

Wave propagation methods for hyperbolic problems on mapped grids

Keh-Ming Shyue

Department of Mathematics
National Taiwan University
Taiwan

Outline

- Review a high-resolution **wave propagation** method for solving hyperbolic problems on mapped grids (which is basic integration scheme implemented in CLAWPACK)
- Describe an **interpolation-type adaptive moving mesh** approach that is an easy generalization of the above method for improving numerical resolution
- Show sample results

Mapped grid methods

To begin with, we consider system of conservation laws

$$\partial_t \mathbf{q} + \nabla \cdot \mathbf{f}(\mathbf{q}) = 0 \quad (1)$$

with suitable initial condition for \mathbf{q} in $N_d \geq 1$ spatial domain

Here $\mathbf{q} \in \mathbb{R}^m$ denotes vector of m conserved quantities,

$\mathbf{f} = (f_1, f_2, \dots, f_{N_d}) \in \mathbb{R}^{m \times N_d}$ denotes flux matrix; $f_j \in \mathbb{R}^m$

It is known integral form of (1) over any control volume C is

$$\frac{d}{dt} \int_C \mathbf{q} \, d\mathbf{x} = - \int_{\partial C} \mathbf{f}(\mathbf{q}) \cdot \mathbf{n} \, ds,$$

where \mathbf{n} is outward-pointing normal vector at boundary ∂C

Mapped grid methods

Using above integral conservation law, a finite volume method on a control volume C can be written as

$$Q^{n+1} = Q^n - \frac{\Delta t}{\mathcal{M}(C)} \sum_{j=1}^{N_s} h_j \check{F}_j,$$

where $\mathcal{M}(C)$ is measure (**area** in 2D or **volume** in 3D) of C ,

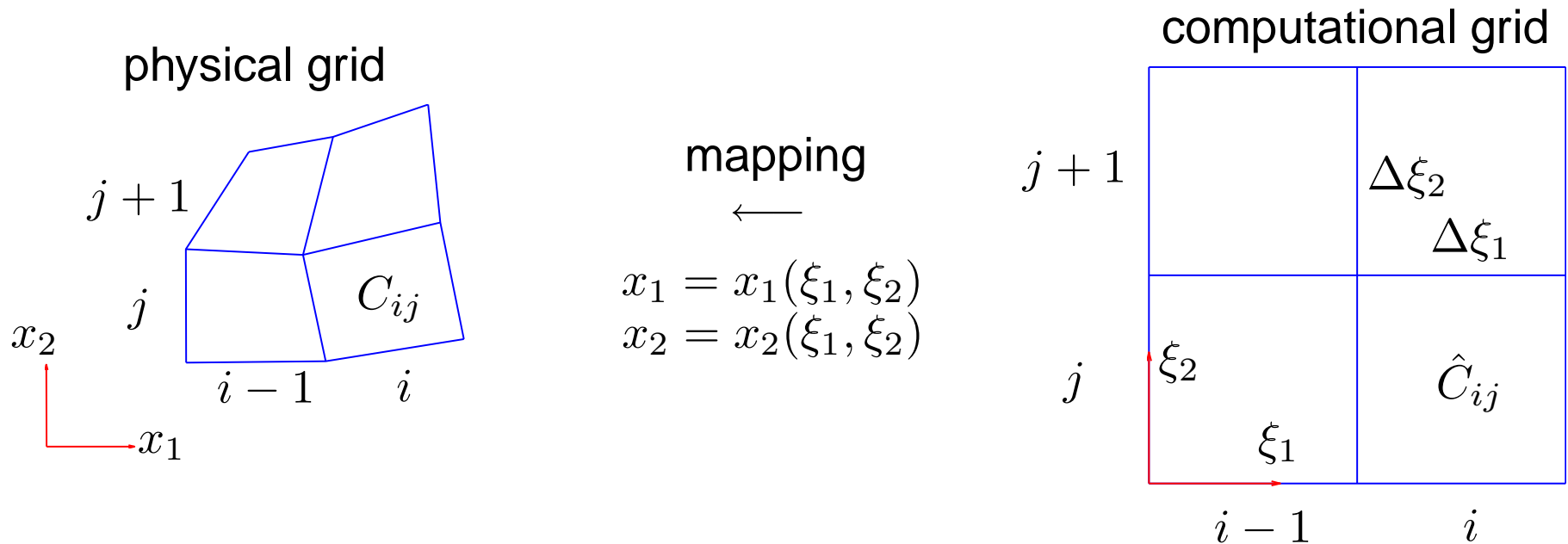
N_s is number of **sides**,

h_j is **length** of j -th side (in 2D) or **area** of cell edge (in 3D) measured in physical space,

\check{F}_j is numerical approximation to **normal flux in average** across j -th side of grid cell

Mapped grid methods

In the following we assume that our **mapped grids** are **logically rectangular**, & will restrict our consideration to two-dimensional case $N_d = 2$ as illustrated below



Then on a curvilinear grid, a finite volume method takes

$$Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\kappa_{ij} \Delta \xi_1} \left(F_{i+\frac{1}{2},j}^1 - F_{i-\frac{1}{2},j}^1 \right) - \frac{\Delta t}{\kappa_{ij} \Delta \xi_2} \left(F_{i,j+\frac{1}{2}}^2 - F_{i,j-\frac{1}{2}}^2 \right)$$

Mapped grid methods

On a curvilinear grid, a finite volume method takes

$$Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\kappa_{ij} \Delta \xi_1} \left(F_{i+\frac{1}{2},j}^1 - F_{i-\frac{1}{2},j}^1 \right) - \frac{\Delta t}{\kappa_{ij} \Delta \xi_2} \left(F_{i,j+\frac{1}{2}}^2 - F_{i,j-\frac{1}{2}}^2 \right)$$

where $\Delta \xi_1, \Delta \xi_2$ denote **equidistant discretization** of computational domain,

$\kappa_{ij} = \mathcal{M}(C_{ij}) / \Delta \xi_1 \Delta \xi_2$ is **area ratio** between area of grid cell in physical space & area of a computational grid,

$F_{i-\frac{1}{2},j}^1 = \gamma_{i-\frac{1}{2},j} \check{F}_{i-\frac{1}{2},j}$, $F_{i,j-\frac{1}{2}}^2 = \gamma_{i,j-\frac{1}{2}} \check{F}_{i,j-\frac{1}{2}}$ are fluxes per unit length in computational space with $\gamma_{i-\frac{1}{2},j} = h_{i-\frac{1}{2},j} / \Delta \xi_1$ &

$\gamma_{i,j-\frac{1}{2}} = h_{i,j-\frac{1}{2}} / \Delta \xi_2$ representing **length ratios**

Mapped grid methods

In this setup, **first order wave propagation** method is a **Godunov-type** finite volume method that takes form

$$Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\kappa_{ij} \Delta \xi_1} \left(\mathcal{A}_1^+ \Delta Q_{i-\frac{1}{2},j} + \mathcal{A}_1^- \Delta Q_{i+\frac{1}{2},j} \right) - \frac{\Delta t}{\kappa_{ij} \Delta \xi_2} \left(\mathcal{A}_2^+ \Delta Q_{i,j-\frac{1}{2}} + \mathcal{A}_2^- \Delta Q_{i,j+\frac{1}{2}} \right)$$

with right-, left-, up-, and down-moving fluctuations $\mathcal{A}_1^+ \Delta Q_{i-\frac{1}{2},j}$, $\mathcal{A}_1^- \Delta Q_{i+\frac{1}{2},j}$, $\mathcal{A}_2^+ \Delta Q_{i,j-\frac{1}{2}}$, & $\mathcal{A}_2^- \Delta Q_{i,j+\frac{1}{2}}$ that are entering into grid cell

To **determine** these **fluctuations**, we need to **solve** one-dimensional **Riemann problems** normal to cell edges

Computing fluctuations

Let $\mathbf{n}_{i-\frac{1}{2},j} = (n^1, n^2)$ & $\mathbf{t}_{i-\frac{1}{2},j} = (t^1, t^2)$ be normalized normal and tangential vectors to cell edge $(i - \frac{1}{2}, j)$ between cells $(i - 1, j)$ & (i, j) . Then rotation matrix which rotates velocity components (e.g., for **shallow water** or **acoustic** equations) has form

$$\mathcal{R}_{i-\frac{1}{2},j} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & n^1 & n^2 \\ 0 & t^1 & t^2 \end{pmatrix}$$

Fluctuations $\mathcal{A}_1^\pm \Delta Q_{i-\frac{1}{2},j}$ can be calculated by steps:

1. Determine \check{Q}_L & \check{Q}_R by rotating velocity components,

$$\check{Q}_L = \mathcal{R}_{i-\frac{1}{2},j} Q_{i-1,j}, \quad \check{Q}_R = \mathcal{R}_{i-\frac{1}{2},j} Q_{i,j}$$

Computing fluctuations

2. Solve Riemann problem for 1D conservation law

$$\partial_t \mathbf{q} + \nabla \cdot \mathbf{f}_1 = 0$$

with initial data

$$q(\xi_1, 0) = \begin{cases} \check{Q}_L & \text{if } x < x_{i-\frac{1}{2}} \\ \check{Q}_R & \text{if } x > x_{i-\frac{1}{2}} \end{cases}$$

This results in **waves** $\check{W}_{i-\frac{1}{2},j}^{1,k}$ that are moving with **speeds** $\check{\lambda}_{i-\frac{1}{2},j}^{1,k}$, $k = 1, \dots, N_w$, N_w is the number of waves. Here jumps are decomposed into waves as $\check{Q}_R - \check{Q}_L = \sum_{k=1}^{N_w} \check{W}_{i-\frac{1}{2},j}^{1,k}$ & solution at cell edge can be expressed

$$Q_{i-\frac{1}{2},j}^* = \check{Q}_L + \sum_{k:\lambda^{1,k} < 0} \check{W}_{i-\frac{1}{2},j}^{1,k}$$

Computing fluctuations

3. Define **scaled speeds**

$$\lambda_{i-\frac{1}{2},j}^{1,k} = \gamma_{i-\frac{1}{2},j} \check{\lambda}_{i-\frac{1}{2},j}^{1,k}$$

and **rotate waves** back to Cartesian coordinates by

$$\mathcal{W}_{i-\frac{1}{2},j}^{1,k} = \mathcal{R}_{i-\frac{1}{2},j}^T \check{\mathcal{W}}_{i-\frac{1}{2},j}^{1,k}, \quad k = 1, \dots, N_w$$

4. Determine left- and right-moving fluctuations in the form

$$\mathcal{A}_1^\pm \Delta Q_{i-\frac{1}{2},j} = \sum_{k=1}^{N_w} \left(\lambda_{i-\frac{1}{2},j}^{1,k} \right)^\pm \mathcal{W}_{i-\frac{1}{2},j}^{1,k}$$

5. In an analogous way up and down-moving fluctuations at cell edge $(i, j - \frac{1}{2})$ can be calculated

High resolution corrections

The **speeds** and **limited** versions of **waves** are used to calculate **second order** correction terms. These terms are added to the method in flux difference form as

$$Q_{ij}^{n+1} := Q_{ij}^{n+1} - \frac{1}{\kappa_{ij}} \frac{\Delta t}{\Delta \xi_1} \left(\tilde{\mathcal{F}}_{i+\frac{1}{2},j}^1 - \tilde{\mathcal{F}}_{i-\frac{1}{2},j}^1 \right) - \frac{1}{\kappa_{ij}} \frac{\Delta t}{\Delta \xi_2} \left(\tilde{\mathcal{F}}_{i,j+\frac{1}{2}}^2 - \tilde{\mathcal{F}}_{i,j-\frac{1}{2}}^2 \right)$$

For example, at cell edge $(i - \frac{1}{2}, j)$ correction flux takes

$$\tilde{\mathcal{F}}_{i-\frac{1}{2},j}^1 = \frac{1}{2} \sum_{k=1}^{N_w} \left| \lambda_{i-\frac{1}{2},j}^{1,k} \right| \left(1 - \frac{\Delta t}{\kappa_{i-\frac{1}{2},j} \Delta \xi_1} \left| \lambda_{i-\frac{1}{2},j}^{1,k} \right| \right) \tilde{\mathcal{W}}_{i-\frac{1}{2},j}^{1,k}$$

where $\kappa_{i-\frac{1}{2},j} = (\kappa_{i-1,j} + \kappa_{ij})/2$. To avoid oscillations near discontinuities, a **wave limiter** is applied leading to limited waves $\tilde{\mathcal{W}}$

