Deterministic Methods for the Boltzmann Equation

A. Narayan and A. Klöckner

AM281.1 Convection-Dominated Problems
October 15, 2007

Outline

Outline

The Boltzmann Equation

The Boltzmann Equation:

- Is an equation of *Statistical* Mechanics.

The Boltzmann Equation

The Boltzmann Equation:

- Is an equation of *Statistical* Mechanics.
- Describes a rarefied gas.

The Boltzmann Equation

The Boltzmann Equation:

- Is an equation of *Statistical* Mechanics.
- Describes a rarefied gas.

Rarefied gas?

The Boltzmann Equation

The Boltzmann Equation:

- Is an equation of *Statistical* Mechanics.
- Describes a rarefied gas.

Rarefied gas?
$\longrightarrow$ Enough space that particles in the same volume element may have differing velocities.

The Boltzmann Equation

The Boltzmann Equation:

- Is an equation of *Statistical* Mechanics.
- Describes a rarefied gas.

Rarefied gas?
$\longrightarrow$ Enough space that particles in the same volume element may have differing velocities.

Contrast: *Continuum* Mechanics

The Boltzmann Equation

The Boltzmann Equation:

- Is an equation of *Statistical* Mechanics.
- Describes a rarefied gas.

Rarefied gas?
$\longrightarrow$ Enough space that particles in the same volume element may have differing velocities.

Contrast: *Continuum* Mechanics
$\longrightarrow$All particles in a volume element are moving with constant velocity.

The Boltzmann Equation

The Boltzmann Equation:

- Is an equation of *Statistical* Mechanics.
- Describes a rarefied gas.

Rarefied gas?
$\rightarrow$ Enough space that particles in the same volume element may have differing velocities.

Contrast: *Continuum* Mechanics
$\rightarrow$All particles in a volume element are moving with constant velocity.
(Navier-Stokes Equations)

Density Functions

The Boltzmann equation is solved by a *density function* $f(x, v, t)$

Density Functions

The Boltzmann equation is solved by a *density function* $f(x, v, t)$

$$f : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}_0^+.$$

Density Functions

The Boltzmann equation is solved by a *density function* $f(x, v, t)$

$$f : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}_0^+.$$

Density function:

Density Functions

The Boltzmann equation is solved by a *density function* $f(x, v, t)$

$$f : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}_0^+.$$

Density function:

$$\int_{\mathbb{R}^3 \times \mathbb{R}^3} f(v, x, t) \mathrm{d}x \mathrm{d}v = n(t).$$

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

Also with Forces:

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

Also with Forces:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

Also with Forces:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

- LHS: particle transport according to v ("*Vlasov Equation*")

Forces may come from:

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

Also with Forces:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

- LHS: particle transport according to v ("*Vlasov Equation*")
- RHS: describes binary collisions between gas particles

Forces may come from:

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

Also with Forces:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

- LHS: particle transport according to v ("*Vlasov Equation*")
- RHS: describes binary collisions between gas particles

Forces may come from:

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

Also with Forces:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

- LHS: particle transport according to v ("*Vlasov Equation*")
- RHS: describes binary collisions between gas particles

Forces may come from: Maxwell's Equations (Lorentz force),

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f,f), \qquad x, v \in \mathbb{R}^3$$

Also with Forces:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \frac{1}{k_n} Q(f,f), \qquad x, v \in \mathbb{R}^3$$

- LHS: particle transport according to v ("*Vlasov Equation*")
- RHS: describes binary collisions between gas particles

Forces may come from: Maxwell's Equations (Lorentz force), Poisson Equation (electrostatic force),

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f,f), \qquad x, v \in \mathbb{R}^3$$

Also with Forces:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \frac{1}{k_n} Q(f,f), \qquad x, v \in \mathbb{R}^3$$

- LHS: particle transport according to v ("*Vlasov Equation*")
- RHS: describes binary collisions between gas particles

Forces may come from: Maxwell's Equations (Lorentz force), Poisson Equation (electrostatic force), etc.

The Boltzmann Equation

The Boltzmann equation is an Integro-PDE.

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

Also with Forces:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \frac{1}{k_n} Q(f, f), \qquad x, v \in \mathbb{R}^3$$

- LHS: particle transport according to v ("*Vlasov Equation*")
- RHS: describes binary collisions between gas particles

Forces may come from: Maxwell's Equations (Lorentz force), Poisson Equation (electrostatic force), etc. Force PDEs depend on f, making the whole system nonlinear.

