Adaptive moving mesh methods for hyperbolic problems

Perspective to astrophysical applications

Keh-Ming Shyue

Department of Mathematics

National Taiwan University

Taiwan

Objective

Discuss adaptive moving mesh method for sharp & accurate numerical resolution of discontinuous solutions (shock waves & interfaces) for hyperbolic balance laws

$$\partial_t q + \nabla \cdot f(q) = \psi(q)$$

in more than one space dimension

$q \in \mathbb{R}^m$, $f \in \mathbb{R}^{m \times N_d}$, & $\psi \in \mathbb{R}^m$ denote vector of m conserved quantities, flux matrix, & source terms

Hyperbolicity of system means any linear combination of Jacobian matrix of column vector of flux matrix f has real eigenvalues & complete set of eigenvectors

Content

1. Cartesian cut-cell approach
   - Marker-and-cell (MAC) front tracking method
   - Volume-of-fluid (VOF) interface tracking method
   - Application to cosmic-ray modified shock waves, detonation waves, & compressible multiphase flow

2. Mapped-grid approach (variant of ALE method)
   - Interpolation-based method
   - Interpolation-free method

3. Future research direction

Cartesian cut-cell method

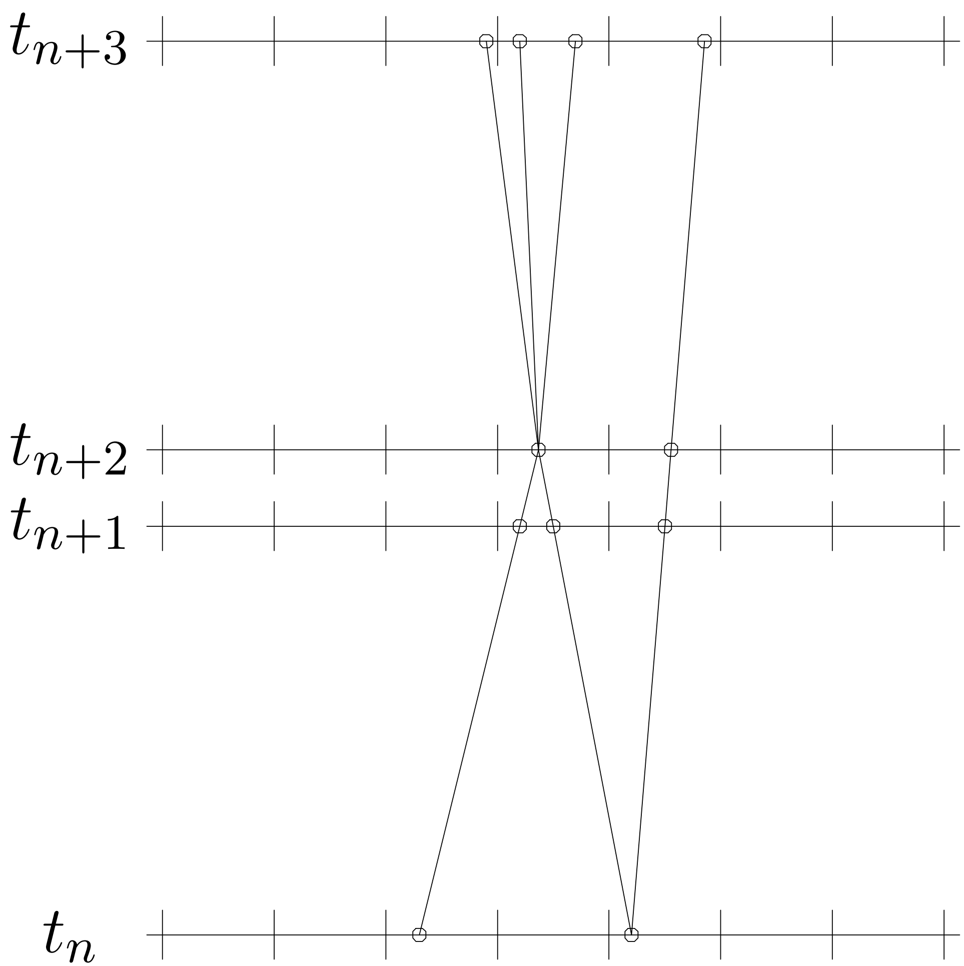
Basic algorithmic features:

- Use uniform underlying grid
- Introduce additional grid interfaces (points in 1D, curves in 2D, surfaces in 3D) which represent discontinuities moving freely through underlying grid
- Employ a finite volume method on a grid which contains tracked discontinuities for solution update

This method is unlike a mapped grid method (to be discussed later) where underlying grid is adjusted to fit location of tracked discontinuities

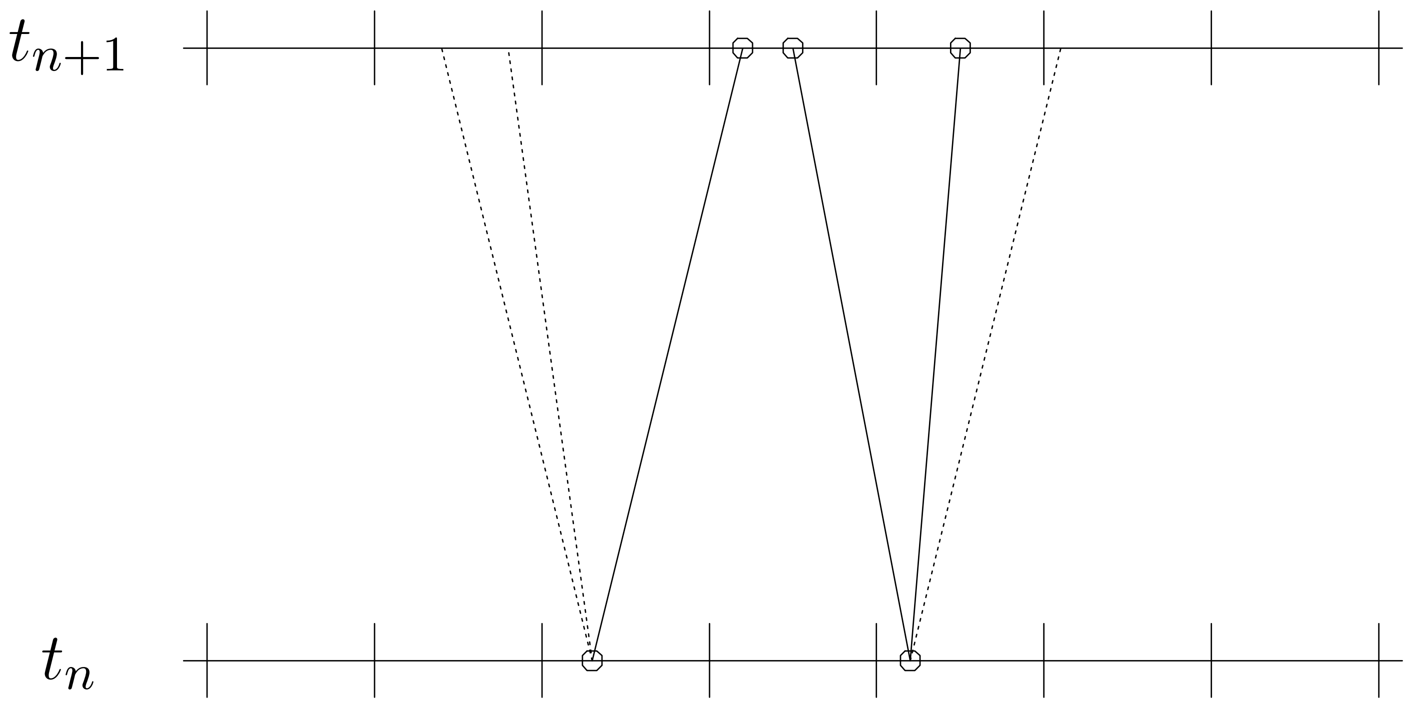
MAC front tracking in 1D

Our grid system is time-varying that consists of two parts: regular & irregular cells, 1D sample grid is shown below



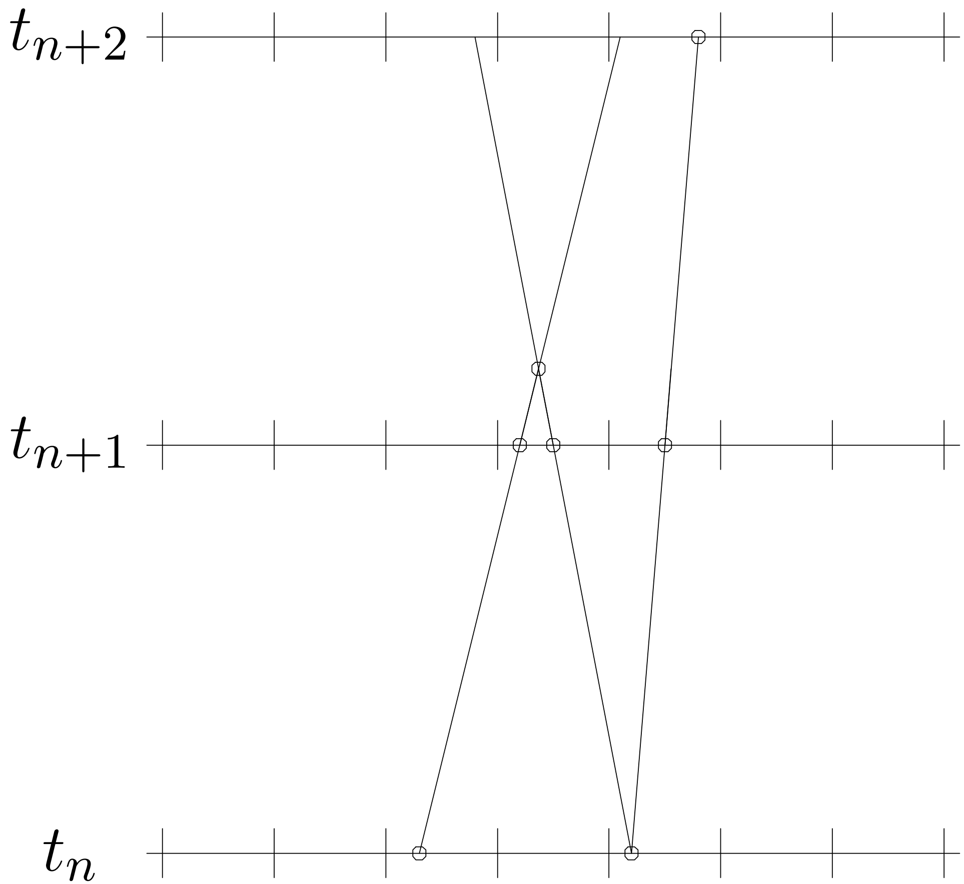
MAC front moving procedure

1. Solve Riemann problems at each grid point
2. Check strength of resulting Riemann solutions; only strong wave (solid line) is tracked & weak wave (dashed line) is captured



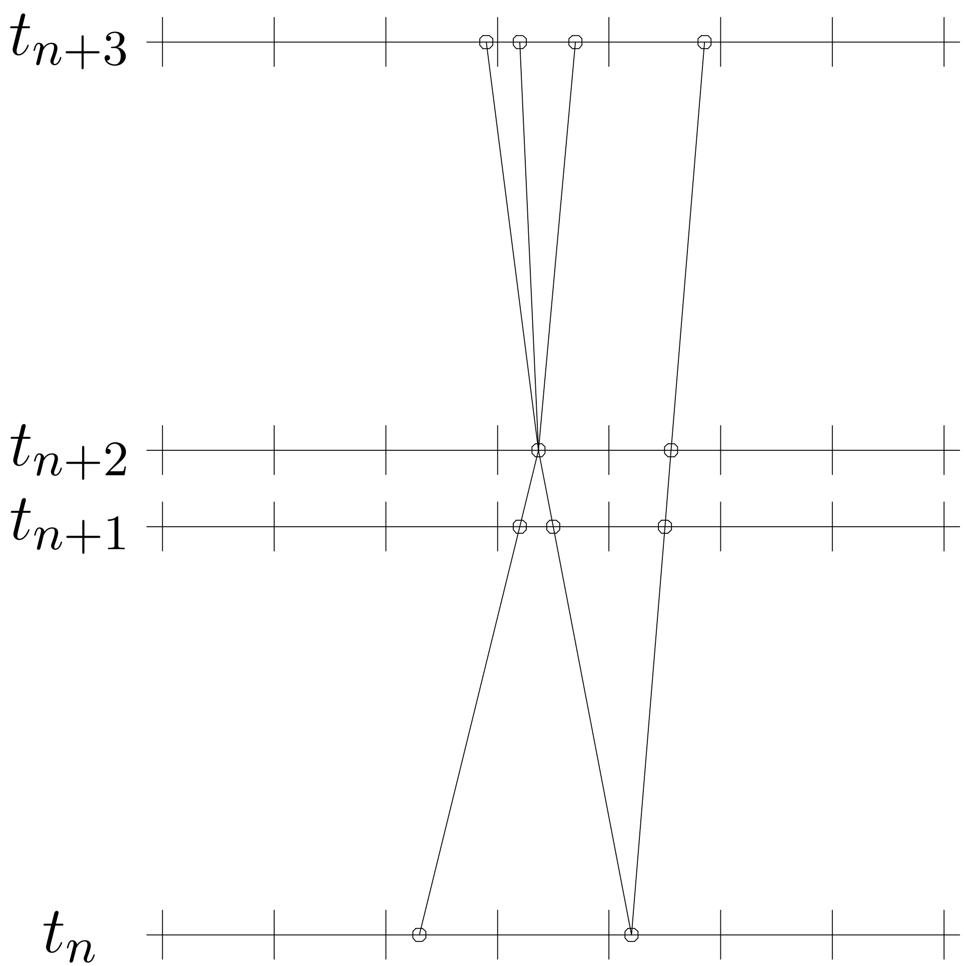
MAC front moving procedure

Two tracked waves collide at a point over $[t_{n+1}, t_{n+2}]$



MAC front moving procedure

Front collision case: adjust time step to collision point for accurate resolution of solution after wave interaction



Solution update

Method uses finite-volume formulation in that approximate value of cell average of solution over jth cell at a time t_n is

$$Q_j^n \approx \frac{1}{\mathcal{M}(C_j^n)} \int_{C_j^n} q(x, t_n)\, dx$$

C_j^n denotes region occupied by grid cell j at t_n & $\mathcal{M}(C_j^n)$ is measure (length) of C_j^n

Choose "large" time step Δt based on CFL condition $\nu_{\Delta x}$ but is not restricted one based on $\nu_{\Delta x_{\min}}$ as

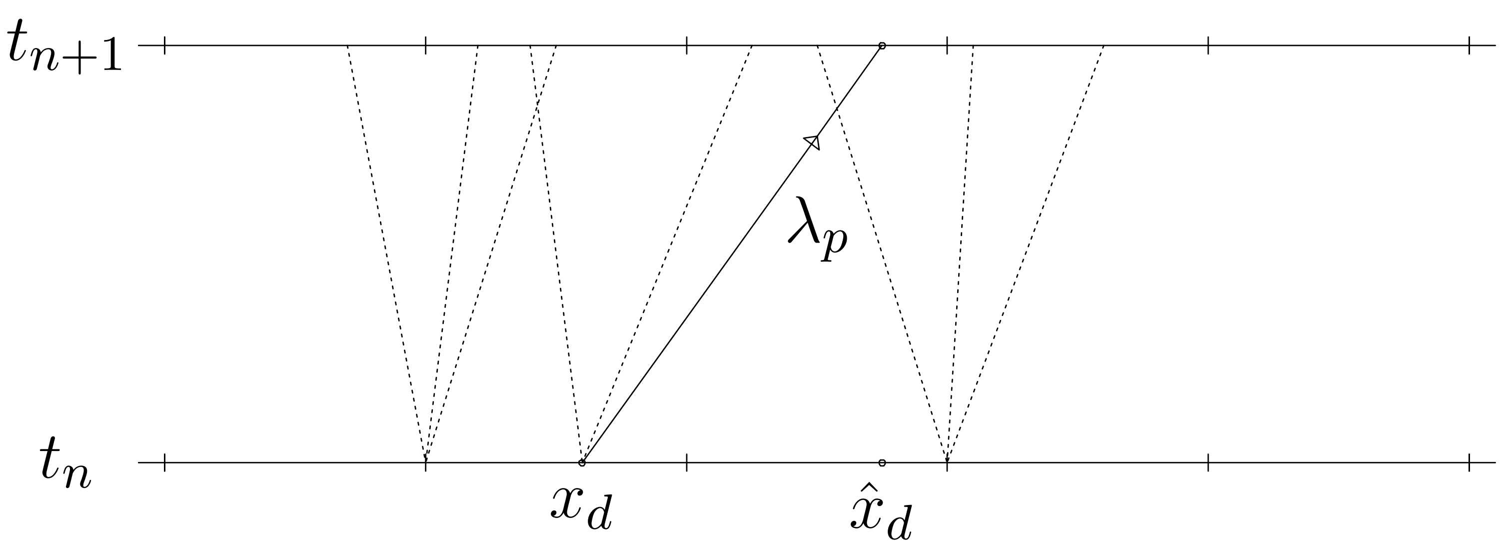
$$\nu_{\Delta x} = \frac{\Delta t}{\Delta x} \max_{p,j} |\lambda_{pj}| \leq 1 \quad \& \quad \nu_{\Delta x_{\min}} = \frac{\Delta t}{\Delta x_{\min}} \max_{p,j} |\lambda_{pj}| \leq 1$$