High resolution corrections

To ensure second order accuracy & also improve stability, a **transverse wave propagation** is included in the algorithm that left- & right-going fluctuations $\mathcal{A}_1^\pm \Delta Q_{i-\frac{1}{2},j}$ are each split into two transverse fluctuations: up- & down-going $\mathcal{A}_2^\pm \mathcal{A}_1^+ \Delta Q_{i-\frac{1}{2},j}$ & $\mathcal{A}_2^\pm \mathcal{A}_1^- \Delta Q_{i-\frac{1}{2},j}$

This wave propagation method can be shown to be **conservative** & **stable** under a variant of CFL (Courant-Friedrichs-Lewy) condition of form

$$\nu = \Delta t \max_{i,j,k} \left(\frac{|\lambda_{i-\frac{1}{2},j}^{1,k}|}{J_{i_p,j} \Delta \xi_1}, \frac{|\lambda_{i,j-\frac{1}{2}}^{2,k}|}{J_{i,j_p} \Delta \xi_2} \right) \leq 1,$$

where $i_p = i$ if $\lambda_{i-\frac{1}{2},j}^{1,k} > 0$ & $i - 1$ if $\lambda_{i-\frac{1}{2},j}^{1,k} < 0$

Accuracy test in 2D

- Consider 2D compressible Euler equations with ideal gas law as governing equations
- Take smooth vortex flow with initial condition

$$\rho = \left(1 - \frac{25(\gamma - 1)}{8\gamma\pi^2} \exp(1 - r^2) \right)^{1/(\gamma-1)},$$

$$p = \rho^\gamma,$$

$$u_1 = 1 - \frac{5}{2\pi} \exp((1 - r^2)/2) (x_2 - 5),$$

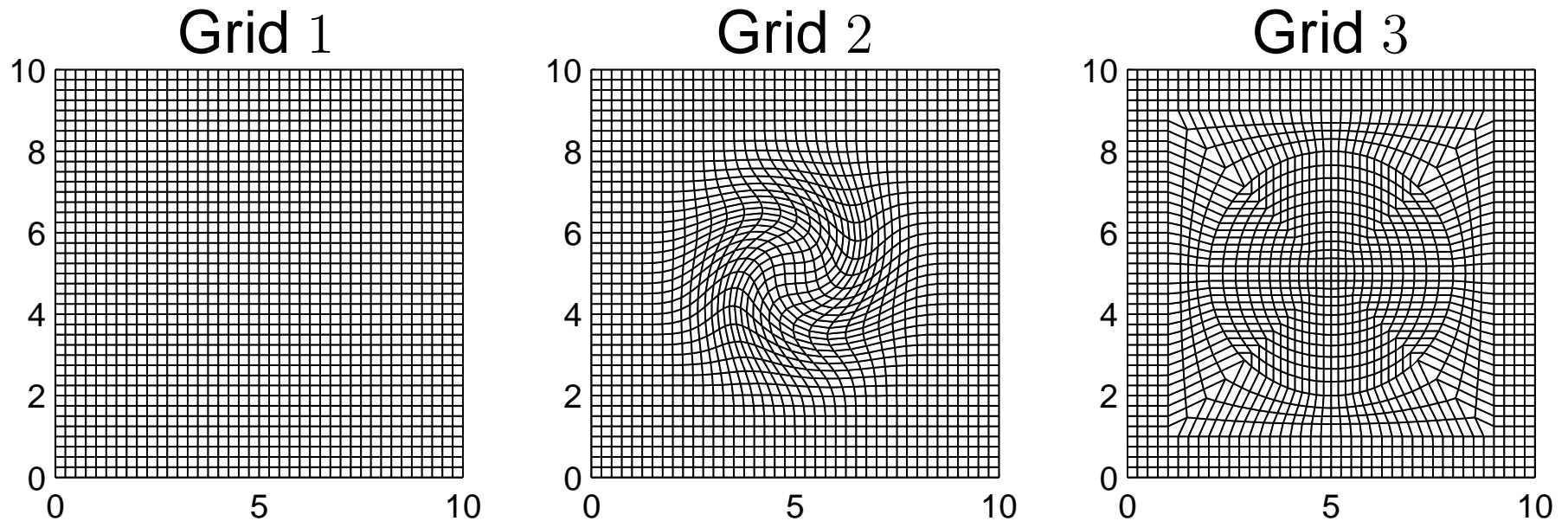
$$u_2 = 1 + \frac{5}{2\pi} \exp((1 - r^2)/2) (x_1 - 5),$$

& periodic boundary conditions as an example, where

$$r = \sqrt{(x_1 - 5)^2 + (x_2 - 5)^2}$$

Accuracy test in 2D

- Grids used for this smooth vortex flow test



- $\|\mathcal{E}_z\|_{1,\infty} = \|z_{\text{comput}} - z_{\text{exact}}\|_{1,\infty}$ denotes discrete 1- or maximum-norm error for state variable z
- Results shown below are at time $t = 10$ on $N \times N$ mesh

Accuracy results in 2D: Grid 1

N	$\mathcal{E}_1(\rho)$	Order	$\mathcal{E}_1(u_1)$	Order	$\mathcal{E}_1(u_2)$	Order	$\mathcal{E}_1(p)$	Order
40	0.6673		2.3443		1.7121		0.8143	
80	0.1792	1.90	0.6194	1.92	0.4378	1.97	0.2128	1.94
160	0.0451	1.99	0.1537	2.01	0.1104	1.99	0.0536	1.99
320	0.0113	2.00	0.0384	2.00	0.0276	2.00	0.0134	2.00

N	$\mathcal{E}_\infty(\rho)$	Order	$\mathcal{E}_\infty(u_1)$	Order	$\mathcal{E}_\infty(u_2)$	Order	$\mathcal{E}_\infty(p)$	Order
40	0.1373		0.3929		0.1810		0.1742	
80	0.0377	1.87	0.1014	1.95	0.0502	1.85	0.0482	1.85
160	0.0093	2.02	0.0248	2.03	0.0123	2.03	0.0119	2.02
320	0.0022	2.07	0.0062	2.00	0.0030	2.04	0.0029	2.04

Accuracy results in 2D: Grid 2

N	$\mathcal{E}_1(\rho)$	Order	$\mathcal{E}_1(u_1)$	Order	$\mathcal{E}_1(u_2)$	Order	$\mathcal{E}_1(p)$	Order
40	0.9298		2.6248		2.1119		1.2104	
80	0.2643	1.81	0.7258	1.85	0.5296	2.00	0.3277	1.89
160	0.0674	1.97	0.1833	1.99	0.1309	2.02	0.0845	1.96
320	0.0169	2.00	0.0458	2.00	0.0327	2.00	0.0212	1.99

N	$\mathcal{E}_\infty(\rho)$	Order	$\mathcal{E}_\infty(u_1)$	Order	$\mathcal{E}_\infty(u_2)$	Order	$\mathcal{E}_\infty(p)$	Order
40	0.1676		0.4112		0.2259		0.2111	
80	0.0471	1.83	0.1242	1.73	0.0645	1.79	0.0586	1.85
160	0.0126	1.91	0.0333	1.90	0.0162	2.02	0.0149	1.97
320	0.0033	1.93	0.0085	1.97	0.0040	2.00	0.0038	1.98

Accuracy results in 2D: Grid 3

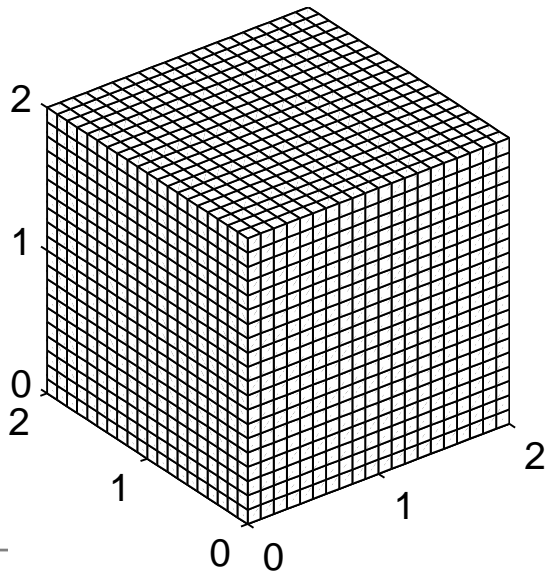
N	$\mathcal{E}_1(\rho)$	Order	$\mathcal{E}_1(u_1)$	Order	$\mathcal{E}_1(u_2)$	Order	$\mathcal{E}_1(p)$	Order
40	4.8272		4.7734		5.3367		5.4717	
80	1.5740	1.62	1.5633	1.61	1.5660	1.77	1.5634	1.81
160	0.4536	1.79	0.4559	1.78	0.4537	1.79	0.4560	1.78
320	0.1215	1.90	0.1221	1.90	0.1222	1.89	0.1221	1.90

N	$\mathcal{E}_\infty(\rho)$	Order	$\mathcal{E}_\infty(u_1)$	Order	$\mathcal{E}_\infty(u_2)$	Order	$\mathcal{E}_\infty(p)$	Order
40	0.4481		0.4475		0.4765		0.4817	
80	0.1170	1.94	0.1181	1.92	0.1196	1.99	0.1191	2.02
160	0.0434	1.43	0.0431	1.45	0.0442	1.43	0.0440	1.44
320	0.0117	1.89	0.0119	1.86	0.0119	1.89	0.0118	1.89

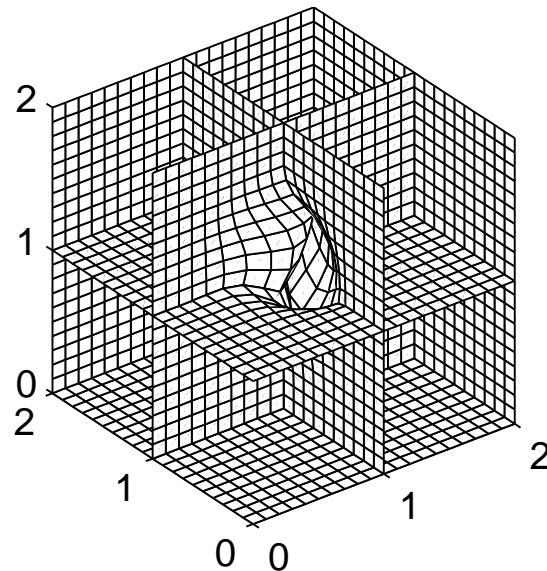
Accuracy test in 3D

- Consider 3D compressible Euler equations with ideal gas law as governing equations
- Take smooth radially-symmetric flow with the flow condition that is at rest initially with density $\rho(r) = 1 + \exp(-30(r - 1)^2)/10$ & pressure $p(r) = \rho^\gamma$
- Grids used for smooth radially-symmetric flow test

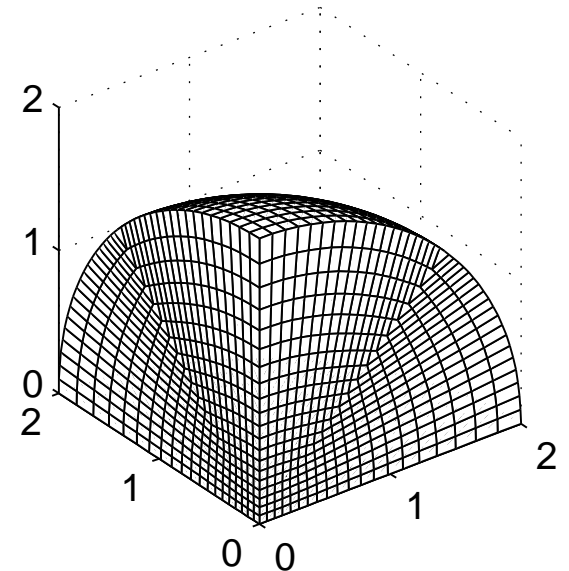
Grid 1



Grid 2



Grid 3



Accuracy results in 3D: Grid 1

N	$\mathcal{E}_1(\rho)$	Order	$\mathcal{E}_1(\vec{u})$	Order	$\mathcal{E}_1(p)$	Order
20	$7.227 \cdot 10^{-3}$		$8.920 \cdot 10^{-3}$		$1.019 \cdot 10^{-2}$	
40	$2.418 \cdot 10^{-3}$	1.58	$2.558 \cdot 10^{-3}$	1.80	$3.415 \cdot 10^{-3}$	1.58
80	$6.356 \cdot 10^{-4}$	1.93	$6.754 \cdot 10^{-4}$	1.92	$8.980 \cdot 10^{-4}$	1.93
160	$1.616 \cdot 10^{-4}$	1.98	$1.718 \cdot 10^{-4}$	1.97	$2.282 \cdot 10^{-4}$	1.98

