On semilagrangian methods for kinetic equations

Giovanni Russo

Dipartimento di Matematica e Informatica
Università di Catania

Numerical methods for PDEs: optimal control, games and image processing
In honor of the 60th birthday of Maurizio Falcone
University of Rome, La Sapienza
December 4-5, 2014

Main collaborators:
Jingmei Qiu, University of Houston
Maria Groppi, Giuseppe Stracquadanio, Univ. of Parma

Outline

Motivation: Vlasov-Poisson system

The collisionless and non-relativistic plasma may be described by the well-known Vlasov-Poisson (VP) system,

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \mathbf{E}(t, \mathbf{x}) \cdot \nabla_{\mathbf{v}} f = 0, \tag{1}$$

$$\mathbf{E}(t, \mathbf{x}) = -\nabla_{\mathbf{x}} \phi(t, \mathbf{x}), \quad -\triangle_{\mathbf{x}} \phi(t, \mathbf{x}) = \rho(t, \mathbf{x}). \tag{2}$$

$f(t, \mathbf{x}, \mathbf{v})$: probability density of finding a particle with velocity $\mathbf{v}$ at position $\mathbf{x}$ at time t.
$\rho(t, \mathbf{x}) = \int f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}$ - 1: charge density

Numerical methods

Numerical methods

- Eulerian: fixed numerical mesh, CFL restriction
  e.g., finite difference method, finite volume method, disc. Galerkin

Numerical methods

- Eulerian: fixed numerical mesh, CFL restriction
  e.g., finite difference method, finite volume method, disc. Galerkin
- Lagrangian: follow characteristics (particles), no CFL restriction.

Numerical methods

- Eulerian: fixed numerical mesh, CFL restriction
  e.g., finite difference method, finite volume method, disc. Galerkin
- Lagrangian: follow characteristics (particles), no CFL restriction.
- Semi-Lagrangian: fixed numerical mesh; update solution by following characteristics, no CFL restriction
  - with operator splitting:
    easy to trace characteristics, but subject to splitting error
  - without operator splitting:
    difficult to trace characteristics with high order accuracy, no splitting error

Numerical methods

- Eulerian: fixed numerical mesh, CFL restriction
  e.g., finite difference method, finite volume method, disc. Galerkin
- Lagrangian: follow characteristics (particles), no CFL restriction.
- Semi-Lagrangian: fixed numerical mesh; update solution by following characteristics, no CFL restriction
  - with operator splitting:
    easy to trace characteristics, but subject to splitting error
  - without operator splitting:
    difficult to trace characteristics with high order accuracy, no splitting error

Other subdivision is between

- Forward SL schemes (FSL)
- Backward SL schemes (BSL)

We only adopt Backward Semi Lagrangian (BSL) here.

Semi-Lagrangian: without splitting

How to update solution: $\{f_{i,j}^n\} \Rightarrow \{f_{i,j}^{n+1}\}$?

1. Tracing characteristics: locate the foot of characteristics $(x_{i,j}^*, v_{i,j}^*) = (x(t^n), v(t^n))$ subject to the following final value problem

$$\begin{cases} \frac{dx(t)}{dt} = v(t); & \frac{dv(t)}{dt} = E(x(t), t), \\ x(t^{n+1}) = x_i; & v(t^{n+1}) = v_j \end{cases} \tag{3}$$

2. High order interpolation:

$$f_{i,j}^{n+1} = f(x_{i,j}^*, v_{i,j}^*, t^n),$$

where the R.H.S. is approximated by high-dimensional interpolation (for example WENO).

Tracing Characteristics

- *First order accuracy:*

$$x_i^{(1)} = x_i - v_j \Delta t; \quad v_j^{(1)} = v_j - E_i^n \Delta t$$

from which one can update $f^{(1)}$, $\rho^{(1)}$ and $E^{(1)}$ at t^{n+1}. [1,2]

[1] superscript (1) means first order approximation
[2] $E_i^n = E(x_i, t^n)$

Tracing Characteristics

- *First order accuracy:*

$$x_i^{(1)} = x_i - v_j \Delta t; \quad v_j^{(1)} = v_j - E_i^n \Delta t$$

from which one can update $f^{(1)}$, $\rho^{(1)}$ and $E^{(1)}$ at t^{n+1}. [1,2]

- *Second order accuracy:*

$$x_i^{(2)} = x_i - \frac{1}{2}(v_j + v_j^{(1)})\Delta t;$$

$$v_j^{(2)} = v_j - \frac{1}{2}(E(x_i^{(1)}, t^n) + E^{(1)}(x_i, t^{n+1}))\Delta t$$

from which one can update $f^{(2)}$, $\rho^{(2)}$ and $E^{(2)}$ at t^{n+1}.

[1] superscript (1) means first order approximation
[2] $E_i^n = E(x_i, t^n)$

Tracing Characteristics (cont.)

- *Third order accuracy:*

$$x_i^{(3)} = x_i - v_j \Delta t + \frac{\Delta t^2}{2}\left(\frac{2}{3}E^{(2)}(x_i, t^{n+1}) + \frac{1}{3}E(x_i^{(2)}, t^n)\right);$$

$$\begin{aligned} v_j^{(3)} &= v_j - E^{(2)}(x_i, t^{n+1})\Delta t \\ &\quad + \frac{\Delta t^2}{2}\left(\frac{2}{3}\frac{d}{dt}E^{(2)}(x_i, t^{n+1}) + \frac{1}{3}\frac{d}{dt}E(x_i^{(2)}, t^n)\right); \end{aligned}$$

from which one can update $f^{(3)}$, $\rho^{(3)}$ and $E^{(3)}$ at t^{n+1}.

Simulation results: two-stream instability

Consider the symmetric two stream instability, the VP system with initial condition

$$f(x,v,t=0) = \frac{2}{7\sqrt{2\pi}}(1+5v^2)(1+\alpha((\cos(2kx) + \cos(3kx))/1.2 + \cos(kx))\exp(-\frac{v^2}{2}),$$

with $\alpha = 0.01$, $k = 0.5$ and the length of domain in $x-$direction $L = \frac{2\pi}{k}$.

Order or accuracy

Table: Order of accuracy in time: two stream instability with sixth order WENO interpolation $Nx = Nv = 160$ and $T = 5$.

	first order		second order		third order	
CFL	L^1 error	order	L^1 error	order	L^1 error	order
6	1.17E-4	–	2.40E-6	–	1.13E-7	–
7	1.40E-4	1.13	2.80E-6	2.04	1.79E-7	3.02
8	1.63E-4	1.16	3.69E-6	2.07	2.69E-7	3.02
9	1.87E-4	1.16	4.69E-6	2.04	3.84E-7	3.03
10	2.12E-4	1.20	5.84E-6	2.08	5.31E-7	3.06

VP system: weak Landau damping $\alpha = 0.01$

Consider the VP system with initial condition,

$$f(x, v, t = 0) = \frac{1}{\sqrt{2\pi}}(1 + \alpha \cos(kx))exp(-\frac{v^2}{2}),$$

with $\alpha = 0.01$, $k = 2$.

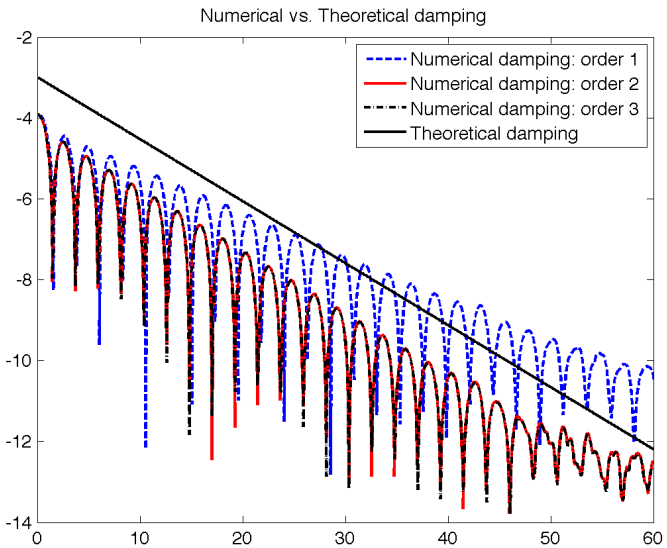


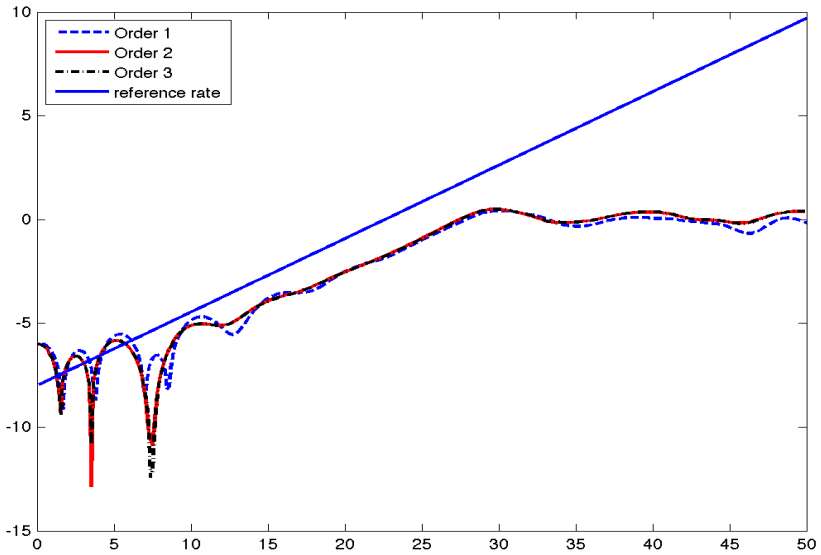
Figure: Time evolution of L^2 norm of electric field.

