Numerical Analysis
A particle method for conservation laws

Aditya Kashi

McGill University

December 18, 2016

Table of Contents

Background - numerical solution of conservation laws

State-of-the-practice (engineering): finite volume or finite difference methods with limiters, ENO/WENO or artificial dissipation terms.

- second-order accurate only in smooth regions
- between first and second-order accurate at shocks
- TVD can usually be guaranteed only for scalar case

Based on 'Eulerian' method, usually.

Aim

Solve scalar 1D conservation laws with a second-order TVD scheme.

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{1}$$

$$u(x, 0) = u_0(x) \tag{2}$$

Burgers' equation: $f(u) = \frac{u^2}{2}$.

Table of Contents

The method

The characteristic equations are key. [2]

$$\dot{x} = f'(u) \tag{3}$$

$$\dot{u} = 0 \tag{4}$$

The solution is a curve along which u is constant, while u is smooth. We can think in terms of 'particles'.

Conservative particle management and interpolation

'Particle management' is needed when particles collide or get too far apart.

- Based on local conservation of total 'amount of u' between two particles.
- Rate of change of amount of u contained between two particles

$$\frac{d}{dt}\int_{x_1(t)}^{x_2(t)} u(x,t)dx \tag{5}$$

- However, care must be taken because the limits of integration are now time-dependent.

Particle Management

After some calculation it can be shown, for space-independent flux functions f, that

$$\frac{d}{dt}\int_{x_1(t)}^{x_2(t)} u(x,t)dx = (x_2(t) - x_1(t))a_f(u_1, u_2) \tag{6}$$

where

$$a_f(u_1, u_2) = \frac{[f'(u) - f(u)]_{u_1}^{u_2}}{[f'(u)]_{u_1}^{u_2}}. \tag{7}$$

For Burgers' equation is

$$a_f(u_1, u_2) = \frac{u_1 + u_2}{2} \tag{8}$$

This is the basis of conservative particle management and conservative interpolation.

It has been shown that the method is second-order accurate away from shocks and TVD.

A ‘shock location’ step has to be performed to achieve second-order accuracy in presence of shocks. However, for Burgers’ equation, second order accuracy seems to be achieved irrespective of this.

Implementation

- Since particles may need to be added or deleted anywhere, an array is not a good data structure. A linked list was coded, which allows efficient addition and deletion, at the cost of more expensive iteration through the particles.
- Conservative particle management is cheap only for quadratic flux functions. For others, a Newton iteration is required to solve for the new particles.
- Implementation in Julia programming language [1]