N	$\mathcal{E}_\infty(\rho)$	Order	$\mathcal{E}_\infty(\vec{u})$	Order	$\mathcal{E}_\infty(p)$	Order
20	$1.096 \cdot 10^{-2}$		$1.200 \cdot 10^{-2}$		$1.569 \cdot 10^{-2}$	
40	$4.085 \cdot 10^{-3}$	1.42	$4.381 \cdot 10^{-3}$	1.45	$5.848 \cdot 10^{-3}$	1.42
80	$1.235 \cdot 10^{-3}$	1.73	$1.263 \cdot 10^{-3}$	1.79	$1.765 \cdot 10^{-3}$	1.73
160	$3.517 \cdot 10^{-4}$	1.81	$3.349 \cdot 10^{-4}$	1.91	$5.030 \cdot 10^{-4}$	1.81

Accuracy results in 3D: Grid 2

N	$\mathcal{E}_1(\rho)$	Order	$\mathcal{E}_1(\vec{u})$	Order	$\mathcal{E}_1(p)$	Order
20	$7.227 \cdot 10^{-3}$		$8.920 \cdot 10^{-3}$		$1.019 \cdot 10^{-2}$	
40	$2.418 \cdot 10^{-3}$	1.58	$2.558 \cdot 10^{-3}$	1.80	$3.415 \cdot 10^{-3}$	1.58
80	$6.356 \cdot 10^{-4}$	1.93	$6.754 \cdot 10^{-4}$	1.92	$8.980 \cdot 10^{-4}$	1.93
160	$1.616 \cdot 10^{-4}$	1.98	$1.718 \cdot 10^{-4}$	1.97	$2.282 \cdot 10^{-4}$	1.98

N	$\mathcal{E}_\infty(\rho)$	Order	$\mathcal{E}_\infty(\vec{u})$	Order	$\mathcal{E}_\infty(p)$	Order
20	$7.227 \cdot 10^{-3}$		$8.920 \cdot 10^{-3}$		$1.019 \cdot 10^{-2}$	
40	$2.418 \cdot 10^{-3}$	1.58	$2.558 \cdot 10^{-3}$	1.80	$3.415 \cdot 10^{-3}$	1.58
80	$6.356 \cdot 10^{-4}$	1.93	$6.754 \cdot 10^{-4}$	1.92	$8.980 \cdot 10^{-4}$	1.93
160	$1.616 \cdot 10^{-4}$	1.98	$1.718 \cdot 10^{-4}$	1.97	$2.282 \cdot 10^{-4}$	1.98

Accuracy results in 3D: Grid 3

N	$\mathcal{E}_1(\rho)$	Order	$\mathcal{E}_1(\vec{u})$	Order	$\mathcal{E}_1(p)$	Order
20	$1.290 \cdot 10^{-2}$		$1.641 \cdot 10^{-2}$		$1.816 \cdot 10^{-2}$	
40	$4.694 \cdot 10^{-3}$	1.46	$4.999 \cdot 10^{-3}$	1.71	$6.623 \cdot 10^{-3}$	1.46
80	$1.257 \cdot 10^{-3}$	1.90	$1.379 \cdot 10^{-3}$	1.86	$1.774 \cdot 10^{-3}$	1.90
160	$3.209 \cdot 10^{-4}$	1.97	$3.546 \cdot 10^{-4}$	1.96	$4.527 \cdot 10^{-4}$	1.97

N	$\mathcal{E}_\infty(\rho)$	Order	$\mathcal{E}_\infty(\vec{u})$	Order	$\mathcal{E}_\infty(p)$	Order
20	$1.632 \cdot 10^{-2}$		$1.984 \cdot 10^{-2}$		$2.316 \cdot 10^{-2}$	
40	$5.819 \cdot 10^{-3}$	1.49	$6.745 \cdot 10^{-3}$	1.56	$8.307 \cdot 10^{-3}$	1.48
80	$1.823 \cdot 10^{-3}$	1.67	$4.290 \cdot 10^{-3}$	0.65	$2.710 \cdot 10^{-3}$	1.67
160	$5.053 \cdot 10^{-4}$	1.85	$3.271 \cdot 10^{-3}$	0.39	$7.237 \cdot 10^{-4}$	1.85

Extension to moving mesh

One simple way to extend mapped grid method described above to solution adaptive moving grid method is to take approach proposed by

- H. Tang & T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal., 2003

In each time step, this moving mesh method consists of three basic steps:

1. **Mesh redistribution**
2. **Conservative interpolation** of solution state
3. **Solution update** on a fixed mapped grid

Mesh redistribution scheme

- **Winslow's** approach (1981)

$$\text{Solve } \nabla \cdot (D \nabla \xi_j) = 0, \quad j = 1, \dots, N_d$$

for $\xi(\mathbf{x})$. Coefficient D is a **positive definite matrix** which may depend on solution gradient

- **Variational** approach (Tang & many others)

$$\text{Solve } \nabla_{\xi} \cdot (D \nabla_{\xi} x_j) = 0, \quad j = 1, \dots, N_d$$

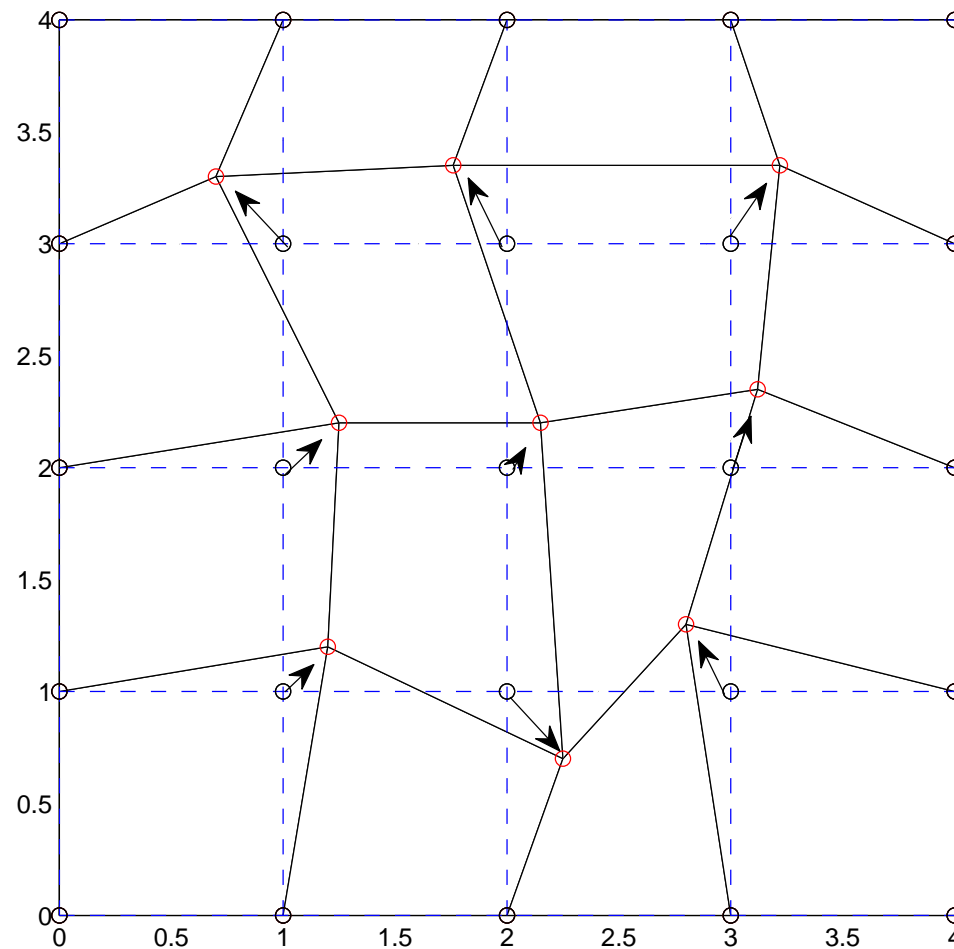
for $\mathbf{x}(\xi)$ that minimizes the “energy” functional

$$\mathcal{E}(\mathbf{x}(\xi)) = \frac{1}{2} \int_{\Omega} \sum_{j=1}^{N_d} \nabla_{\xi}^T D \nabla x_j d\xi$$

- **Lagrangian** (ALE)-type approach (e.g., CAVEAT code)

Mesh redistribution

Dashed lines represent **initial mesh** & **solid lines** represent **new mesh** after a redistribution step



Conservative interpolation

Numerical solutions need to be updated conservatively, *i.e.*

$$\sum \mathcal{M}(C^{k+1}) Q^{k+1} = \sum \mathcal{M}(C^k) Q^k$$

after each mesh redistribution iterate k . This can be done

- Finite-volume approach (Tang & Tang, SIAM 03)

$$\mathcal{M}(C^{k+1}) Q^{k+1} = \mathcal{M}(C^k) Q^k - \sum_{j=1}^{N_s} h_j \check{G}_j, \quad \check{G} = (\dot{\mathbf{x}} \cdot \mathbf{n}) Q$$

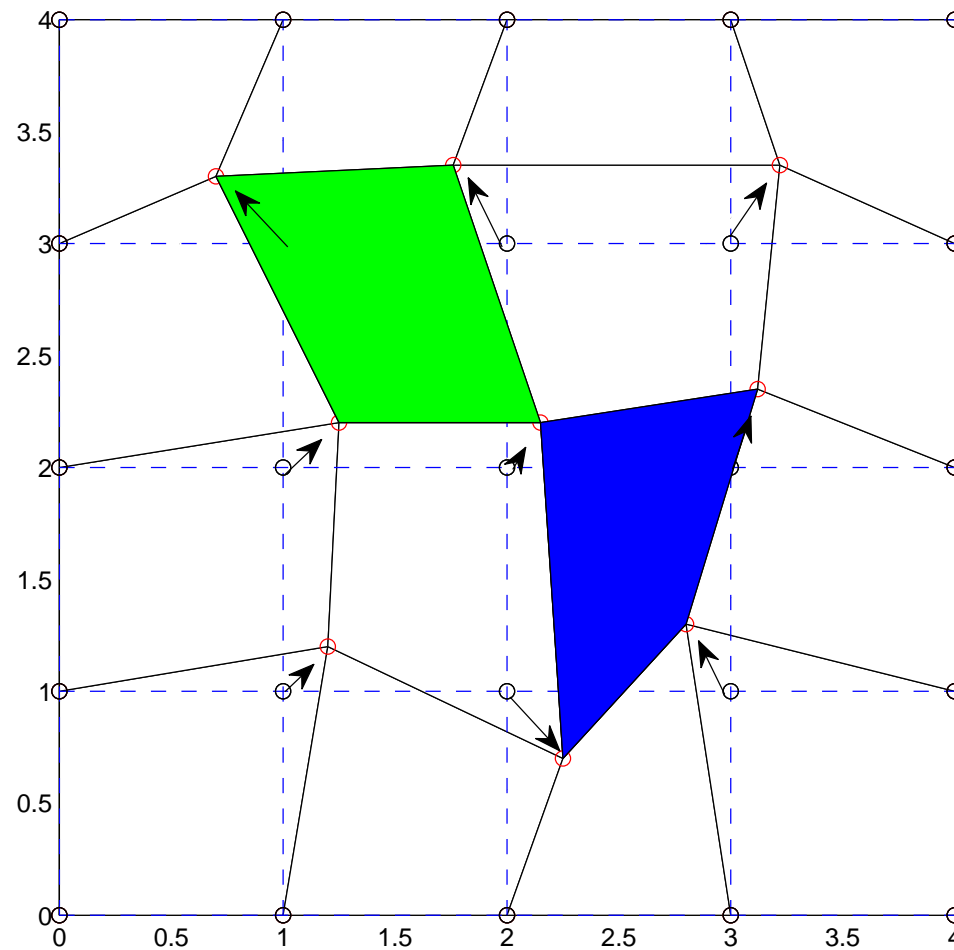
- Geometric approach (Shyue 2010 & others)

$$\left[\sum_S \mathcal{M}(C_p^{k+1} \cap S_p^k) \right] Q_C^{k+1} = \sum_S \mathcal{M}(C_p^{k+1} \cap S_p^k) Q_S^k$$

C_p, S_p are polygonal regions occupied by cells C & S

Conservative interpolation

Geometric way to perform conservative interpolation



Interpolation-free moving mesh

One way to derive an **interpolation-free** moving mesh method is to consider coordinate change of equations via $(\mathbf{x}, t) \mapsto (\xi, t)$. In this case, the transformed conservation law reads

$$\partial_t \tilde{\mathbf{q}} + \nabla_{\xi} \cdot \tilde{\mathbf{f}} = \mathcal{G} \quad (2)$$

