\dagger$

- Using a QR decomposition of A:
  - $A = QR$, with $Q^t = Q^{-1}$ and R triangular superior

$$A^\dagger = ((QR)^t (QR))^{-1} A^t \Rightarrow A^\dagger = (R^t R)^{-1} A^t$$

- Other possibilities: SVD, etc.
- Pre-processing: pseudo-inverse matrices are stored

Polynomial reconstruction

Idea for resolution

$$A\mathcal{R} = \bar{U} \Leftrightarrow A^t A\mathcal{R} = A^t \bar{U} \Leftrightarrow \mathcal{R} = (A^t A)^{-1} A^t \bar{U} \Leftrightarrow \mathcal{R} = A^\dagger \bar{U}$$

How to get $A^\dagger$

- Using a QR decomposition of A:
  - $A = QR$, with $Q^t = Q^{-1}$ and R triangular superior

$$A^\dagger = ((QR)^t (QR))^{-1} A^t \Rightarrow A^\dagger = (R^t R)^{-1} A^t$$

- Other possibilities: SVD, etc.
- Pre-processing: pseudo-inverse matrices are stored

Runge-Kutta 3 TVD

Higher-order discretization in time

- The spatial high-order Finite Volume scheme

$$U_h^{n+1} = U_h^n + \Delta t \mathcal{H}^R(U_h^n), \quad U_h^n = \sum_{i,\ \text{cells}} U_i^n \mathbb{1}_{\Omega_i}$$

Abstract form that includes relax. step and energies correction.

- The 3^{rd}-order Runge–Kutta time discretization

$$\left.\begin{array}{rcl} U_h^{(1)} & = & U_h^n + \Delta t\ \mathcal{H}^R\left(U_h^n\right) \\ U_h^{(2)} & = & U_h^{(1)} + \Delta t\ \mathcal{H}^R(U_h^{(1)}) \\ U_h^{(3)} & = & \bar{U}_h^{(2)} + \Delta t\ \mathcal{H}^R(\bar{U}_h^{(2)}) \end{array}\right\} \Longrightarrow U_h^{n+1} = \frac{U_h^n + 2U_h^{(3)}}{3}$$

where $\bar{U}_h^{(2)} = \dfrac{3U_h^n + U_h^{(2)}}{4}$.

$\Longrightarrow$ MOOD is performed on each substep!

HLLC Riemann Solver

Waves speeds estimates

$$\begin{aligned} S_L &= \min(u_L - c_L, u_R - c_R), \\ S_R &= \max(u_L + c_L, u_R + c_R), \\ S^\star &= \frac{p_R - p_L + (\rho u)_L(S_L - u_L) - (\rho u)_R(S_R - u_R)}{\rho_L(S_L - u_L) - \rho_R(S_R - u_R)}. \end{aligned}$$

The HLLC is given by

$$\mathbb{F}_{HLLC}(U_L, U_R) = \begin{cases} F_L & \text{if} \quad 0 \le S_L, \\ F_L + S_L(U_L^\star - U_L) & \text{if} \quad S_L \le 0 \le S^\star, \\ F_R + S_R(U_R^\star - U_R) & \text{if} \quad S^\star \le 0 \le S_R, \\ F_R & \text{if} \quad 0 \ge S_R, \end{cases}$$

where for $K = \{L, R\}$, F_K is the physical flux and

$$U_K^\star = \left(\frac{S_K - u_K}{S_K - S^\star}\right) \begin{pmatrix} (\alpha_1\rho_1)_K \\ (\alpha_2\rho_2)_K \\ \rho_K S^\star \\ \rho_K E_K + \rho_K(S^\star - u_K)\left(S^\star + \dfrac{p_K}{\rho_K(S_K - u_K)}\right) \end{pmatrix}$$