$\Delta x_{\min} = \min_j \Delta x_j$, λ_{pj} wave speed in pth family

Wave propagation method

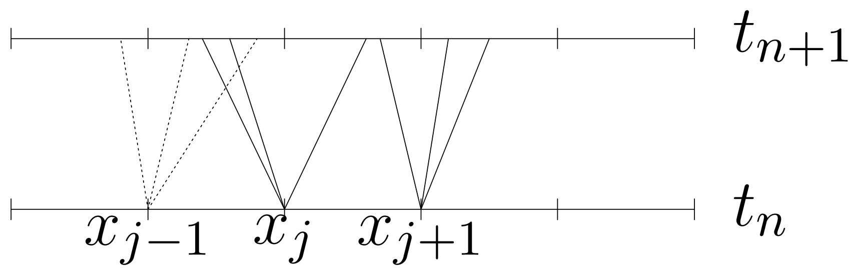
Method is of Godunov-type in that

- Propagate waves (obtained using shock-only approximate Riemann solver) independently
- Allow waves to propagate more than one cell to maintain stability even in presence of small cells
  - wave interaction in cell is handled linearly
- No averaging error & so smearing of tracked waves

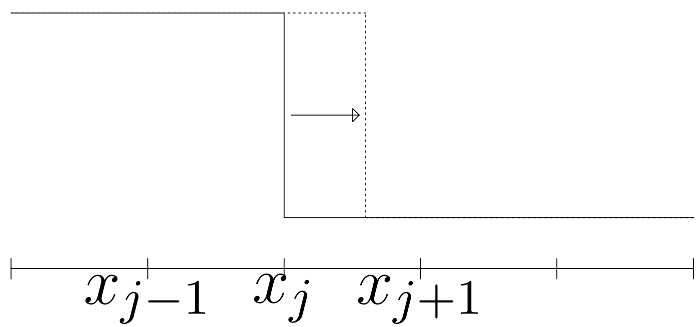


Wave propagation (graphical view)

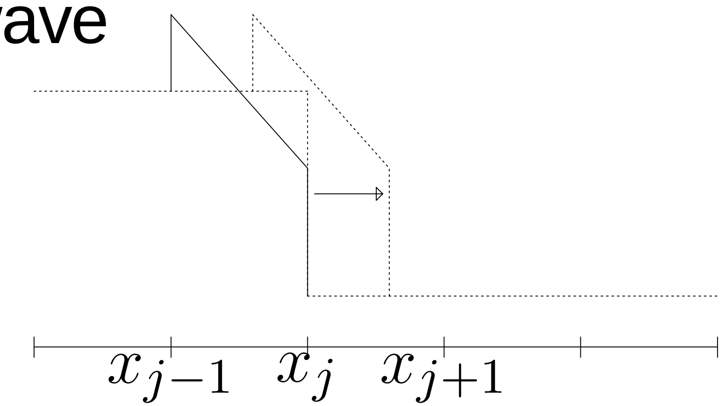
Wave structure in x-t space



Piecewise constant wave arising at x_j



Piecewise linear wave



Wave propagation method

On uniform grid, first order method takes form

$$Q_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x} \sum_{p=1}^{m_w} \left(\lambda_p^- \mathcal{W}_p\right)_{j+1}^n + \left(\lambda_p^+ \mathcal{W}_p\right)_j^n$$

while high resolution method (slope limiter type) takes

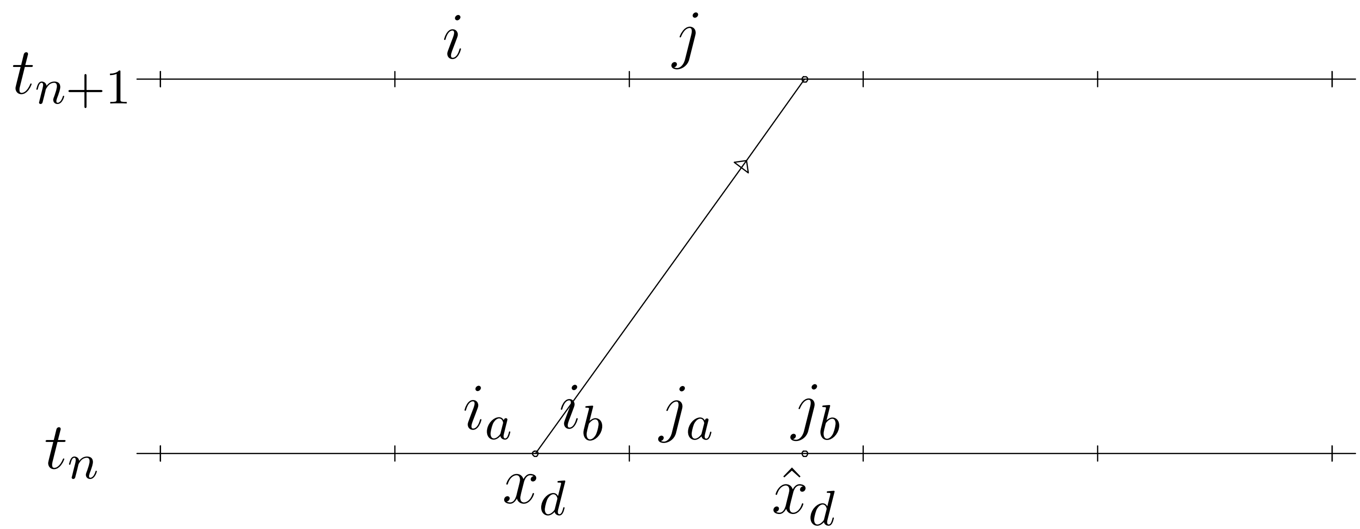
$$Q_j^{n+1} := Q_j^{n+1} - \frac{\Delta t}{\Delta x} \left(\widetilde{\mathcal{F}}_{j+1} - \widetilde{\mathcal{F}}_j\right)$$

with $$\widetilde{\mathcal{F}}_{j+1} = \frac{1}{2} \sum_{p=1}^{m_w} \left[|\lambda_p| \left(1 - |\lambda_p| \frac{\Delta t}{\Delta x}\right) \widetilde{\mathcal{W}}_p \right]_{j+1}^n$$

$\widetilde{\mathcal{W}}_{kj}$ is limited version of wave $\mathcal{W}_{kj}$ (jumps in Riemann solution across λ_{pj}), $\lambda^+ = \max\{\lambda, 0\}$, $\lambda^- = \min\{\lambda, 0\}$, m_w is number of waves in total, *e.g.*, $m_w = 3$ for 1D Euler eq.

Split & merge grid cells

A tracked wave propagating from cell i to cell $j = i + 1$ leads to a subdivision of cells i and j



At t_n, split cell j in two, setting $Q^n_{j_a} = Q^n_{j_b} = Q^n_j$, while at t_{n+1}, remove old tracked point in cell i, using conservative weighted average

$$Q_i^{n+1} := \frac{x_d - x_i}{\Delta x} Q_{i_a}^{n+1} + \frac{x_{i+1} - x_d}{\Delta x} Q_{i_b}^{n+1}$$

MAC front tracking algorithm

In summary, in each time step, algorithm consists of

1. Flag tracked points by checking Riemann solutions
2. Determine time step Δt & location of tracked points at next time step
3. Modify current grid by inserting these new tracked points. Some cells will be subdivided & values in each subcell must be initialized
4. Take Δt as in step 2, employ a conservative finite volume method to update cell averages on this nonuniform grid
5. Delete old tracked points from previous time step. Some subcells will be combined & value in combined cell must be determined from subcell values

Cosmic-ray hydrodynamics

Consider two-fluid model for cosmic-ray modified flows proposed by Axford *et al.* 1977 & Drury & Völk 1981 in that

- Cosmic-rays (energetic charged particles) are assumed to be a hot low-density gas with negligible mass density, mass flux, & momentum density compared to that of thermal gas
- Cosmic rays are assumed to be scattered by waves or turbulence traveling in background flow
- Dynamics of flow system are governed by overall mass, momentum, & energy conservation equations
- Transfer of energy between cosmic rays & background flow is described by diffusive transport equation

Two-fluid cosmic-ray model

Two-fluid model for cosmic-ray-modified flows

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

$$\partial_t (\rho \mathbf{u}) + \nabla \cdot [\rho \mathbf{u} \otimes \mathbf{u} + (p_g + p_c) \mathbf{I}] = 0$$

$$\partial_t \left(\frac{1}{2} \rho \mathbf{u}^2 + E_g + E_c \right) + \nabla \cdot \left[\left(\frac{1}{2} \rho \mathbf{u}^2 + E_g + p_g \right) \mathbf{u} + \mathbf{F}_c \right] = 0$$

$$E_g = \frac{p_g}{\gamma_g - 1}, \qquad E_c = \frac{p_c}{\gamma_c - 1}$$

ρ, $\mathbf{u}$, p_g, E_g, γ_g, p_c, E_c, γ_c, $\mathbf{F}_c$, & $\mathbf{I}$, denote thermal gas density, velocity, pressure, energy density, adiabatic index, cosmic-ray pressure , energy density, adiabatic index, energy flux, & unit 3×3 dyadic

Cosmic-ray energy equation

Classical two-fluid model consists in using diffusive transport eq.

$$\partial_t E_c + \nabla \cdot \mathbf{F}_c = \mathbf{u} \cdot \nabla p_c$$

for energy density E_c carried by energetic particles in that energy flux F_c is defined by

$$\mathbf{F}_c = (E_c + p_c)\,\mathbf{u} - \kappa \cdot \nabla E_c$$

κ is mean hydrodynamical diffusion tensor

Cosmic-ray distribution function

Recent model concerns cosmic-ray particles described by distribution function $f(\mathbf{x}, p, t)$ that follows convection-diffusion equation of form

$$\partial_t f + \mathbf{u} \cdot \nabla f = \frac{1}{3} (\nabla \cdot \mathbf{u}) \, \partial_p f + \nabla \cdot (\kappa \nabla f)$$

p denotes momentum. We compute E_c, p_c, & γ_c by

$$E_c = 4\pi \int_{p_1}^{\infty} p^2 \left[\left(p^2 + 1\right)^{1/2} - 1 \right] f(\mathbf{x}, p, t) \, dp$$

$$p_c = \frac{4\pi}{3} \int_{p_1}^{\infty} p^4 \left(p^2 + 1\right)^{-1/2} f(\mathbf{x}, p, t) \, dp$$

$$\gamma_c = 1 + \frac{p_c}{E_c}$$

p_1 injection momentum

Numerical resolution of CR-hydro

Diffusion of cosmic rays pressure would tend to decelerate & compress flow into shock, forming a shock precursor

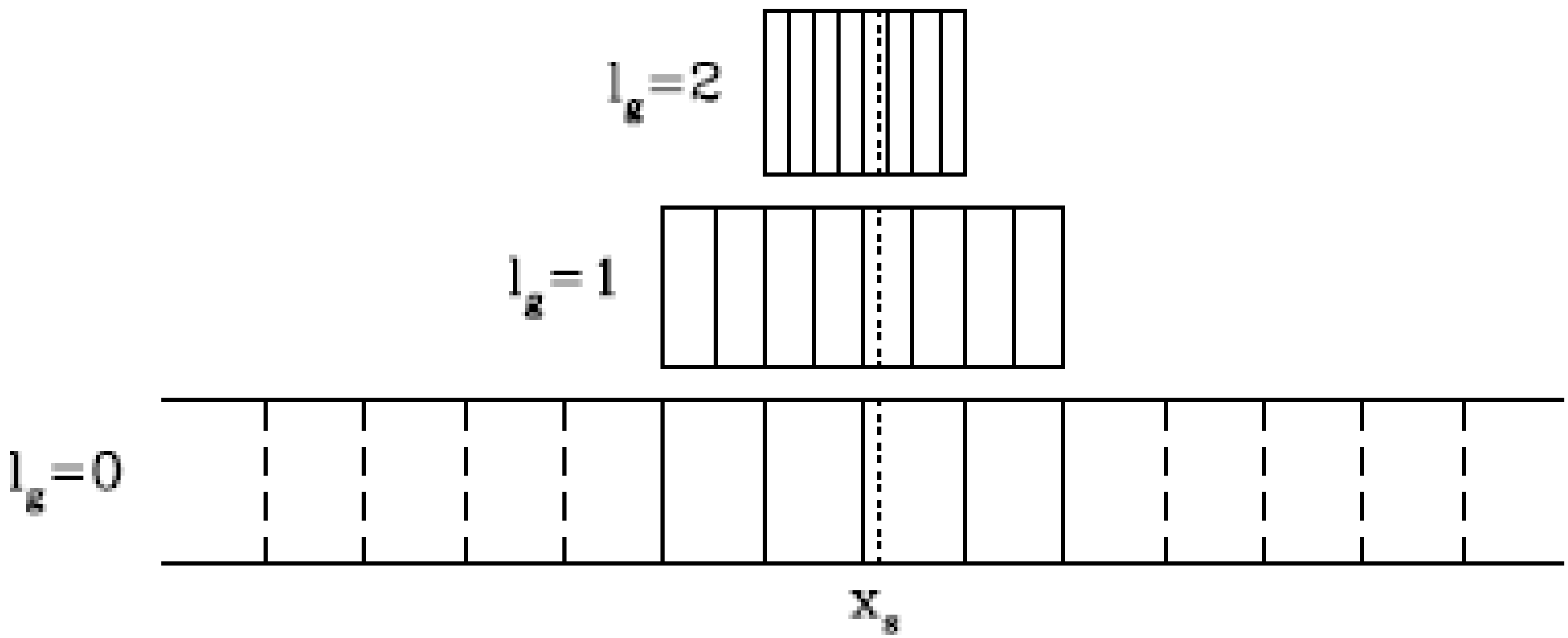
Spatial scale of flow within precursor can be characterized by so-called diffusion length $D_{\text{diff}} = \kappa(p)/u$, power law $\kappa(p) \propto p^s$ with $s \sim 1$-2 is of practical interest

Accurate solutions to CR convection-diffusion equation require a grid spacing significantly smaller than D_{diff}, typically $\Delta x \approx 5 \times 10^{-2} D_{\text{diff}}(p)$

CRASH (Cosmic-Ray Amr SHock) code developed by Kang & Jones for CR-related flow using front tracking method with AMR in region near shock