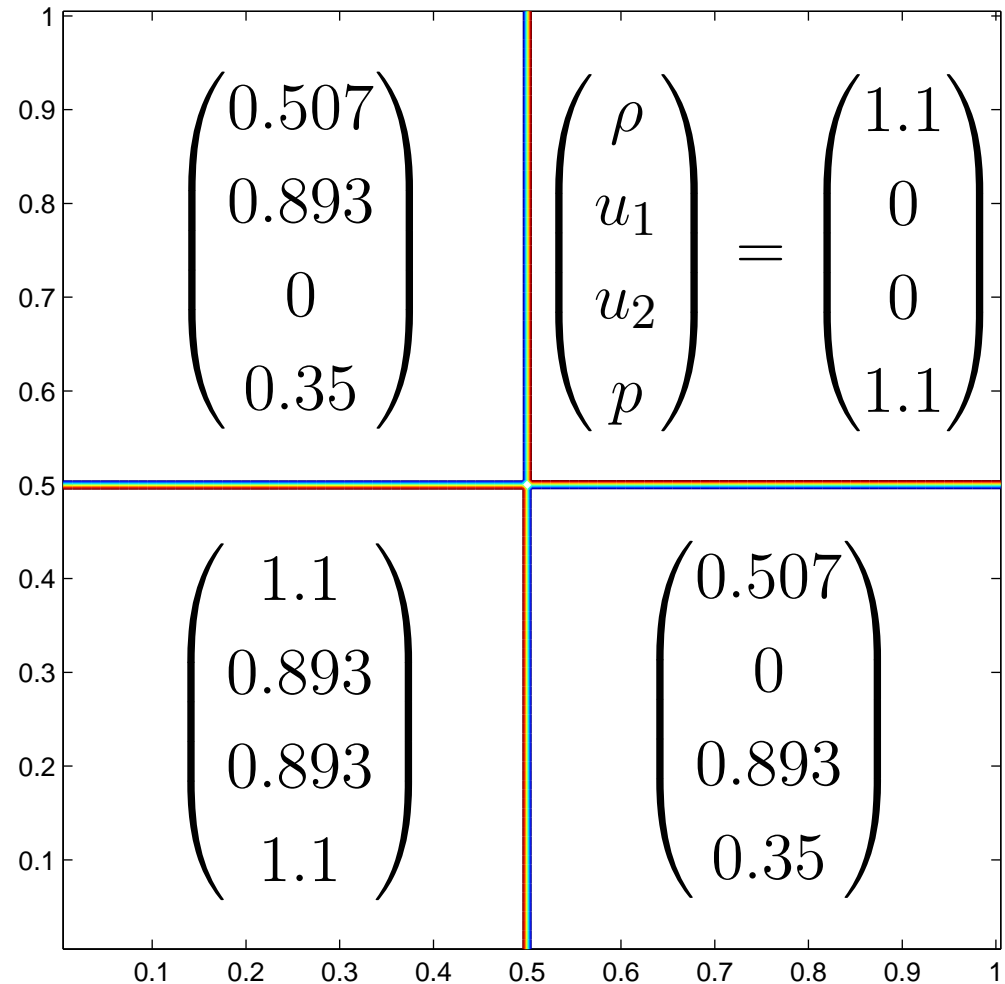
with

$$\begin{aligned} \tilde{q} &= Jq, & J &= \det(\partial\xi/\partial\mathbf{x})^{-1} \\ \tilde{f}_j &= J(q \partial_t \xi_j + \nabla \xi_j \cdot \mathbf{f}) \\ \mathcal{G} &= q [\partial_t J + \nabla_{\xi} \cdot (J \partial_t \xi_j)] + \sum_{j=1}^N f_j \nabla_{\xi} \cdot (J \partial_{x_j} \xi_k) \\ &= 0 \quad (\text{if GCL \& SCL are satisfied}) \end{aligned}$$

We may then devise numerical method to solve (2) (Shyue: HYP08)

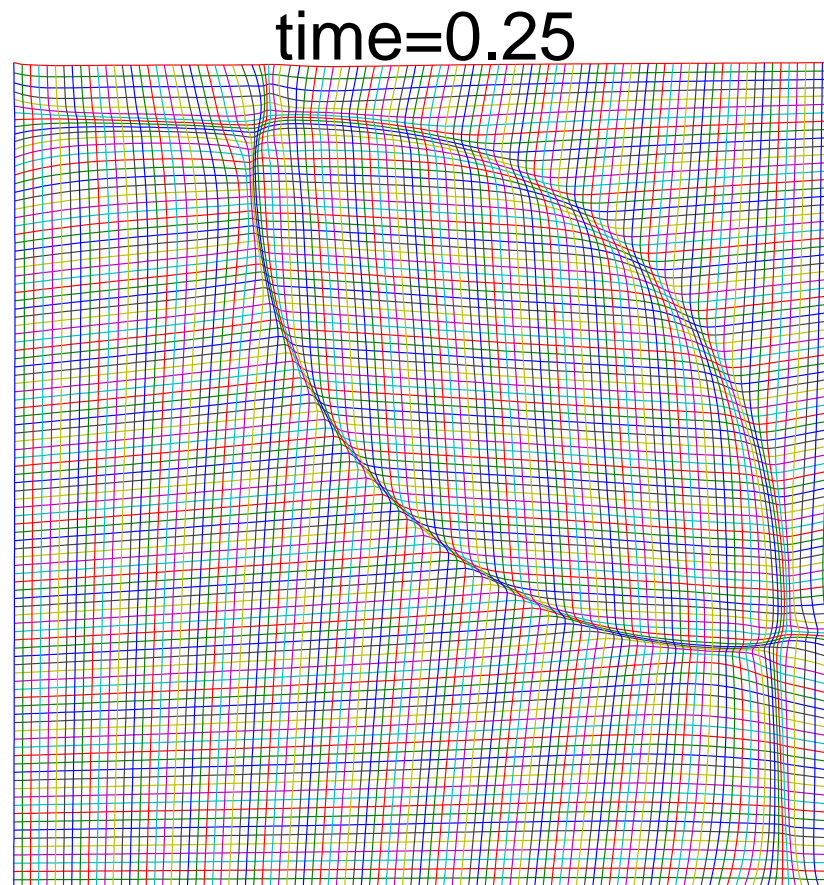
2D Riemann Problem

- 4-shock wave pattern initially



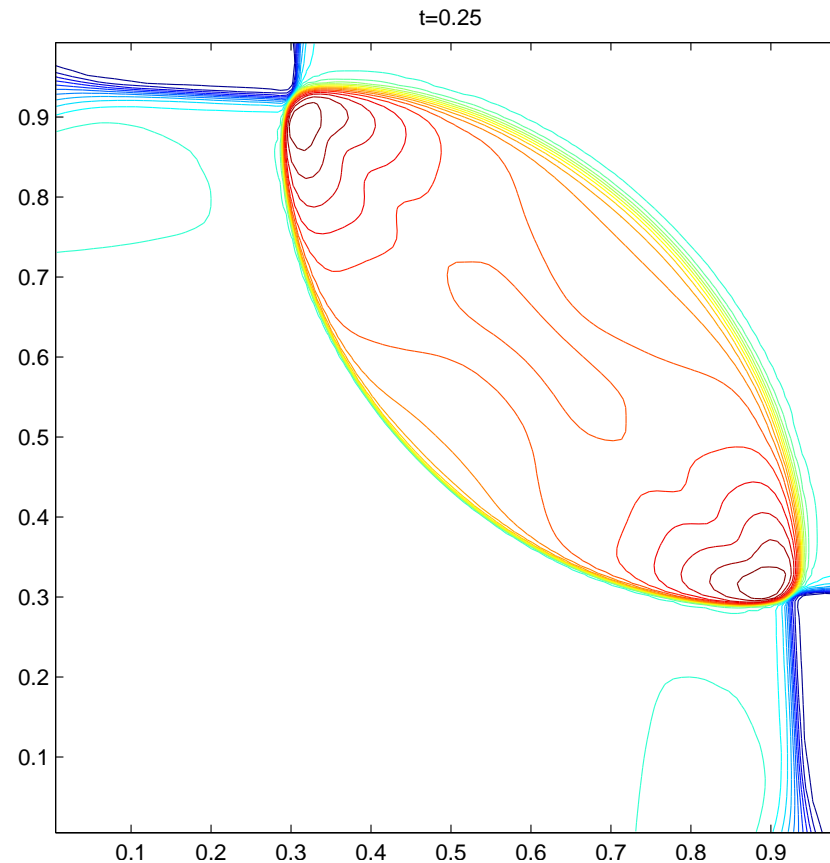
2D Riemann Problem

- Mesh redistribution scheme: variational approach
- Grid system



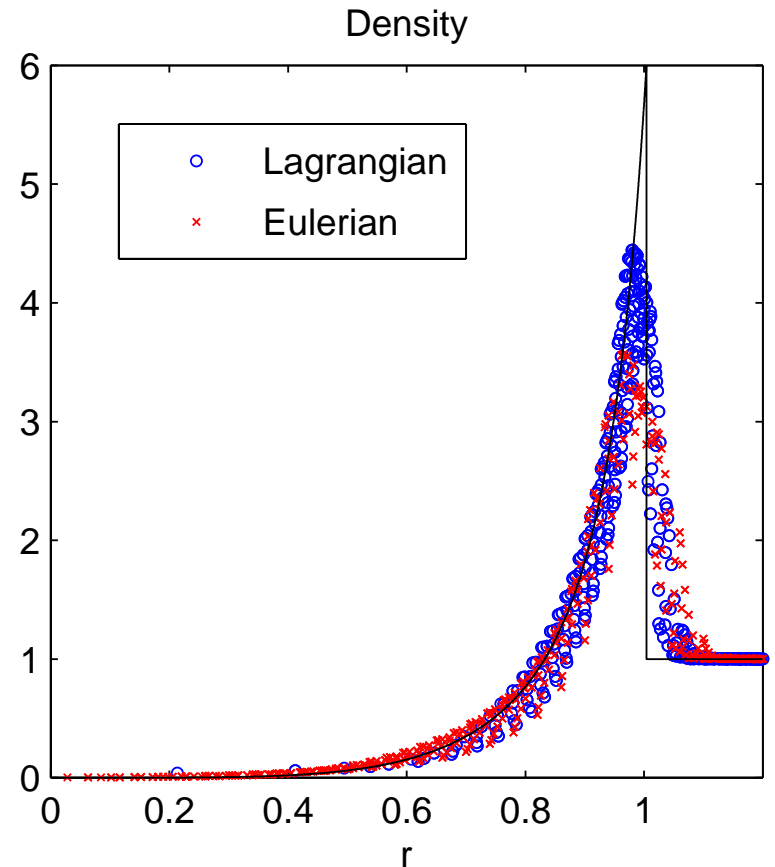
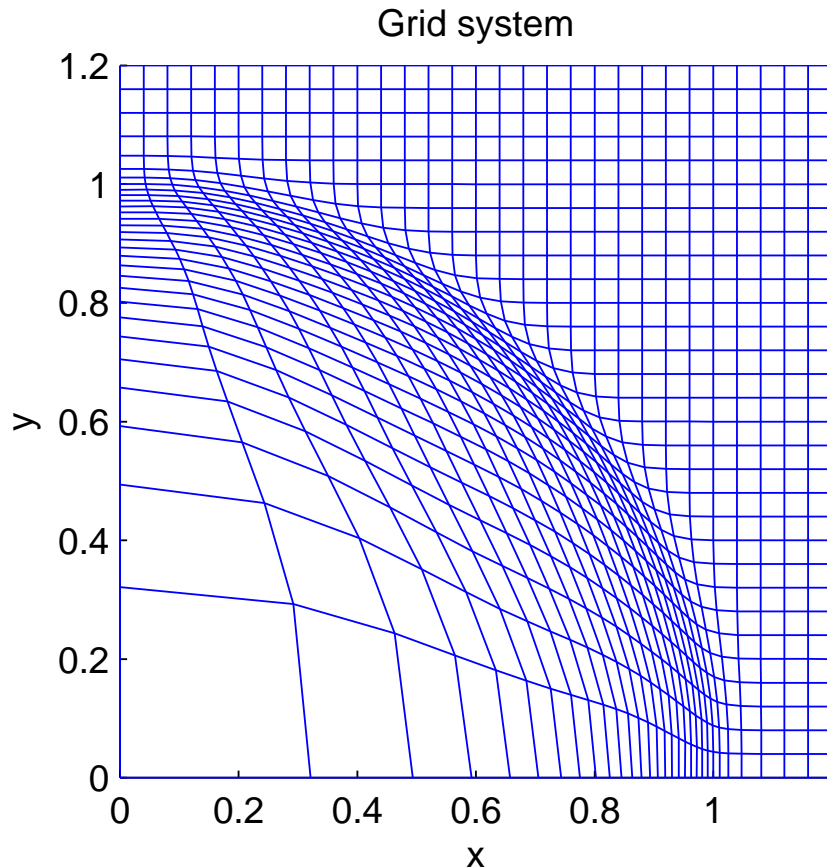
2D Riemann Problem