HLLC Riemann Solver

Additional variables

Volume fractions *(constant along trajectories)*

$$(\alpha_k)^\star_K = (\alpha_k)_K$$

Material densities

$$(\rho_k)^\star_K = (\rho_k)_K \left(\frac{S_K - u_K}{S_K - S^\star} \right)$$

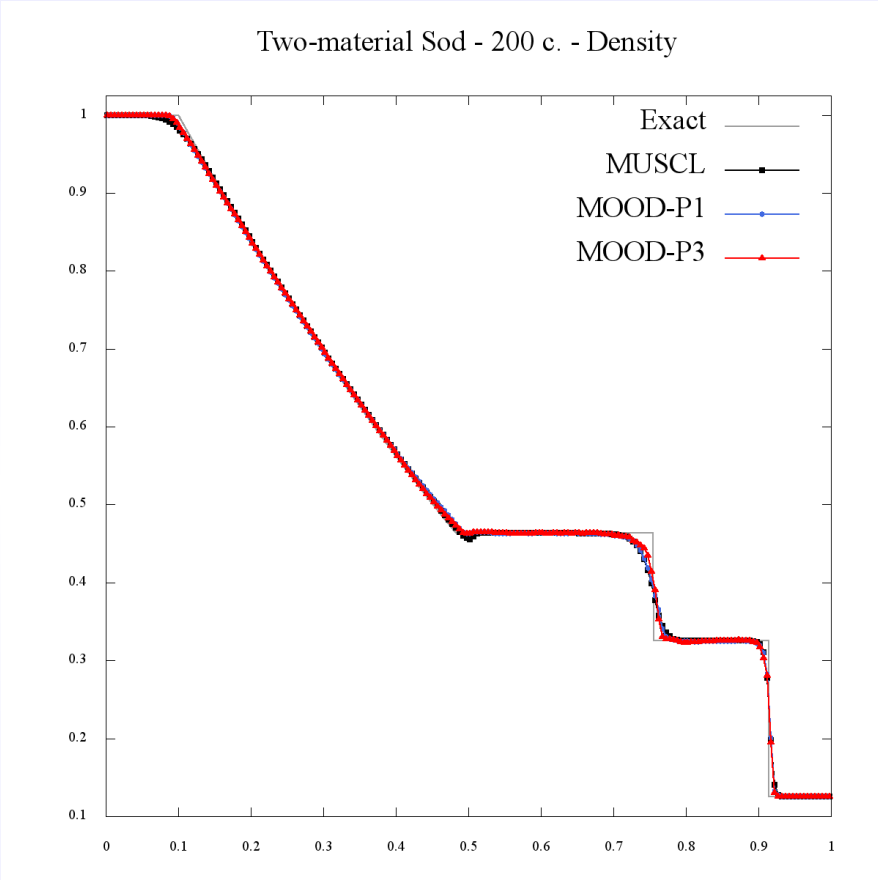