Knudsen Number

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f)$$

What is k_n?

Knudsen Number

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f)$$

What is k_n? $\rightarrow$ The *Knudsen Number*.

Knudsen Number

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f)$$

What is k_n? $\rightarrow$ The *Knudsen Number*.

$$k_n := \frac{\lambda}{L} := \frac{\text{mean free path}}{\text{representative physical length scale}}$$

Knudsen Number

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f)$$

What is k_n? $\rightarrow$ The *Knudsen Number*.

$$k_n := \frac{\lambda}{L} := \frac{\text{mean free path}}{\text{representative physical length scale}}$$

$k_n \rightarrow 0$:

Knudsen Number

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{k_n} Q(f, f)$$

What is k_n? $\rightarrow$ The *Knudsen Number*.

$$k_n := \frac{\lambda}{L} := \frac{\text{mean free path}}{\text{representative physical length scale}}$$

$k_n \rightarrow 0$: Continuum limit.

Collision Term

The collision term splits into *gain* and *loss*.

Collision Term

The collision term splits into *gain* and *loss*.

$$Q(f,f) = Q^{+}(f,f) - L[f]f$$

Collision Term

The collision term splits into *gain* and *loss*.

$$Q(f,f) = Q^{+}(f,f) - L[f]f$$

Gain Term:

Collision Term

The collision term splits into *gain* and *loss*.

$$Q(f,f) = Q^{+}(f,f) - L[f]f$$

Gain Term:

$$Q^{+}(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B(|v-v_*|,\cos\theta)f(v')f(v'_*)\mathrm{d}\omega\mathrm{d}v_*,$$

Collision Term

The collision term splits into *gain* and *loss*.

$$Q(f,f) = Q^{+}(f,f) - L[f]f$$

Gain Term:

$$Q^{+}(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B(|v - v_*|, \cos\theta) f(v') f(v'_*) \mathrm{d}\omega \mathrm{d}v_*,$$

Loss Term:

Collision Term

The collision term splits into *gain* and *loss*.

$$Q(f,f) = Q^+(f,f) - L[f]f$$

Gain Term:

$$Q^+(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B(|v - v_*|, \cos\theta) f(v') f(v'_*) \mathrm{d}\omega \mathrm{d}v_*,$$

Loss Term:

$$L[f] = \int_{\mathbb{R}^3}\int_{S^2} B(|v - v_*|, \cos\theta) f(v_*) \mathrm{d}\omega \mathrm{d}v_*.$$

Collision Term

The collision term splits into *gain* and *loss*.

$$Q(f,f) = Q^+(f,f) - L[f]f$$

Gain Term:

$$Q^+(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B(|v-v_*|,\cos\theta) f(v')f(v'_*)\mathrm{d}\omega\mathrm{d}v_*,$$

Loss Term:

$$L[f] = \int_{\mathbb{R}^3}\int_{S^2} B(|v-v_*|,\cos\theta) f(v_*)\mathrm{d}\omega\mathrm{d}v_*.$$

Arggh. Too much notation.

Collision Term

The collision term splits into *gain* and *loss*.

$$Q(f,f) = Q^+(f,f) - L[f]f$$

Gain Term:

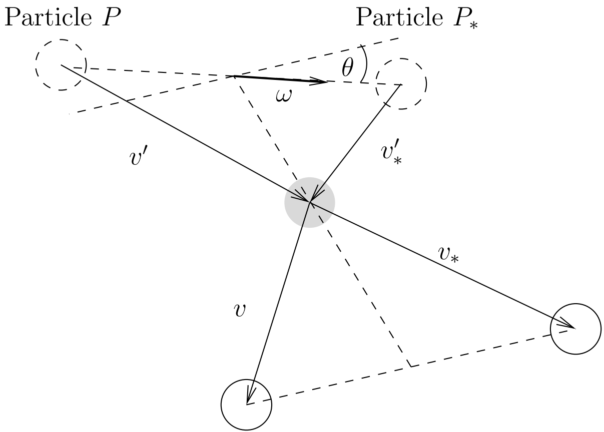
$$Q^+(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B(|v-v_*|, \cos\theta) f(v') f(v'_*) \mathrm{d}\omega \mathrm{d}v_*,$$

Loss Term:

$$L[f] = \int_{\mathbb{R}^3}\int_{S^2} B(|v-v_*|, \cos\theta) f(v_*) \mathrm{d}\omega \mathrm{d}v_*.$$

Arggh. Too much notation. What are we describing, anyway?