- Mesh redistribution scheme: variational approach
- Density contours



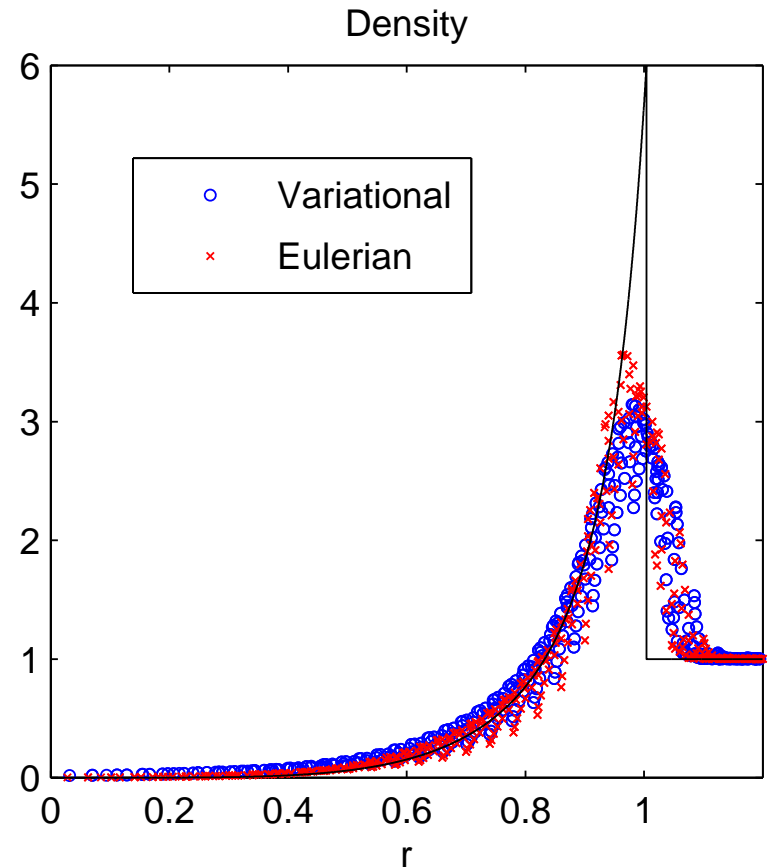
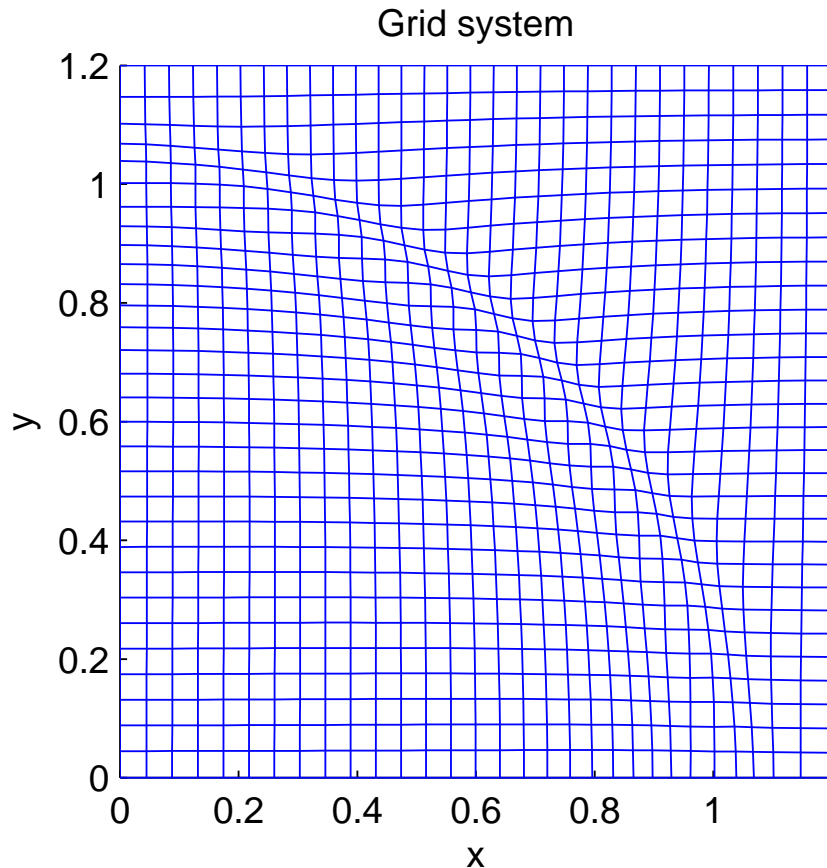
Sedov problem

- Mesh redistribution scheme: Lagrangian approach
- 30×30 mesh grid



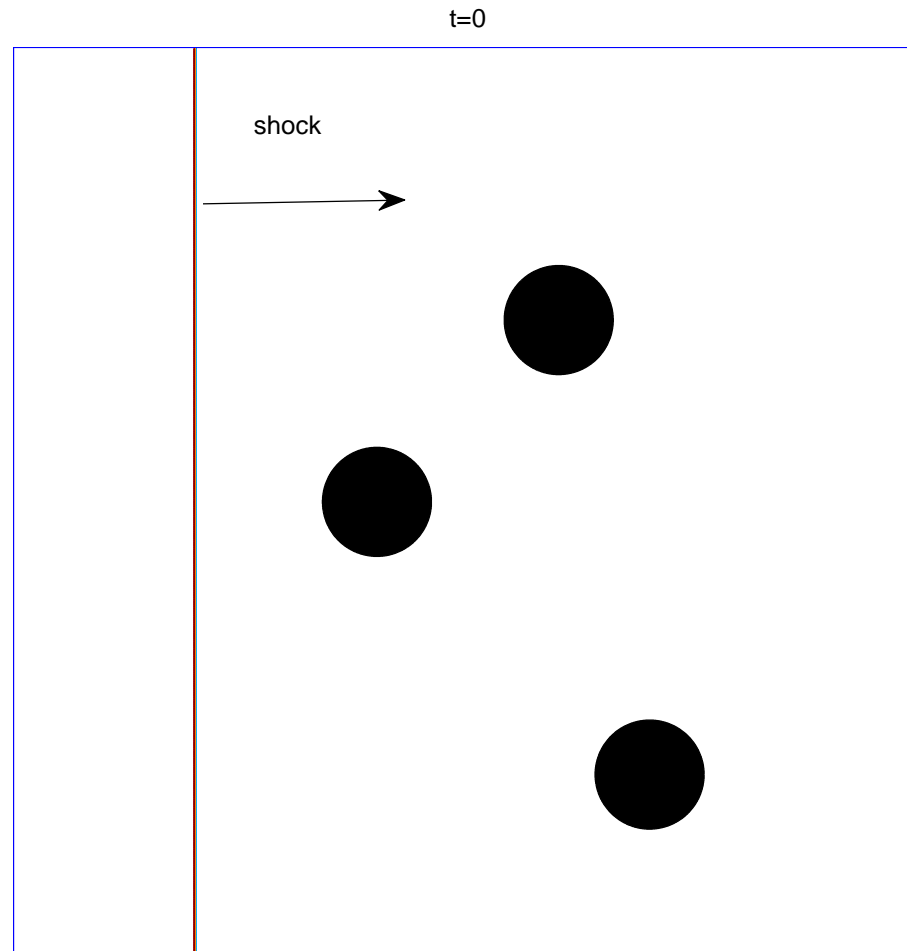
Sedov problem

- Mesh redistribution scheme: variational approach
- “Monitor” function D based on density gradient