Symmetric two stream instability

Initial condition:

$$f(x,v,t=0) = \frac{1}{\sqrt{8\pi}v_{th}}\left[\exp\left(-\frac{(v-u)^2}{2v_{th}^2}\right)+\exp\left(-\frac{(v+u)^2}{2v_{th}^2}\right)\right](1+0.0005\cos(kx)) \tag{4}$$

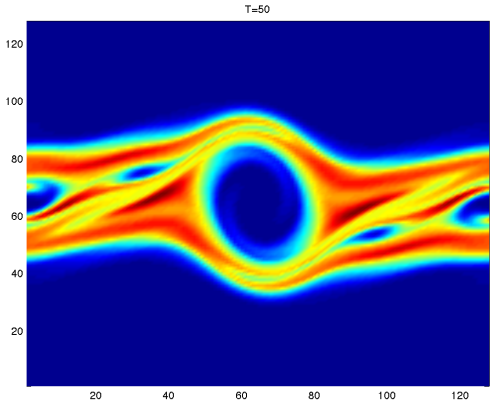
with $u = 5\sqrt{3}/4$, $v_{th} = 0.5$ and $k = 0.2$.
Constant background ion distribution chosen so that net charge density is zero.



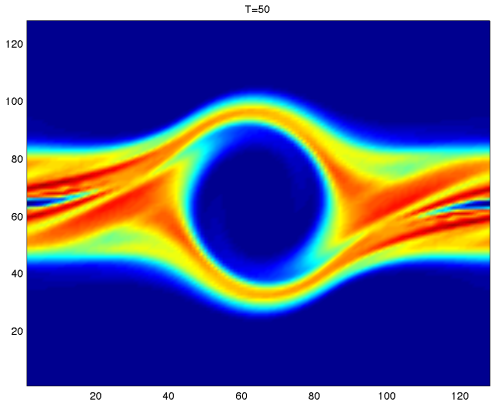
Time evolution of electric field in L^2 norm (from J. Banks and J. Hittinger, 2010)

2 stream instab.: phase space portraits

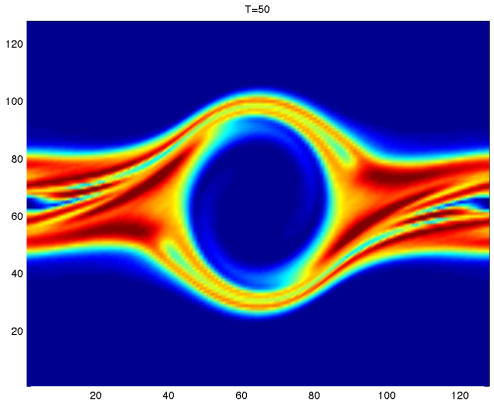
First order, CFL = 5.0



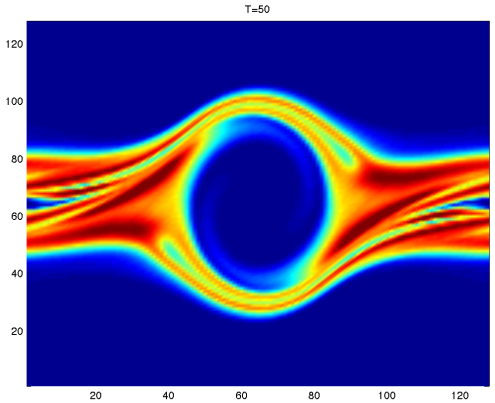
First order, CFL = 0.1



Second order, CFL = 5.0



Third order, CFL = 5.0



BGK model

The BGK model (Bhatnagar-Gross-Krook '54) approximates Boltzmann equation for the evolution of a rarefied gas.

The main variable is the distribution function f of the particles, as in the Boltzmann equation. The evolution of f is given by:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{\epsilon}(M[f] - f) \tag{5}$$

with initial condition $f(x, v, 0) = f_0(x, v)$.
Here ϵ represents the non dimensional collision time. Hydrodynamic regime $\rightarrow \epsilon \ll 1$
Rarefied regime $\rightarrow \epsilon \sim O(1)$

BGK model

The BGK model (Bhatnagar-Gross-Krook '54) approximates Boltzmann equation for the evolution of a rarefied gas.

The main variable is the distribution function f of the particles, as in the Boltzmann equation. The evolution of f is given by:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{\epsilon}(M[f] - f) \tag{5}$$

with initial condition $f(x, v, 0) = f_0(x, v)$.
Here ϵ represents the non dimensional collision time. Hydrodynamic regime $\rightarrow \epsilon \ll 1$
Rarefied regime $\rightarrow \epsilon \sim O(1)$

Here we present a numerical method for the BGK equation based on a Semi-Lagrangian formulation using BDF (Stracquadanio, Russo, Groppi, in progress).
The method is compared with a RK-based approach (Russo, Santagati, 2008).

Semi-Lagrangian formulation

Simplified model: 1D in space and velocity:

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} = \frac{1}{\epsilon}(M[f] - f). \tag{6}$$

with

$$M[f] = \frac{\rho}{(2\pi RT)^{1/2}} exp\left(- \frac{(v-u)^2}{2RT} \right)$$

$t \geq 0, \quad x, v \in \mathbb{R}.$

Semi-Lagrangian: follow the evolution along the characteristics.

$$\begin{aligned} &\frac{df(x,v,t)}{dt} = \frac{1}{\epsilon}\Big(M[f](x,v,t) - f(x,v,t) \Big), \\ &\frac{dx}{dt} = v, \quad x(0) = \tilde{x}, \quad f(0,t,v) = f_0(\tilde{x}, v) \quad t \geq 0, \quad x, v \in \mathbb{R}. \end{aligned} \tag{7}$$

Note that x becomes a time dependent variable and its equation gives:

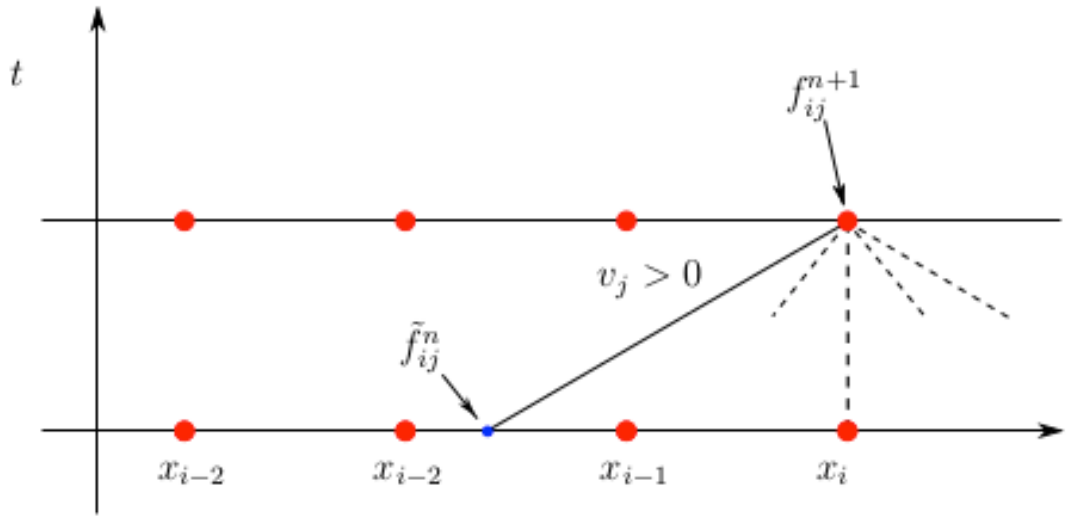
$$x(t) = \tilde{x} + vt, \quad t \geq 0, \quad x, v \in \mathbb{R}, \qquad \text{(characteristic straight lines)}.$$

Implicit first order Semi-Lagrangian scheme

Let $f_{ij}^n \approx f(x_i, v_j, t^n)$ be approximate solution.
Possible stiffness (small ϵ) $\Rightarrow$ implicit formulation.

$$f_{ij}^{n+1} = \tilde{f}_{ij}^n + \frac{\Delta t}{\epsilon}(M_{ij}^{n+1} - f_{ij}^{n+1}), \tag{8}$$

Here $\tilde{f}_{ij}^n = f(t^n, \tilde{x}_i = x_i - v_j \Delta t, v_j)$ can be calculated by (linear) interpolation from $\{f_{.j}^n\}$.



Solution of the implicit step

Equation (8) is non linear.
Indeed $M[f]_{i,j}^{n+1}$ depends on f_{ij}^{n+1} through its moments.

Solution of the implicit step

Equation (8) is non linear.
Indeed $M[f]_{i,j}^{n+1}$ depends on f_{ij}^{n+1} through its moments.

Let $\phi(v)$ be the vector $\phi(v) = (1, v, v^2)^T$. Compute the moments of f_{ij}^{n+1}:

$$\langle f_{ij}^{n+1}\phi\rangle = \langle \tilde{f}_{ij}^{n}\phi\rangle + \frac{\Delta t}{\epsilon}\langle (M_{ij}^{n+1} - f_{ij}^{n+1})\phi\rangle.$$

Solution of the implicit step

Equation (8) is non linear.
Indeed $M[f]_{i,j}^{n+1}$ depends on f_{ij}^{n+1} through its moments.

