Arbitrary Lagrangian Eulerian Discontinuous Galerkin Method for 1-D Euler Equations

Jayesh Badwaik
jayesh@math.tifrbng.res.in

Center for Applicable Mathematics,
Tata Institute of Fundamental Research,
Bangalore, INDIA - 560065

Conference on Computational PDE, 21-23 Dec, 2015

Supported by Airbus Chair on Mathematics of Complex Systems at TIFR-CAM

December 22, 2015

Euler Equations in 1-D

Conservation laws for mass, momentum and energy

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = 0, \qquad \mathbf{u} = \begin{bmatrix} \rho \\ \rho v \\ E \end{bmatrix}, \qquad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho v \\ p + \rho v^2 \\ (E + p) v \end{bmatrix}$$

$$\rho = \text{density}, \qquad v = \text{velocity}, \qquad p = \text{pressure}$$

$$E = \text{total energy/volume} = \rho e + \frac{1}{2}\rho v^2$$

Equation of state: $p = p(\rho, e)$; for a calorically ideal gas

$$p = (\gamma - 1)\rho e \quad \Longrightarrow \quad p = (\gamma - 1)\left[E - \frac{1}{2}\rho v^2\right]$$

Non-linear system of hyperbolic conservation laws

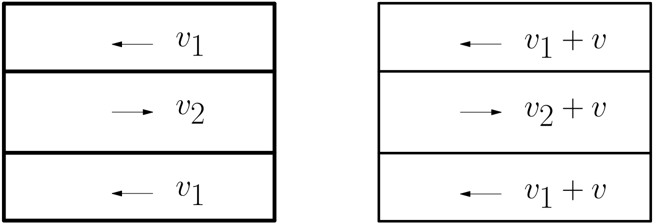
Schemes for conservation laws

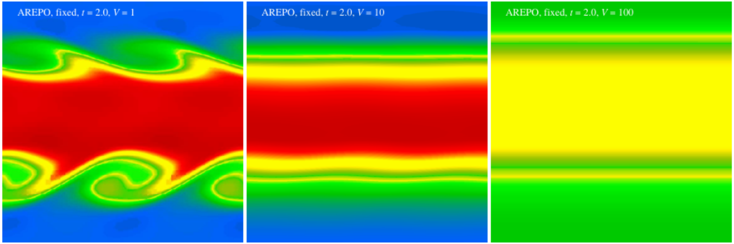
- Hyperbolic equations: eigenvalues

$$v - c, \ v, \ v + c, \qquad c = \text{speed of sound}$$

- Solutions can be discontinuous: look for weak solutions
- Finite volume method
  - based on integral formulation, hence capable of computing weak solutions
  - piecewise constant solution
  - Riemann problems solved exactly/approximately to obtain flux
  - Higher order scheme via local solution reconstruction
- Discontinuous Galerkin method
  - piecewise polynomial solutions, possibly discontinuous across cells
  - Riemann solver technology can be used
  - high order accuracy possible (no need for reconstruction)

Numerical Dissipation in Fixed Mesh Methods





$V = 1$ $V = 10$ $V = 100$

Kelvin-Helmholtz problem at time $t = 2.0$ with different boost velocities V on a fixed mesh (Springel)

Numerical viscosity

- Upwind scheme for a linear convection equation $u_t + au_x = 0$,

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} + \max(a,0)\frac{u_j - u_{j-1}}{h} + \min(a,0)\frac{u_{j+1} - u_j}{h} = 0$$

- Modified partial differential equation

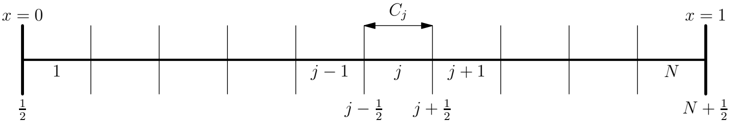
$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \frac{1}{2}|a|h(1-\nu)\frac{\partial^2 u}{\partial x^2} + \mathcal{O}(h^2), \qquad \nu = \frac{|a|\,\Delta t}{h}$$

  Numerical viscosity is proportional to $|a|$
- Euler equations: numerical viscosity proportional to $|v| + c$
- Not Galilean invariant, adds too much dissipation if large relative velocities are present
- Frame moving with velocity w, largest eigenvalue $= |v - w| + c$
- Idea is to construct scheme with $w \approx v$
  $\Longrightarrow$ move the mesh along with the fluid

Mesh

- Partition the domain into disjoint cells

$$\Omega(t) = \bigcup_{i=0}^{N} C_j(t), \qquad C_j(t) = \left(x_{j-\frac{1}{2}}(t), x_{j+\frac{1}{2}}(t)\right)$$