Binary Collisions



Collision Term

Ok, let's look at this again.

Collision Term

Ok, let's look at this again.

$$Q(f,f) = Q^{+}(f,f) - L[f]f$$

Collision Term

Ok, let's look at this again.

$$Q(f,f) = Q^+(f,f) - L[f]f$$

Gain Term:

Collision Term

Ok, let's look at this again.

$$Q(f,f) = Q^{+}(f,f) - L[f]f$$

Gain Term:

$$Q^{+}(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B(|v - v_*|, \cos\theta) f(v') f(v'_*) \mathrm{d}\omega \mathrm{d}v_*,$$

Collision Term

Ok, let's look at this again.

$$Q(f,f) = Q^{+}(f,f) - L[f]f$$

Gain Term:

$$Q^{+}(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B(|v - v_*|, \cos\theta) f(v') f(v'_*) \mathrm{d}\omega \mathrm{d}v_*,$$

Loss Term:

Collision Term

Ok, let's look at this again.

$$Q(f,f) = Q^{+}(f,f) - L[f]f$$

Gain Term:

$$Q^{+}(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B(|v - v_*|, \cos\theta) f(v') f(v'_*) \mathrm{d}\omega \mathrm{d}v_*,$$

Loss Term:

$$L[f]f = \int_{\mathbb{R}^3}\int_{S^2} B(|v - v_*|, \cos\theta) f(v) f(v_*) \mathrm{d}\omega \mathrm{d}v_*.$$

Collision Term

Ok, let's look at this again.

$$Q(f,f) = Q^+(f,f) - L[f]f$$

Gain Term:

$$Q^+(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B(|v - v_*|, \cos\theta) f(v') f(v'_*) \mathrm{d}\omega \mathrm{d}v_*,$$

Loss Term:

$$L[f]f = \int_{\mathbb{R}^3}\int_{S^2} B(|v - v_*|, \cos\theta) f(v) f(v_*) \mathrm{d}\omega \mathrm{d}v_*.$$

Q is a spatially local operator!

Collision Kernels

- *Maxwellian gas:* $B(|v - v_*|, \cos\theta) = \text{const.}$

Collision Kernels

- *Maxwellian gas:* $B(|v - v_*|, \cos\theta) = \text{const.}$
- *Hard Sphere gas:* $B(|v - v_*|, \cos\theta) = \text{const} \cdot |v - v_*|.$

Collision Kernels

- *Maxwellian gas:* $B(|v - v_*|, \cos\theta) = \text{const}.$
- *Hard Sphere gas:* $B(|v - v_*|, \cos\theta) = \text{const} \cdot |v - v_*|.$
- *Variable Hard Sphere (VHS) gas:*
  $B(|v - v_*|, \cos\theta) = \text{const} \cdot |v - v_*|^\alpha.$ (generalizes both cases above)

Collision Kernels

- *Maxwellian gas:* $B(|v - v_*|, \cos\theta) = \text{const}.$
- *Hard Sphere gas:* $B(|v - v_*|, \cos\theta) = \text{const} \cdot |v - v_*|.$
- *Variable Hard Sphere (VHS) gas:* $B(|v - v_*|, \cos\theta) = \text{const} \cdot |v - v_*|^\alpha.$ (generalizes both cases above)

Of course, one can cook up many more complicated collision kernels.

Challenges

- Dimensionality: Eleven nested loops (naively) $\rightarrow$ horrible!