Let $\phi(v)$ be the vector $\phi(v) = (1, v, v^2)^T$. Compute the moments of f_{ij}^{n+1}:

$$\langle f_{ij}^{n+1}\phi\rangle = \langle \tilde{f}_{ij}^{n}\phi\rangle + \frac{\Delta t}{\epsilon}\langle (M_{ij}^{n+1} - f_{ij}^{n+1})\phi\rangle.$$

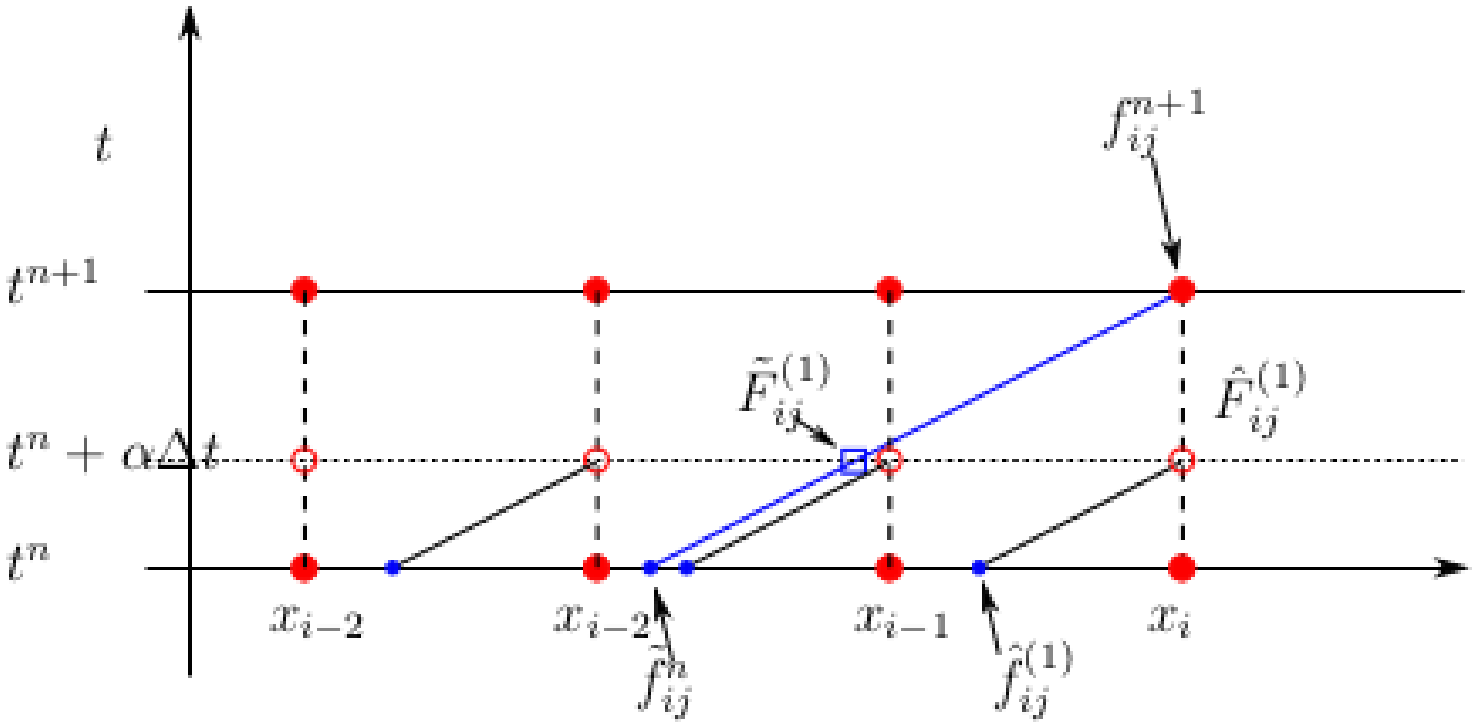
From the conservation, we have

$$\langle (M_{ij}^{n+1} - f_{ij}^{n+1})\phi\rangle = 0 \qquad \Rightarrow \qquad \langle f_{ij}^{n+1}\phi\rangle = \langle \tilde{f}_{ij}^{n}\phi\rangle$$

Hence we immediately find the macroscopic variables ρ_i^{n+1}, u_i^{n+1} and T_i^{n+1} corresponding to f_{ij}^{n+1} using $\tilde{f}_{ij}^{n}$ and with these values the approximated Maxwellian is updated.

Higher order: Runge-Kutta

Classical RK schemes can be adopted.
Stage values are computed along the characteristics.
First they are computed at grid position x_i (empty cycles) and then the value of f (or the RK flux) is interpolated on the characteristics (empty squares)



High order BDF schemes

Runge-Kutta methods may be expensive.
The BDF (Backward Difference Formula) methods allow same order of accuracy at lower cost.

We will show some numerical results concerning the BDF methods with 2 (BDF2) and 3 (BDF3) steps. Applying these methods to the Lagrangian formulation of the BGK model we obtain the following schemes:

$$f_{ij}^{n+1} = \frac{4}{3}\overset{(1)}{f}{}_{ij}^{n} - \frac{1}{3}\overset{(2)}{f}{}_{ij}^{n-1} + \frac{\Delta t}{\epsilon}(M_{ij}^{n+1} - f_{ij}^{n+1})$$

$$f_{ij}^{n+1} = \frac{11}{18}\overset{(1)}{f}{}_{ij}^{n} - \frac{9}{11}\overset{(2)}{f}{}_{ij}^{n-1} + \frac{2}{11}\overset{(3)}{f}{}_{ij}^{n-2} + \frac{\Delta t}{\epsilon}(M_{ij}^{n+1} - f_{ij}^{n+1})$$

where $\overset{(s)}{f}{}^{n}{}_{ij} = f^n(x_i - sv_j\Delta t, v_j),\ s = 1, 2, 3$, obtained by interpolation.

High order BDF schemes

Runge-Kutta methods may be expensive.
The BDF (Backward Difference Formula) methods allow same order of accuracy at lower cost.

We will show some numerical results concerning the BDF methods with 2 (BDF2) and 3 (BDF3) steps. Applying these methods to the Lagrangian formulation of the BGK model we obtain the following schemes:

$$f_{ij}^{n+1} = \frac{4}{3}\overset{(1)}{f}{}_{ij}^{n} - \frac{1}{3}\overset{(2)}{f}{}_{ij}^{n-1} + \frac{\Delta t}{\epsilon}(M_{ij}^{n+1} - f_{ij}^{n+1})$$

$$f_{ij}^{n+1} = \frac{11}{18}\overset{(1)}{f}{}_{ij}^{n} - \frac{9}{11}\overset{(2)}{f}{}_{ij}^{n-1} + \frac{2}{11}\overset{(3)}{f}{}_{ij}^{n-2} + \frac{\Delta t}{\epsilon}(M_{ij}^{n+1} - f_{ij}^{n+1})$$

where $\overset{(s)}{f}{}^{n}{}_{ij} = f^n(x_i - sv_j\Delta t, v_j),\ s = 1, 2, 3$, obtained by interpolation.
High order in space is obtained by WENO reconstruction.

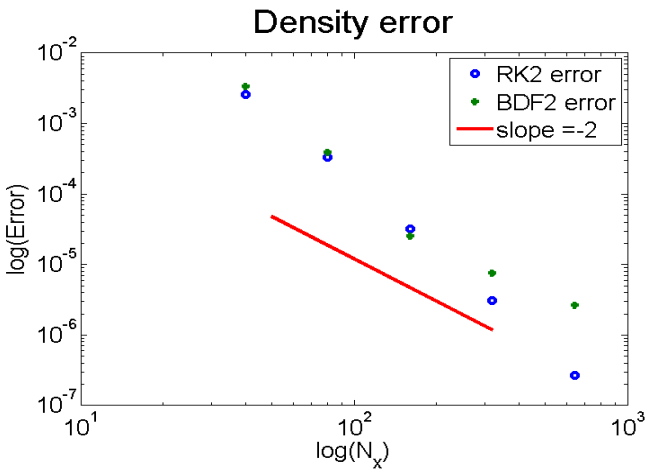
Numerical test for the problem 1+1D

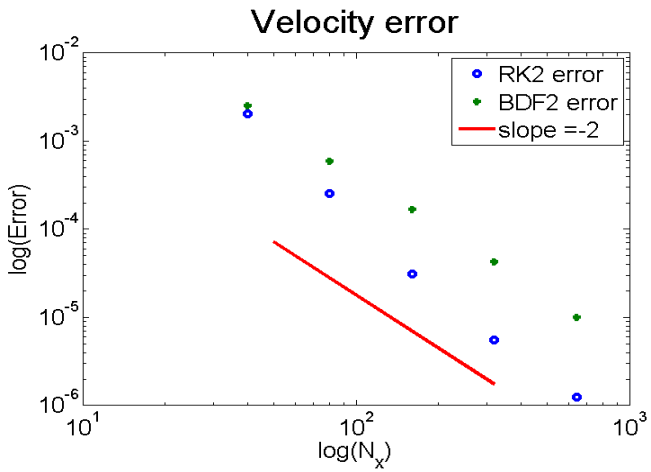
We have considered two numerical test:

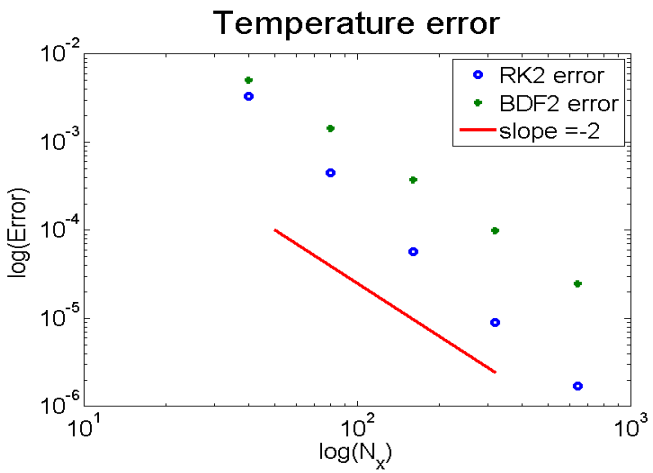
1. Smooth initial data
   $\left(f_0 = M[v, \rho = 1, u = 0.1exp(-(10x-1)^2) - 2exp(-(10x+3)^2), T = 1]\right)$
   - time interval [0,0.04];
   - space interval [-1,1];
   - velocity interval [-10,10];
   - $N_v = 40$;
   - $\Delta t = CFL\,\Delta x/|v_{max}|$;
2. Riemann problem (jump in $x = 0.5$):
   - $(\rho_L, u_L, T_L) = (2.25, 0, 1.125)$, $(\rho_R, u_R, T_R) = (3/7, 0, 1/6)$
   - time interval [0,0.16];
   - space interval [0,1];
   - velocity interval [-10,10];
   - $N_x = 100$;
   - $N_v = 60$;
   - $\Delta t = CFL\,\Delta x/|v_{max}|$;

For each test the cases $\epsilon = 10^{-2}$ and $\epsilon = 10^{-6}$ have been studied.