- Discrete time levels given by $\{t_n\}$
- Time steps given by $\Delta t_n = t_{n+1} - t_n$
- Velocity of cell boundaries are assumed constant in a time step Δt_n

$$\frac{\mathrm{d}}{\mathrm{d}t} x_{j+\frac{1}{2}}(t) = w_{j+\frac{1}{2}}(t) = w^n_{j+\frac{1}{2}}, \qquad t_n \le t \le t_{n+1}$$

$$\implies x_{j+\frac{1}{2}}(t) = x^n_{j+\frac{1}{2}} + (t - t_n) w^n_{j+\frac{1}{2}}, \qquad t_n \le t \le t_{n+1}$$

Mesh

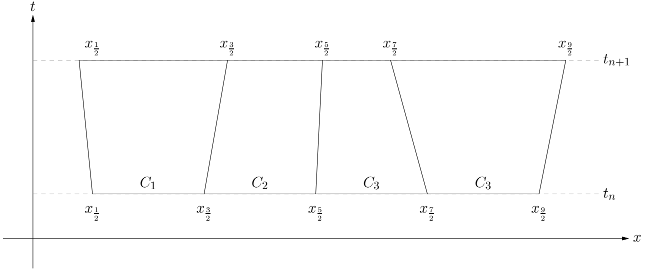
- Center of the cell $x_j(t)$ and length $h_j(t)$ are given by

$$x_j(t) = \frac{1}{2}\left(x_{j-\frac{1}{2}}(t) + x_{j+\frac{1}{2}}(t)\right), \qquad h_j(t) = x_{j+\frac{1}{2}}(t) - x_{j-\frac{1}{2}}(t)$$

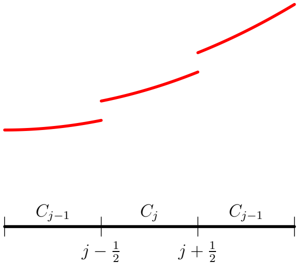
- Velocity at the interior points is given by linear interpolation

$$w(x,t) = \frac{x_{j+\frac{1}{2}}(t) - x}{h_j(t)} w^n_{j-\frac{1}{2}} + \frac{x - x_{j+\frac{1}{2}}(t)}{h_j(t)} w^n_{j+\frac{1}{2}}$$

- Example of moving cell



Solution Space



- Solution is approximated by piecewise polynomials.
- allowed to be discontinuous at cell boundaries

- For degree $k \geq 0$, the solution in the j-th cell is given by

$$\mathbf{u}_h(x,t) = \sum_{m=0}^{k} \mathbf{u}_{j,m}(t)\varphi(x,t)$$

$$\varphi_m(x,t) = \widehat{\varphi}_m(\xi) = \sqrt{2m+1}P_m(\xi), \qquad \xi(x,t) = \frac{x - x_j(t)}{\frac{1}{2}h_j(t)}$$

Solution Space

- orthogonality property

$$\int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \varphi_l(x,t)\varphi_m(x,t)\mathrm{d}x = h_j(t)\delta_{lm}$$

- This allows us to write the expression for the "moments" as

$$\mathbf{u}_{j,m}(t) = \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \boldsymbol{u}_h(x,t)\varphi_l(x,t)\mathrm{d}x$$

Derivation of the ALE-DG scheme

Introduce the change of variable $(x, t) \to (\xi, \tau)$ by

$$\tau = t, \qquad \xi = \frac{x - x_j(t)}{\frac{1}{2} h_j(t)}$$

Calculate the rate of change of moments of the solution starting from

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \boldsymbol{u}_h(x,t) \varphi_l(x,t) \mathrm{d}x &= \frac{\mathrm{d}}{\mathrm{d}\tau} \int_{-1}^{+1} \boldsymbol{u}_h(\xi, \tau) \hat{\varphi}_l(\xi) \frac{1}{2} h_j(\tau) \mathrm{d}\xi \\
&= \frac{1}{2} \int_{-1}^{+1} \left[h_j(\tau) \frac{\partial \boldsymbol{u}_h}{\partial \tau} + \boldsymbol{u}_h \frac{\mathrm{d} h_j}{\mathrm{d}\tau} \right] \hat{\varphi}_l(\xi) \mathrm{d}\xi
\end{aligned}$$

But we have

$$\frac{\partial \boldsymbol{u}_h}{\partial \tau}(\xi, \tau) = \frac{\partial \boldsymbol{u}_h}{\partial t}(x, t) + w(x, t) \frac{\partial \boldsymbol{u}_h}{\partial x}(x, t)$$

Derivation of the ALE-DG scheme

and

$$\frac{\mathrm{d}h_j}{\mathrm{d}\tau} = w_{j+\frac{1}{2}} - w_{j-\frac{1}{2}} = h_j \frac{\partial w}{\partial x} \qquad \text{since } w(x,t) \text{ is linear in } x$$

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \boldsymbol{u}_h(x,t)\varphi_l(x,t)\mathrm{d}x &= \int_{-1}^{+1}\left[\frac{\partial \boldsymbol{u}_h}{\partial t} + w\frac{\partial \boldsymbol{u}_h}{\partial x} + \boldsymbol{u}_h\frac{\partial w}{\partial x}\right]\hat{\varphi}_l(\xi)\frac{1}{2}h_j\mathrm{d}\xi \\
&= \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)}\left[-\frac{\partial \boldsymbol{f}(\boldsymbol{u}_h)}{\partial x} + \frac{\partial}{\partial x}(w\boldsymbol{u}_h)\right]\varphi_l(x,t)\mathrm{d}x
\end{aligned}$$