Table of Contents

Initial condition

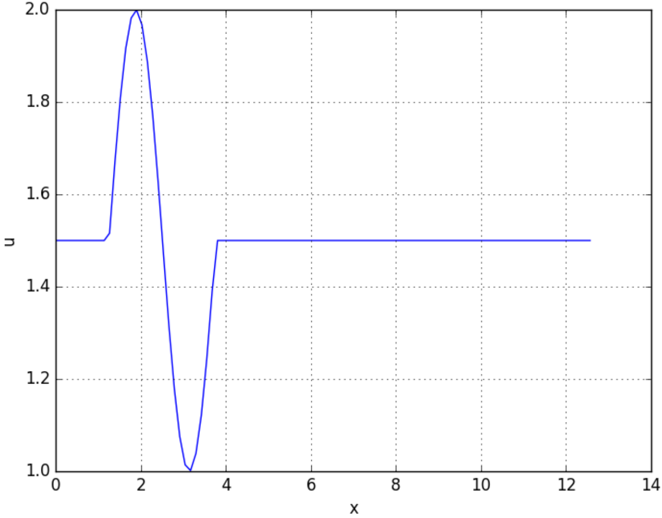

Figure: Initial condition

Before shock formation

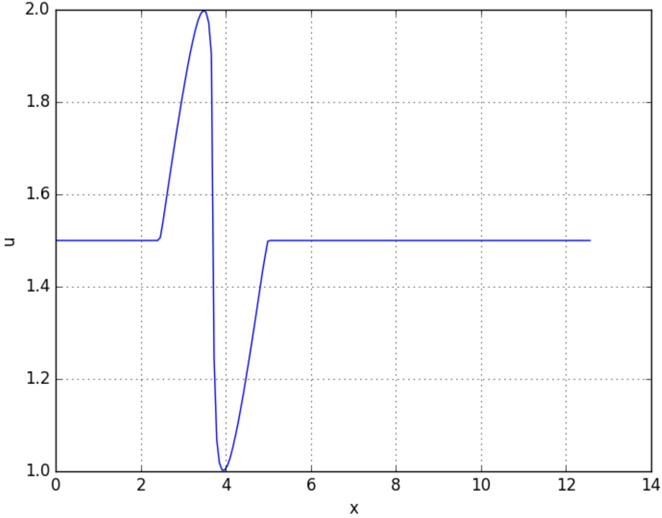

Figure: t = 0.1

After shock formation

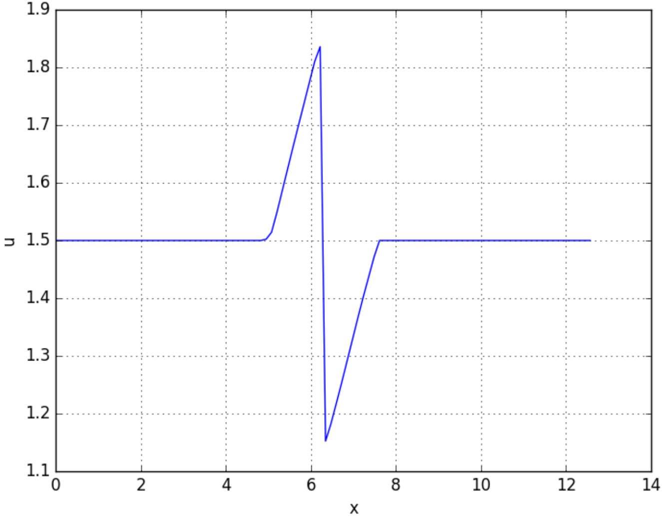

Figure: $t = 2.5$

Grid convergence

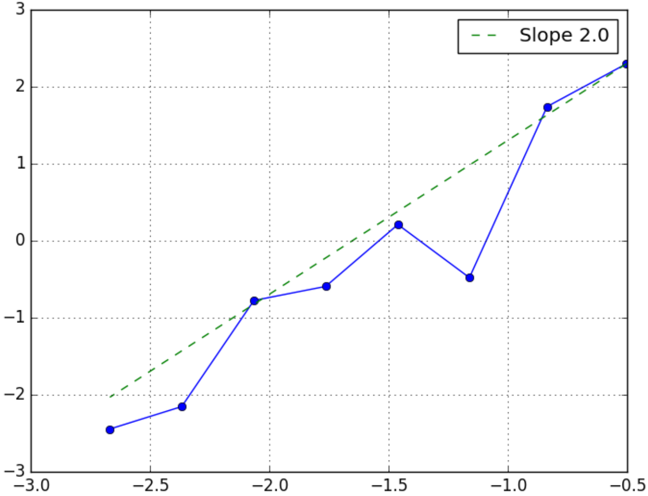

Figure: t = 2.5

Grid convergence without post-processing the shock

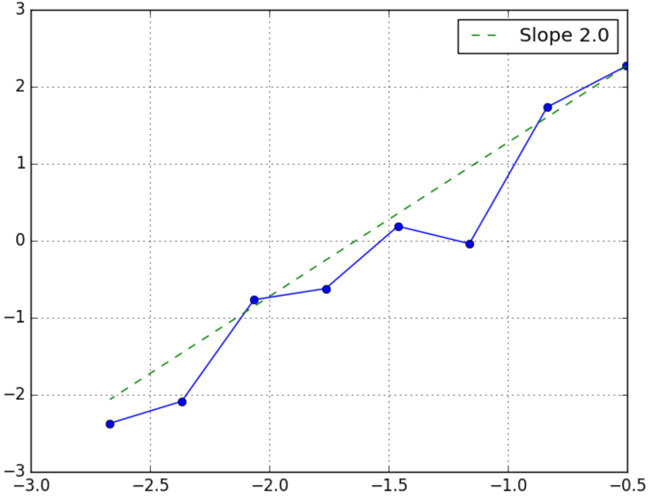

Figure: t = 2.5; no post-processing

Conclusion

- The particle method provides a way to achieve second-order accuracy for non-smooth solutions, at least for scalar conservation laws.
- An easy post-processing step is needed to maintain second-order accuracy in presence of shocks.
- However, extension to multi-dimensional problems and systems of conservation laws is not straightforward.

Table of Contents

Systems of conservation laws

- $\mathbf{f}'(\mathbf{u})$ is now a matrix with N eigenvalues
- Speed of the particle?
- One possible solution: use N different sets of particles.
- Can TVD still be guaranteed?

Multidimensional problems

- In 1D, we have ‘next’ and ‘previous’ particles. Not so in multi-D.
- One could compute a Voronoi tessellation to get ‘neighbors’ for each particle, and apply the 1D merge/insert for each pair thus generated.
- Implementation - data structures?

Others

- Steady-state solutions?
- Implicit time stepping?

Table of Contents

Jeff Bezanson et al. “Julia: A Fast Dynamic Language for Technical Computing”. In: *CoRR* abs/1209.5145 (2012). URL: http://arxiv.org/abs/1209.5145.

Y. Farjoun and B. Seibold. “An exactly conservative particle method for one-dimensional scalar conservation laws”. In: *Journal of Computational Physics* 228.14 (2009), pp. 5298–5315.