Material pressures

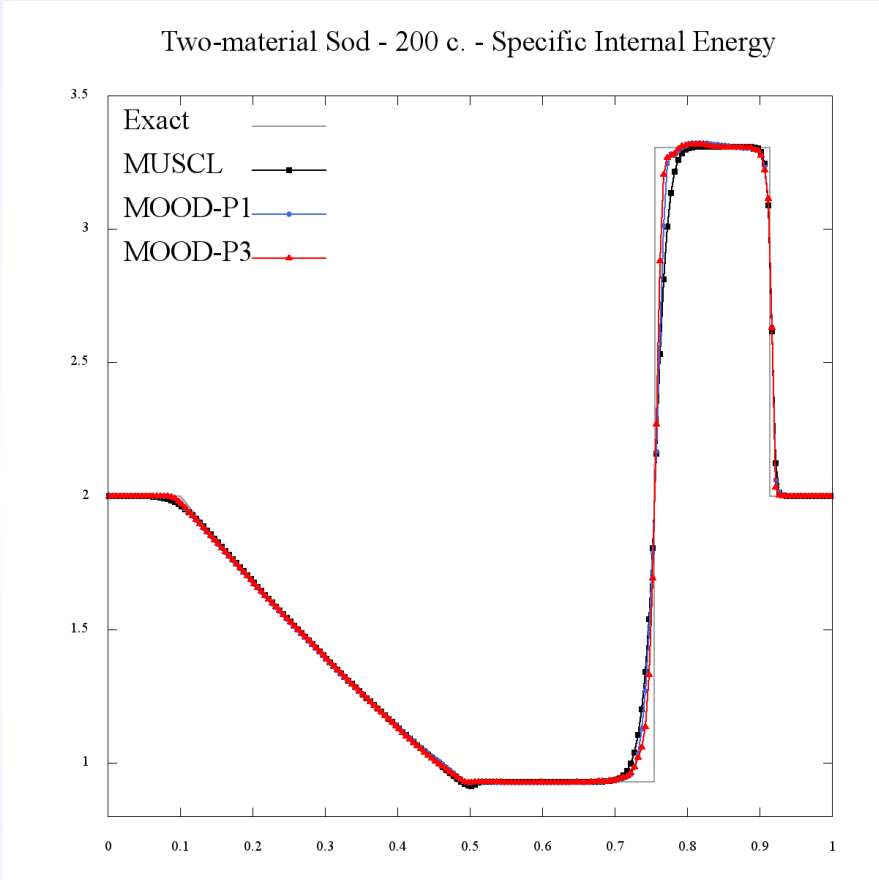
$$(p_k)^\star_K = \Big((p_k)_K + \pi_k \Big) \frac{(\gamma_k - 1)(\rho_k)_K - (\gamma_k + 1)(\rho_k)^\star_K}{(\gamma_k - 1)(\rho_k)^\star_K - (\gamma_k + 1)(\rho_k)_K} - \pi_k$$