Define the flux

$$\boldsymbol{g}(\boldsymbol{u},w) = \boldsymbol{f}(\boldsymbol{u}) - w\boldsymbol{u}$$

Performing an integration by parts in the x variable, we obtain

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \boldsymbol{u}_h(x,t)\varphi_l(x,t)\mathrm{d}x &= \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \boldsymbol{g}(\boldsymbol{u}_h,w)\frac{\partial}{\partial x}\varphi_l(x,t)\mathrm{d}x \\
&\quad + \hat{\boldsymbol{g}}_{j-\frac{1}{2}}(\boldsymbol{u}_h(t))\varphi_l(x^+_{j-\frac{1}{2}},t) - \hat{\boldsymbol{g}}_{j+\frac{1}{2}}(\boldsymbol{u}_h(t))\varphi_l(x^-_{j+\frac{1}{2}},t)
\end{aligned}$$

Derivation of the ALE-DG scheme

where we have introduced the numerical flux

$$\hat{\boldsymbol{g}}_{j+\frac{1}{2}}(\boldsymbol{u}_h(t)) = \hat{\boldsymbol{g}}(\boldsymbol{u}^-_{j+\frac{1}{2}}, \boldsymbol{u}^+_{j+\frac{1}{2}}, w_{j+\frac{1}{2}})$$

Integrating over the time interval (t_n, t_{n+1}) we obtain

$$\begin{aligned} h_j^{n+1}\boldsymbol{u}_{j,l}^{n+1} &= h_j^n \boldsymbol{u}_{j,l}^n + \int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \boldsymbol{g}(\boldsymbol{u}_h, w) \frac{\partial}{\partial x}\varphi_l(x,t)\mathrm{d}x\mathrm{d}t \\ &\quad + \int_{t_n}^{t_{n+1}} [\hat{\boldsymbol{g}}_{j-\frac{1}{2}}(t)\varphi_l(x^+_{j-\frac{1}{2}}, t) - \hat{\boldsymbol{g}}_{j+\frac{1}{2}}(t)\varphi_l(x^-_{j+\frac{1}{2}}, t)]\mathrm{d}t \end{aligned}$$

This has an implicit nature; $\boldsymbol{u}_h$ is known only at $t = t_n$ but we need it over the interval $[t_n, t_{n+1}]$

Derivation of the ALE-DG scheme

Assume that we can get a *predicted solution* $\boldsymbol{U}_h$; then using quadratures, the fully discrete scheme

$$
\begin{aligned}
h_j^{n+1}\boldsymbol{u}_{j,l}^{n+1} \quad = \quad & h_j^n \boldsymbol{u}_{j,l}^n + \Delta t_n \sum_r \theta_r h_j(\tau_r) \sum_q \eta_q \boldsymbol{g}(\boldsymbol{U}_h(x_q,\tau_r), w(x_q,\tau_r)) \frac{\partial}{\partial x}\varphi_l(x_q,\tau_r) \\
& + \Delta t_n \sum_r \theta_r [\hat{\boldsymbol{g}}_{j-\frac{1}{2}}(\boldsymbol{U}_h(\tau_r))\varphi_l(x_{j-\frac{1}{2}}^+,\tau_r) - \hat{\boldsymbol{g}}_{j+\frac{1}{2}}(\boldsymbol{U}_h(\tau_r))\varphi_l(x_{j+\frac{1}{2}}^-,\tau_r)]
\end{aligned}
$$

$$\tau_r, \theta_r = \text{nodes and weights for time qudrature}$$

$$x_q, \eta_q = \text{nodes and weights for spatial quadrature}$$

Spatial quadrature: use $q = k + 1$ point Gauss quadrature.
Time quadrature: use mid-point rule for $k = 1$, two point Gauss quadrature for $k = 2, 3$, etc.

Mesh velocity

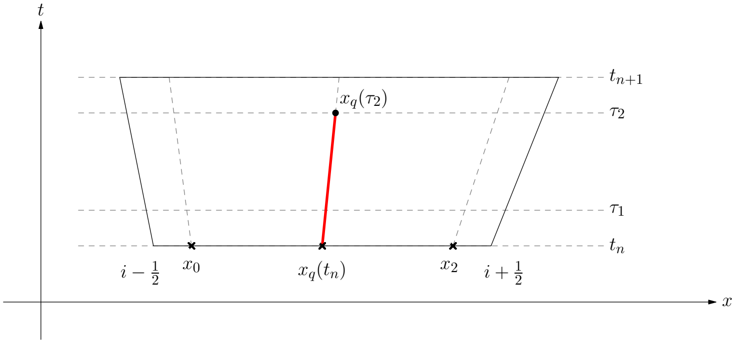
- Mesh velocity must be close to the local fluid velocity
- simple choice is to take an average

$$\tilde{w}^n_{j+\frac{1}{2}} = \frac{1}{2}[v(x^-_{j+\frac{1}{2}}, t_n) + v(x^+_{j+\frac{1}{2}}, t_n)]$$