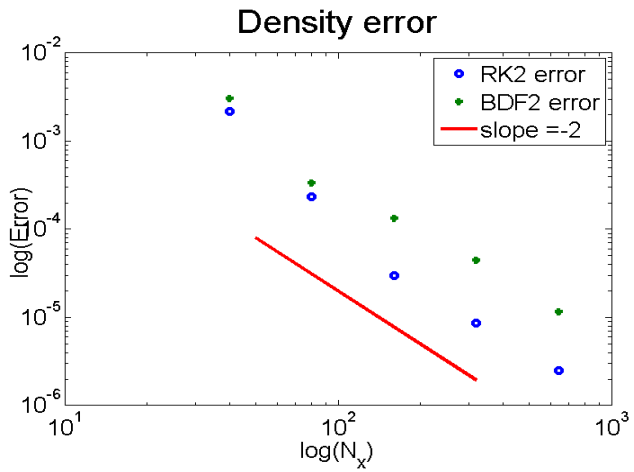
RK2 and BDF2 accuracy 1+1D in rarefied regime ($\epsilon = 10^{-2}$)

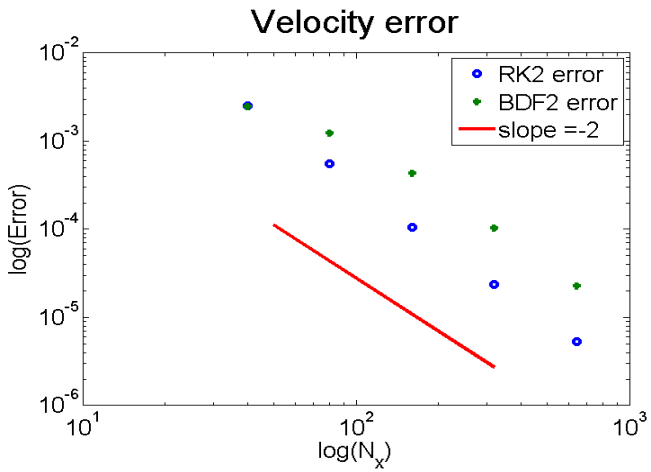


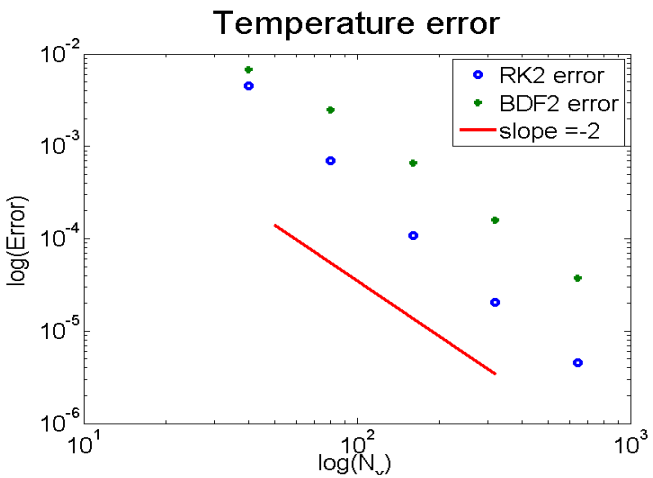




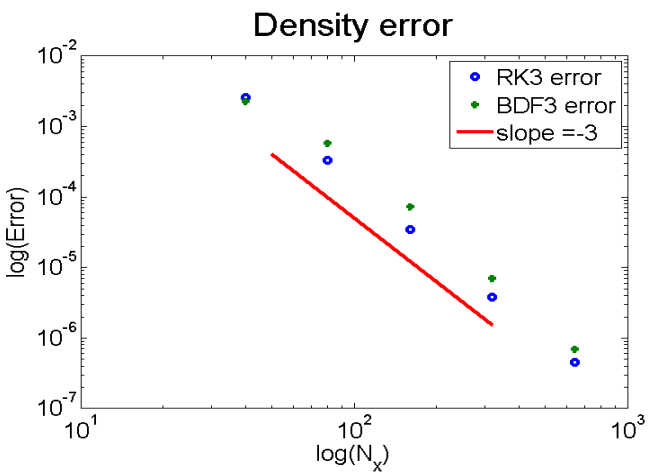
RK2 and BDF2 accuracy 1+1D in hydrodynamic regime ($\epsilon = 10^{-6}$)

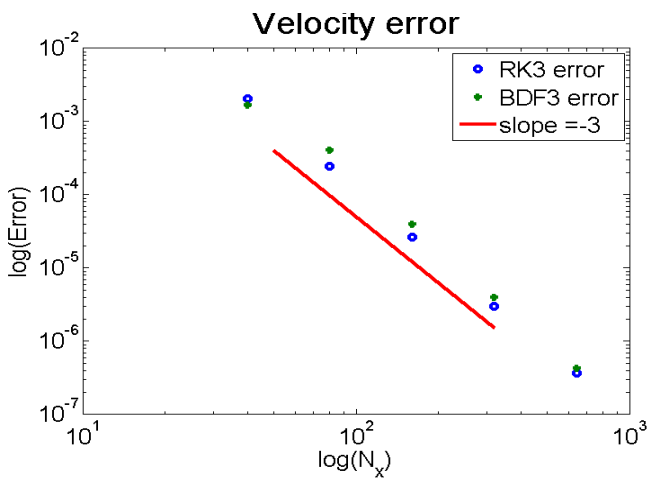


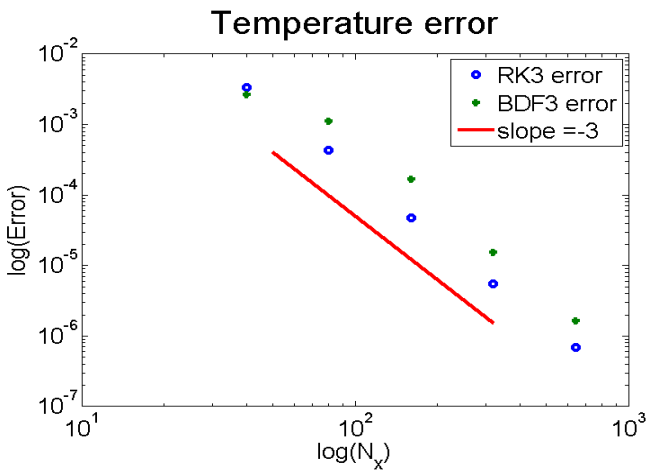




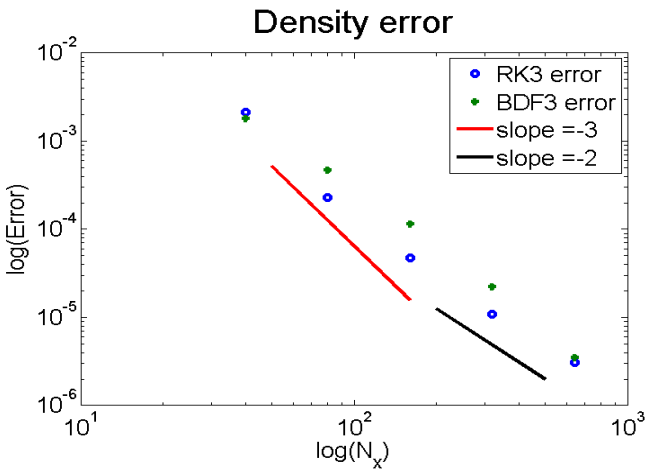
RK3 and BDF3 accuracy 1+1D in rarefied regime ($\epsilon = 10^{-2}$)

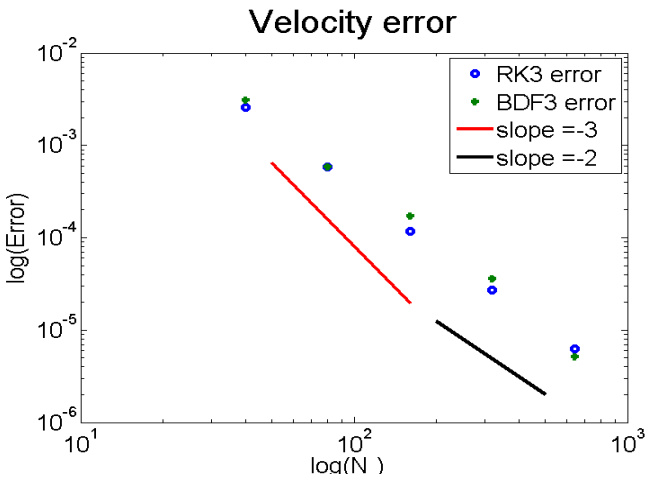


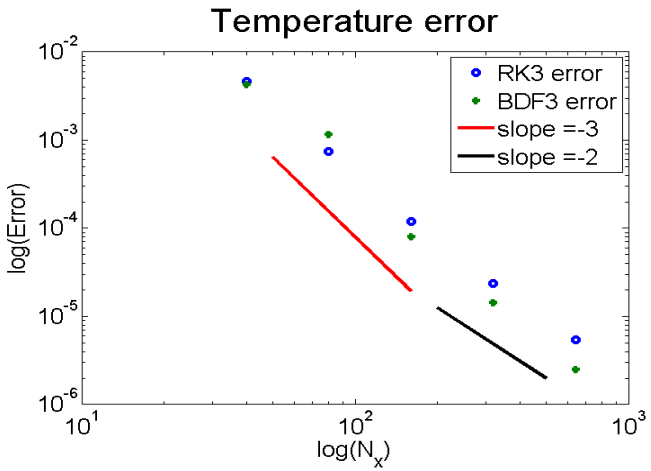




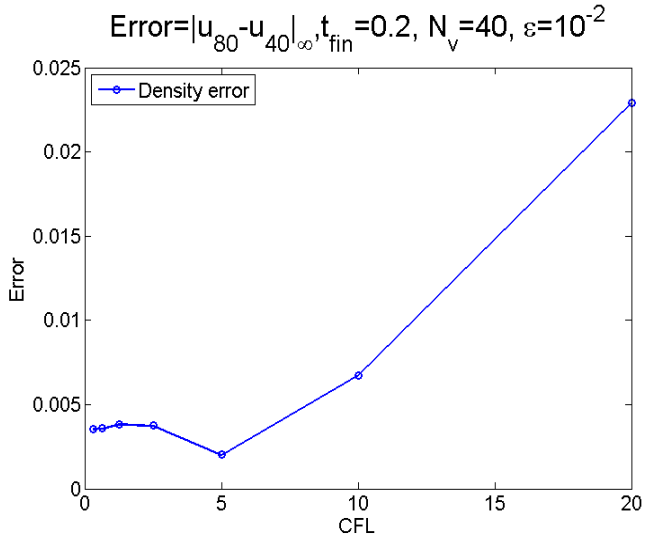
RK3 and BDF3 accuracy 1+1D in hydrodynamic regime ($\epsilon = 10^{-6}$)