Material internal energies

$$(e_k)^\star_K = (e_k)^\star_K \Big((\rho_k)^\star_K, (p_k)^\star_K \Big)$$

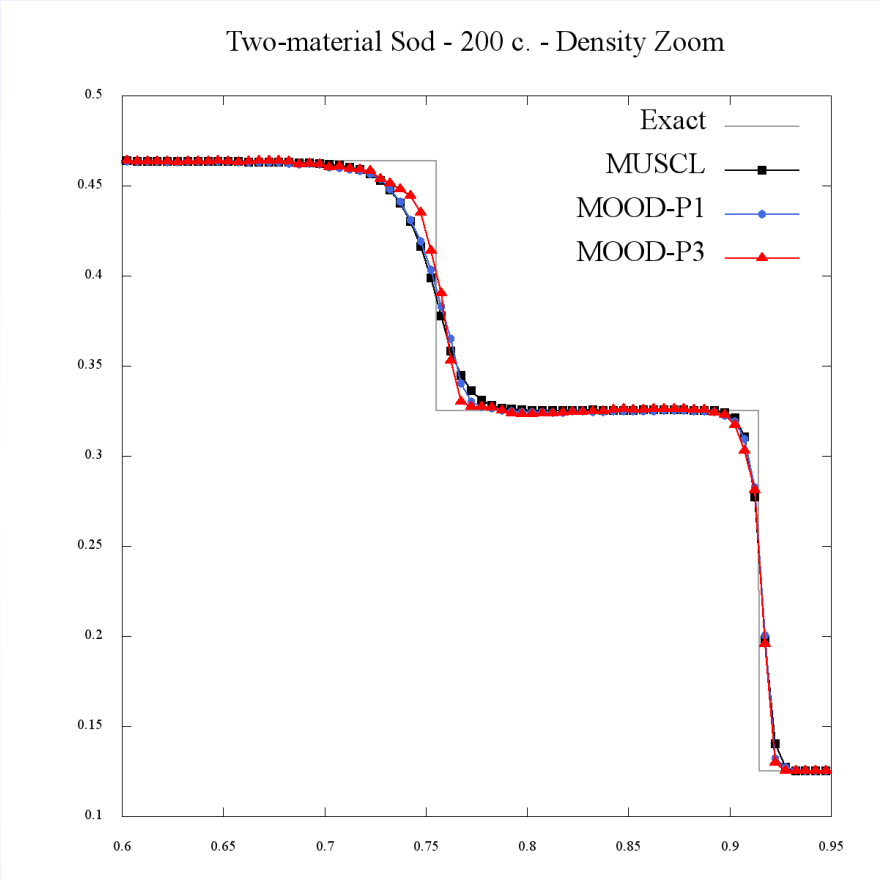
2-material Sod - 200 cells - Solution

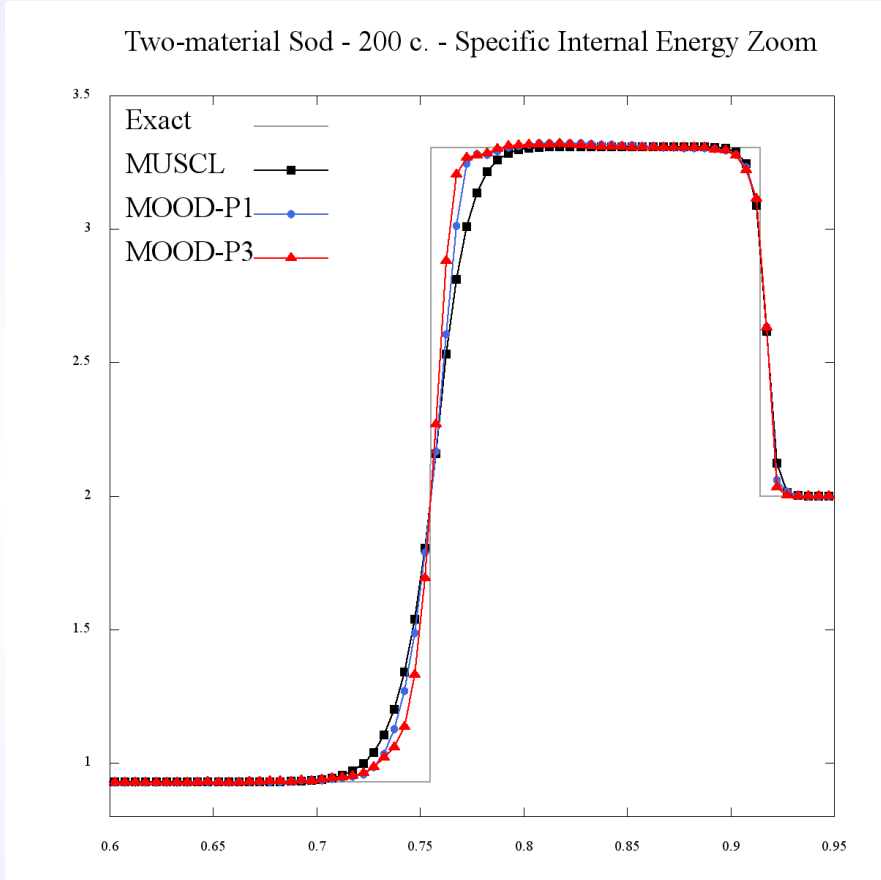




$t_{end} = 0.2s$	ρ	u	p	γ	π	Boundary Cond.
Left State on $[0.0; 0.5]$	1	0	2	2.0	0	Reflective at x=0.0
Right State on $[0.5; 1.0]$	0.125	0	1	1.4	0	Reflective at x=1.0