- perform some smoothing of the mesh velocity, e.g.,

$$w^n_{j+\frac{1}{2}} = \frac{1}{3}(\tilde{w}^n_{j-\frac{1}{2}} + \tilde{w}^n_{j+\frac{1}{2}} + \tilde{w}^n_{j+\frac{3}{2}})$$

Predictor via Taylor expansion



The Taylor expansion around (X_q, t_n) is

$$\begin{aligned} \boldsymbol{u}(x_q, \tau_r) &= \boldsymbol{u}(X_q, t_n) + (\tau_r - t_n)\frac{\partial \boldsymbol{u}}{\partial t}(X_q, t_n) + (x_q - X_q)\frac{\partial \boldsymbol{u}}{\partial x}(X_q, t_n) \\ &\quad + O(\tau_r - t_n)^2 + O(x_q - X_q)^2 \end{aligned}$$

Predictor via Taylor expansion

and hence the predicted solution is

$$\boldsymbol{U}(x_q, \tau_r) = \boldsymbol{u}_h(X_q, t_n) + (\tau_r - t_n)\frac{\partial \boldsymbol{u}_h}{\partial t}(X_q, t_n) + (x_q - X_q)\frac{\partial \boldsymbol{u}_h}{\partial x}(X_q, t_n)$$

Using the conservation law, the time derivative is written as $\frac{\partial \boldsymbol{u}}{\partial t} = -\frac{\partial \boldsymbol{f}}{\partial x} = -A\frac{\partial \boldsymbol{u}}{\partial x}$ so that predictor is given by

$$\boldsymbol{U}_h(x_q, \tau_r) = \boldsymbol{u}_h^n(X_q) - (\tau_r - t_n)\left[A(\boldsymbol{u}_h^n(X_q)) - w_q I\right]\frac{\partial \boldsymbol{u}_h^n}{\partial x}(X_q)$$

The above predictor is used for the case of polynomial degree $k = 1$.

This procedure can be extended to higher orders by including more terms in the Taylor expansion but the algebra becomes complicated.

Predictor using Runge-Kutta

Idea: Apply RK scheme to obtain solution in $[t_n, t_{n+1}]$

Choose a set of $(k+1)$ distinct nodes, e.g., Gauss-Legendre or Gauss-Lobatto nodes, which uniquely define the polynomial of degree k.

Nodes are moving with velocity $w(x,t)$, the time evolution of the solution at $x = x_m$ is governed by

$$
\begin{aligned}
\frac{\mathrm{d}\boldsymbol{U}_m}{\mathrm{d}t} &= \frac{\partial}{\partial t}\boldsymbol{U}_h(x_m,t) + w(x_m,t)\frac{\partial}{\partial x}\boldsymbol{U}_h(x_m,t) \\
&= -\frac{\partial}{\partial x}\boldsymbol{f}(\boldsymbol{U}_h(x_m,t)) + w(x_m,t)\frac{\partial}{\partial x}\boldsymbol{U}_h(x_m,t) \\
&= -[A(\boldsymbol{U}_m(t)) - w_m(t)I]\frac{\partial}{\partial x}\boldsymbol{U}_h(x_m,t) =: \boldsymbol{K}_m(t)
\end{aligned}
$$

with initial condition

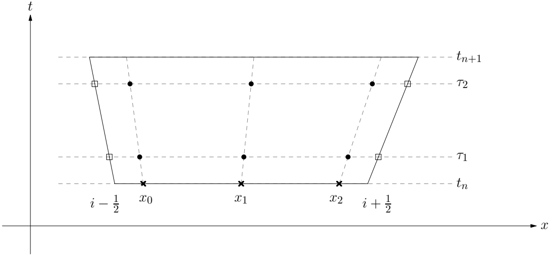
$$
\boldsymbol{U}_m(t_n) = \boldsymbol{u}_h(x_m,t_n) = \boldsymbol{u}_h^n(x_m)
$$

Predictor using Runge-Kutta

Using a Runge-Kutta scheme of sufficient order, we will approximate the solution at these nodes as

$$\boldsymbol{U}_m(t) = \boldsymbol{u}_h(x_m, t_n) + \sum_{s=1}^{n_s} b_s((t - t_n)/\Delta t_n)\boldsymbol{K}_{m,s}, \quad t \in [t_n, t_{n+1}), \quad m = 0, 1, ..$$

$$\boldsymbol{K}_{m,s} = \boldsymbol{K}_m(t_n + \tau_s), \qquad \tau_s = \text{stage time}$$



Quadrature points for third order scheme

Predictor using Runge-Kutta

Once the predictor is computed as above, it must be evaluated at the quadrature point (x_q, τ_r) as follows. For each time quadrature point $\tau_r \in [t_n, t_{n+1}]$,

1. Compute nodal values $\boldsymbol{U}_m(\tau_r)$, $m = 0, 1, \ldots, k$
2. Convert nodal values to modal coefficients $\boldsymbol{u}_{m,r}$, $m = 0, 1, \ldots, k$
3. Evaluate predictor $\boldsymbol{U}_h(x_q, \tau_r) = \sum_{m=0}^{k} \boldsymbol{u}_{m,r} \varphi_m(x_q, \tau_r)$

The predictor is also computed at the cell boundaries using the above procedure.