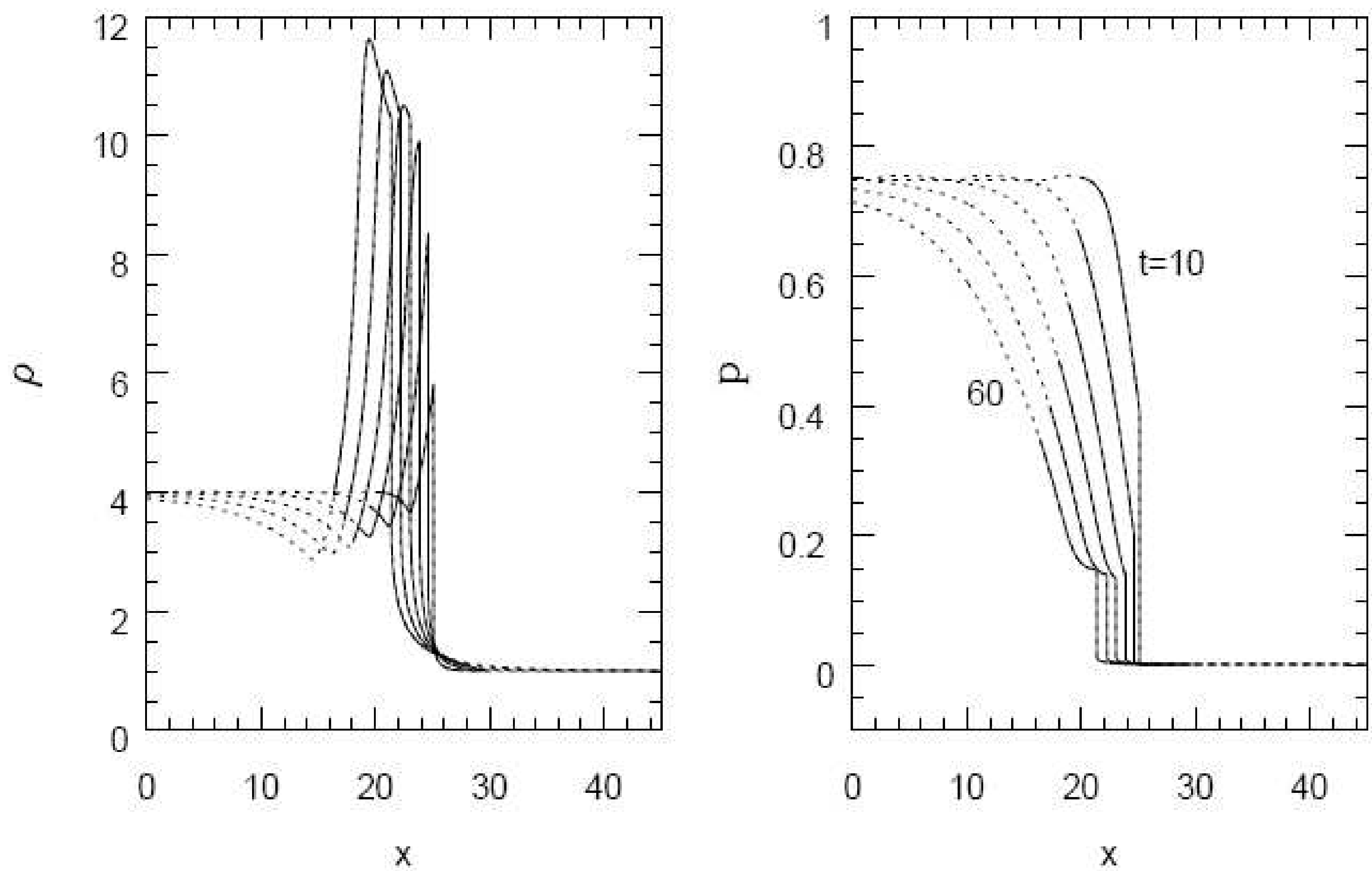
Test for CR modified plane shock

CRASH code basic grid setup: Shock tracking with AMR



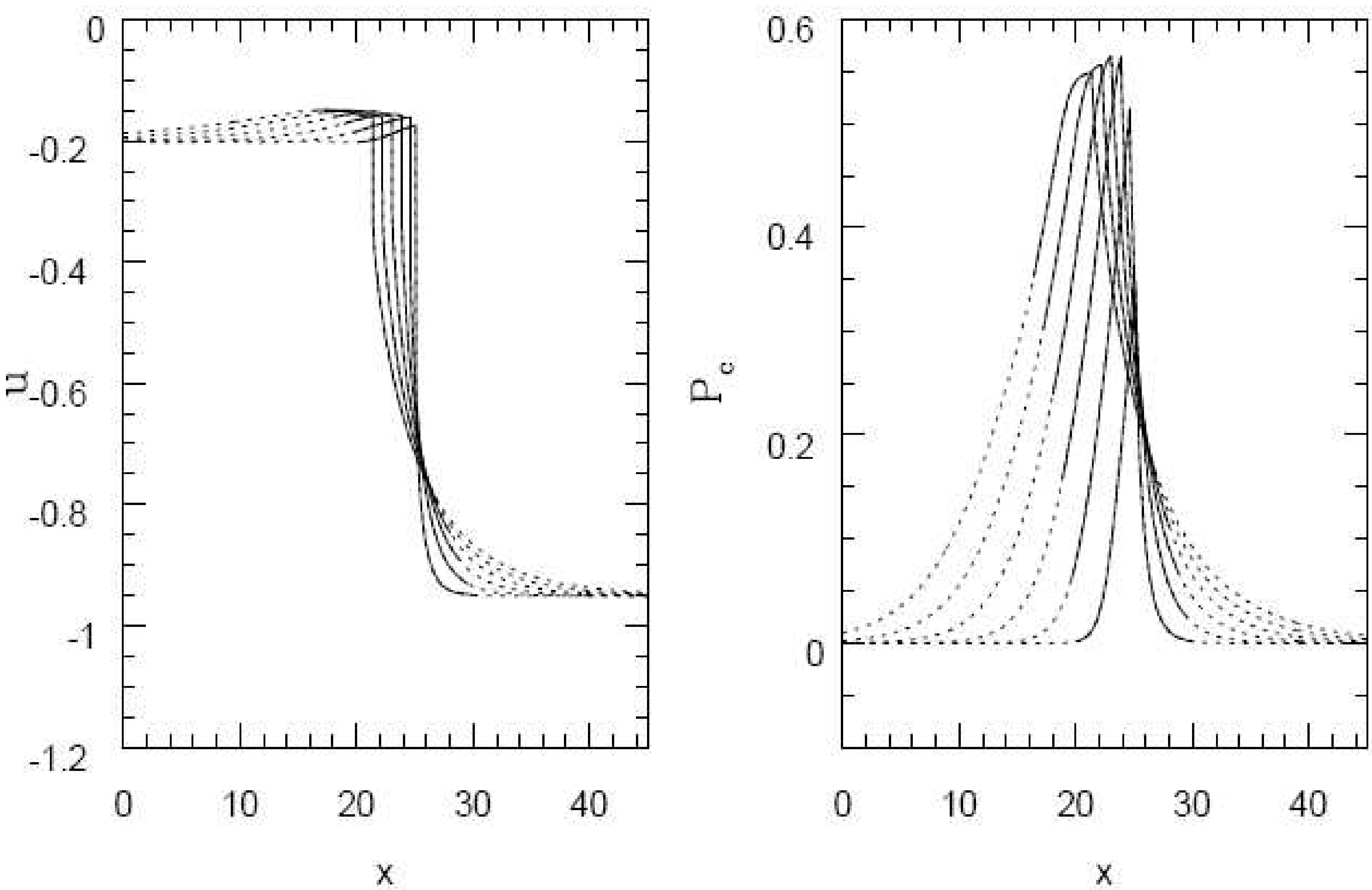
CR modified plane shock

Density & pressure obtained using CRASH code at six different times $t = 10, 20, \ldots, 60$



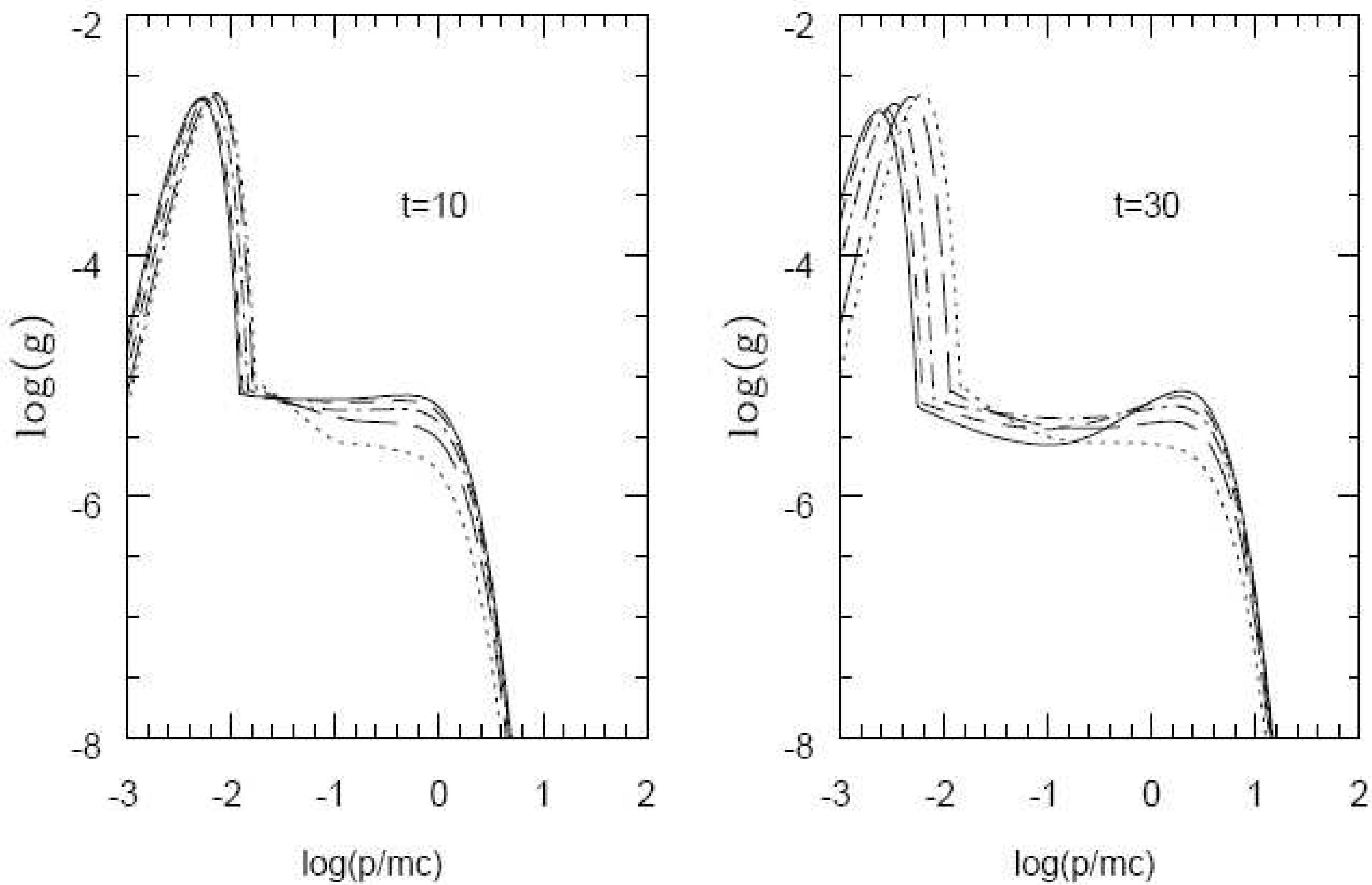
CR modified plane shock

Velocity & cosmic-ray pressure obtained using CRASH code at six different times $t = 10, 20, \ldots, 60$



CR modified plane shock

Distribution functions $g = fp^4$ at time $t = 10$ & 30 obtained using CRASH code with 4 different mesh sizes



Unstable detonation wave

Toy model for supernovae explosion

- Equation of motion

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

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

$$\partial_t E + \nabla \cdot (E\mathbf{u} + p\mathbf{u}) = 0$$

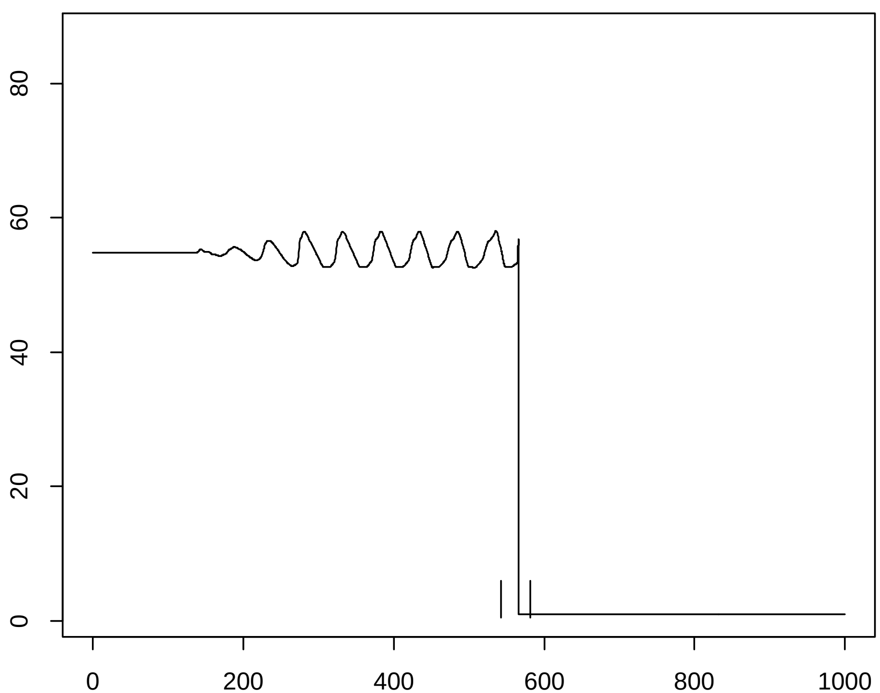
$$\partial_t (\rho Y) + \nabla \cdot (\rho Y \mathbf{u}) = -K(T)\rho Y$$

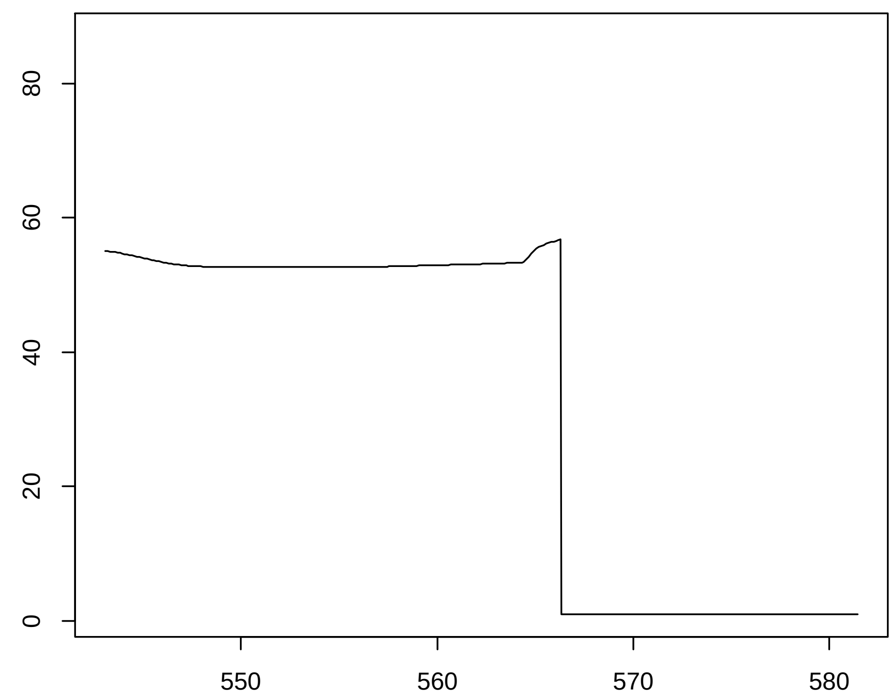
- Combustion model: unburnt gas $\xrightarrow{K(T)}$ burnt gas
  e.g., Arrhenius relation $K(T) = K_0 T^{\alpha} e^{-E^{+}/T}$
- EOS: $p = (\gamma - 1)(\rho e - q_0 Z)$, q_0: heat release

E: total energy, Y: unburnt gas mass fraction, $T = p/\rho R$

Detonation wave spatial structure

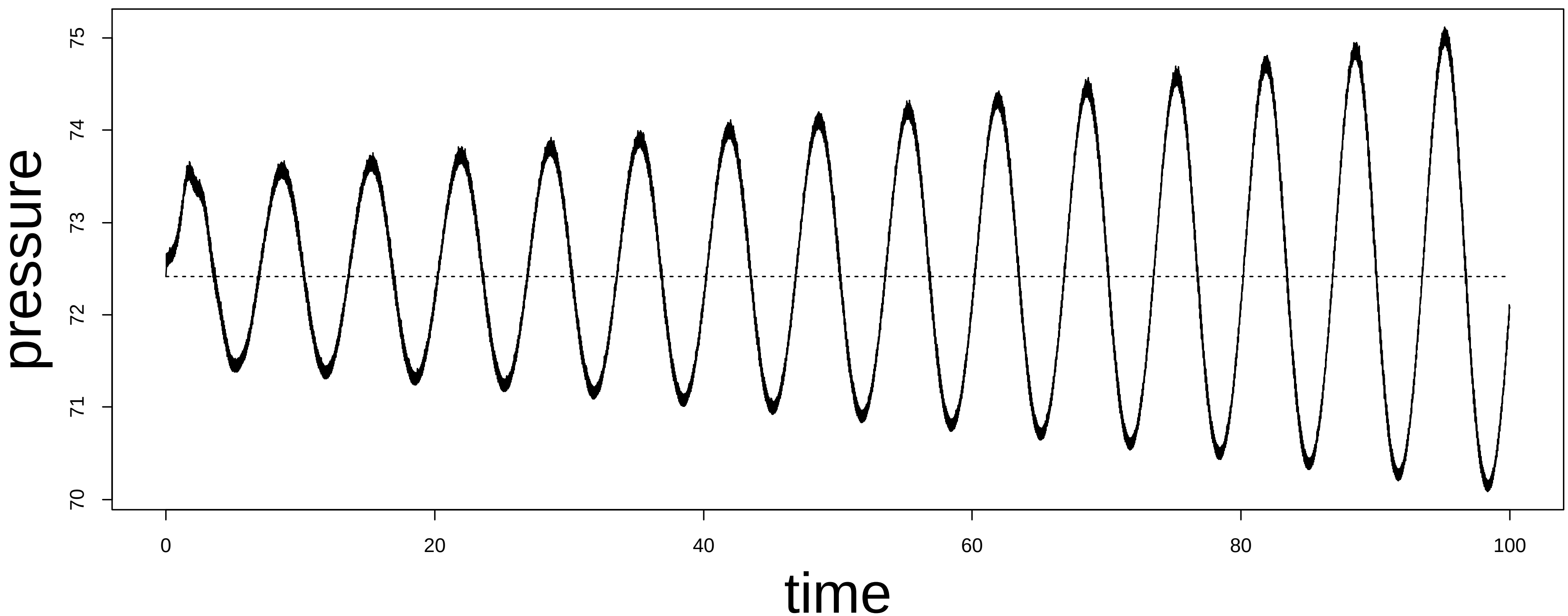
Spatial resolution of pressure for unstable (left) & stable (right) detonation waves