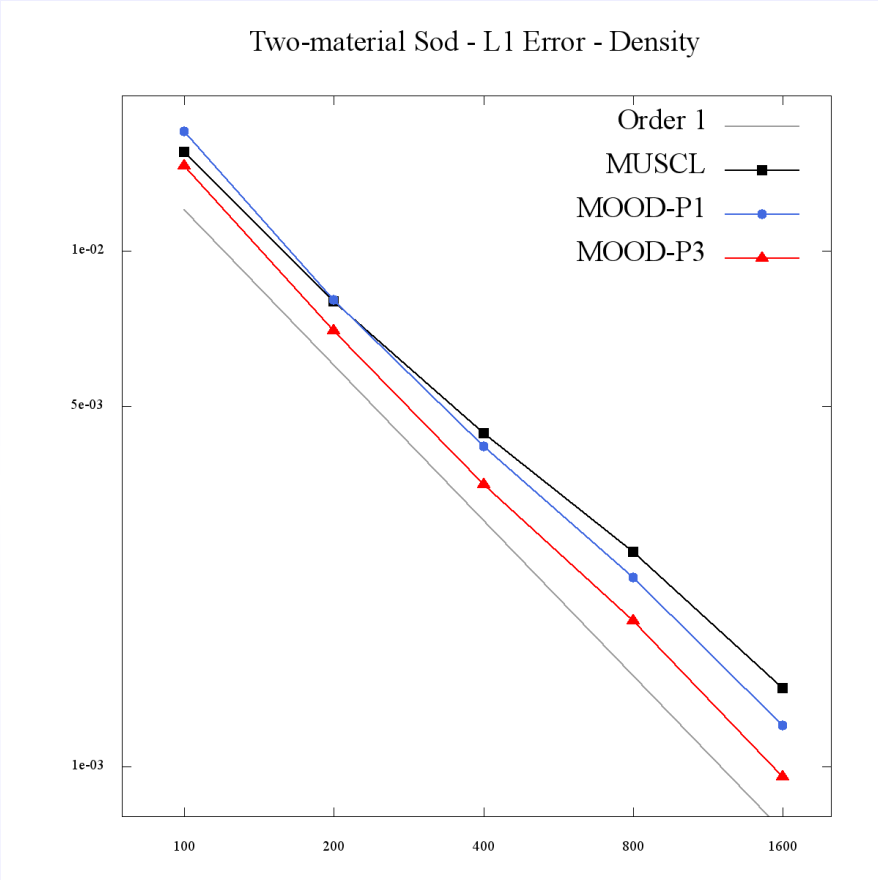
2-material Sod - 200 cells - Solution

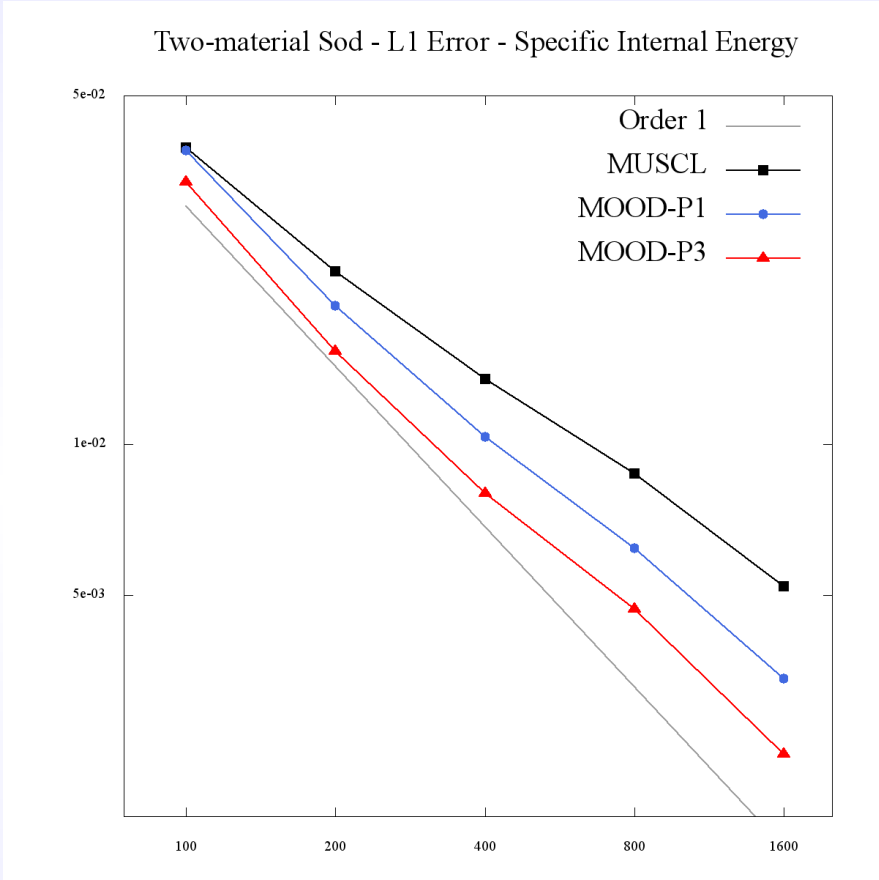




$t_{end} = 0.2s$	ρ	u	p	γ	π	Boundary Cond.
Left State on $[0.0; 0.5]$	1	0	2	2.0	0	Reflective at x=0.0
Right State on $[0.5; 1.0]$	0.125	0	1	1.4	0	Reflective at x=1.0