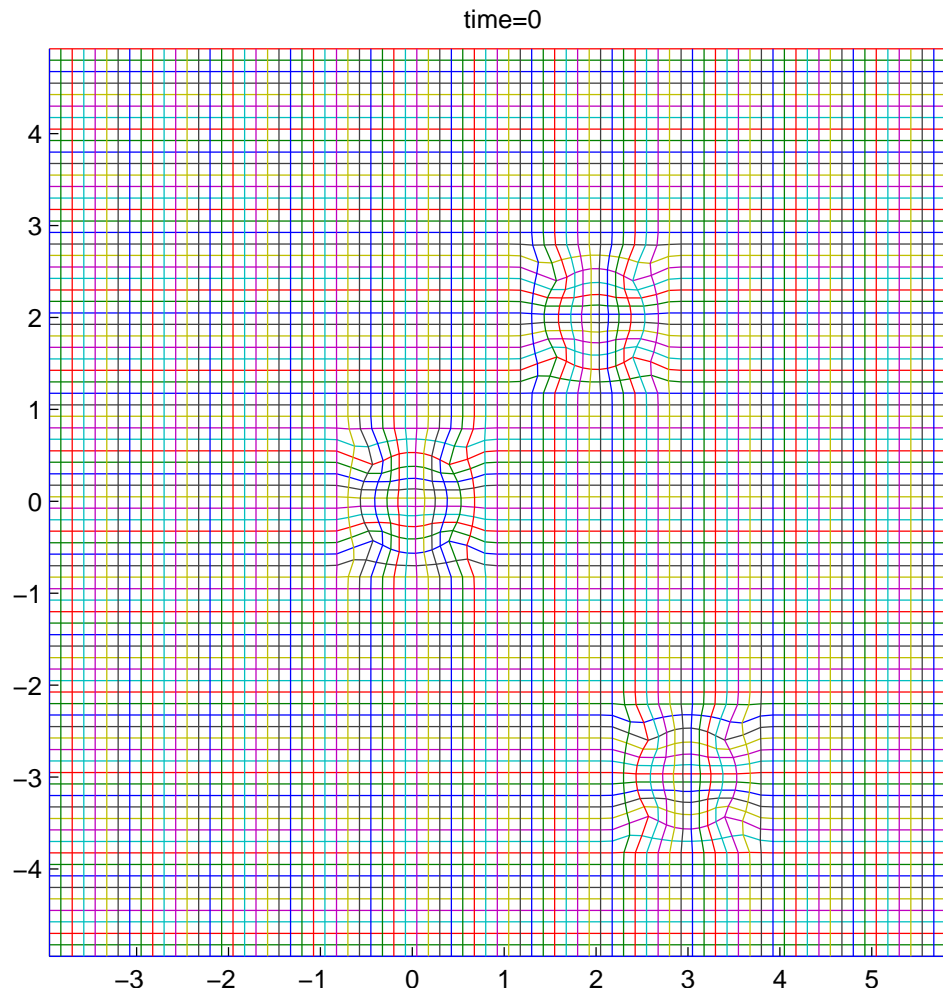
Shock waves over circular array

- A Mach 1.42 shock wave in water over a circular array



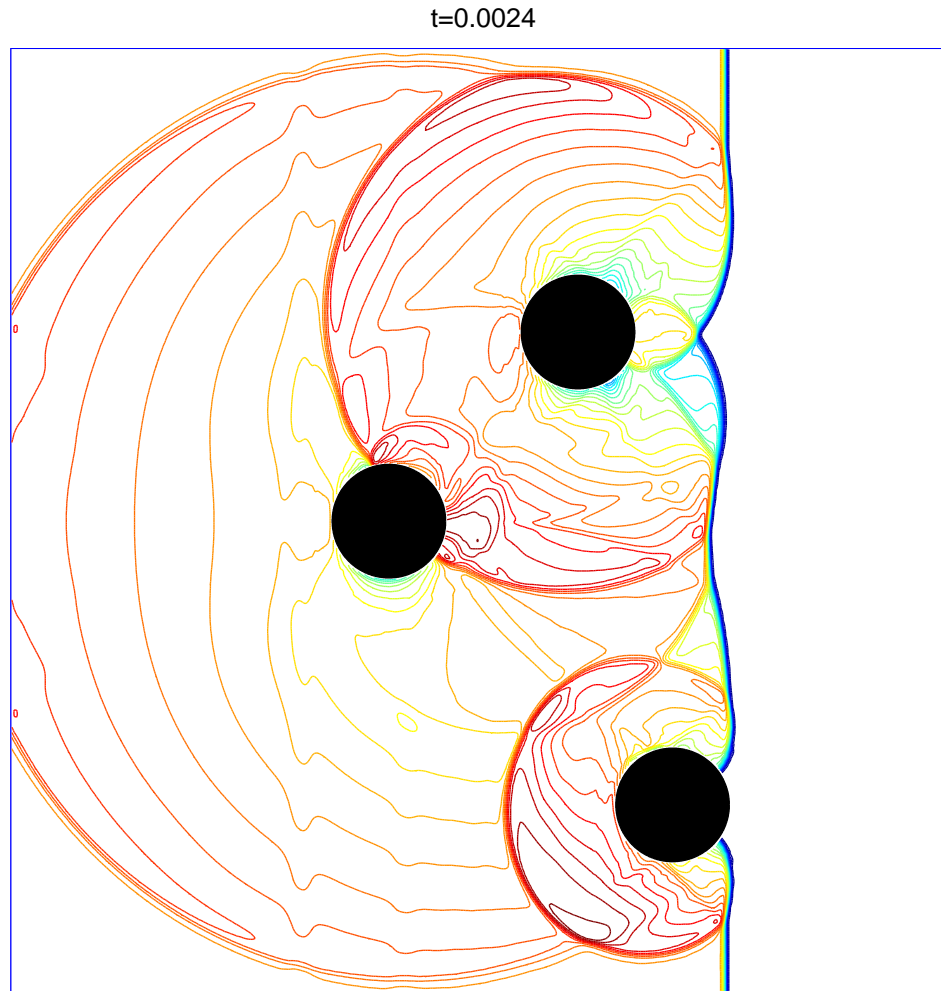
Shock waves over circular array

● Grid system



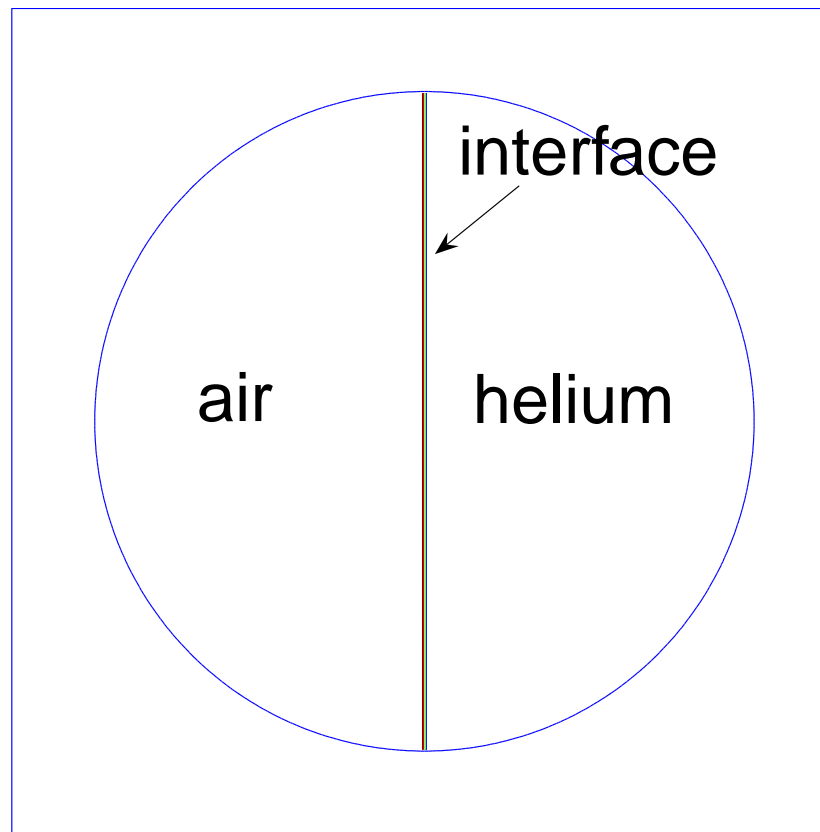
Shock waves over circular array

- Contours for density



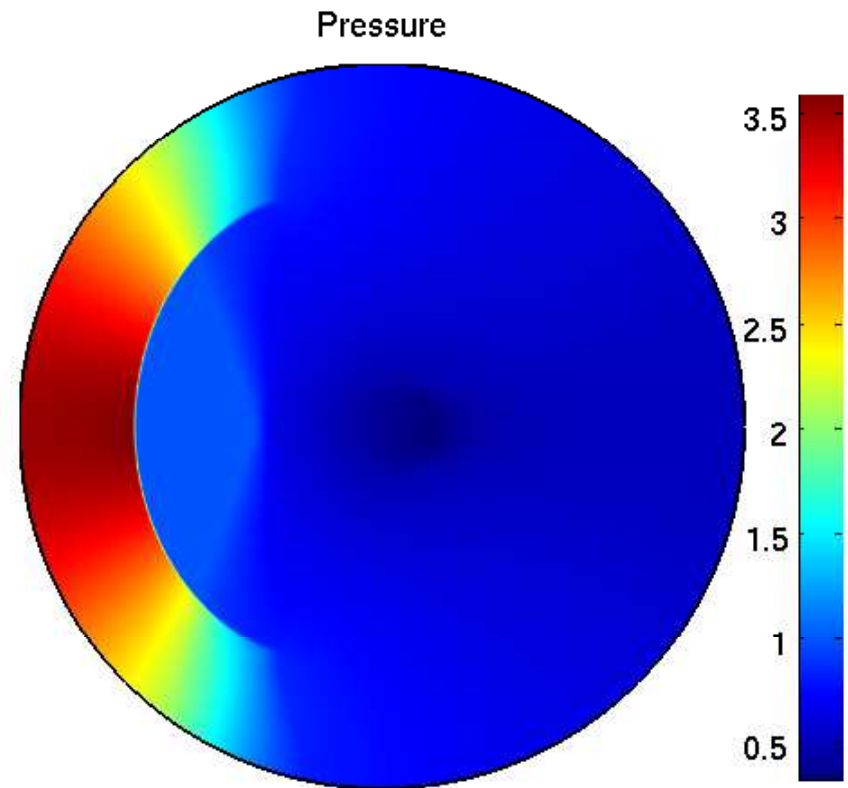
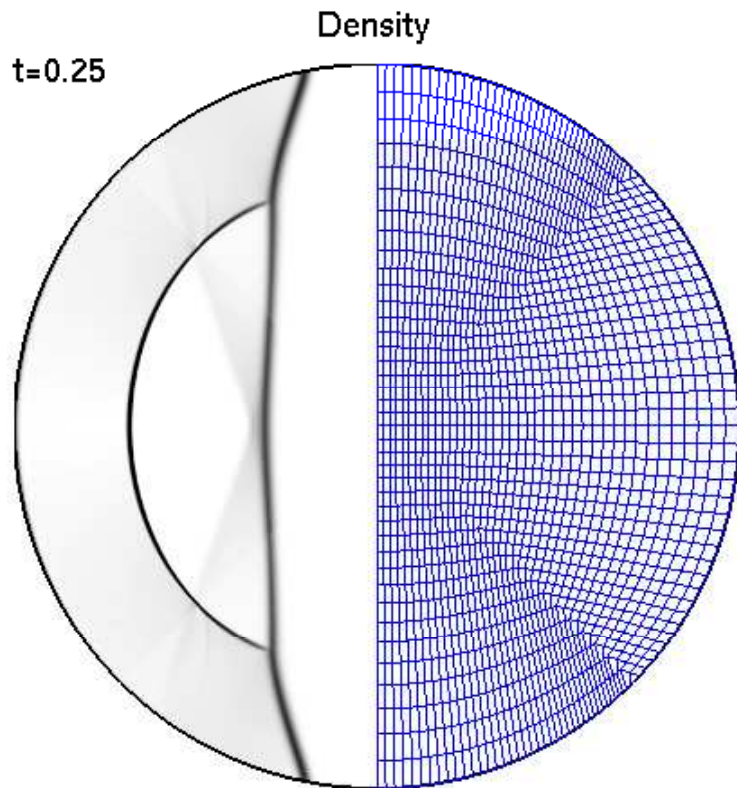
Moving cylindrical vessel

- Impose uniform flow velocity $(u_1, u_2) = (-1, 0)$ (*i.e.*, in the frame of vessel moving with speed one in x_1 -direction)