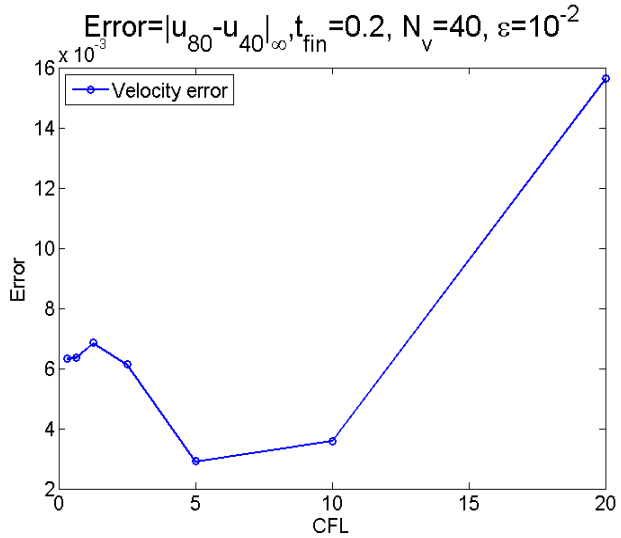


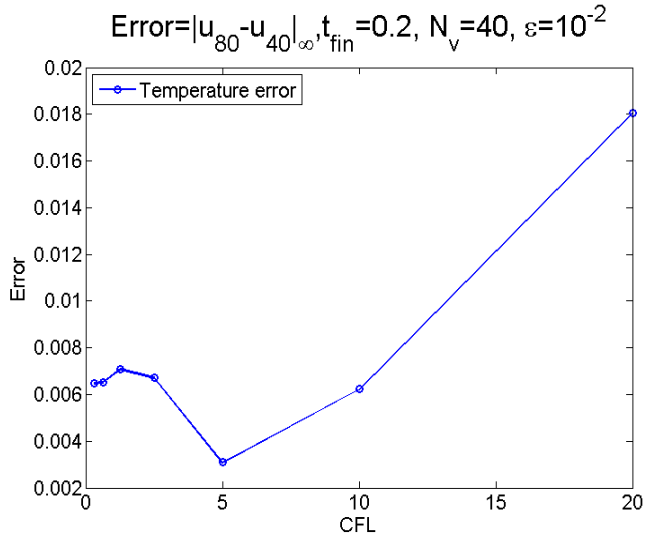




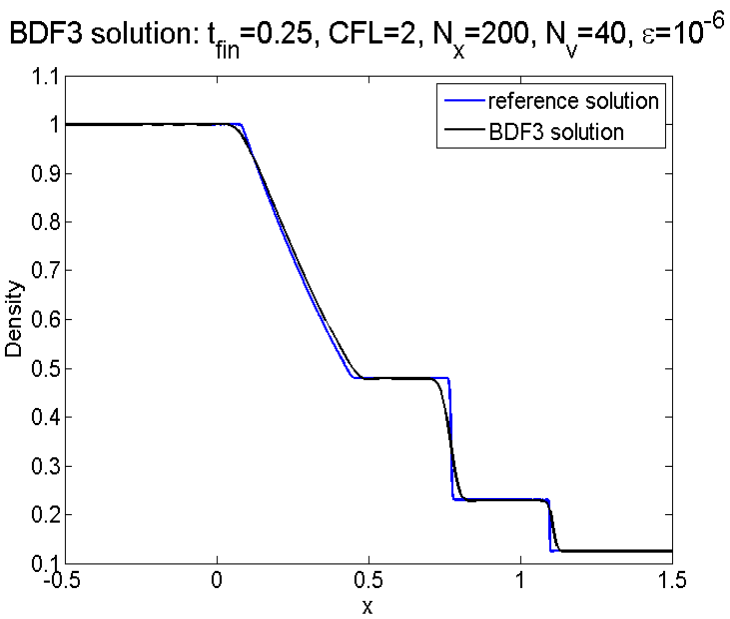
BDF3-weno3-5, CFL-Error for the problem 1D+1D

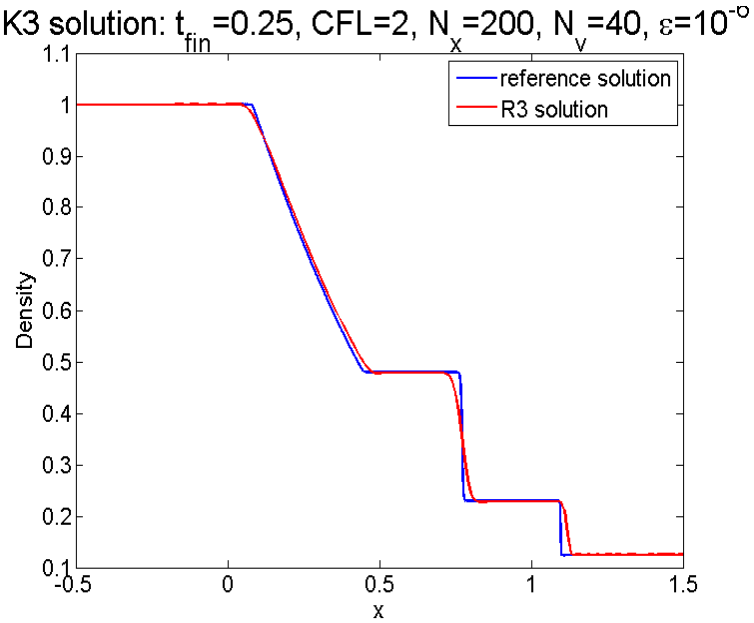




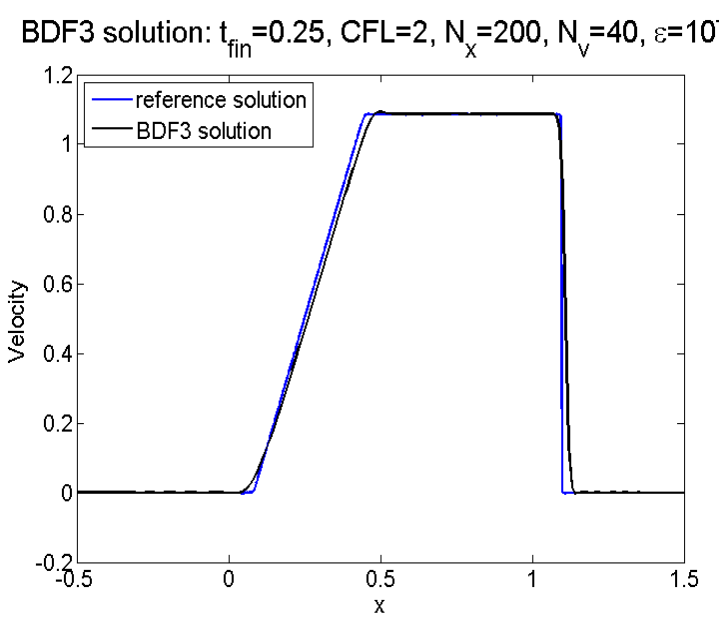


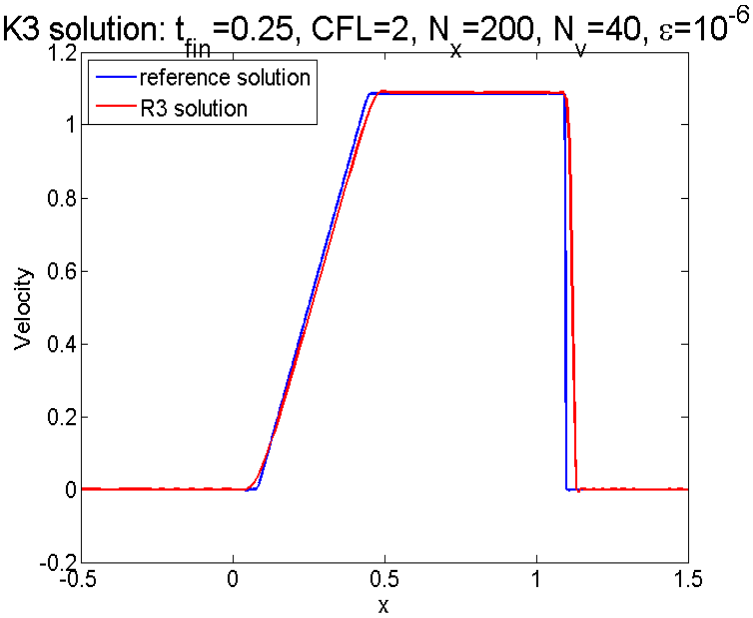
Comparison with the solution of gas dynamics: density given by BDF3 and RK3 for 1+3D



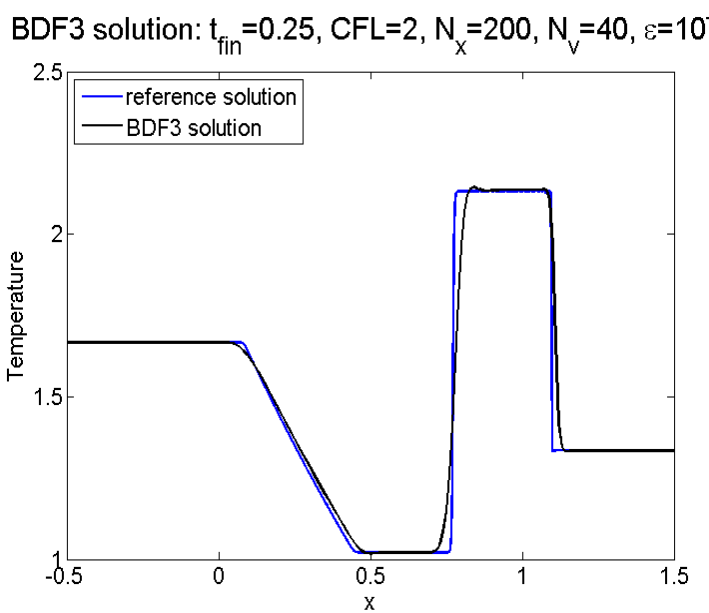


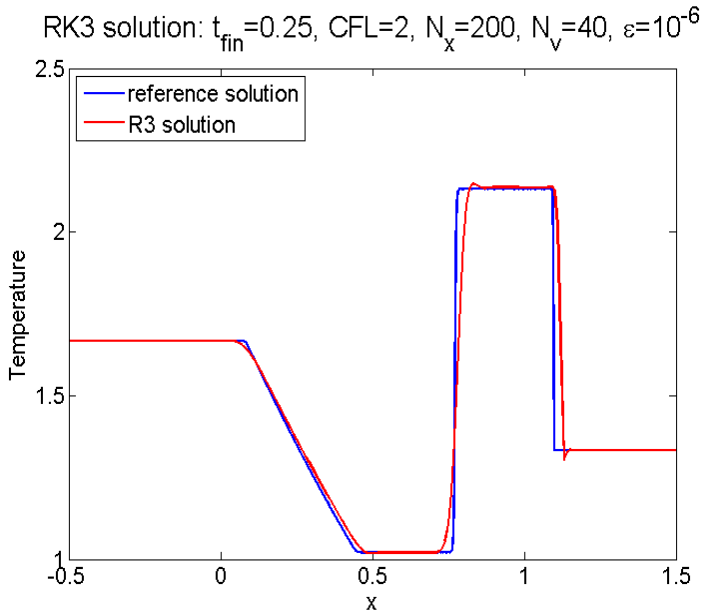
Comparison with the solution of gas dynamics: velocity given by BDF3 and RK3 for 1+3D