2-material Sod - 200 cells - Convergence



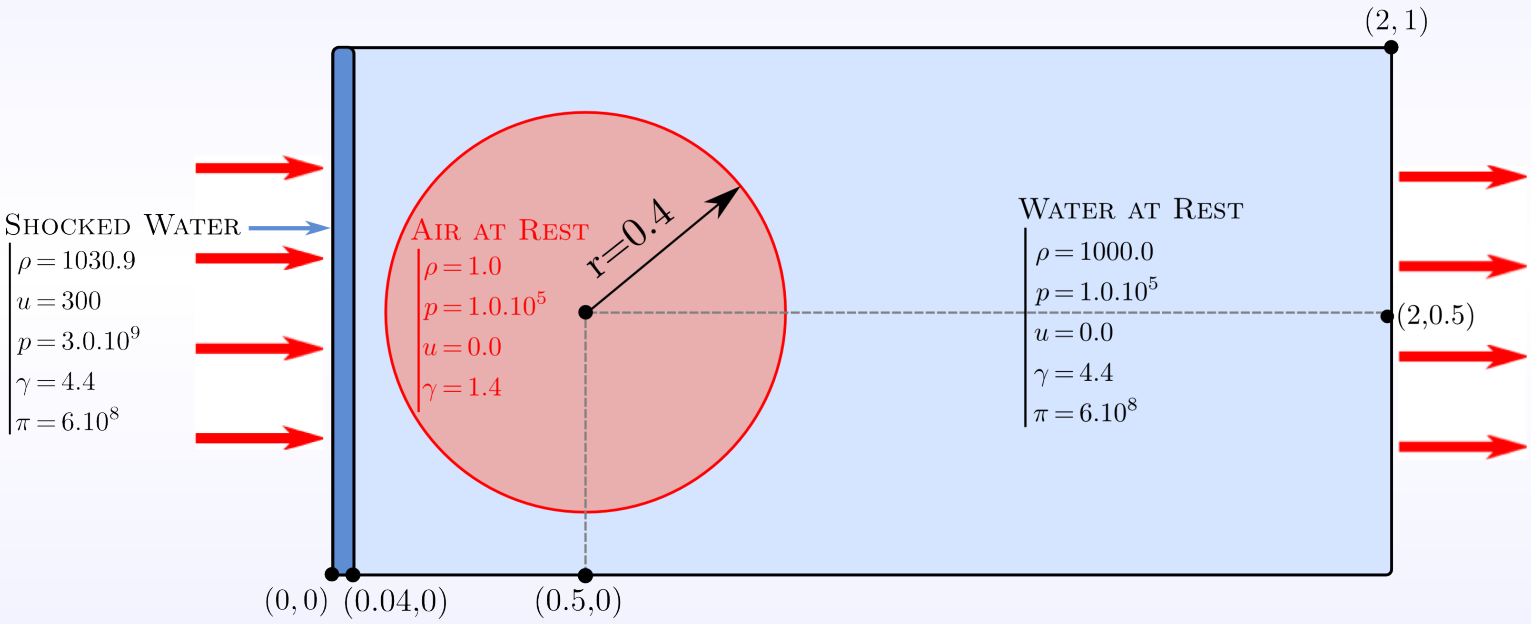


1600 cells	MUSCL Rel.Err.	MUSCL CPU time	MOOD-ℙ1 Rel.Err.	MOOD-ℙ1 CPU time	MOOD-ℙ3 Rel.Err.	MOOD-ℙ3 CPU time
Density	1	10s.	1/1.2	15s. (14.4s.)	1/2.2	27s. (19.6s.)
Sp.Int.E.	1	10s.	1/1.5	15s. (22.5s.)	1/2.7	27s. (48.4s.)