Challenges

- Dimensionality: Eleven nested loops (naively) $\rightarrow$ horrible!
- Timestepping: Different for transport and collision
  $\rightarrow$ Splitting (next section)

Challenges

- Dimensionality: Eleven nested loops (naively) $\longrightarrow$ horrible!
- Timestepping: Different for transport and collision
  $\longrightarrow$ Splitting (next section)
- Shock Waves (Transport $\longrightarrow$ Section 3)

Challenges

- Dimensionality: Eleven nested loops (naively) $\to$ horrible!
- Timestepping: Different for transport and collision
  $\to$ Splitting (next section)
- Shock Waves (Transport $\to$ Section 3)
- Collision operator: Deepest dimensionality. Can we do better than five nested loops at each point? (Collision $\to$ Section 4)

Challenges

- Dimensionality: Eleven nested loops (naively) $\longrightarrow$ horrible!
- Timestepping: Different for transport and collision
  $\longrightarrow$ Splitting (next section)
- Shock Waves (Transport $\longrightarrow$ Section 3)
- Collision operator: Deepest dimensionality. Can we do better than five nested loops at each point? (Collision $\longrightarrow$ Section 4)
- Positivity (if there is time)

Questions?

?

Outline

General Framework

- Semi-discrete form of Boltzmann equation

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f_1(y) + f_2(y)$$

- f_1 and f_2 may have significantly different properties
  - f_1 represents convection
  - f_2 is usually a stiff discretization of collision

General Framework

- Semi-discrete form of Boltzmann equation

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f_1(y) + f_2(y)$$

- f_1 and f_2 may have significantly different properties
  - f_1 represents convection
  - f_2 is usually a stiff discretization of collision
- Possible nonlinearity of f_1: high-order accurate explicit method
- Highly stiff f_2: implicit method
- Conservation? Positivity?

Splitting Methods

- Strategy 1: operator splitting
  - $S_1^{\Delta t}$ and $S_2^{\Delta t}$ are solution operators for

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f_1(y) \qquad \frac{\mathrm{d}y}{\mathrm{d}t} = f_2(y), \tag{1}$$

    respectively
  - Attempt to approximate $S^{\Delta t}$ with $S_1^{\Delta t}$ and $S_2^{\Delta t}$:

$$S^{\Delta t} \approx p(S_1^{\Delta t}, S_2^{\Delta t})$$

Splitting Methods

- Strategy 1: operator splitting
  - $S_1^{\Delta t}$ and $S_2^{\Delta t}$ are solution operators for

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f_1(y) \qquad \frac{\mathrm{d}y}{\mathrm{d}t} = f_2(y), \tag{1}$$

    respectively
  - Attempt to approximate $S^{\Delta t}$ with $S_1^{\Delta t}$ and $S_2^{\Delta t}$:

$$S^{\Delta t} \approx p(S_1^{\Delta t}, S_2^{\Delta t})$$

- Strategy 2: Runge-Kutta split methods
  - Create two separate Runge-Kutta methods for each of the ODE's in (1), but evolve them together
  - High-order accuracy requires coupling conditions between the two methods
  - Disadvantage: requires semi-discrete form of PDE

Operator Splitting (1/2)

- First guess: sequential application

$$S^{\Delta t} \approx S_1^{\Delta t} S_2^{\Delta t}$$

  - First-order in time
  - Very simple to formulate and elementary to implement

Operator Splitting (1/2)

- First guess: sequential application

$$S^{\Delta t} \approx S_1^{\Delta t} S_2^{\Delta t}$$

  - First-order in time
  - Very simple to formulate and elementary to implement
- "Strang splitting"

$$S^{\Delta t} \approx S_1^{\Delta t/2} S_2^{\Delta t} S_1^{\Delta t/2}$$

  - Second-order in time
  - Not much more difficult than first-order splitting
  - Probably the most popular operator splitting algorithm used today

Operator Splitting (2/2)

- Higher-order accuracy?
  - Methods exist, but suffer from stability problems
  - Dia has developed 3rd, 4th order schemes

Operator Splitting (2/2)

- Higher-order accuracy?
  - Methods exist, but suffer from stability problems
  - Dia has developed 3rd, 4th order schemes
- Conservation?
  - Ohwada: second-order scheme

$$S^{\Delta t} \approx \frac{1}{2}\left(I + S_2^{\Delta t} S_2^{\Delta t}\right) S_1^{\Delta t}$$

  - Preserves mass, momemtum, energy (assuming the discrete collision operator does as well)
  - Slightly more work compared to Strang splitting