Limiter

- Discontinuous solutions obtained from high order schemes suffer from numerical oscillations: loss of TVD property
- Post process the DG solution with a TVD or TVB limiter (Cockburn & Shu)
- To make density/pressure postive, apply positivity limiter of Zhang & Shu

Grid coarsening

- Grid cells can become small in size, e.g., around shocks
- Time step is reduced due to CFL condition
- If $h_j^n < h_{min}$, then merge this cell with one of its neighbouring cells. Transfer solution by L^2 projection.

Choosing the time step

- Geometrical constraint: cell size must not change by more than a fraction β

$$(1-\beta)h_j^n \le h_j^{n+1} \le (1+\beta)h_j^n \qquad \text{e.g., } \beta = 0.1$$

$$\implies \Delta t_n \le \frac{\beta h_j^n}{|w_{j+\frac{1}{2}}^n - w_{j-\frac{1}{2}}^n|}$$

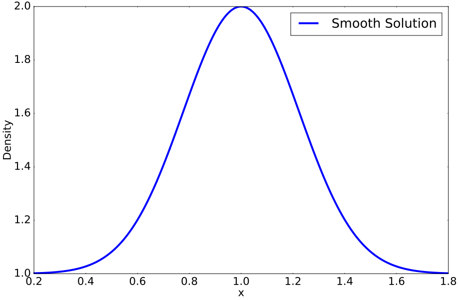
- First order scheme with Rusanov flux is positive if

$$\Delta t_n \le \Delta t_n^{(1)} := \min_j \left\{ \frac{(1-\frac{1}{2}\beta)h_j^n}{\frac{1}{2}(\lambda_{j-\frac{1}{2}}^n + \lambda_{j+\frac{1}{2}}^n)}, \frac{\beta h_j^n}{|w_{j+\frac{1}{2}}^n - w_{j-\frac{1}{2}}^n|} \right\}$$

- In general, when degree k polynomials are used

$$\Delta t_n = \frac{\mathrm{CFL}}{2k+1}\Delta t_n^{(1)}, \qquad \mathrm{CFL} \approx 0.5$$

Order of Accuracy
Smooth Solution Test Case



$$\rho(x,0) = 1 + \exp(-10x^2), \qquad v(x,0) = 1, \qquad p(x,0) = 1$$

Order Accuracy

Fixed Mesh, Lax Friedrichs Flux, L^2 Errors

NC	Taylor Error	Rate	CERK2 Error	Rate	CERK3 Error	Rate
100	4.370E-02		3.498E-03		3.883E-04	
200	6.611E-03	2.725	4.766E-04	2.876	1.620E-05	4.583
400	1.332E-03	2.518	6.415E-05	2.885	9.376E-07	4.347
800	3.151E-04	2.372	8.246E-06	2.910	5.763E-08	4.239
1600	7.846E-05	2.280	1.031E-06	2.932	3.595E-09	4.180

Table: Order of accuracy study on static mesh

Order Accuracy

Moving Mesh, Lax Friedrichs Flux, L^2 Errors

NC	Taylor Error	Rate	CERK2 Error	Rate	CERK3 Error	Rate
100	2.331E-02		3.979E-03		8.633E-04	
200	6.139E-03	1.925	4.0582E-04	3.294	1.185E-05	6.186
400	1.406E-03	2.0258	5.250E-05	3.122	7.079E-07	5.126
800	3.375E-04	2.0366	6.626E-06	3.077	4.340E-08	4.760
1600	8.278E-05	2.0344	8.304E-07	3.057	2.689E-09	4.573

Table: Order of accuracy study on moving mesh

Order Accuracy

Fixed Mesh, HLLC Flux, L^2 Errors

NC	Taylor Error	Rate	CERK2 Error	Rate	CERK3 Error	Rate
100	4.582E-02		3.952E-03		3.464E-04	
200	9.611E-03	2.253	4.048E-04	3.287	2.058E-05	4.073
400	2.052E-03	2.240	4.640E-05	3.206	1.287E-06	4.036
800	4.803E-04	2.192	5.623E-06	3.152	8.061E-08	4.023
1600	1.184E-04	2.149	6.929E-07	3.119	5.050E-09	4.016

