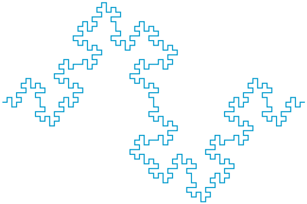
Parallel Semi-Lagrangian Scheme

For solving advection equations

Rongting Zhang, Yimin Zhong
Dept. Mathematics, University of Texas at Autin



April 10, 2014

Dynamical System in $\mathbb{R}^d$

It is widely used in numerical weather prediction and various simulation.

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = \mathbf{V}(\mathbf{X}(t), t) \tag{1}$$

or equivalently finding *characteristic* of advection equation

$$\frac{\partial f(\mathbf{x}, t)}{\partial t} + \mathbf{V}(\mathbf{x}, t) \cdot \nabla_x f = 0 \tag{2}$$

Method

- Semi-Lagrangian

$$f(\mathbf{x}_m, t_n + \Delta t) = f(\mathbf{x}_m - 2\alpha_m, t_n - \Delta t) \tag{3}$$

- Second order accuracy and iteration

$$\alpha_m = \Delta t \, \mathbf{V}(\mathbf{x_m} - \alpha_m, t_n) \tag{4}$$

Pro

- No need to worry about CFL condition.

Con

- This scheme is at best second order accurate in approximating the backtracking point, if **V** is variable.

SEMI-LAGRANGIAN ENO/WENO

General Steps

- 1D problem:
  - Backtracking in time.
  - Spacial interpolation using ENO/WENO.
- Multi-D problem:
  Strang splitting: decompose into severl 1D problems.

SEQUENTIAL METHOD

- If **V** is constant vector, then it is fast to know the solution by simply interpolation and implement high order accuracy scheme by ENO/WENO.
- To achieve $2k+1$ order accuracy in $\mathbf{R}^d$, with number of time steps as N and mesh size M, by using WENO. Time complexity is $O(kNM^d)$. (Qiu & Shu 2011) =======
- If **V** is constant vector, then it is fast to know the solution by simply interpolation, and easy to implement arbitrary accuracy scheme by ENO/WENO.
- To achieve $2k+1$ order accuracy in $\mathbf{R}^d$, with time step as N and mesh size M, by using WENO. Time complexity is $O(kNM^d)$.

SEQUENTIAL METHOD

- If **V** is variable and divergence free. Tracking back is no longer an easy job within high accuracy. We shall use some high order time integrator to implement this.
- For example, Semi-Lagrangian method with Runge-Kutta high order solver(time integrator) for back tracking. The time complexity is $O(M^d N \times T(SL))$ with second order time splitting and WENO. (Qiu & Shu 2011)

Parallel Method

Most of running time are consumed in Semi-Lagrangian back tracking and interpolation. And this can be parallelized perfectly, since back tracking with different end points $\mathbf{x}_m$ are independent process. Work-depth model time complexity here is $O(T(SL) \times N)$.

MILESTIONES

- 1D Sequential Semi-Lagrangian ENO/WENO implementation. (April 12)
- 3D Semi-Lagrangian ENO/WENO implementation. (April 18)
- 3D shared memory implementation. (April 25)
- 3D distributed memory implementation. (May 9)