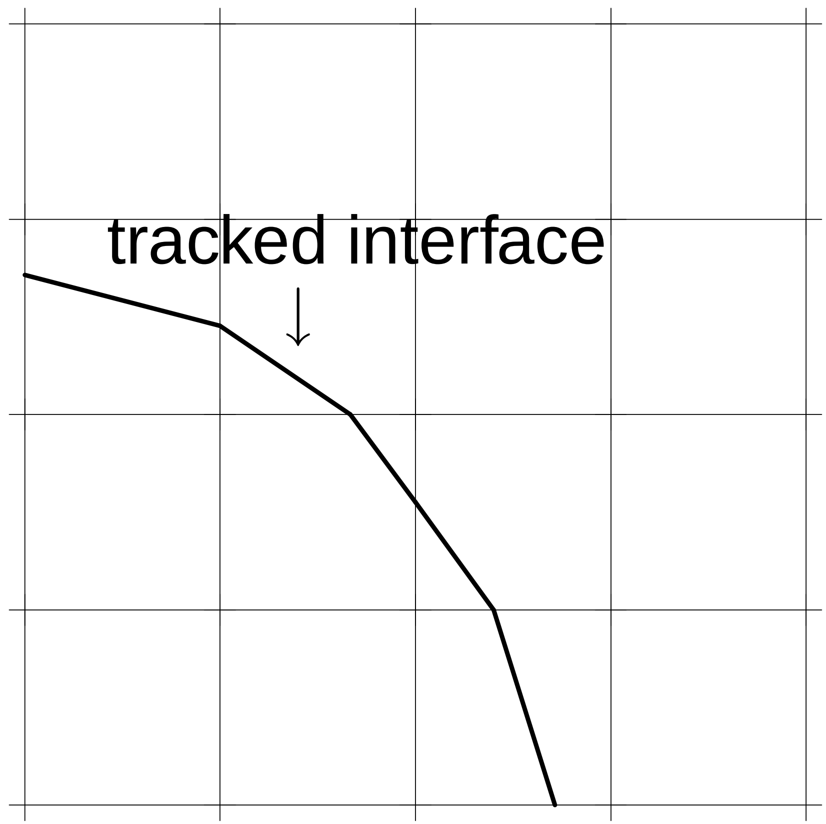
Unstable detonation wave

Shock front pressure history for unstable detonation with underdriven parameter $f = s/s_{\mathrm{CJ}} = 1.72$ (shock tracking with AMR is required)



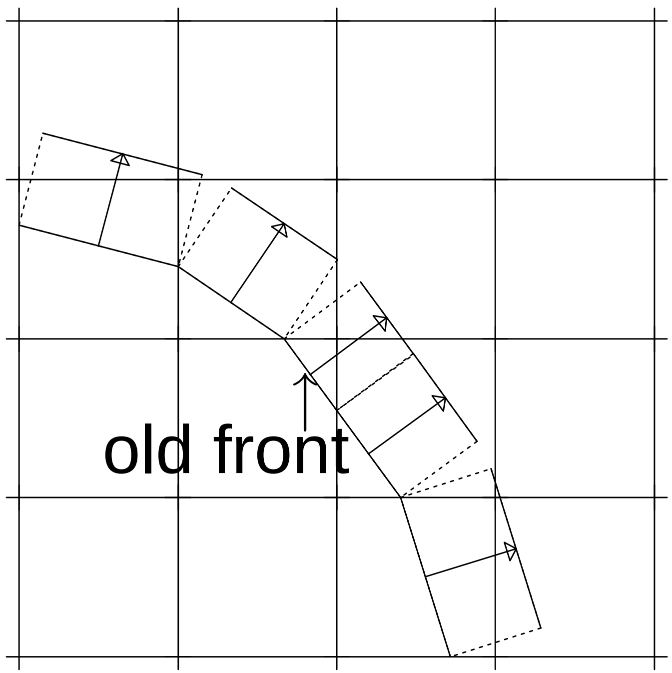
Cartesian cut-cell method in $2D$

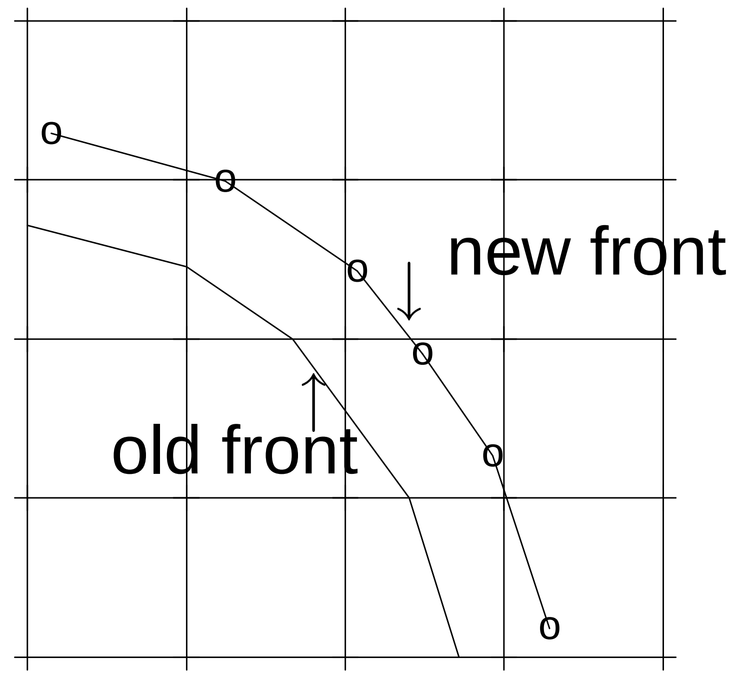
As before, our grid system consists of two parts: regular & irregular cells. Tracked interfaces are represented by piecewise linear segments.



MAC front moving procedure

1. Solve Riemann problems normal to tracked interfaces
2. Detect & follow strong waves of step 1 over time step Δt
3. Interpolate to get new front location





This approach works good for simple front but is not robust for complex topological change of front

VOF interface moving procedure

1. Volume fraction update
   Take a time step on current grid to update cell averages of volume fractions α governed by
   $$\partial_t \alpha + \mathbf{u} \cdot \nabla \alpha = 0$$
   at next time step

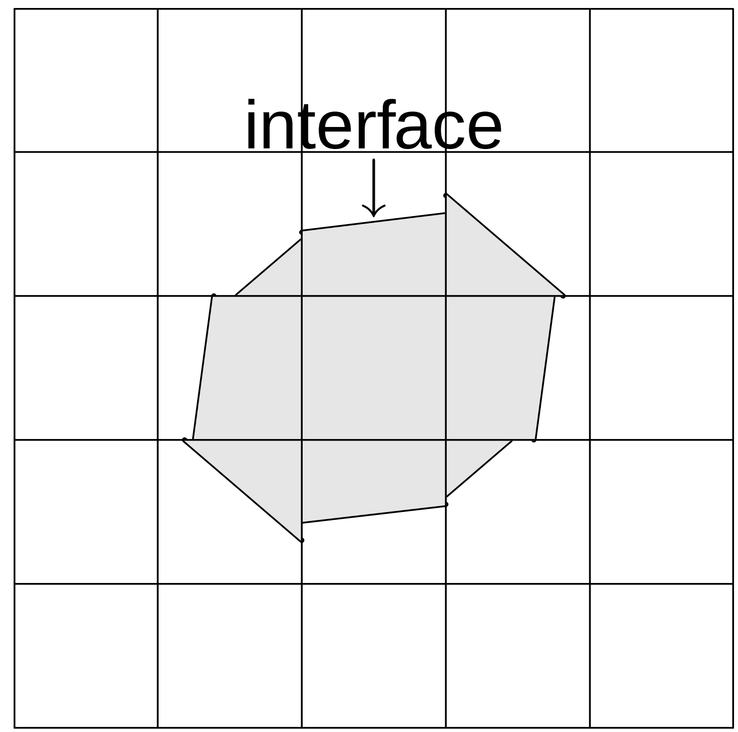
2. Interface reconstruction
   Given volume fractions on current grid, piecewise linear interface reconstruction (PLIC) method does:
   (a) Compute interface normal
      - Gradient or least squares method method of Youngs or Puckett

   (b) Determine interface location by iterative bisection

Interface reconstruction: Example

Cell-averaged volume fractions (left) & reconstructed interface (right)

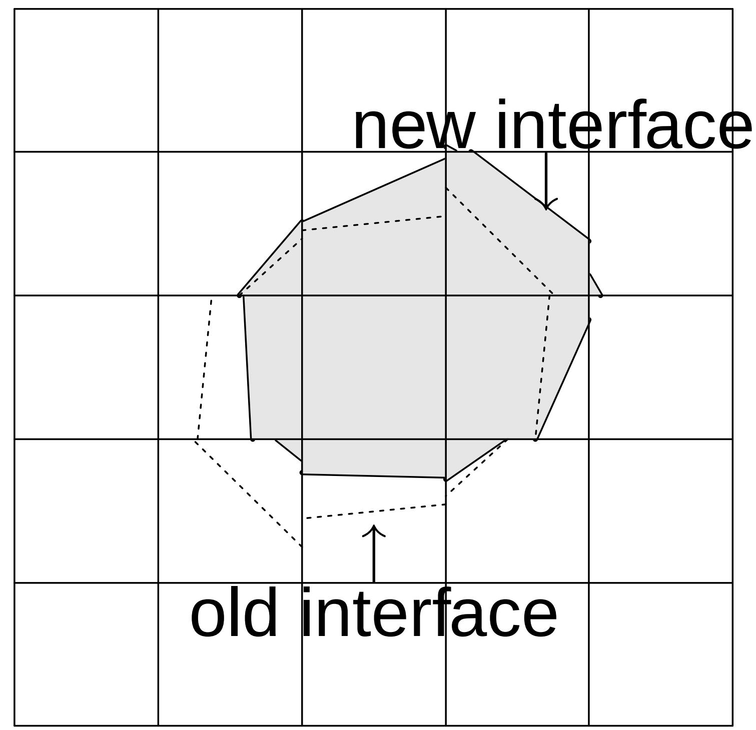
0	0	0	0	0
0	0.09	0.51	0.29	0
0	0.68	1	0.68	0
0	0.29	0.51	0.09	0
0	0	0	0	0



Interface reconstruction: Example

- Updated volume fraction (left) with $\mathbf{u} = (1, 1)$ over a time step $\Delta t = 0.06$, $\kappa_1 = 5.7 \times 10^{-3}$ & $\kappa_2 = 1.3 \times 10^{-3}$
- New reconstructed interface location (right)

0	0	0	κ_2	0
0	0.11	0.72	0.74	κ_1
0	0.38	1	0.85	0
0	0.01	0.25	0.06	0
0	0	0	0	0

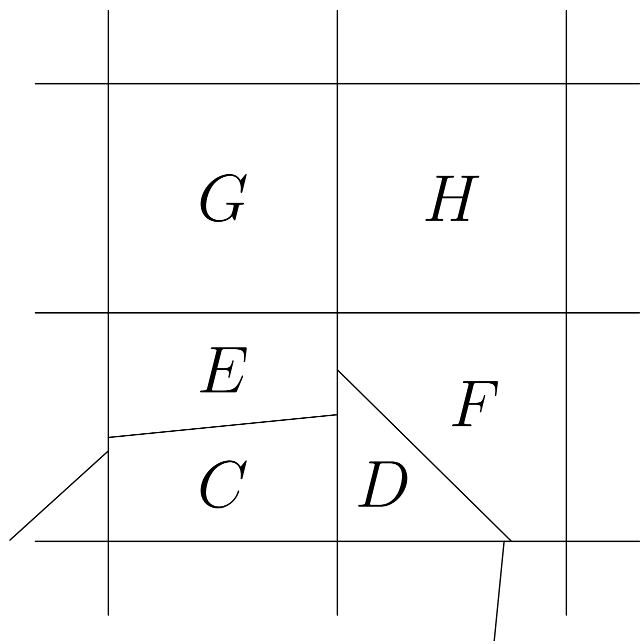


Solution update

Finite volume formulation of wave propagation method, Q_S^n gives approximate value of cell average of solution q over cell S at time t_n

$$Q_S^n \approx \frac{1}{\mathcal{M}(S)} \int_S q(X, t_n)\, dV$$

$\mathcal{M}(S)$: measure (area in 2D or volume in 3D) of cell S

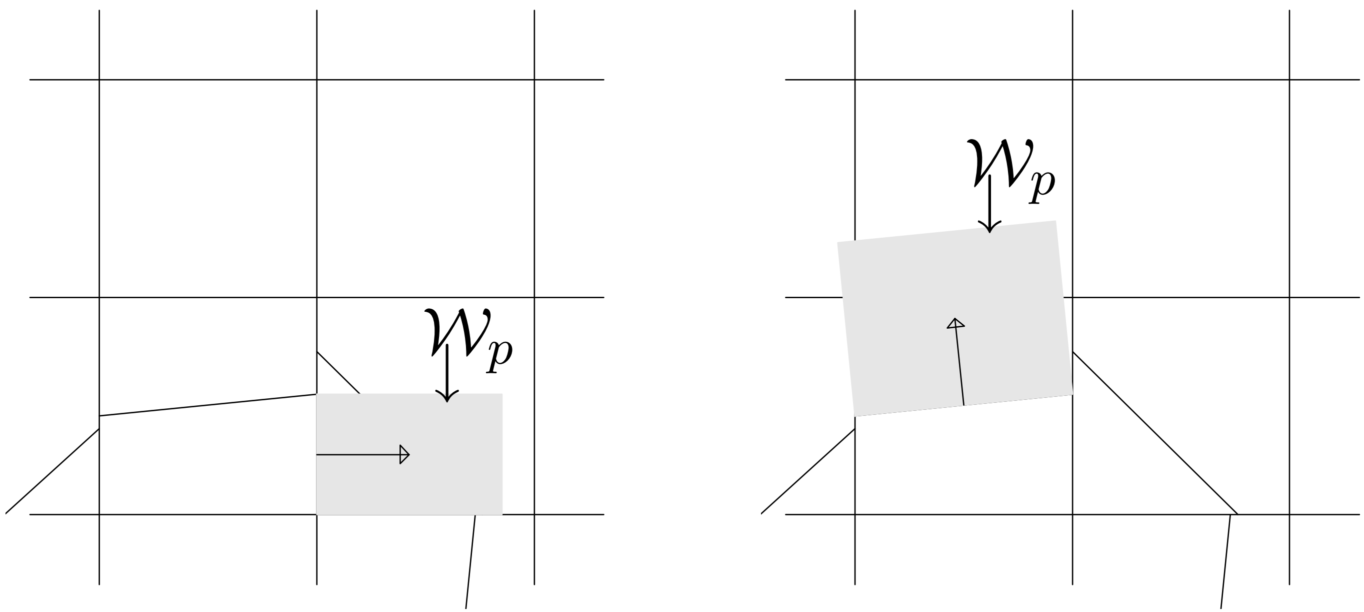


Normal-edge wave propagation

First order version: Piecewise constant wave update

- Fully discretized Godunov-type method: Solve Riemann problem at each cell edge in normal direction & use resulting waves to update cell averages whatever cells they affect