Table: Order of accuracy study on static mesh

Order Accuracy

Moving Mesh, HLLC Flux, L^2 Errors

NC	Taylor Error	Order	CERK2 Error	Order	CERK3 Error	Order
100	1.590E-02		1.626E-03		1.962E-04	
200	4.042E-03	1.977	2.072E-04	2.972	1.269E-05	3.950
400	1.014E-03	1.985	2.605E-05	2.982	7.983E-07	3.971
800	2.538E-04	1.990	3.261E-06	2.988	4.997E-08	3.980
1600	6.349E-05	1.992	4.077E-07	2.991	3.124E-09	3.985

Table: Order of accuracy study on moving mesh

Sod Shocktube

Problem

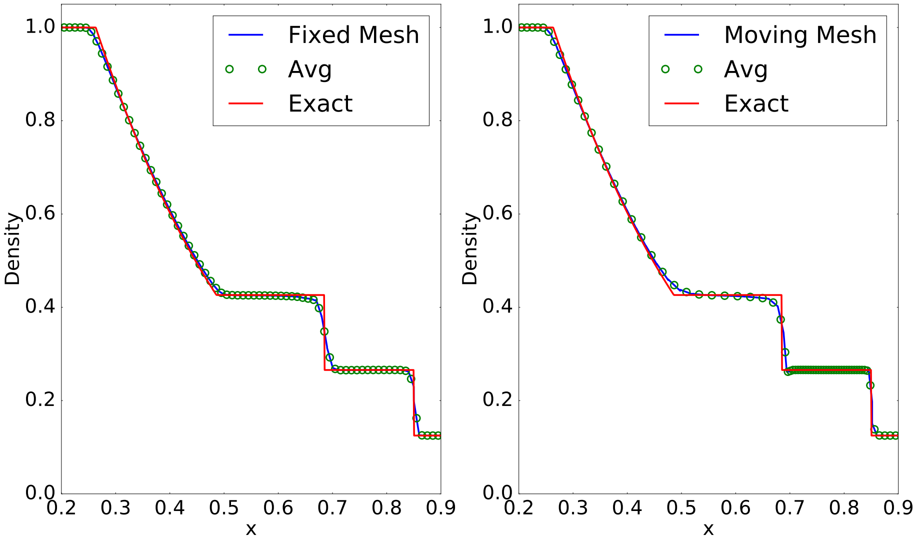
The initial conditions are given by

$$(\rho, v, p) = \begin{cases} (1.0, 0.0, 1.0) & x < 0.5 \\ (0.125, 0.0, 0.1) & x > 0.5 \end{cases}$$

- $T = 0.2$.
- Number of cells $= 100$.

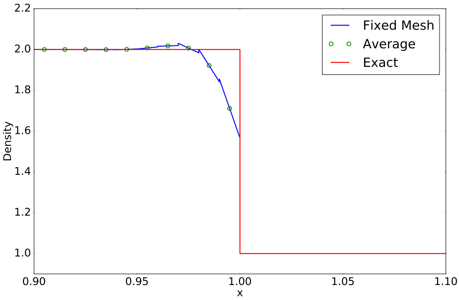
Sod Shocktube

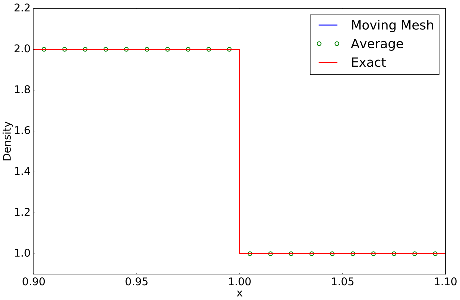
Lax Friedrichs Flux



Contact

$$(\rho, v, p) = \begin{cases} (2.0, 1.0, 1.0) & x < 0.5 \\ (1.0, 1.0, 1.0) & x > 0.5 \end{cases}$$





Shu-Osher

Problem

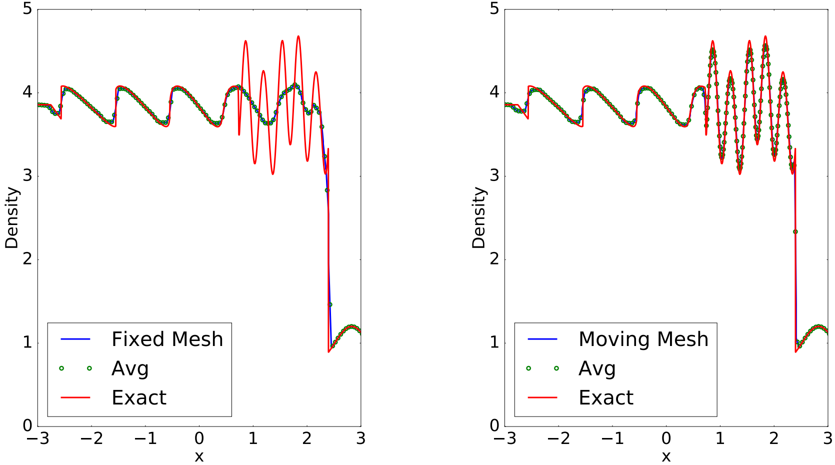
The initial conditions are given by

$$(\rho, v, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & x < -4 \\ (1 + 0.2\sin(5x), 0.0, 1.0) & x > -4 \end{cases}$$

- $T = 1.8$
- Number of cells is 200.