Runge-Kutta methods

- "Usual" s-stage Runge-Kutta methods have the form

$$y^{n+1} = y^n + \Delta t \sum_{j=1}^{s} b_j f(t^n + c_j \Delta t, Y_j)$$

$$Y_i = y^n + \Delta t \sum_{j=1}^{s} a_{ij} f(t^n + c_j \Delta t, Y_j)$$

where a_{ij} is an $s \times s$ matrix and b_j and c_j are $s \times 1$ matrices

- Can define explicit and implicit RK methods in this way

More Runge-Kutta

- Implicit-explicit (IMEX) RK idea: use different RK methods for each different term:

$$
\begin{aligned}
y^{n+1} =& y^n + \Delta t \sum_{j=1}^{s} \tilde{b}_j f_1(t^n + \tilde{c}_j \Delta t, Y_j) + \\
& \Delta t \sum_{j=1}^{s} b_j f_2(t^n + c_j \Delta t, Y_j), \\
Y_i =& y^n + \Delta t \sum_{j=1}^{s} \tilde{a}_{ij} f_1(t^n + \tilde{c}_j \Delta t, Y_j) + \\
& \Delta t \sum_{j=1}^{s} a_{ij} f_2(t^n + c_j \Delta t, Y_j)
\end{aligned}
$$

IMEX methods

- What kind of RK methods to use for each operator?
  - Use an implicit RK method for the collision (f_2)
  - use an explicit RK method for the transport (f_1)

IMEX methods

- What kind of RK methods to use for each operator?
  - Use an implicit RK method for the collision (f_2)
  - use an explicit RK method for the transport (f_1)
- Some examples of IMEX methods
  - First-order Euler explicit-implicit
  - Second-order methods: 'midpoint' splitting, Strang splitting, Jin splitting, ...
  - Many higher-order methods

IMEX methods

- What kind of RK methods to use for each operator?
  - Use an implicit RK method for the collision (f_2)
  - use an explicit RK method for the transport (f_1)
- Some examples of IMEX methods
  - First-order Euler explicit-implicit
  - Second-order methods: 'midpoint' splitting, Strang splitting, Jin splitting, ...
  - Many higher-order methods
- IMEX methods for the Boltzmann equation
  - Require semi-discrete form of the PDE
  - Desire positivity
  - TVD/SSP property 'in the hyperbolic limit' would be nice....

IMEX methods

- What kind of RK methods to use for each operator?
  - Use an implicit RK method for the collision (f_2)
  - use an explicit RK method for the transport (f_1)
- Some examples of IMEX methods
  - First-order Euler explicit-implicit
  - Second-order methods: ‘midpoint’ splitting, Strang splitting, Jin splitting, ...
  - Many higher-order methods
- IMEX methods for the Boltzmann equation
  - Require semi-discrete form of the PDE
  - Desire positivity
  - TVD/SSP property ‘in the hyperbolic limit’ would be nice....
- Can’t have your cake and eat it, too

What do people actually do?

- Mostly operator splitting (Strang)

What do people actually do?

- Mostly operator splitting (Strang)
- Splitting is, of course, not the only answer
- Could use just about any ODE discretization you want

Questions?

?

Outline

1. Introduction
   - The Equation
   - Collisions
2. Time-Step Splitting
   - Operator splitting
   - Runge-Kutta
3. Transport
   - Flux-Balance Methods
   - Semi-Lagrangian Methods
4. Collision Operator
   - Discretization
   - Bobylev-Rjasanow's Integral Transform Method
   - Pareschi-Russo's Spectral Method
   - Mouhot-Pareschi's Sub-N^6 Method

The Vlasov Equation

- Recall that the transport part of the Boltzmann equation is the Vlasov equation:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = 0.$$

  - Nonlinear: F depends on f
  - Conservation law: v is x-independent and F is v-independent

The Vlasov Equation

- Recall that the transport part of the Boltzmann equation is the Vlasov equation:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = 0.$$

  - Nonlinear: F depends on f
  - Conservation law: v is x-independent and F is v-independent
- Many methods exist to solve these types of equations
- What we'll talk about: methods designed specifically for the Vlasov/Boltzmann equation

Argh, More Splitting?

- Complexity: six-dimensional transport equation → two 3-dimensional systems is more tractable

$$\frac{\partial f}{\partial t} = v \cdot \nabla_x f \qquad \frac{\partial f}{\partial t} = F \cdot \nabla_v f$$

Argh, More Splitting?