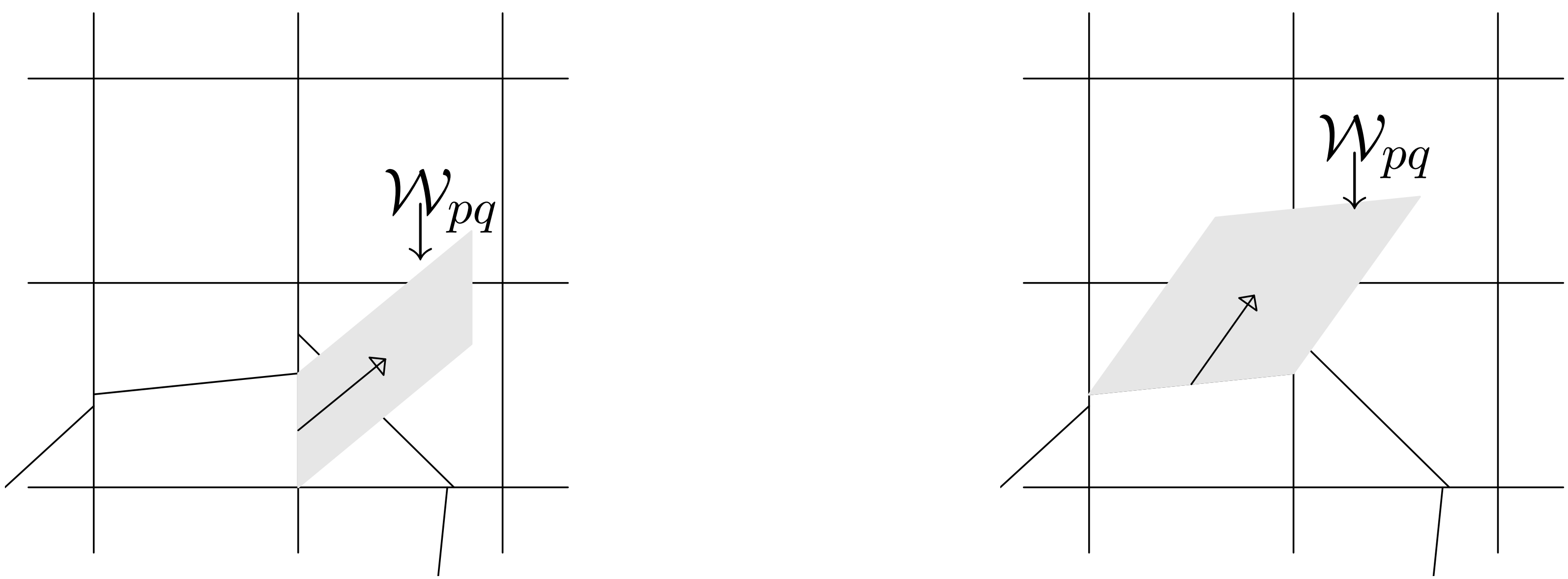
$$Q_S^{n+1} := Q_S^{n+1} - \frac{\mathcal{M}\left(\mathcal{W}_p \cap S\right)}{\mathcal{M}(S)} R_p, \qquad R_p \text{ being jump from RP}$$



Transverse wave propagation

First order version: Transverse-wave included

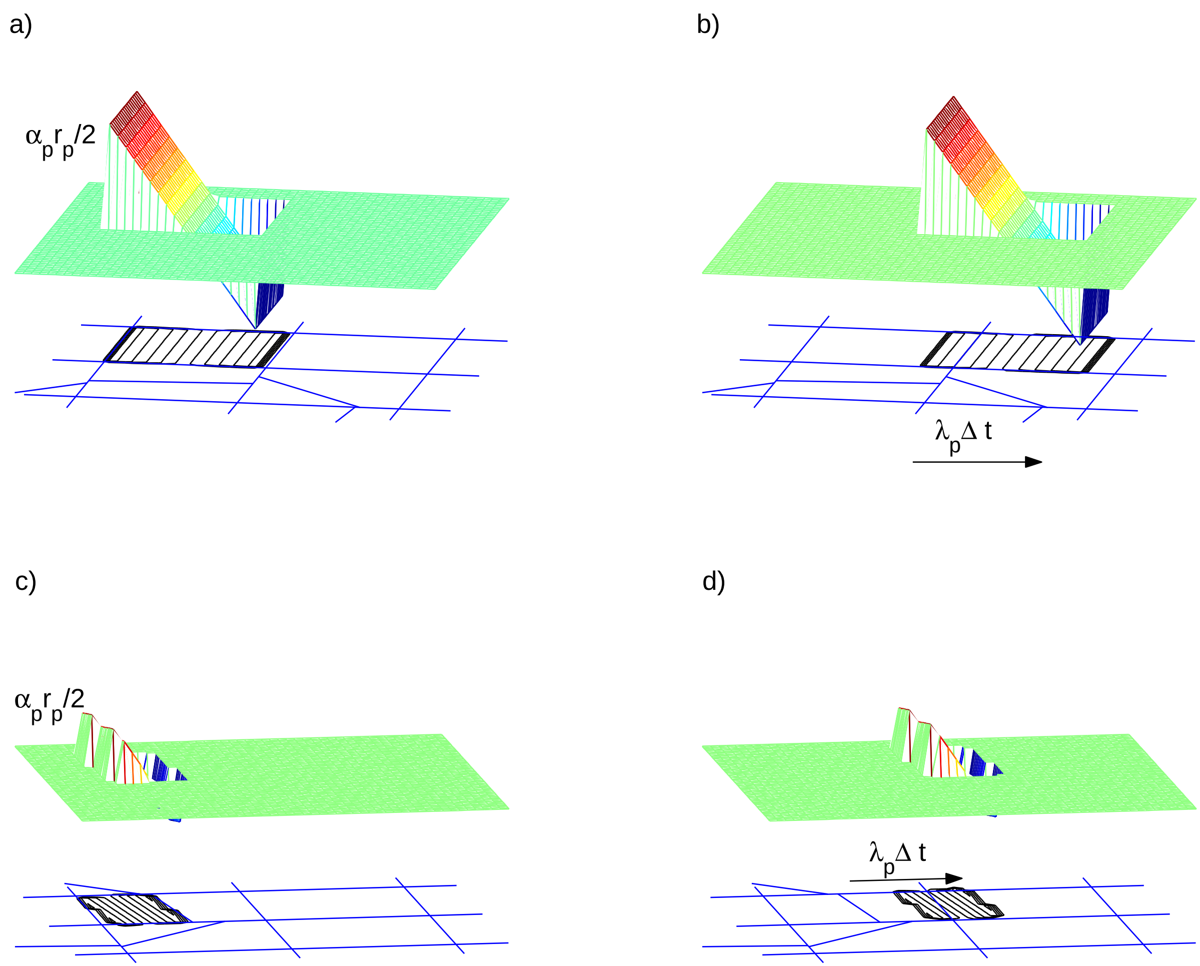
- Use transverse portion of equation, solve Riemann problem in transverse direction, & use resulting waves to update cell averages as usual
- Stability of method is typically improved, while conservation of method is maintained



High resolution correction

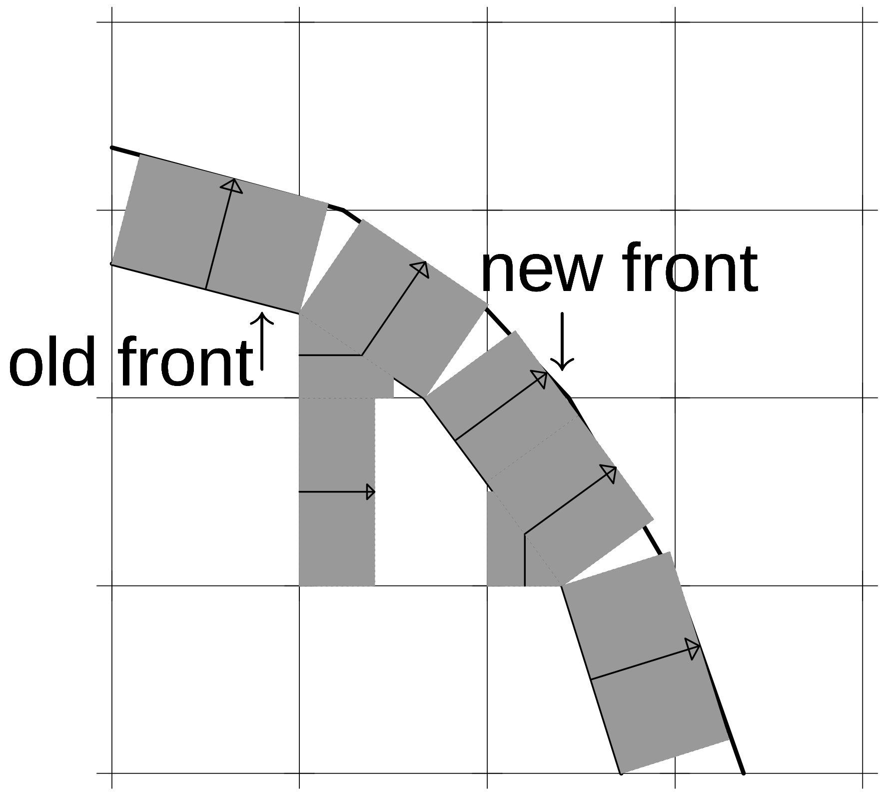
High resolution version: Piecewise linear wave update

wave before propagation after propagation



Cartesian cut-cell method: Remark

Little or no smearing of physical states in tracked wave family as illustrated below



Method remains stable with “large” time step chosen by

$$\nu = \Delta t \max_{p,q} (\lambda_p, \mu_q) \,/\, \min(\Delta x, \Delta y) \le 1$$

Front tracking: Advantages

- Tracked wave remain sharp
  - Avoid anomalous oscillations due to numerical smearing in a capturing method for interfaces such as slip line & material line, for example
- Provide valuable information on fronts for hybrid method (*e.g.*, couple front tracking with AMR) to solve multiscale problems
  - Useful for problems involving internal structure near discontinuities such as cosmic-ray modified flow & chemically-reacting detonation waves, or many MHD, RMHD, GRMHD flow

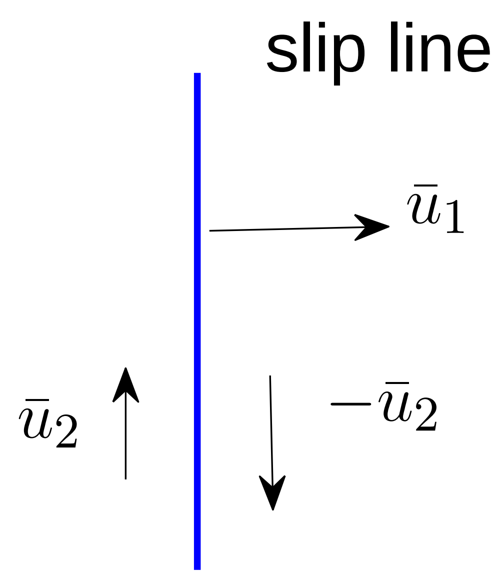
Numerical challenges to front tracking

- Small cell problems
  - Stringent limits on time step in presence of small cells created by tracked front cutting through grid
- Conservation of algorithm
- Second order accuracy near tracked front without post-front oscillations
- Front formation & wave interactions in multiple dimensions
- Robust algorithm for front moving, bifurcation & topological changes
- Efficient numerical implementation, in particular, in 3D

Slip line (shear flow) problem

To show anomalous oscillations obtained using state-of-the-art capturing method, we consider a plane right-moving interface for ideal gas in x_1-direction. Interface conditions for this problem are

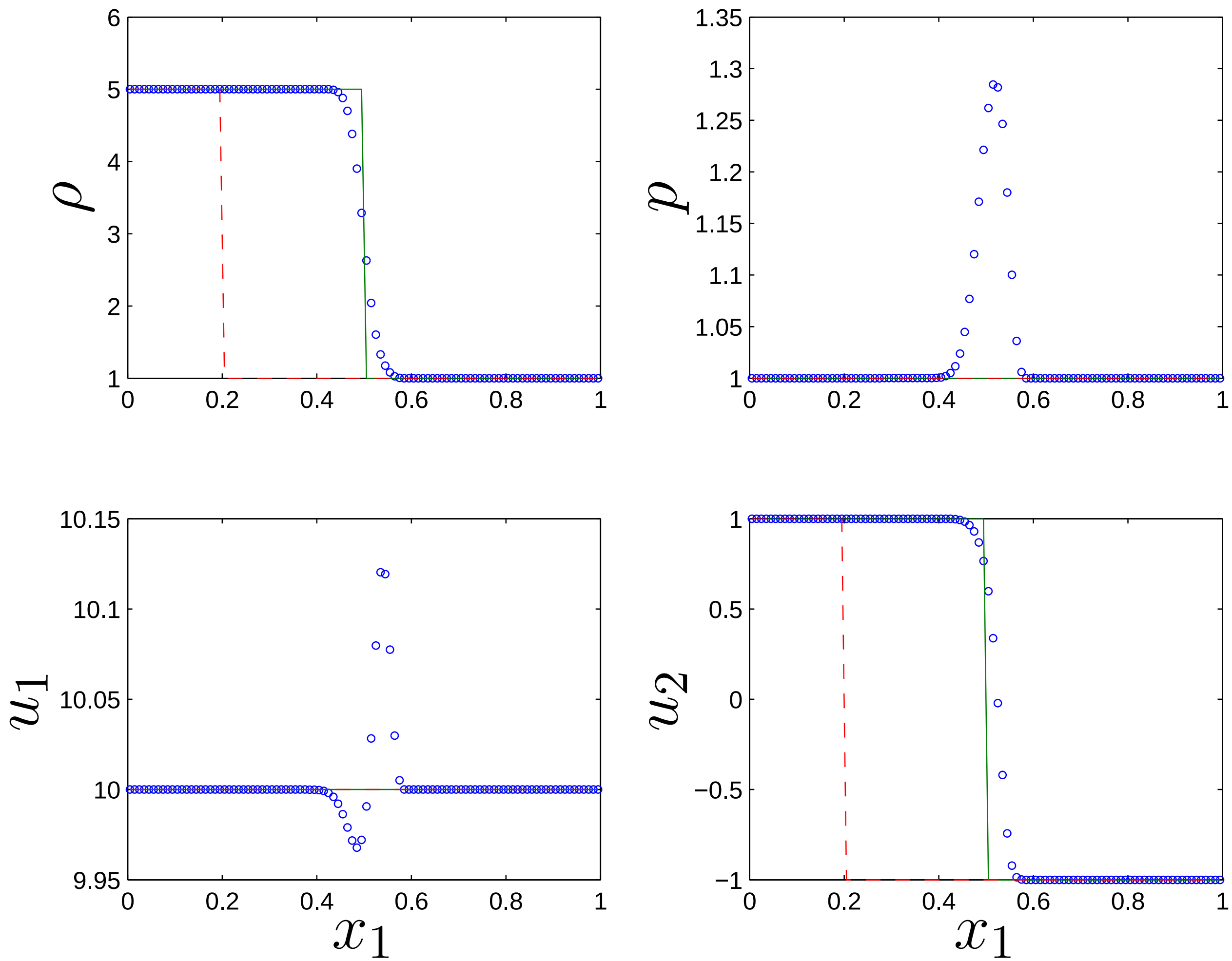
- Dynamic condition: $p_R = p_L$
- Kinematic condition: $u_{1,R} = u_{1,L}$ & $(u_{2,R} - u_{2,L}) \neq 0$



Slip line problem: Example

Example obtained by using a Godunov-type method

- Errors depend strongly on transverse velocity jump



Slip line problem: Source of error

To ensure pressure equilibrium, as it should be for this slip line problem, motion of transverse-kinetic energy $\rho u_2^2/2$ is

$$\partial_t \left(\rho u_2^2/2\right) + \bar{u}_1 \partial_{x_1} \left(\rho u_2^2/2\right) = 0$$

To compute pressure, from EOS using conservative variables,

$$p = (\gamma - 1)\left(E - \sum_{i=1}^{2} (\rho u_i)^2/2\rho\right)$$

while generally $(\rho u_2)^2/2\rho \neq \rho u_2^2/2$

When a slip line is smeared out, yielding loss of pressure equilibrium & so incorrect solution of other variables

Slip line problem: Improvement

To devise a more accurate method for numerical resolution of slip lines, we may use

- Diffuse interface approach
  - Include transverse kinetic energy equation in the model & use its solution for pressure update

$$p = (\gamma - 1)\left(E - \frac{(\rho u_1)^2}{2\rho} + \frac{\rho u_2^2}{2} \right)$$

  This transverse kinetic equation should be modified so that there is no difficulty to work with shock waves

- Sharp interface approach
  - Front tracking or Lagrangian moving grid method

Material Line Problem

Consider a plane material line, separating regions of two different fluid phases. Assume ideal gas law for each phase: $p_k(\rho, e) = (\gamma_k - 1)\rho e$, $\gamma_1 \neq \gamma_2$
To ensure pressure equilibrium, from energy eq.

$$\partial_t \left(\frac{p}{\gamma - 1} \right) + \bar{u}_1 \partial_{x_1} \left(\frac{p}{\gamma - 1} \right) + \partial_t \left(\frac{1}{2} \rho u_2^2 \right) + \bar{u}_1 \partial_{x_1} \left(\frac{1}{2} \rho u_2^2 \right) = 0$$