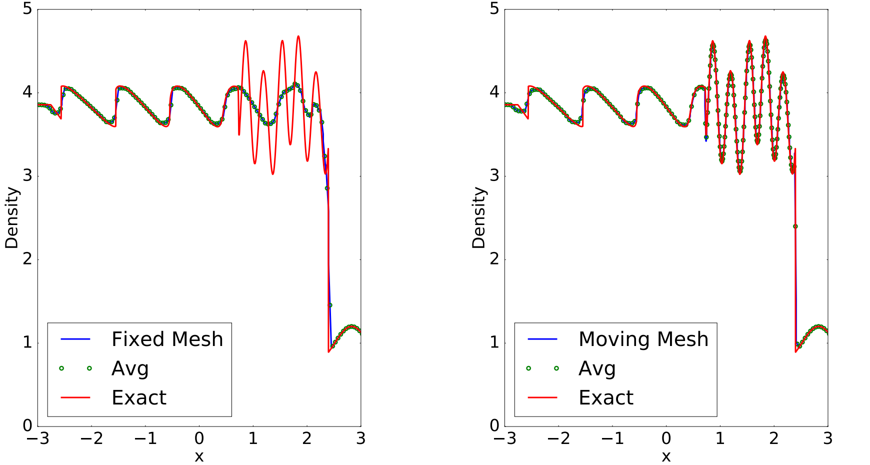
Shu-Osher Problem

Lax Friedrichs



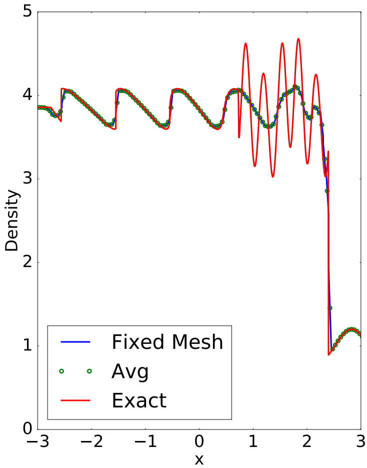
Shu-Osher

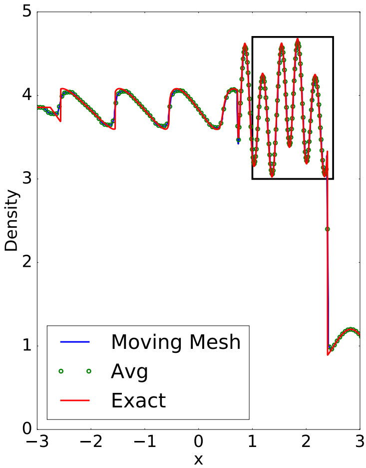
Roe Flux I



Shu-Osher

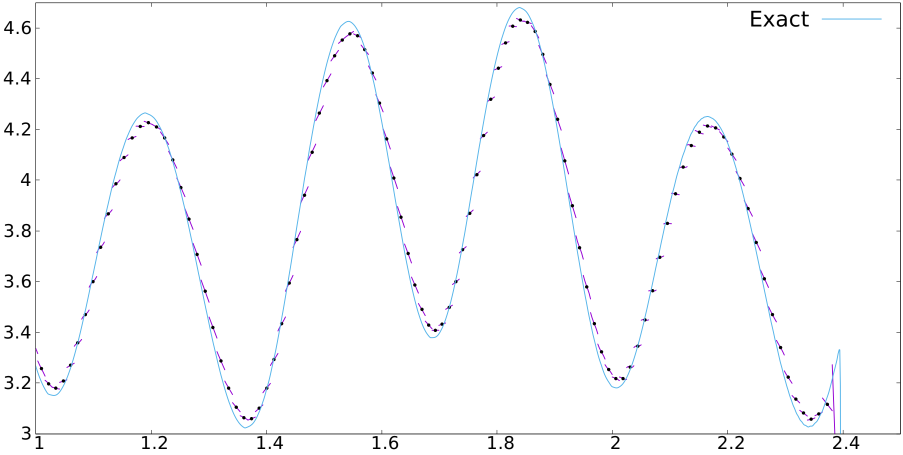
Roe Flux II





Shu-Osher

Roe Flux III



Roe Flux and ALE

Near Zero Eigenvalues

- Eigenvalues for moving mesh

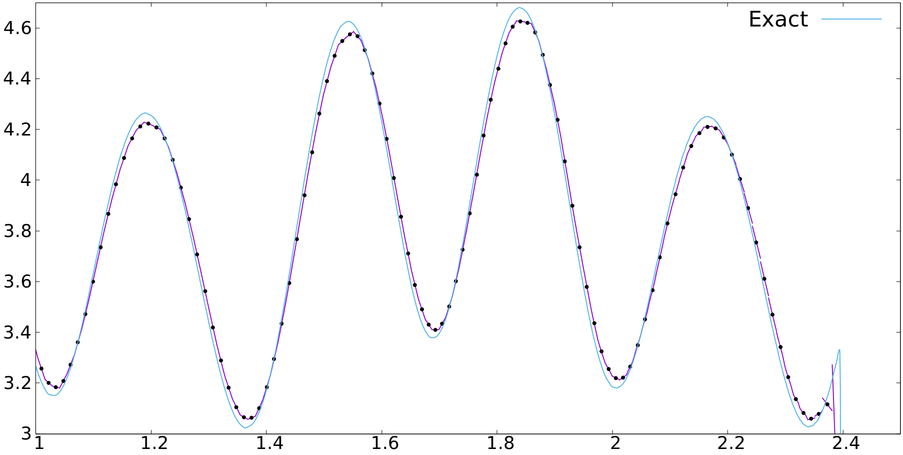
$$|v - w - c|, \qquad |v - w|, \qquad |v - w + c|$$

- For $w \approx v$, $|\lambda_2| = |v - w| \approx 0$
- Dissipation is almost zero for the contact wave
- Modify eigenvalue in Roe flux

$$|\lambda_2| = \begin{cases} |v - w| & |v - w| > \delta \\ \frac{1}{2}(\delta + |v - w|^2/\delta) & \text{otherwise} \end{cases}, \qquad \delta = 0.1c$$

Shu-Osher

Roe Flux with eigenvalue fix



123 Problem

Problem

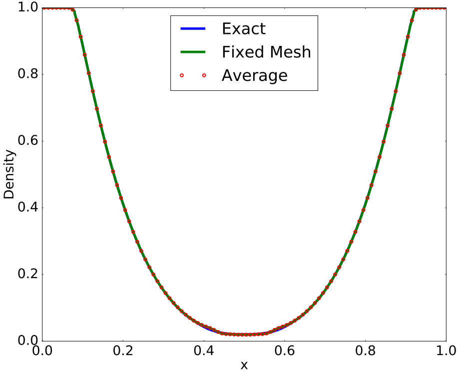
The initial conditions are given by

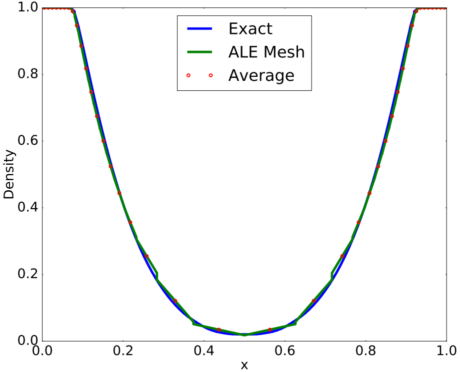
$$(\rho, v, p) = \begin{cases} (1.0, -2.0, 0.4) & x < -4 \\ (1.0, 2.0, 0.4) & x > -4 \end{cases}$$

- Time of simulation is $T = 0.15$

123 Problem