Estimated time for MUSCL to reach same error assuming a perfect order 1

Air Bubble - Initialization

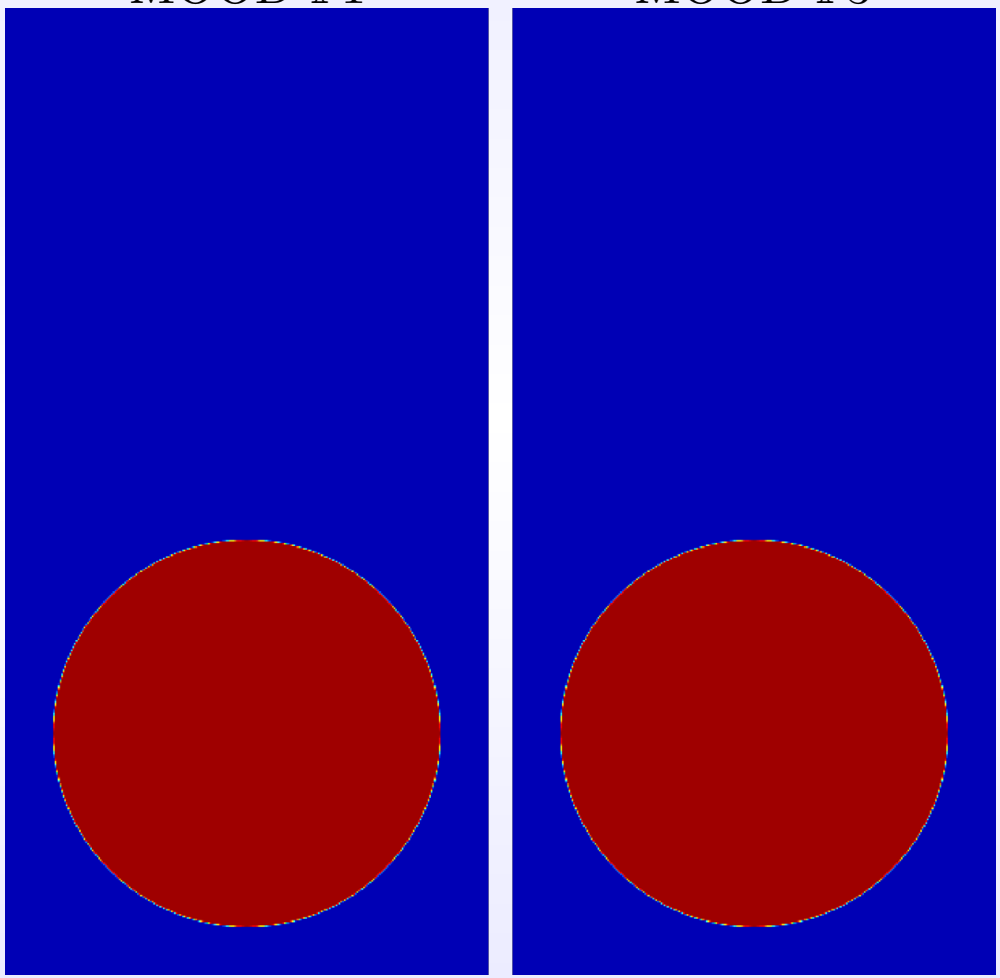
- S.Kokh, F.Lagoutière. "An anti-diffusive numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model"



- Final time = 800.0 μs
- Wall boundary conditions on top/bottom

Air Bubble - 600x300 - Volume Fraction

MOOD-$\mathbb{P}1$ MOOD-$\mathbb{P}3$



Air Bubble - 600x300 - Density

MOOD-$\mathbb{P}1$ MOOD-$\mathbb{P}3$