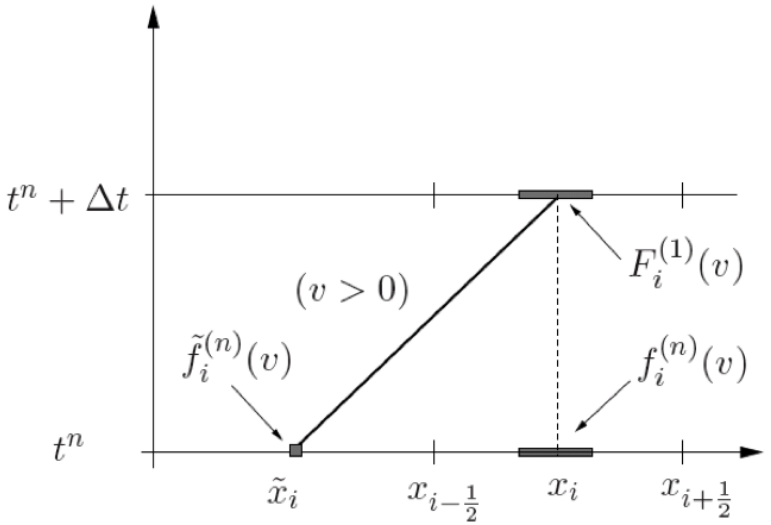
Comparison with the solution of gas dynamics: temperature given by BDF3 and RK3 for 1+3D





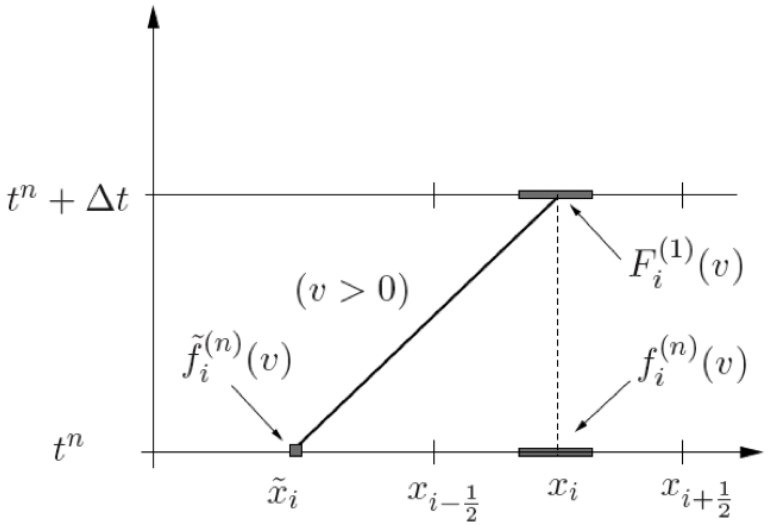
Conservative correction

Can be adopted both at *finite volume* or *conservative finite difference* level (which is what we treat here)



Conservative correction

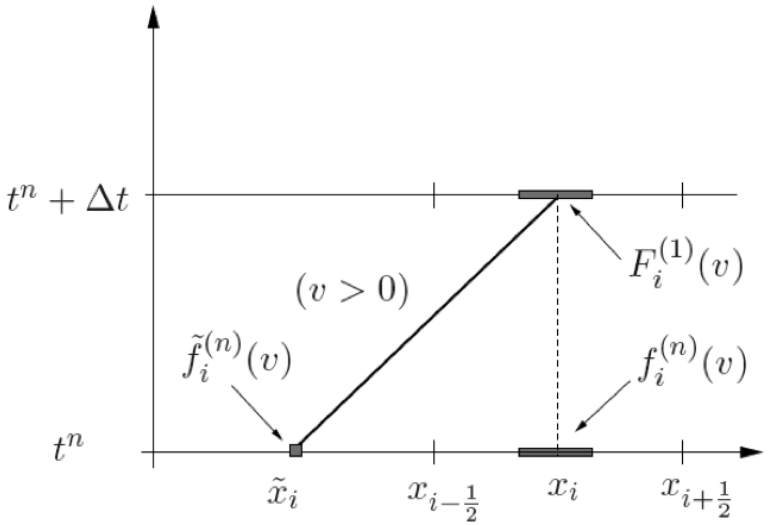
Can be adopted both at *finite volume* or *conservative finite difference* level (which is what we treat here)



1. compute a predictor value at the center of the cell

Conservative correction

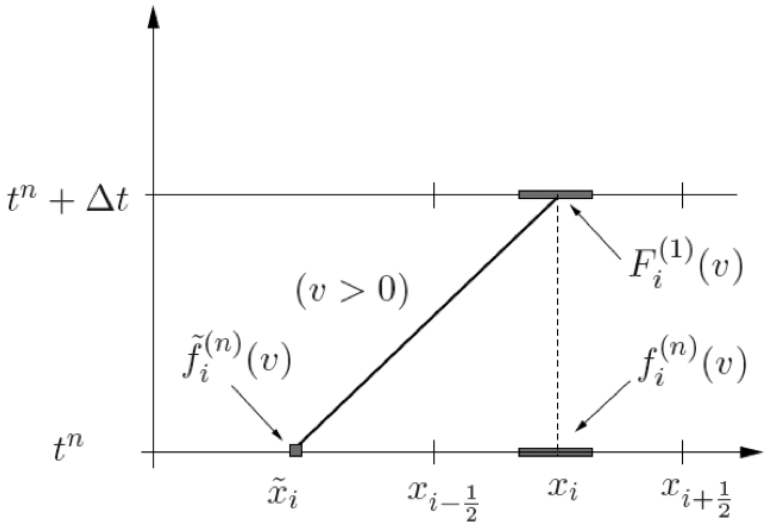
Can be adopted both at *finite volume* or *conservative finite difference* level (which is what we treat here)



1. compute a predictor value at the center of the cell
2. use such a predictor to perform reconstruction of the fluxes, at cell edges

Conservative correction

Can be adopted both at *finite volume* or *conservative finite difference* level (which is what we treat here)



1. compute a predictor value at the center of the cell
2. use such a predictor to perform reconstruction of the fluxes, at cell edges
3. evolve the conservative values according to the computed fluxes

Application to conservation laws

Consider a system of conservation laws

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

Application to conservation laws

Consider a system of conservation laws

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

One can choose a predictor based on

Application to conservation laws

Consider a system of conservation laws

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

One can choose a predictor based on

$$\frac{\partial u}{\partial t} + A(u)\frac{\partial u}{\partial x} = 0$$

Application to conservation laws

Consider a system of conservation laws

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

One can choose a predictor based on

$$\frac{\partial u}{\partial t} + A(u)\frac{\partial u}{\partial x} = 0 \qquad \text{or even} \qquad \frac{\partial v}{\partial t} + B(v)\frac{\partial v}{\partial x} = 0$$

Application to conservation laws

Consider a system of conservation laws

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

One can choose a predictor based on

$$\frac{\partial u}{\partial t} + A(u)\frac{\partial u}{\partial x} = 0 \qquad \text{or even} \qquad \frac{\partial v}{\partial t} + B(v)\frac{\partial v}{\partial x} = 0$$

where $u = U(v)$ is an invertible mapping ($v = V(u)$ is the inverse) and the formulation in v is somehow *simpler*.

Then one can apply a conservative correction using finite volume or finite difference discretization.

skip to stability

Finite volume approach

Finite volume approach

1. from $\{\bar{u}_j^n\}$ compute the pointwise values of $\{v_j^n\}$.

Finite volume approach

1. from $\{\bar{u}_j^n\}$ compute the pointwise values of $\{v_j^n\}$.
2. evolve v_j with a non conservative scheme (e.g. Runge-Kutta with ν stages)

$$v_j^{(l)} = v_j^{(1)} - \Delta t \sum_{k=1}^{l-1} a_{lk} B(v_j^{(k)})(D_x v^{(k)})_j, \quad j = 1, \ldots, N_x, \quad l = 1, \ldots, \nu,$$

$(D_x v^{(k)})_j$: numerical discretization of space derivative of $v(x, t^n + c_k \Delta t)$.