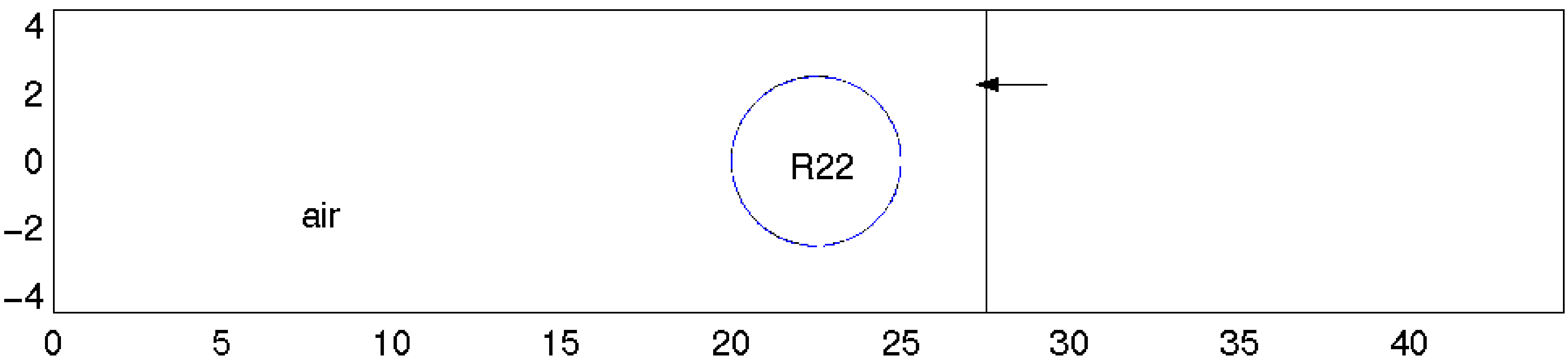
yielding two constraints that should be satisfied numerically,

$$\partial_t \left(\frac{1}{2} \rho u_2^2 \right) + \bar{u}_1 \partial_{x_1} \left(\frac{1}{2} \rho u_2^2 \right) = 0$$

$$\partial_t \left(\frac{1}{\gamma - 1} \right) + \bar{u}_1 \partial_{x_1} \left(\frac{1}{\gamma - 1} \right) = 0$$

Compressible two phase flow

Consider popular shock-bubble interaction for example of compressible fluid mixing



Two-phase flow model

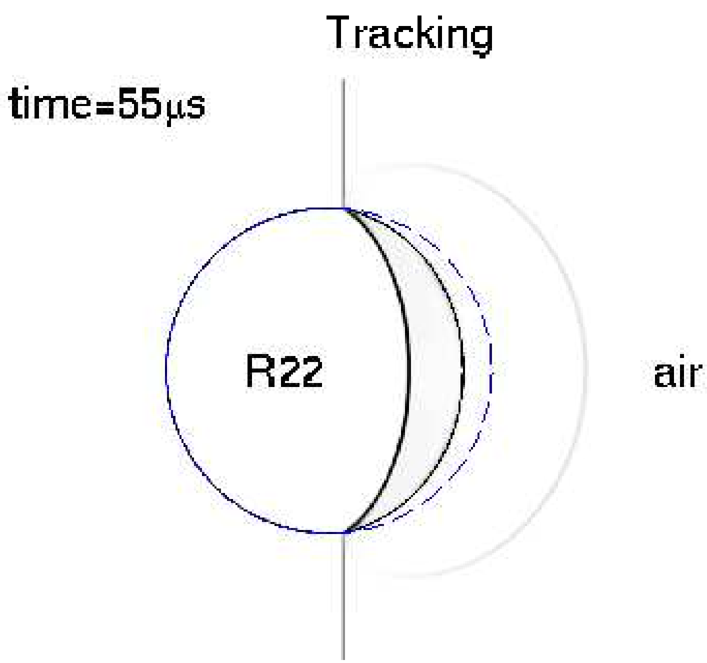
- Equation of motion: Kapila *et al.* two-phase flow model

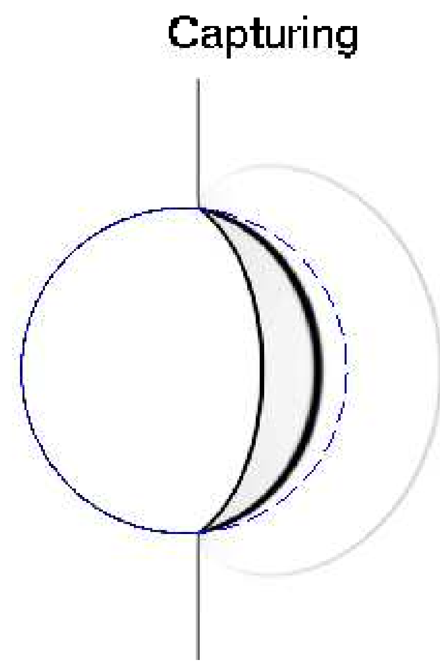
$$\partial_t\left(\alpha_1\rho_1\right)+\nabla\cdot\left(\alpha_1\rho_1\mathbf{u}\right)=0$$
$$\partial_t\left(\alpha_2\rho_2\right)+\nabla\cdot\left(\alpha_2\rho_2\mathbf{u}\right)=0$$
$$\partial_t\left(\rho\mathbf{u}\right)+\nabla\cdot\left(\rho\mathbf{u}\otimes\mathbf{u}+pI\right)=0$$
$$\partial_t E+\nabla\cdot\left(E\mathbf{u}+p\mathbf{u}\right)=0$$
$$\partial_t\alpha_2+\mathbf{u}\cdot\nabla\alpha_2=\alpha_1\alpha_2\left(\frac{\rho_1c_1^2-\rho_2c_2^2}{\sum_{k=1}^{2}\alpha_k\rho_kc_k^2}\right)\nabla\cdot\mathbf{u}$$

- Mixture equation of state: $p=p(\alpha_2,\ \alpha_1\rho_1,\ \alpha_2\rho_2,\ \rho e)$ with isobaric closure: $p_1=p_2=p$

Shock-bubble interaction

Schlieren-type image: front tracking (left) & front capturing (right)





Shock-bubble interaction

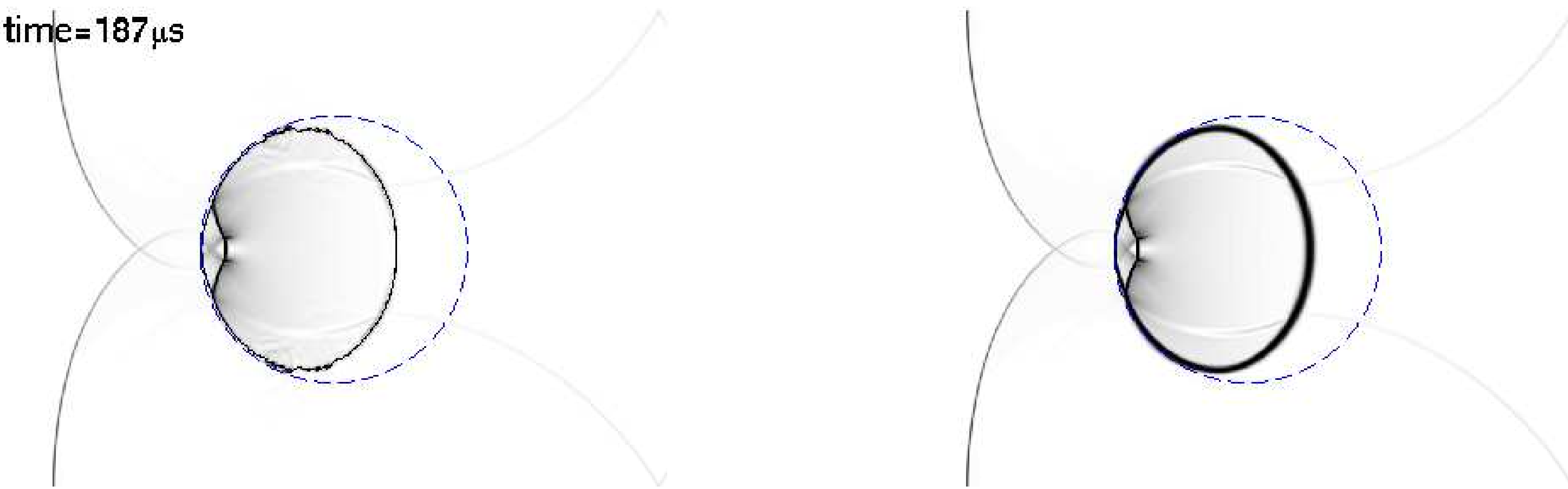
Schlieren-type image: front tracking (left) & front capturing (right)

Shock-bubble interaction

Schlieren-type image: front tracking (left) & front capturing (right)

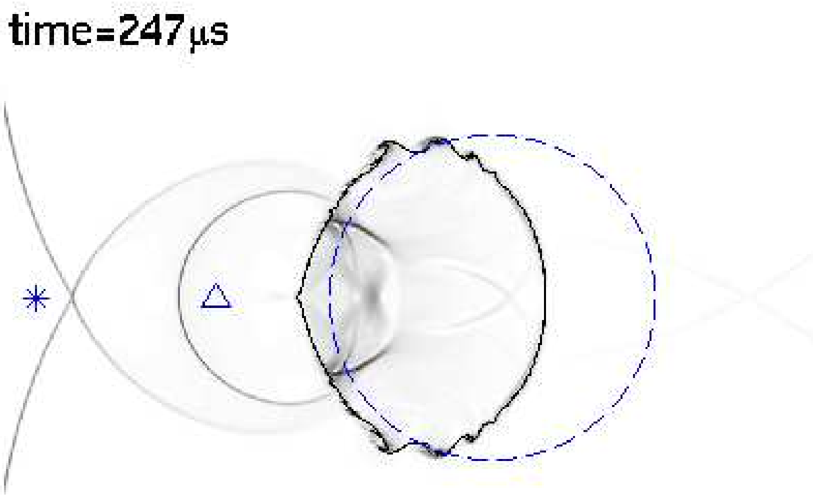
Shock-bubble interaction

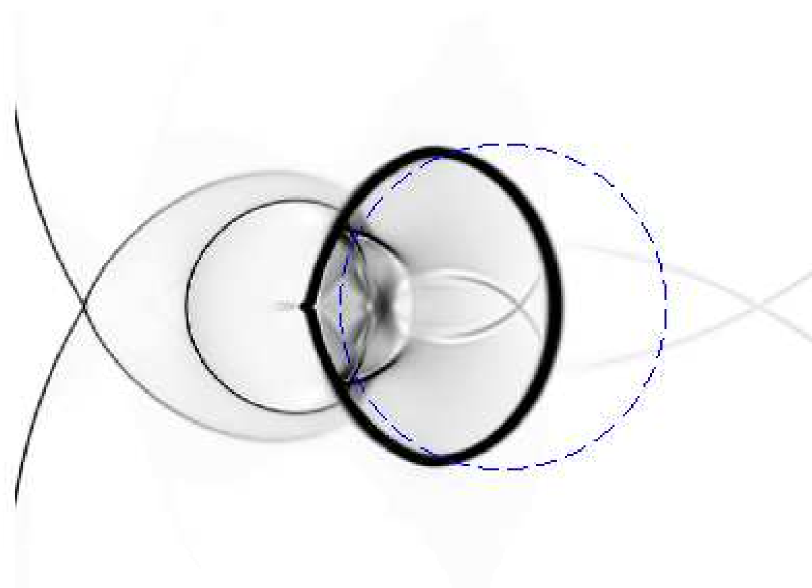
Schlieren-type image: front tracking (left) & front capturing (right)



Shock-bubble interaction

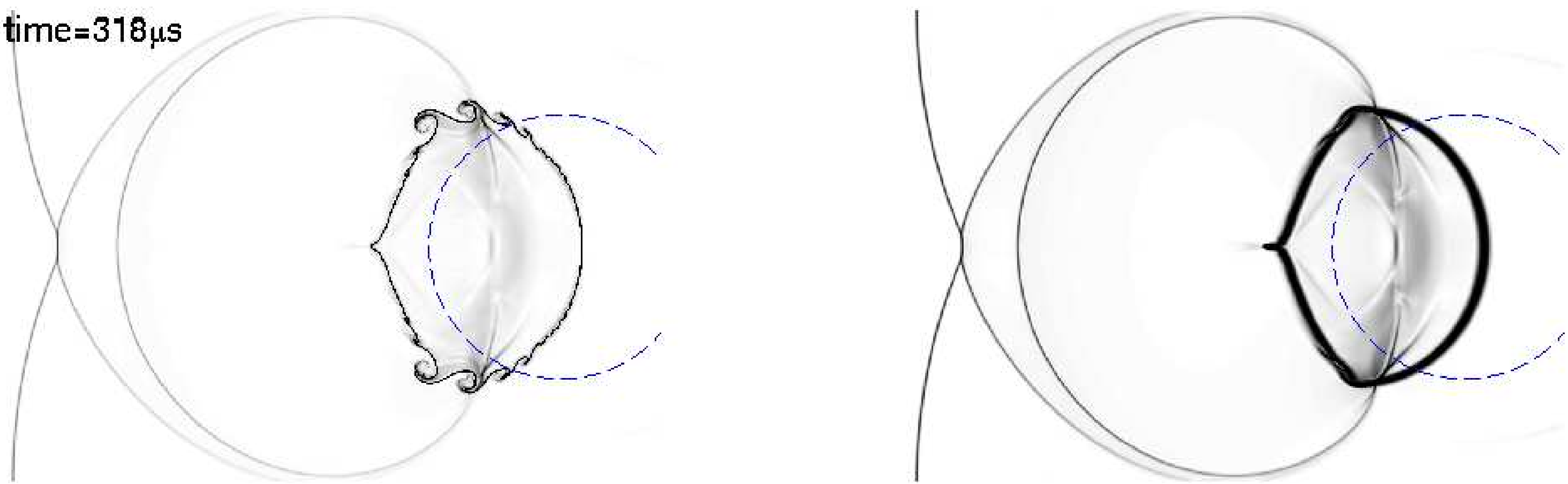
Schlieren-type image: front tracking (left) & front capturing (right)





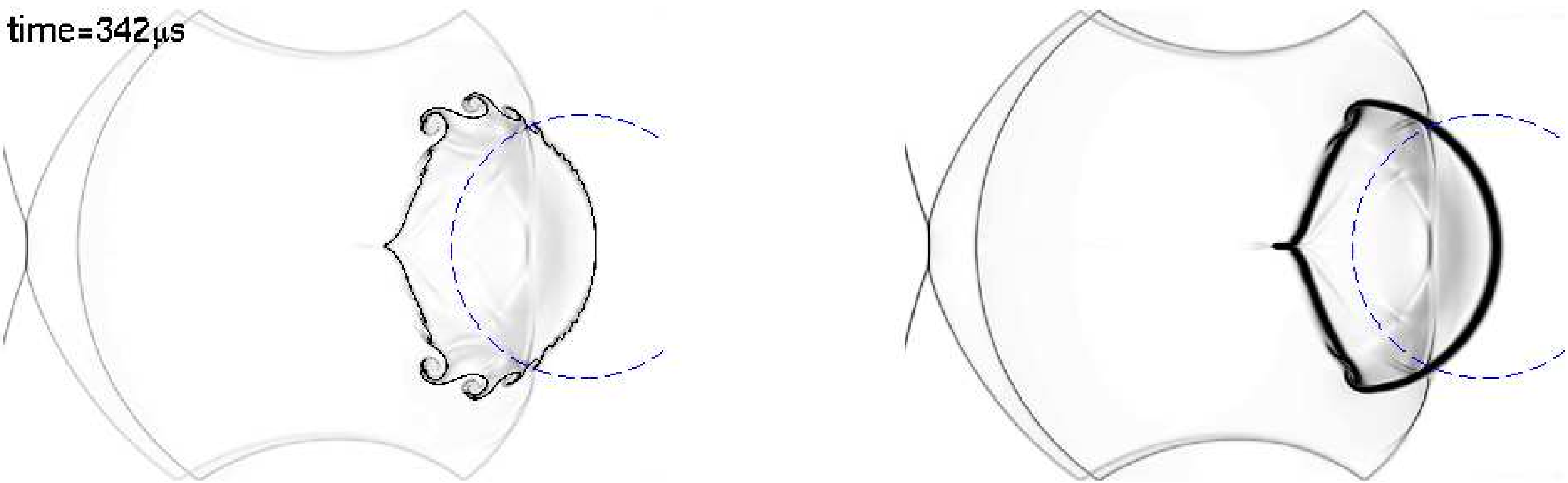
Shock-bubble interaction

Schlieren-type image: front tracking (left) & front capturing (right)