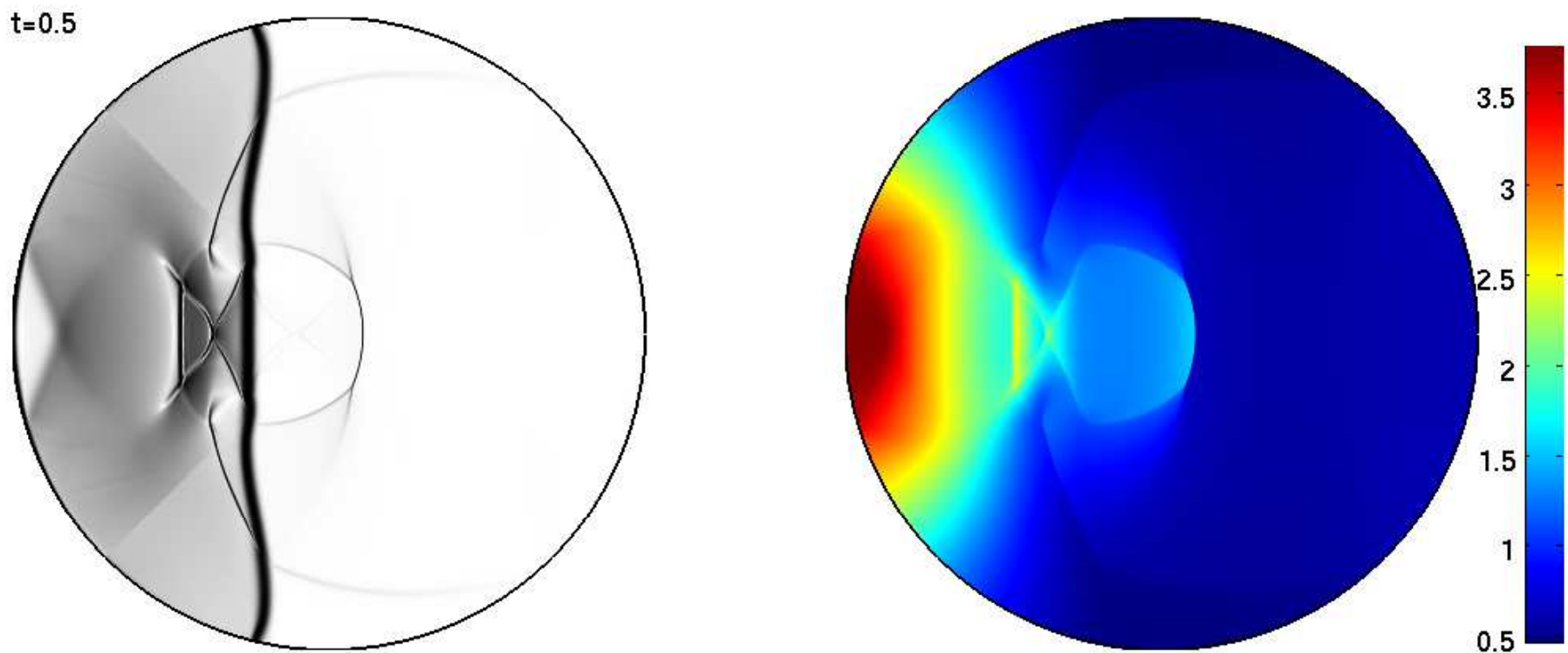
Moving cylindrical vessel

- Solution at time $t = 0.25$



Moving cylindrical vessel

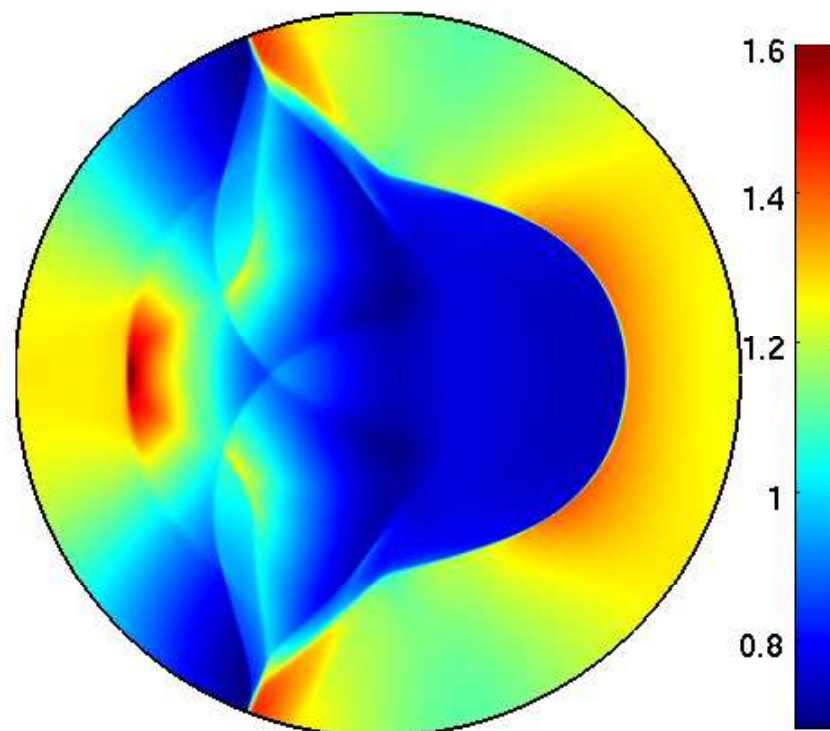
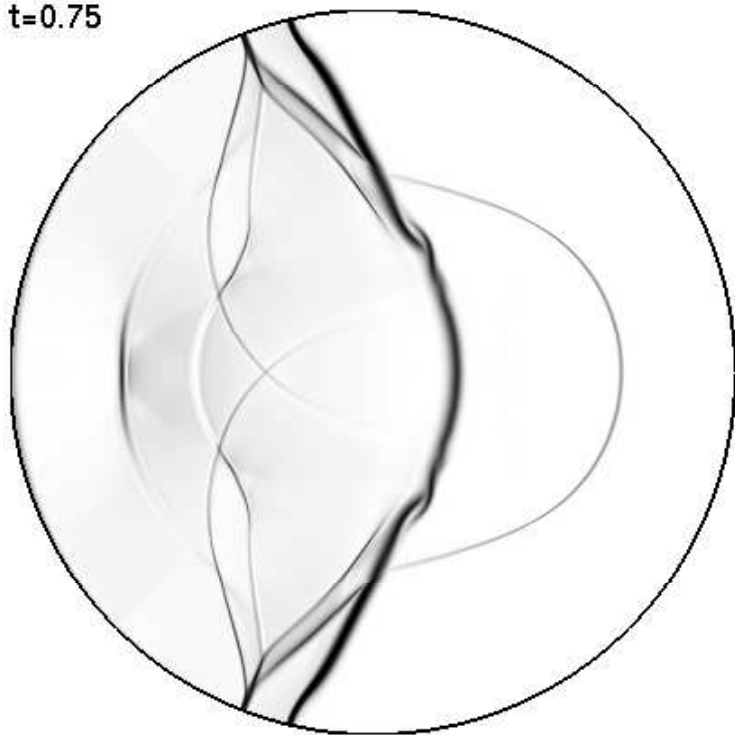
● Solution at time $t = 0.5$



Moving cylindrical vessel

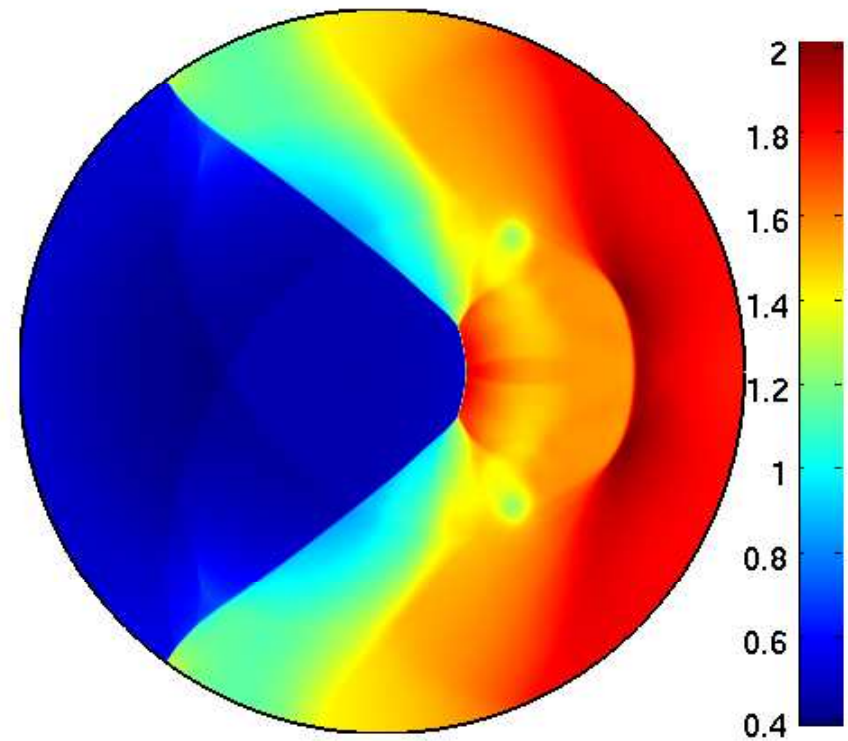
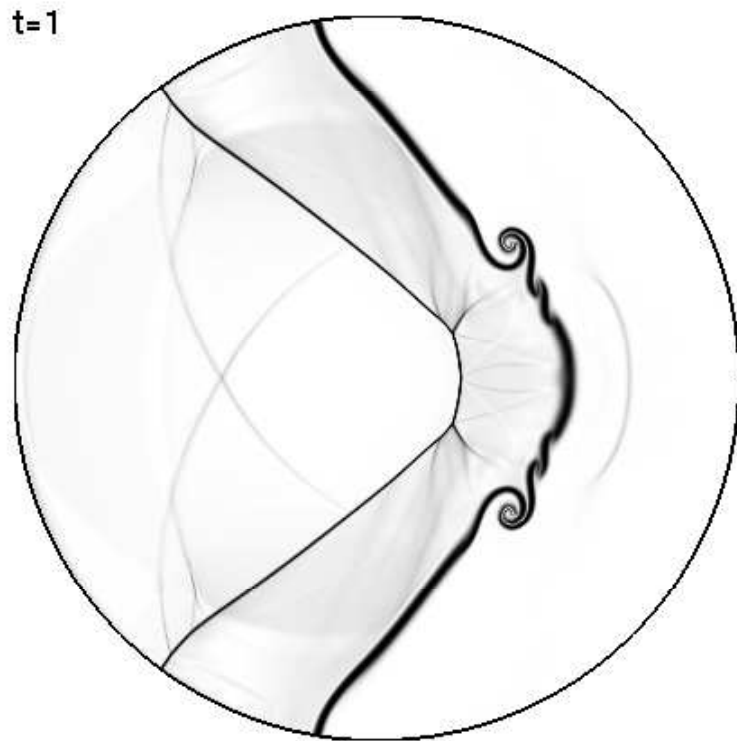
- Solution at time $t = 0.75$

t=0.75



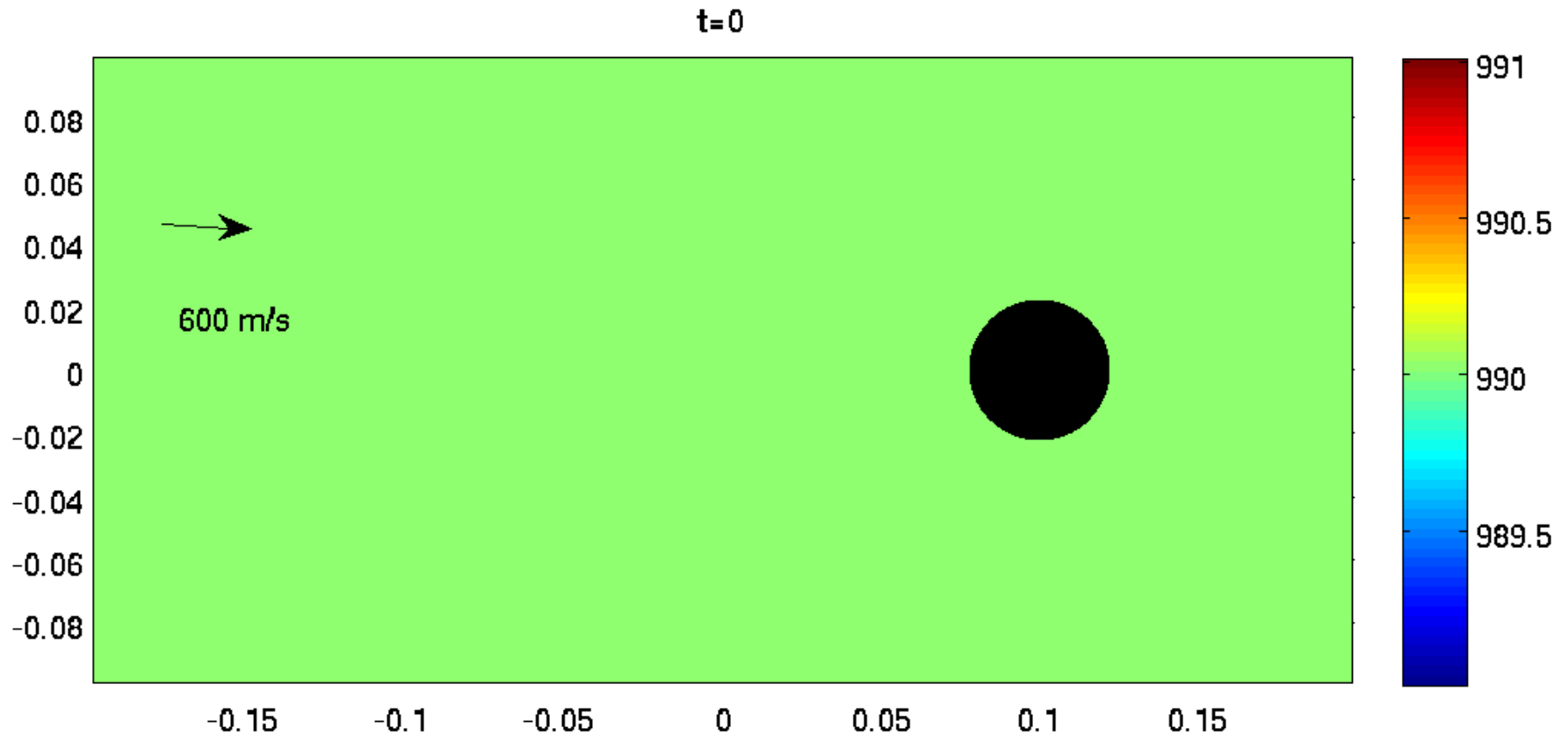
Moving cylindrical vessel

- Solution at time $t = 1$



Cavitation test: 2D steady state

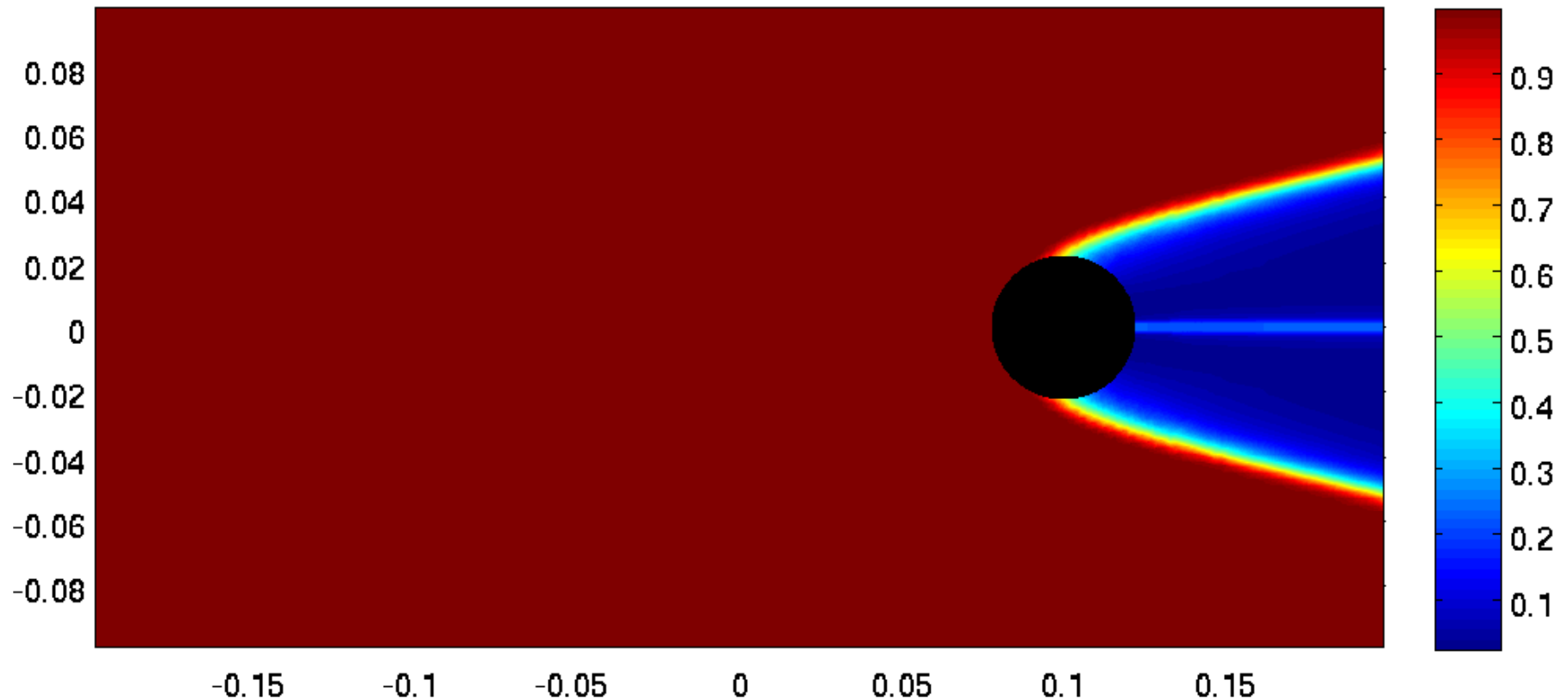
- Uniform gas-liquid mixture with speed 600m /s over a circular region