Finite volume approach

1. from $\{\bar{u}_j^n\}$ compute the pointwise values of $\{v_j^n\}$.
2. evolve v_j with a non conservative scheme (e.g. Runge-Kutta with ν stages)

$$v_j^{(l)} = v_j^{(1)} - \Delta t \sum_{k=1}^{l-1} a_{lk} B(v_j^{(k)})(D_x v^{(k)})_j, \quad j = 1, \ldots, N_x, \quad l = 1, \ldots, \nu,$$

   $(D_x v^{(k)})_j$: numerical discretization of space derivative of $v(x, t^n + c_k \Delta t)$.
3. Reconstruct (pointwise) the nonconservative variables at cell edges $v_{j+1/2}^{(k)\pm}$

Finite volume approach

1. from $\{\bar{u}_j^n\}$ compute the pointwise values of $\{v_j^n\}$.
2. evolve v_j with a non conservative scheme (e.g. Runge-Kutta with ν stages)

$$v_j^{(l)} = v_j^{(1)} - \Delta t \sum_{k=1}^{l-1} a_{lk} B(v_j^{(k)})(D_x v^{(k)})_j, \quad j = 1, \ldots, N_x, \quad l = 1, \ldots, \nu,$$

   $(D_x v^{(k)})_j$: numerical discretization of space derivative of $v(x, t^n + c_k \Delta t)$.
3. Reconstruct (pointwise) the nonconservative variables at cell edges $v_{j+1/2}^{(k)\pm}$
4. Compute the fluxes at cell edges: $f_{j+\frac{1}{2}}^{(k)} = F(u_{j+\frac{1}{2}}^{(k)-}, u_{j+\frac{1}{2}}^{(k)+}) = \tilde{F}(v_{j+\frac{1}{2}}^{(k)-}, v_{j+\frac{1}{2}}^{(k)+})$

Finite volume approach

1. from $\{\bar{u}_j^n\}$ compute the pointwise values of $\{v_j^n\}$.
2. evolve v_j with a non conservative scheme (e.g. Runge-Kutta with ν stages)

$$v_j^{(l)} = v_j^{(1)} - \Delta t \sum_{k=1}^{l-1} a_{lk} B(v_j^{(k)})(D_x v^{(k)})_j, \quad j = 1, \dots, N_x, \quad l = 1, \dots, \nu,$$

$(D_x v^{(k)})_j$: numerical discretization of space derivative of $v(x, t^n + c_k \Delta t)$.

3. Reconstruct (pointwise) the nonconservative variables at cell edges $v_{j+1/2}^{(k)\pm}$
4. Compute the fluxes at cell edges: $f_{j+\frac{1}{2}}^{(k)} = F(u_{j+\frac{1}{2}}^{(k)-}, u_{j+\frac{1}{2}}^{(k)+}) = \tilde{F}(v_{j+\frac{1}{2}}^{(k)-}, v_{j+\frac{1}{2}}^{(k)+})$
5. Evolve the conservative variables

$$\begin{aligned} \bar{u}_j^{n+1} &= \bar{u}_j^n - \frac{\Delta t}{\Delta x} \sum_{l=1}^{\nu} b_l K_l \\ K_l &= f_{j+\frac{1}{2}}^{(l)} - f_{j-\frac{1}{2}}^{(l)} \end{aligned}$$

Finite difference approach

Finite difference approach

1. compute $\{v_j^n = V(u_j)\}$.

Finite difference approach

1. compute $\{v_j^n = V(u_j)\}$.
2. evolve v_j with a non conservative scheme (e.g. Runge-Kutta with ν stages)

$$v_j^{(l)} = v_j^{(1)} - \Delta t \sum_{k=1}^{l-1} a_{lk} B(v_j^{(k)})(D_x v^{(k)})_j, \quad j = 1, \ldots, N_x, \quad l = 1, \ldots, \nu,$$

$(D_x v^{(k)})_j$: numerical discretization of space derivative of $v(x, t^n + c_k \Delta t)$.

Finite difference approach

1. compute $\{v_j^n = V(u_j)\}$.
2. evolve v_j with a non conservative scheme (e.g. Runge-Kutta with ν stages)

$$v_j^{(l)} = v_j^{(1)} - \Delta t \sum_{k=1}^{l-1} a_{lk} B(v_j^{(k)})(D_x v^{(k)})_j, \quad j = 1, \ldots, N_x, \quad l = 1, \ldots, \nu,$$

   $(D_x v^{(k)})_j$: numerical discretization of space derivative of $v(x, t^n + c_k \Delta t)$.
3. Compute the splitter fluxes f^- and f^+ $(f^- + f^+ = f)$ at cell center x_j at each stage l
4. Reconstruct (from cell average to pointwise) the fluxes f^+ and f^- at cell edges $f_{j+\frac{1}{2}}^{(k)} = f^-(x_{j+1/2}^+, t^n + c_k \Delta t) + f^+(x_{j+1/2}^-, t^n + c_k \Delta t)$

Finite difference approach

1. compute $\{v_j^n = V(u_j)\}$.
2. evolve v_j with a non conservative scheme (e.g. Runge-Kutta with ν stages)

$$v_j^{(l)} = v_j^{(1)} - \Delta t \sum_{k=1}^{l-1} a_{lk} B(v_j^{(k)})(D_x v^{(k)})_j, \quad j = 1, \ldots, N_x, \quad l = 1, \ldots, \nu,$$

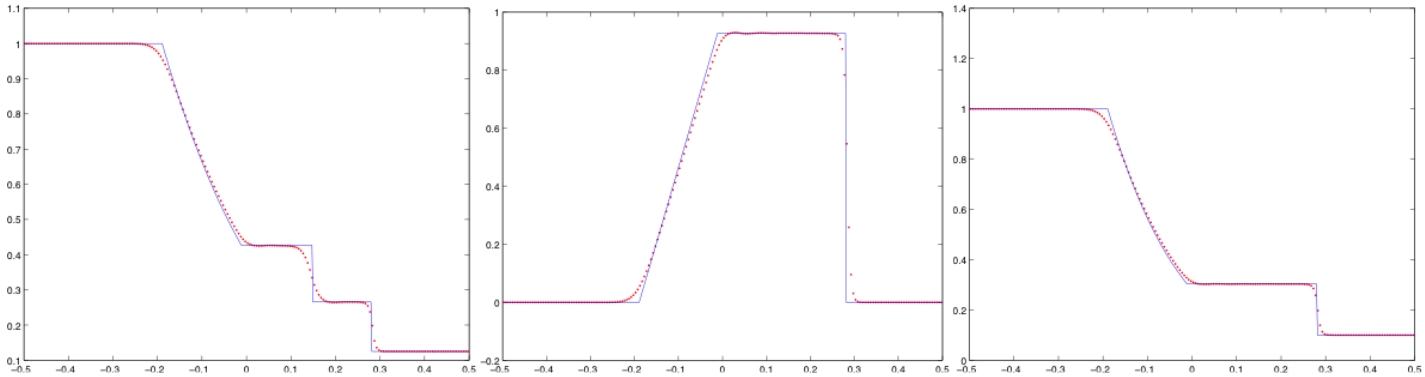
$(D_x v^{(k)})_j$: numerical discretization of space derivative of $v(x, t^n + c_k \Delta t)$.

3. Compute the splitter fluxes f^- and f^+ $(f^- + f^+ = f)$ at cell center x_j at each stage l
4. Reconstruct (from cell average to pointwise) the fluxes f^+ and f^- at cell edges $f_{j+\frac{1}{2}}^{(k)} = f^-(x_{j+1/2}^+, t^n + c_k \Delta t) + f^+(x_{j+1/2}^-, t^n + c_k \Delta t)$
5. Evolve the conservative pointwise variables

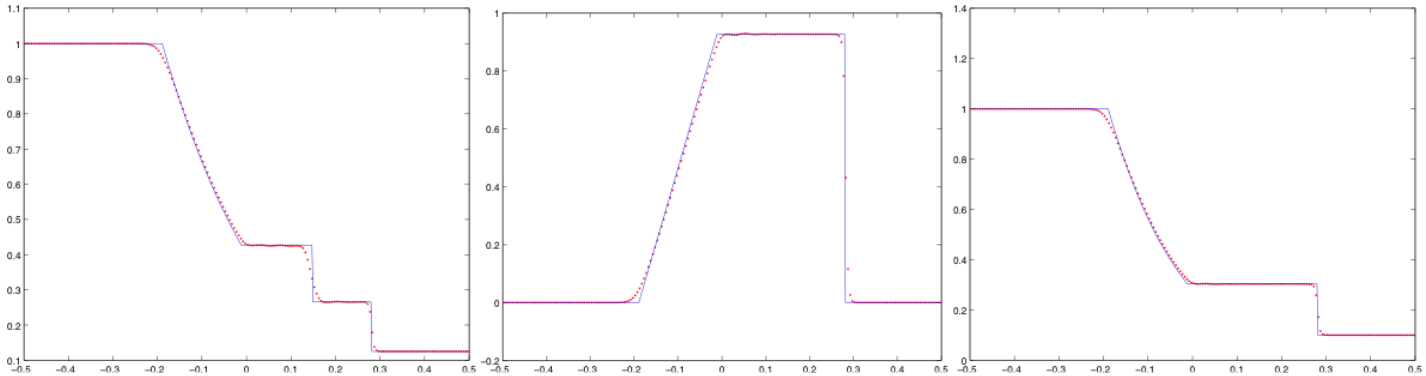
$$\begin{aligned} u_j^{n+1} &= u_j^n - \frac{\Delta t}{\Delta x} \sum_{l=1}^{\nu} b_l K_l \\ K_l &= f_{j+\frac{1}{2}}^{(l)} - f_{j-\frac{1}{2}}^{(l)} \end{aligned}$$

Application to gas dynamics

Classical Sod problem solved using primitive variables as predictor



WENO 2-3



WENO 3-5

Stability analysis

Consider linear convective equation

$$u_t + (vu)_x = 0,$$

Evolve by conservative FD scheme:

$$\frac{du_j}{dt} = -\frac{1}{\Delta x}\left(\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}\right),$$

The numerical solution is computed as

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\sum_{\ell=1}^{s} b_\ell \left(\hat{f}^{(\ell)}_{j+\frac{1}{2}} - \hat{f}^{(\ell)}_{j-\frac{1}{2}}\right).$$

$$u_j^{(\ell)} = u^n(x_j^{(\ell)}), \quad x_j^{(\ell)} = x_j - vc_\ell\Delta t$$

Look for Fourier modes

$$u_j^n[\xi] = \rho^n e^{ij\xi},$$

Stability analysis

Use Fourier interpolation for arbitrary x

$$u^n(x) = \rho^n e^{ix\xi/\Delta x}.$$

Compute the non conservative semilagrangian stages

$$u_j^{(\ell)} = u^n(x_j^{(\ell)}) = \rho^n \exp(i\xi(x_j - v\Delta t c_\ell)/\Delta x) = \rho^n e^{ij\xi} e^{-ic_\ell a\xi},$$

From this obtain the amplification factor

$$\rho = 1 - i\xi a \sum_{\ell=1}^{s} b_\ell \exp(-ic_\ell a\xi).$$

Analogy with A-stability

Test equation for A-stability

$$w'(t) = \lambda w(t), \qquad w(0) = 1$$

Exact solution

$$w(\Delta t) = e^{\lambda \Delta t} = e^z,$$

Identity obtained observing that $\int_0^1 e^{cz}\, dc = (e^z - 1)/z$:

$$e^z = 1 + z \int_0^1 e^{cz}.$$

Using exact Fourier interpolation, the error is due to the use of quadrature rule to compute the integral:

$$R(z) = 1 + z \sum_{\ell=1}^{s} b_\ell e^{c_\ell z}.$$

Therefore:

$$\rho = R(-i\xi a).$$

Optimal quadrature formulas: given s stages, choose the scheme of order s with the largest stability region.