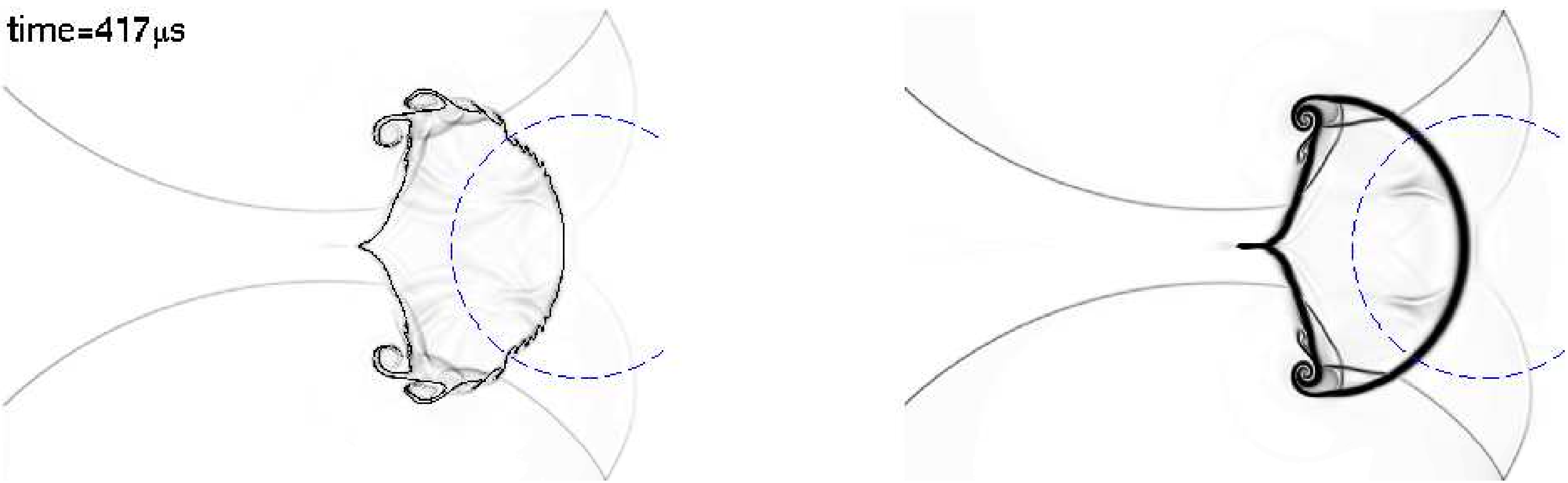
Shock-bubble interaction

Schlieren-type image: front tracking (left) & front capturing (right)



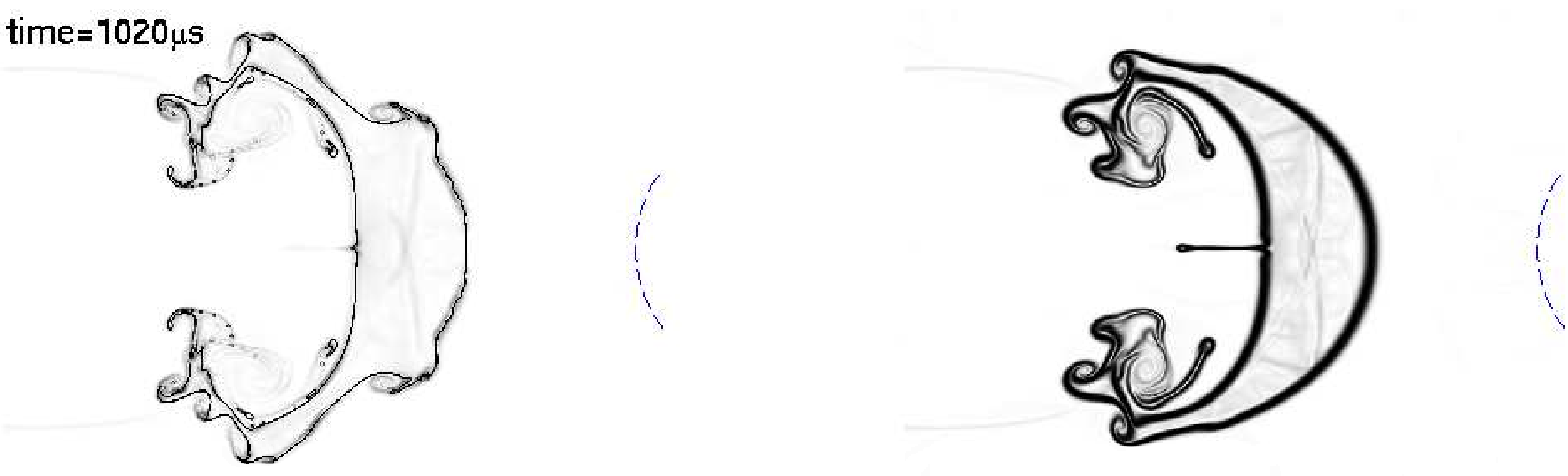
Shock-bubble interaction

Schlieren-type image: front tracking (left) & front capturing (right)



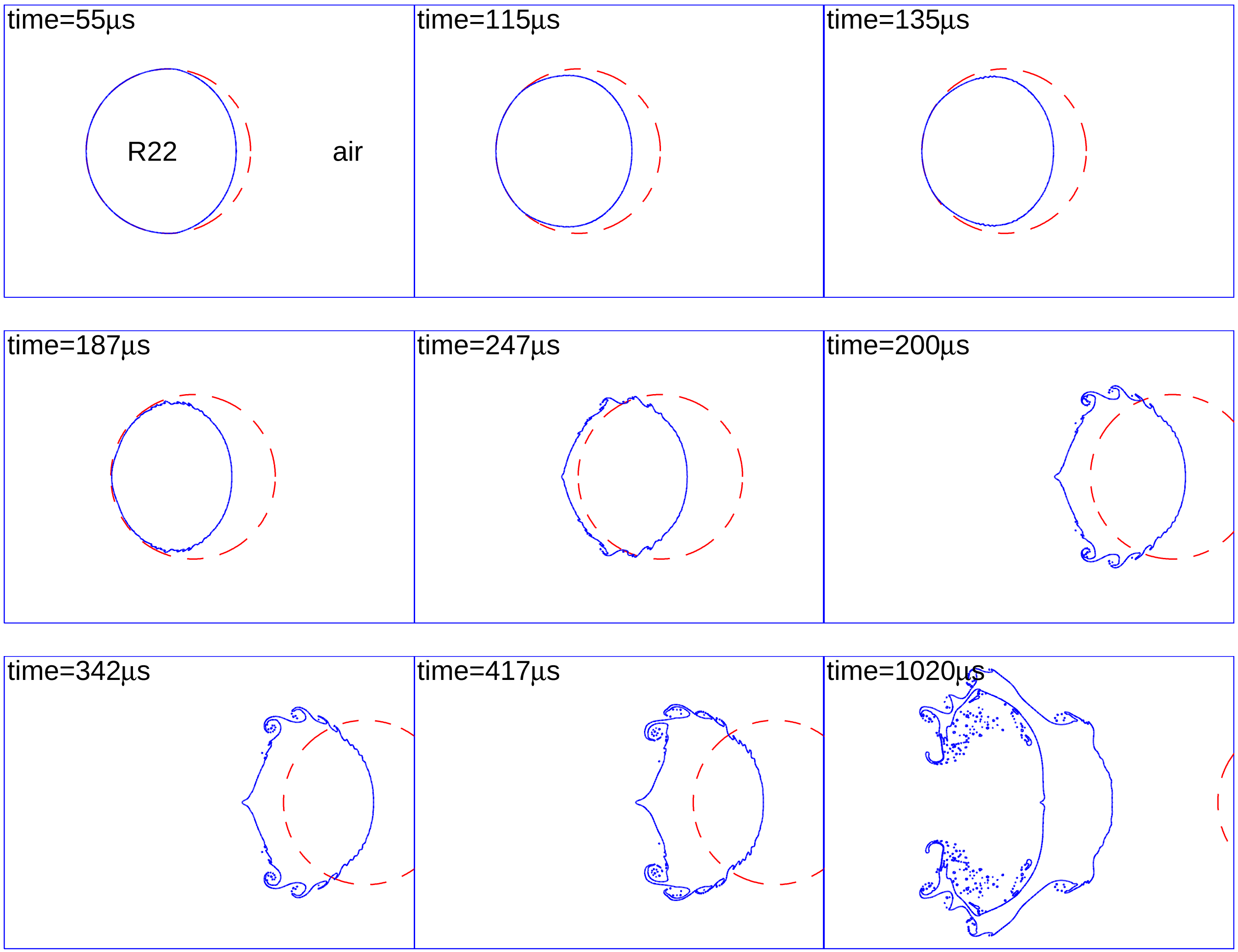
Shock-bubble interaction

Schlieren-type image: front tracking (left) & front capturing (right)



Shock-bubble interaction

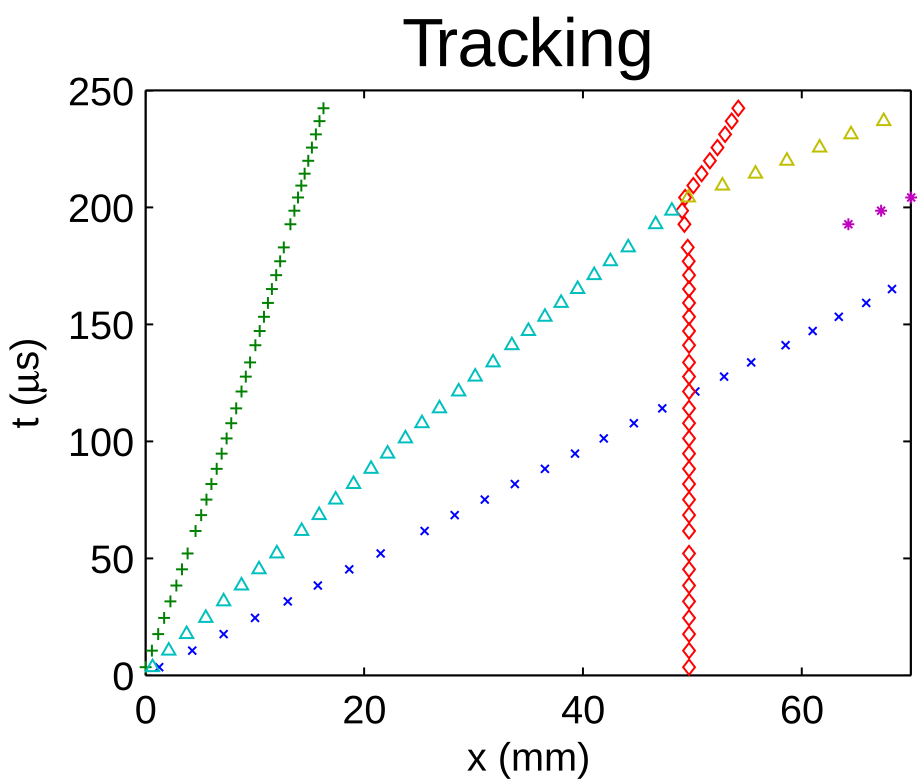
Approximate locations of interfaces

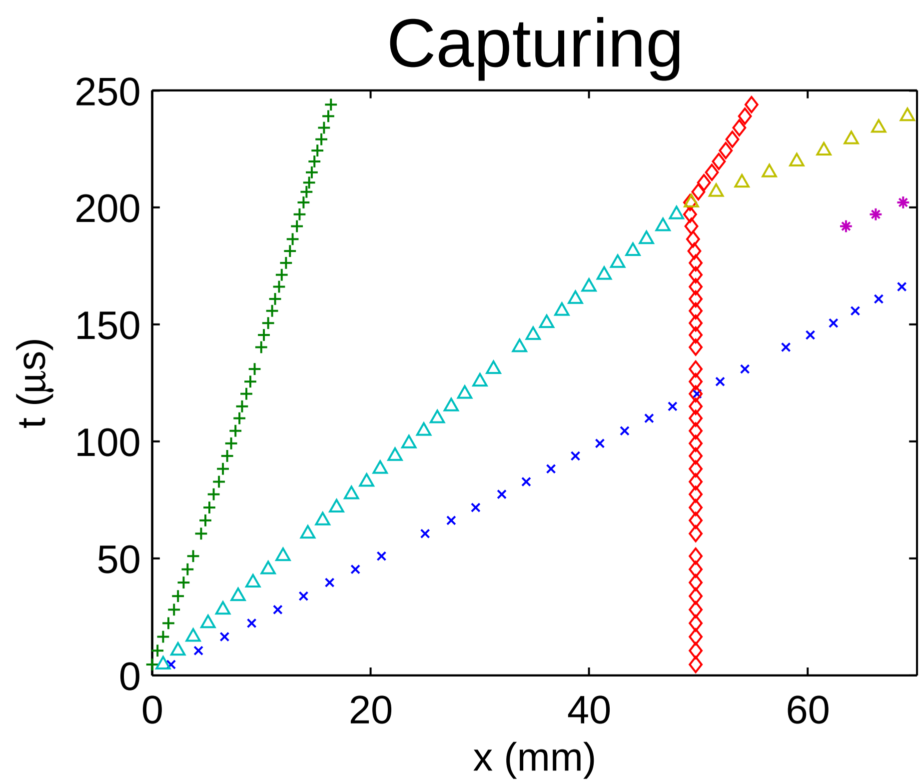


Shock-bubble interaction

Space-time locations of prominent waves

$\times$ (incident shock), $+$ (upstream bubble), $\diamond$ (downstream bubble), $\triangle$ (refracted shock), $*$ & $\triangle$ (transmitted shock)





Shock-bubble interaction

Quantitative assessment of prominent flow velocities

Velocity (m/s)	V_s	V_R	V_T	V_{ui}	V_{uf}	V_{di}	V_{df}
Haas & Sturtevant	415	240	540	73	90	78	78
Quirk & Karni	420	254	560	74	90	116	82
Our result (tracking)	411	243	538	64	87	82	60
Our result (capturing)	411	244	534	65	86	98	76

- V_s (V_R, V_T) Incident (refracted, transmitted) shock speed $t \in [0, 250]\mu$s ($t \in [0, 202]\mu$s, $t \in [202, 250]\mu$s)
- V_{ui} (V_{uf}) Initial (final) upstream bubble wall speed $t \in [0, 400]\mu$s ($t \in [400, 1000]\mu$s)
- V_{di} (V_{df}) Initial (final) downstream bubble wall speed $t \in [200, 400]\mu$s ($t \in [400, 1000]\mu$s)

Mapped grid method

Integral form of conservation laws

$$\partial_t q + \nabla \cdot f(q) = 0$$

over any control volume C is

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

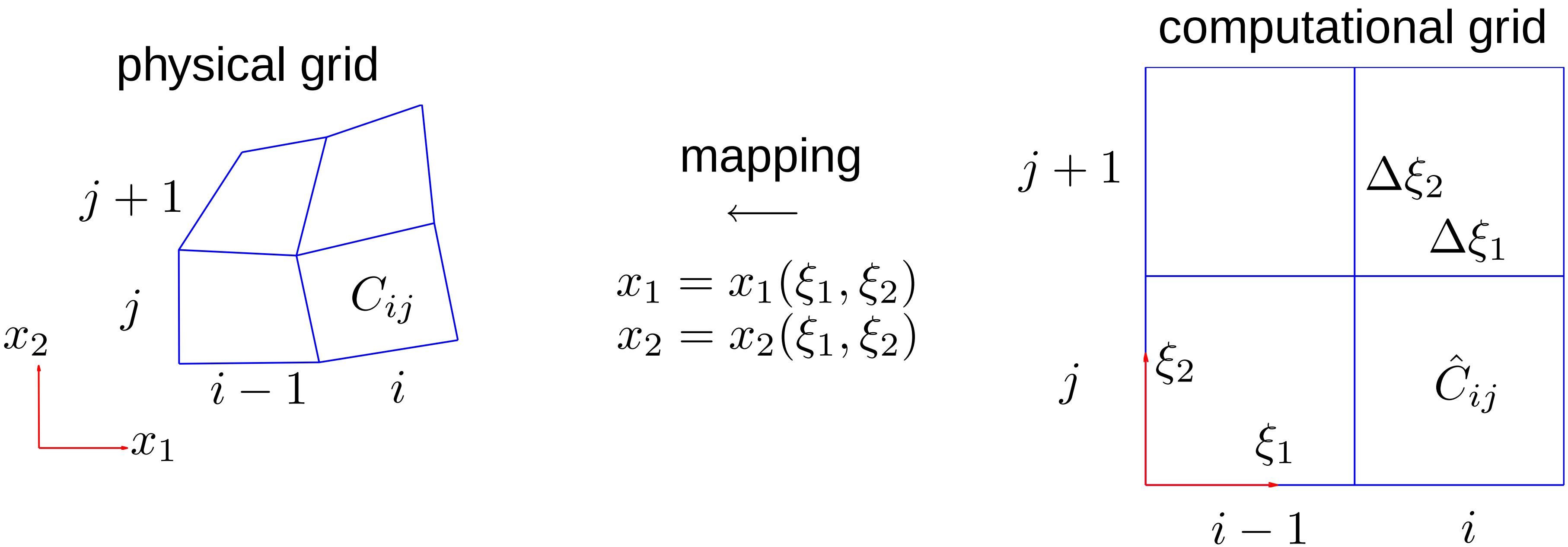
A finite volume method on a control volume C takes

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

$\mathcal{M}(C)$ is measure (area in 2D or volume in 3D) of C, N_s is number of sides, h_j is length (in 2D) or area (in 3D) of j-th side, $\breve{F}_j$ is approx. normal flux in average across j-th side