Cavitation test: 2D steady state

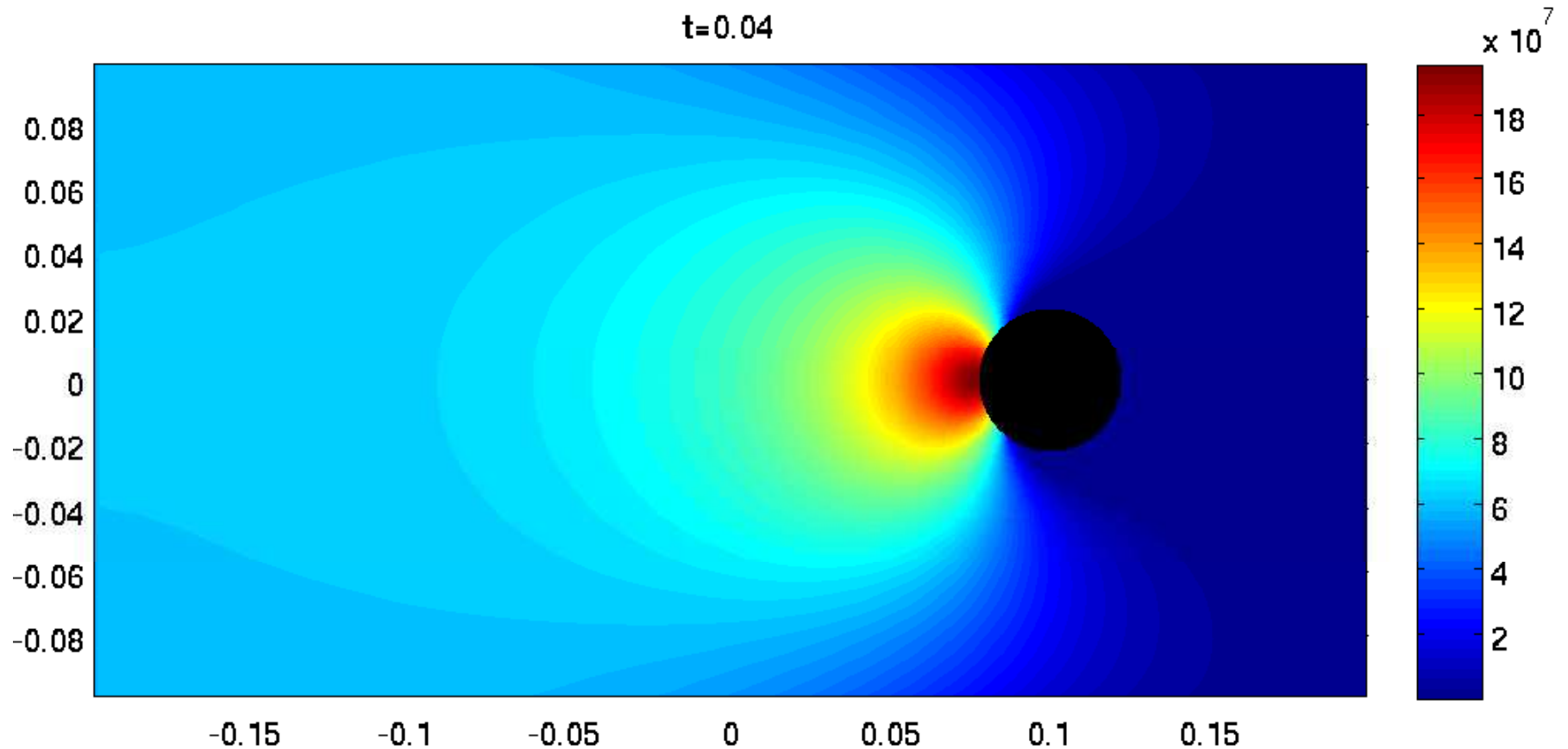
- Pseudo colors of volume fraction
 - Formation of cavitation zone (Onset–shock induced, diffusion, ... ?)

t=0.04



Cavitation test: 2D steady state

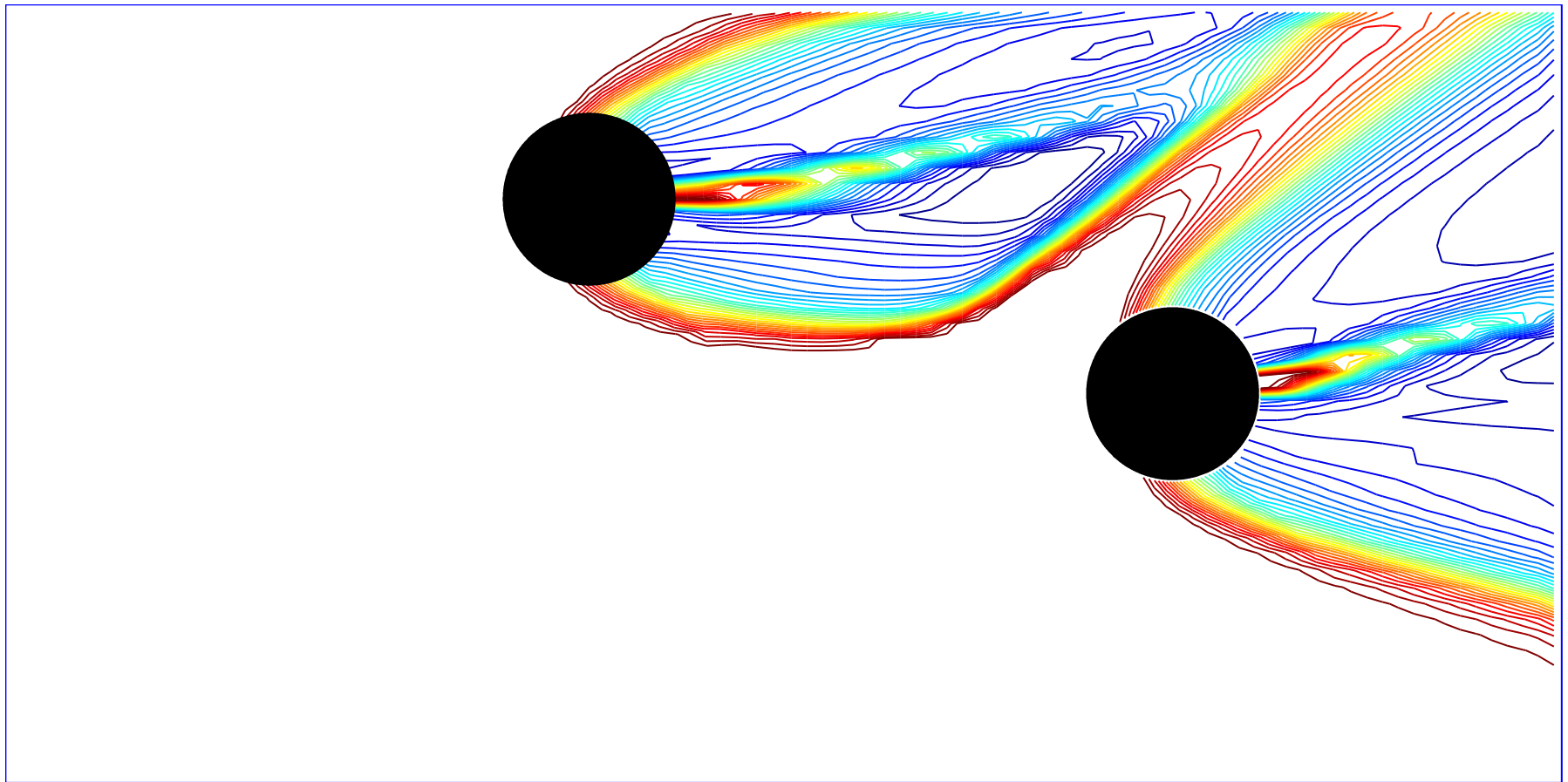
- Pseudo colors of pressure
- Smooth transition across liquid-gas phase boundary



Cavitation test: 2D steady state

- Pseudo colors of volume fraction: 2 circular case
- Convergence of solution as the mesh is refined ?

t=0.05



Thank You

Compressible Multiphase Flow

- Homogeneous equilibrium pressure & velocity across material interfaces
- Volume-fraction based model equations (Shyue JCP '98, Allaire *et al.* JCP '02)

$$\frac{\partial}{\partial t} (\alpha_i \rho_i) + \frac{1}{J} \sum_{j=1}^{N_d} \frac{\partial}{\partial \xi_j} (\alpha_i \rho_i U_j) = 0, \quad i = 1, 2, \dots, m_f$$

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{1}{J} \sum_{j=1}^{N_d} \frac{\partial}{\partial \xi_j} (\rho u_i U_j + p J_{ji}) = 0, \quad i = 1, 2, \dots, N_d,$$

$$\frac{\partial E}{\partial t} + \frac{1}{J} \sum_{j=1}^{N_d} \frac{\partial}{\partial \xi_j} (E U_j + p U_j) = 0,$$

$$\frac{\partial \alpha_i}{\partial t} + \frac{1}{J} \sum_{j=1}^{N_d} U_j \frac{\partial \alpha_i}{\partial \xi_j} = 0, \quad i = 1, 2, \dots, m_f - 1$$

Barotropic two-fluid flow

- Two-phase flow model

$$\partial_t (\alpha_1 \rho_1) + \nabla \cdot (\alpha_1 \rho_1 \mathbf{u}) = 0,$$

$$\partial_t (\alpha_2 \rho_2) + \nabla \cdot (\alpha_2 \rho_2 \mathbf{u}) = 0,$$

$$\partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p \delta) = 0$$

$$\frac{\alpha_1 \rho_1}{\rho_1(p)} + \frac{\alpha_2 \rho_2}{\rho_2(p)} = 1 \quad (\text{iterative solve for } p)$$

Barotropic two-fluid flow

- A relaxation model of Saurel *et al.* (JCP '090)

$$\frac{\partial}{\partial t} (\alpha_1 \rho_1) + \sum_{j=1}^N \frac{\partial}{\partial x_j} (\alpha_1 \rho_1 u_j) = 0,$$

$$\frac{\partial}{\partial t} (\alpha_2 \rho_2) + \sum_{j=1}^N \frac{\partial}{\partial x_j} (\alpha_2 \rho_2 u_j) = 0,$$

$$\frac{\partial}{\partial t} (\rho u_i) + \sum_{j=1}^N \frac{\partial}{\partial x_j} (\rho u_i u_j + p \delta_{ij}) = 0, \quad i = 1, \dots, N$$

$$\frac{\partial \alpha_1}{\partial t} + \sum_{j=1}^N u_j \frac{\partial \alpha_1}{\partial x_j} = \frac{1}{\mu} (p_1(\rho_1) - p_2(\rho_2))$$

Each phasic pressure p_ι is a function of density only

Mixture pressure p satisfies $p = \alpha_1 p_1 + \alpha_2 p_2$, μ parameter

Mixture speed of sound

- Wood formula (stiffness in \bar{c} vs. α)

$$\frac{1}{\rho \bar{c}^2} = \frac{\alpha}{\rho_w c_w^2} + \frac{1 - \alpha}{\rho_g c_g^2}$$

$$\rho_w = 10^3 \text{kg/m}^3, \quad c_w = 1449.4 \text{m/s}, \quad \rho_g = 1.0 \text{kg/m}^3, \quad c_g = 374.2 \text{m/s}$$