Lax Friedrichs Flux





Grid becomes coarse in ALE due to rarefaction

Blast Test Case

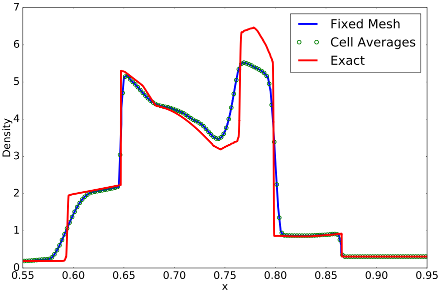
Problem

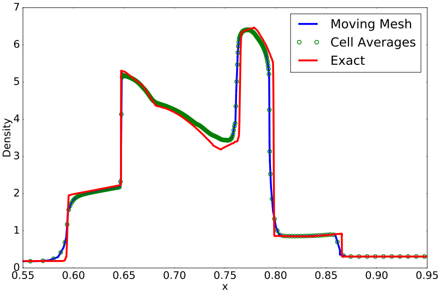
The initial condition is given by

$$(\rho, v, p) = \begin{cases} (1.0, 0.0, 1000.0) & x < 0.1 \\ (1.0, 0.0, 0.01) & 0.1 < x < 0.9 \\ (1.0, 0.0, 100.0) & x > 0.9 \end{cases}$$

$T = 0.038$

Blast Test Case





Moving mesh results have better amplitude but slightly different shock location

Conclusions

- Developed a high order DG on moving meshes
  - 2'nd, 3'rd, 4'th order schemes
  - Single step schemes using a predictor
- Nearly Lagrangian character leads to better solutions
- Good accuracy obtained even with TVD limiters, since mesh is automatically clustered
- Roe flux does not have entropy problem, but contact wave needs a fix
  - fixing contact speed solves this, but exact contact preservation is lost
  - This problem exists with HLLC also

Future Work

- Add mesh refinement to increase accuracy in regions with rarefactions
- Add better limiters
- Extend the scheme to two dimensions

Thank You