Mapped grid method

Assume that our mapped grids are logically rectangular, & will restrict our consideration to 2D as illustrated below



On a curvilinear grid, a finite volume method takes

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

Mapped grid method

On a curvilinear grid, a finite volume method takes

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

$\Delta\xi_1$, $\Delta\xi_2$ denote spatial size of comput. 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 comput. grid

$F^1_{i-\frac{1}{2},j} = \gamma_{i-\frac{1}{2},j}\breve{F}_{i-\frac{1}{2},j}$, $F^2_{i,j-\frac{1}{2}} = \gamma_{i,j-\frac{1}{2}}\breve{F}_{i,j-\frac{1}{2}}$ are fluxes per unit length in comput. 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 method

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-, & 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 (not discussed here)

High resolution corrections

Speeds & limited versions of waves are used to calculate second order correction terms. These terms are added to method in flux difference form as

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

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

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

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

High resolution corrections

To ensure second order accuracy & also improve stability, a transverse wave propagation is included in 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

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

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

Accuracy test in 2D

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

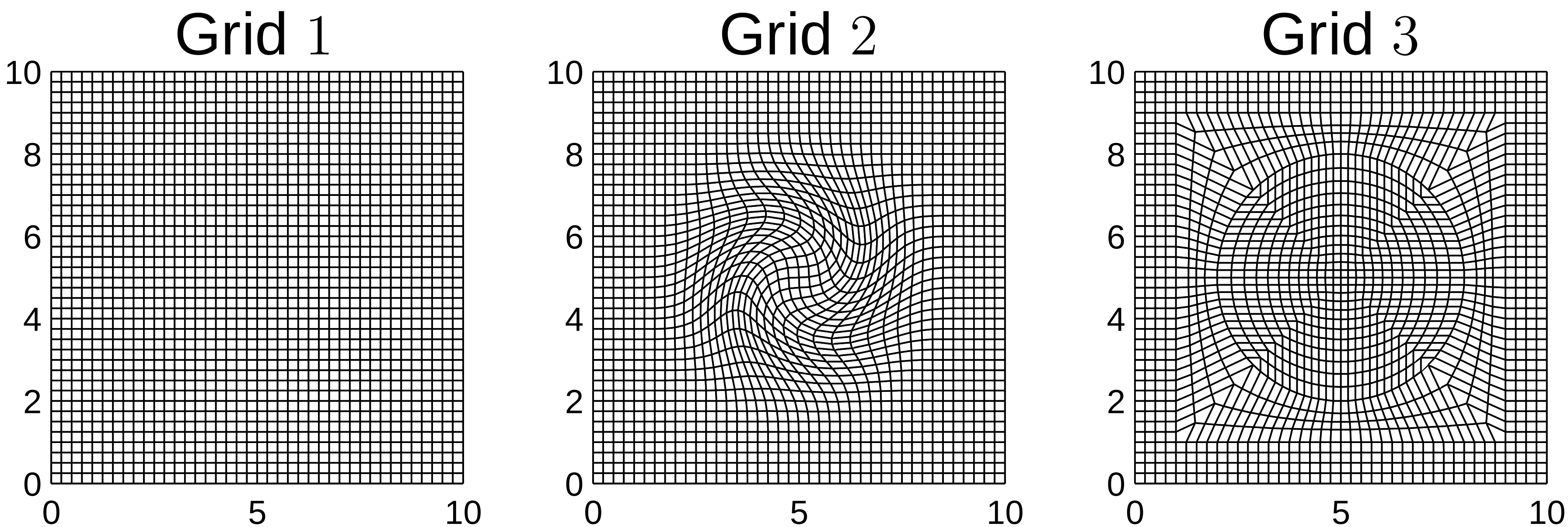
$$
\begin{aligned}
\rho &= \left(1 - \frac{25(\gamma-1)}{8\gamma\pi^2}\exp\left(1-r^2\right)\right)^{1/(\gamma-1)} \\
p &= \rho^\gamma \\
u_1 &= 1 - \frac{5}{2\pi}\exp\left(\left(1-r^2\right)/2\right)\left(x_2-5\right) \\
u_2 &= 1 + \frac{5}{2\pi}\exp\left(\left(1-r^2\right)/2\right)\left(x_1-5\right)
\end{aligned}
$$

& periodic boundary conditions as an example,

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

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

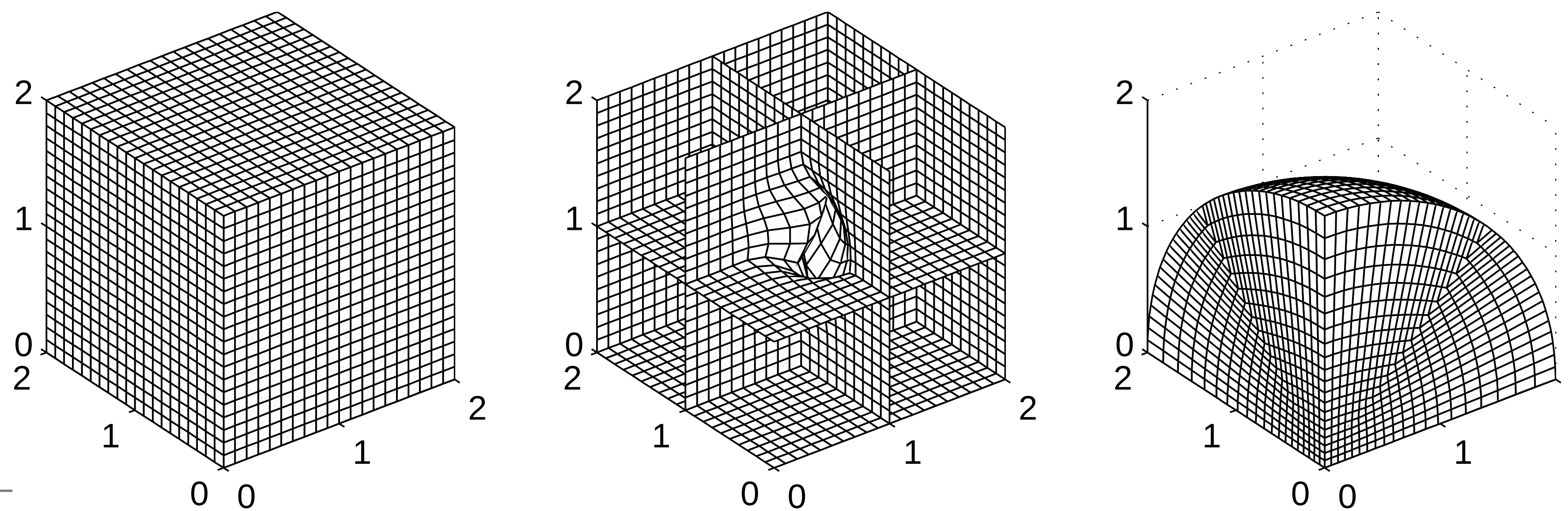
N	$\mathcal{E}_1(\rho)$	Order	$\mathcal{E}_1(u_1)$	Order	$\mathcal{E}_1(u_2)$	Order	$\mathcal{E}_1(p)$	Order
40	4.8272		4.7734		5.3367		5.4717	
80	1.5740	1.62	1.5633	1.61	1.5660	1.77	1.5634	1.81
160	0.4536	1.79	0.4559	1.78	0.4537	1.79	0.4560	1.78
320	0.1215	1.90	0.1221	1.90	0.1222	1.89	0.1221	1.90

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 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	$\mathcal{E}_1(\lvert\vec{u}\rvert)$	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	$\mathcal{E}_\infty(\lvert\vec{u}\rvert)$	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)

  Solve $$\nabla \cdot (D \nabla \xi_j) = 0, \quad j = 1, \ldots, N_d$$

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

- Variational approach (Tang & many others)

  Solve $$\nabla_\xi \cdot (D \nabla_\xi x_j) = 0, \quad j = 1, \ldots, N_d$$

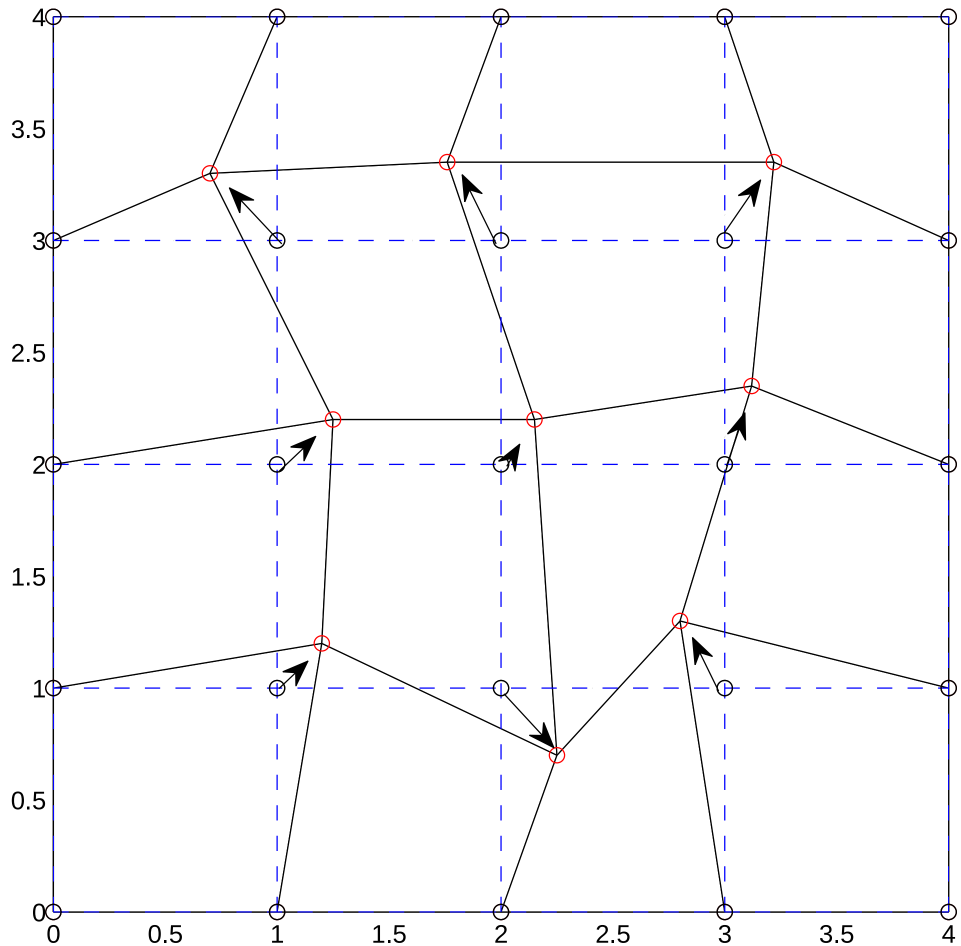
  for $\mathbf{x}(\xi)$ that minimizes "energy" functional

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

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

Mesh redistribution: Example

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.*

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

after each mesh redistribution iterate k. This can be done

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

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

- Geometric approach (Shyue 2010 & others)

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

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

Interpolation-free moving mesh

To avoid averaging error in conservative interpolation step, one approach is to dervise an interpolation-free moving mesh method

To do so, consider coordinate change of equations via $(\mathbf{x},\ t) \mapsto (\xi,\ t)$, yielding transformed conservation law

$$\partial_t \tilde{q} + \nabla_\xi \cdot \tilde{f} = J\psi + \mathcal{G}$$

$$\tilde{q} = Jq, \quad \tilde{f}_j = J\left(q\, \partial_t \xi_j + \nabla \xi_j \cdot f\right), \quad J = \text{det}\left(\partial \xi / \partial \mathbf{x}\right)^{-1}$$

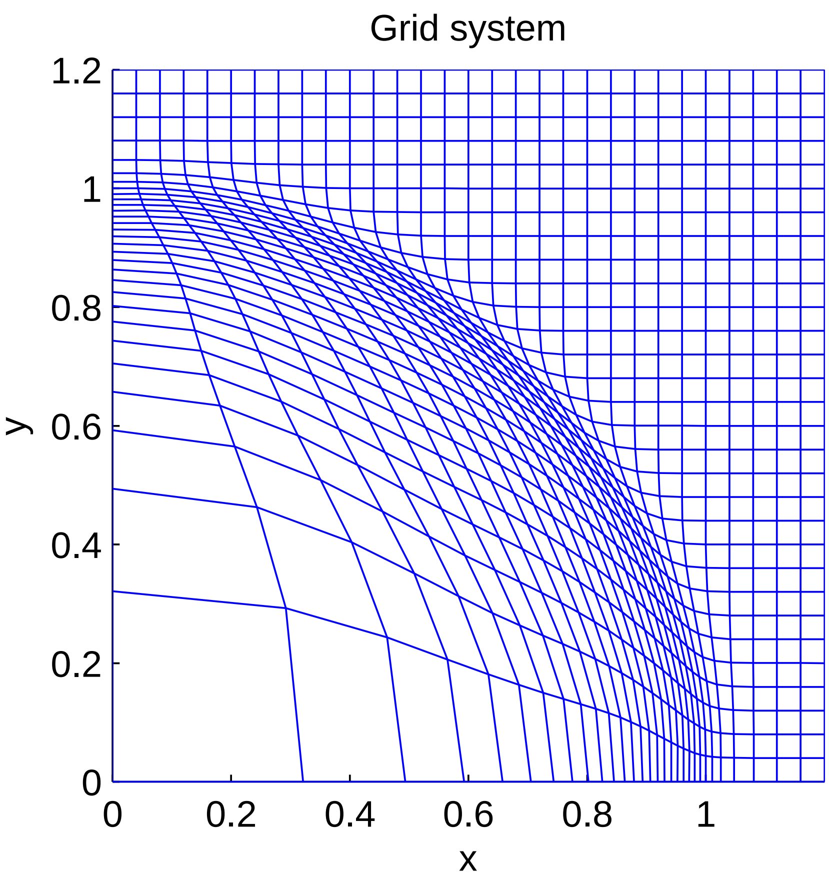
$$\mathcal{G} = q\left[\partial_t J + \nabla_\xi \cdot (J \partial_t \xi_j)\right] + \sum_{j=1}^{N} f_j \nabla_\xi \cdot \left(J \partial_{x_j} \xi_k\right)$$

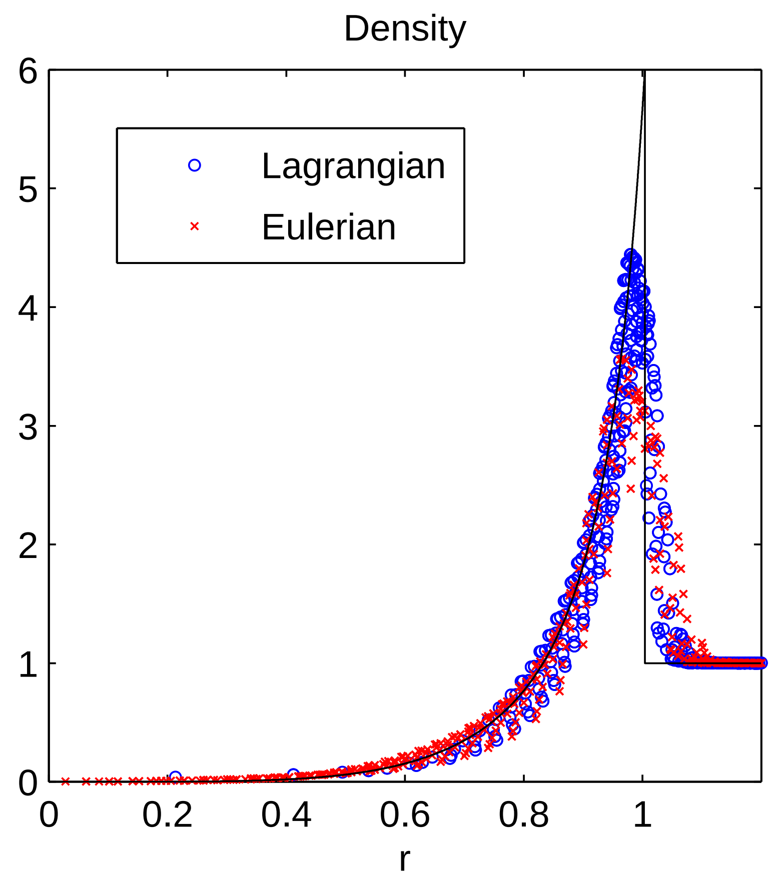
$$= 0 \quad (\text{if GCL \& SCL are satisfied})$$

Model system can be solved by “well-design” method

Sedov problem

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





Future perspective

- Cartesian cut-cell front tracking for shocks & interfaces should be useful tool in astrophysical flows
- Mapped grid method in 3D is applicable for supernovae in spherical geometry (cf. E. Müller using Yin-Yang grid)
- ...
- ...

Thank you