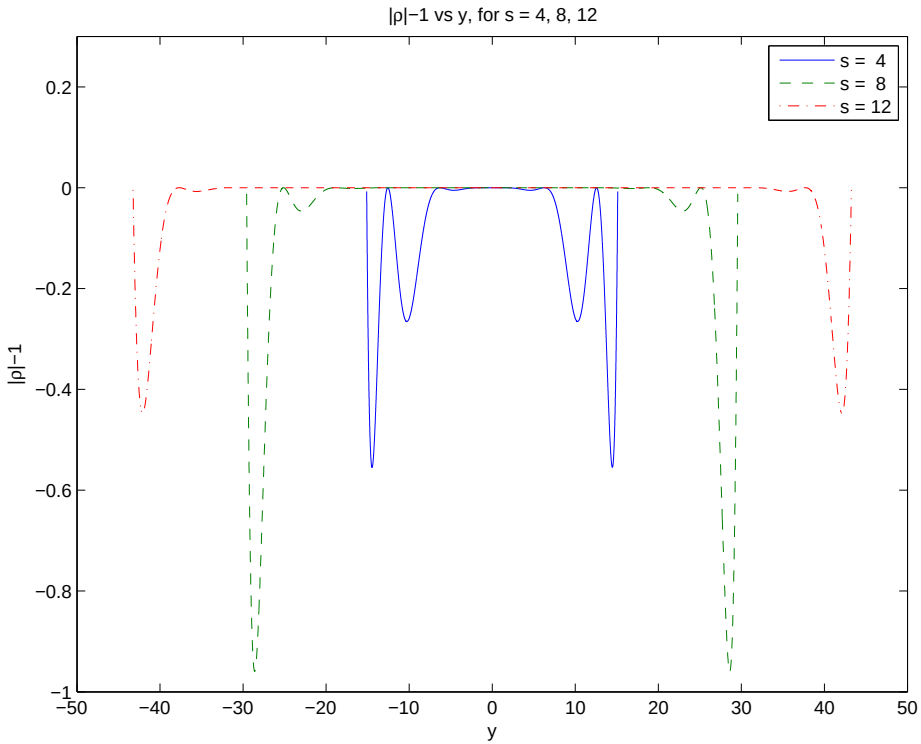
Optimal quadrature formulas: given s stages, choose the scheme of order s with the largest stability region. Simple optimal formulas achieve:

$s = 4,\ a^* = 4.81$
$s = 8,\ a^* = 9.41$
$s = 12,\ a^* = 13.77$

Optimal quadrature formulas: given s stages, choose the scheme of order s with the largest stability region. Simple optimal formulas achieve:

$s = 4,\ a^* = 4.81$
$s = 8,\ a^* = 9.41$
$s = 12,\ a^* = 13.77$

The function $|\rho| - 1$ is given by

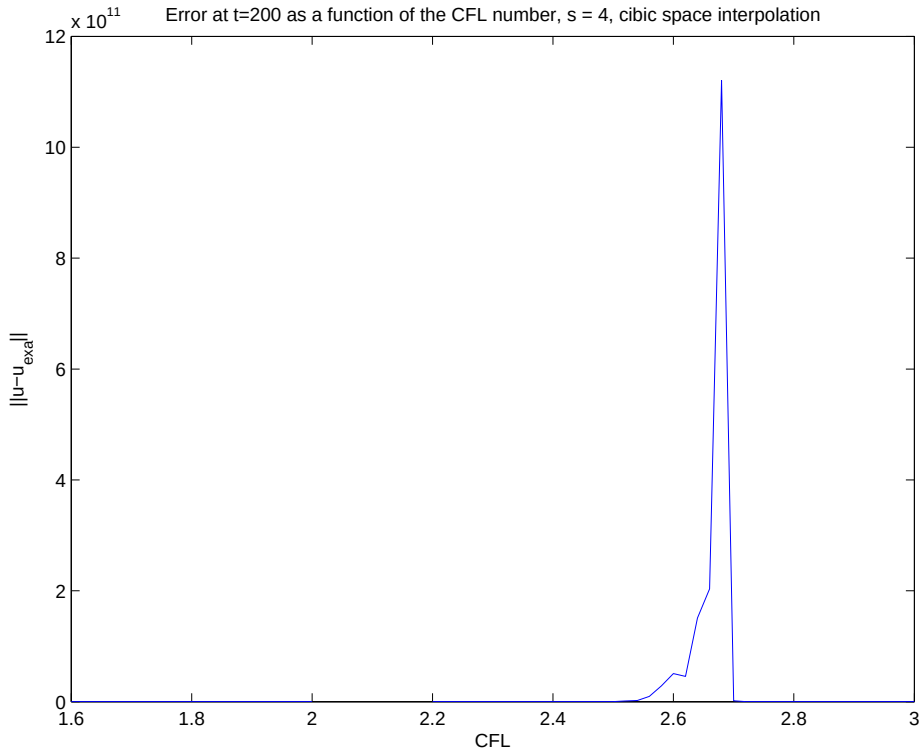


The bad news

Numerical codes for the single scalar equation with such schemes show instabilities for some CFL numbers much smaller than a^*.

The bad news

Numerical codes for the single scalar equation with such schemes show instabilities for some CFL numbers much smaller than a^*.
For example: using a third degree polynomial (4th order space interpolation) rather than Fourier interpolation, $s = 8$, one obtains instability in a neighborhood of $a = 2.6$.



The instability disappears for larger values of a, up to about the theoretical value $a^* = 4.81$.
The stability interval has holes (resonance?)

The instability disappears for larger values of a, up to about the theoretical value $a^* = 4.81$.
The stability interval has holes (resonance?)
Possible solutions (work in progress):

- check stability of different stencils and let nonlinear reconstruction choose the stable stencil
- reformulate the whole stability theory replacing Fourier interpolation by polynomial interpolations

The instability disappears for larger values of a, up to about the theoretical value $a^* = 4.81$.
The stability interval has holes (resonance?)
Possible solutions (work in progress):

- check stability of different stencils and let nonlinear reconstruction choose the stable stencil
- reformulate the whole stability theory replacing Fourier interpolation by polynomial interpolations
- Basic question: is this of any use, or there is no way to make such conservative correction stable in practice?

Conclusions

Conclusions

- Semilagrangian schemes are very promising for kinetic equations because:

Conclusions

- Semilagrangian schemes are very promising for kinetic equations because:
- Non conservative schemes allow high order accuracy and large CFL numbers

Conclusions

- Semilagrangian schemes are very promising for kinetic equations because:
- Non conservative schemes allow high order accuracy and large CFL numbers
- For VP-equations non splitting BSL schemes can be constructed, thus avoiding splitting error

Conclusions

- Semilagrangian schemes are very promising for kinetic equations because:
- Non conservative schemes allow high order accuracy and large CFL numbers
- For VP-equations non splitting BSL schemes can be constructed, thus avoiding splitting error
- For BGK high accuracy in space and time can be reached, using RK or BDF

Conclusions

- Semilagrangian schemes are very promising for kinetic equations because:
- Non conservative schemes allow high order accuracy and large CFL numbers
- For VP-equations non splitting BSL schemes can be constructed, thus avoiding splitting error
- For BGK high accuracy in space and time can be reached, using RK or BDF
- RK are more accurate, but BDF appear are more efficient

Conclusions

- Semilagrangian schemes are very promising for kinetic equations because:
- Non conservative schemes allow high order accuracy and large CFL numbers
- For VP-equations non splitting BSL schemes can be constructed, thus avoiding splitting error
- For BGK high accuracy in space and time can be reached, using RK or BDF
- RK are more accurate, but BDF appear are more efficient
- No formal restriction on CFL. In practice CFL is restricted by the fluid-dynamic CFL condition (to avoid oscillations)

Conclusions

- Semilagrangian schemes are very promising for kinetic equations because:
- Non conservative schemes allow high order accuracy and large CFL numbers
- For VP-equations non splitting BSL schemes can be constructed, thus avoiding splitting error
- For BGK high accuracy in space and time can be reached, using RK or BDF
- RK are more accurate, but BDF appear are more efficient
- No formal restriction on CFL. In practice CFL is restricted by the fluid-dynamic CFL condition (to avoid oscillations)
- A general technique is proposed to construct a conservative scheme starting from a non conservative one

Conclusions

- Semilagrangian schemes are very promising for kinetic equations because:
- Non conservative schemes allow high order accuracy and large CFL numbers
- For VP-equations non splitting BSL schemes can be constructed, thus avoiding splitting error
- For BGK high accuracy in space and time can be reached, using RK or BDF
- RK are more accurate, but BDF appear are more efficient
- No formal restriction on CFL. In practice CFL is restricted by the fluid-dynamic CFL condition (to avoid oscillations)
- A general technique is proposed to construct a conservative scheme starting from a non conservative one
- Deeper analysis is needed to understand and improve the stability property of such characteristic correction.

Conclusions

- Semilagrangian schemes are very promising for kinetic equations because:
- Non conservative schemes allow high order accuracy and large CFL numbers
- For VP-equations non splitting BSL schemes can be constructed, thus avoiding splitting error
- For BGK high accuracy in space and time can be reached, using RK or BDF
- RK are more accurate, but BDF appear are more efficient
- No formal restriction on CFL. In practice CFL is restricted by the fluid-dynamic CFL condition (to avoid oscillations)
- A general technique is proposed to construct a conservative scheme starting from a non conservative one
- Deeper analysis is needed to understand and improve the stability property of such characteristic correction.

Thank you !

Maurizio, Roberto and Giovanni in 2002