- Complexity: six-dimensional transport equation → two 3-dimensional systems is more tractable

$$\frac{\partial f}{\partial t} = v \cdot \nabla_x f \qquad \frac{\partial f}{\partial t} = F \cdot \nabla_v f$$

- Go even farther: split three dimensional problems into three 1-dimensional problems?
- Consider simple 1D conservation law:

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

Flux-Balance Methods

- Finite-volume degrees of freedom:

$$f_i^n = \int_{I_i} f(x,t)\mathrm{d}x$$

Flux-Balance Methods

- Finite-volume degrees of freedom:

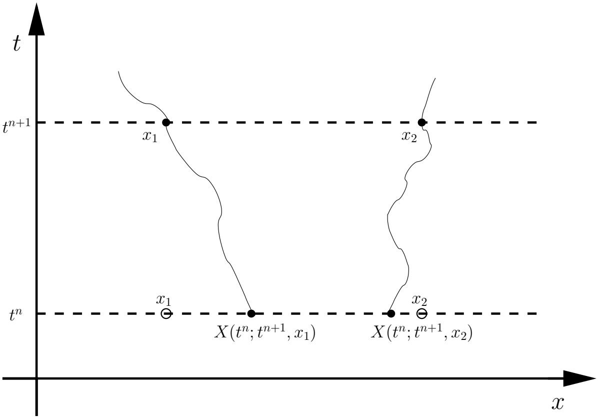
$$f_i^n = \int_{I_i} f(x,t)\mathrm{d}x$$

- Method of characteristics: f is constant along $x = X(t)$, where $X(t)$ solves the ODE

$$\frac{\mathrm{d}X(s)}{\mathrm{d}x} = v(s, X(s))$$
$$X(t^{n+1}) = x.$$

- $f(x, t^{n+1}) = f(X(t^n; t^{n+1}, x), t^n)$

Method of Characteristics



Flux-Balance Methods

- Use characteristics for finite-volume:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} f(t^{n+1}, x)\mathrm{d}x = \int_{X(t^n;t^{n+1},x_{i-1/2})}^{X(t^n,t^{n+1},x_{i+1/2})} f(t^n, x)\mathrm{d}x$$

Flux-Balance Methods

- Use characteristics for finite-volume:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} f(t^{n+1}, x)\mathrm{d}x = \int_{X(t^n;t^{n+1},x_{i-1/2})}^{X(t^n,t^{n+1},x_{i+1/2})} f(t^n, x)\mathrm{d}x$$

- Define flux terms:

$$\Phi_{i+1/2}(t^n) = \int_{x_{i+1/2}-v\Delta t}^{x_{i+1/2}} f(t^n, x)\mathrm{d}x$$

Flux-Balance Methods

- Use characteristics for finite-volume:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} f(t^{n+1},x)\mathrm{d}x = \int_{X(t^n;t^{n+1},x_{i-1/2})}^{X(t^n,t^{n+1},x_{i+1/2})} f(t^n,x)\mathrm{d}x$$

- Define flux terms:

$$\Phi_{i+1/2}(t^n) = \int_{x_{i+1/2}-v\Delta t}^{x_{i+1/2}} f(t^n,x)\mathrm{d}x$$

- Rewrite first equation as

$$\int_{x_{i-1/2}}^{x_{i+1/2}} f(t^{n+1},x)\mathrm{d}x = \int_{x_{i-1/2}}^{x_{i+1/2}} f(t^n,x)\mathrm{d}x + \frac{\Phi_{i-1/2}(t^n) - \Phi_{i+1/2}(t^n)}{\Delta x}$$

Flux-Balance Scheme

- Leads to the finite-volume fully-discrete scheme

$$f_i^{n+1} = f_i^n + \frac{\Phi_{i-1/2}(t^n) - \Phi_{i+1/2}(t^n)}{\Delta x}$$

Flux-Balance Scheme

- Leads to the finite-volume fully-discrete scheme

$$f_i^{n+1} = f_i^n + \frac{\Phi_{i-1/2}(t^n) - \Phi_{i+1/2}(t^n)}{\Delta x}$$

- How to evolve the scheme? Need to compute fluxes Φ.
  - No time-step restriction!
  - Explicitly compute integral definition (reconstruct f)
  - Compute pointwise primitive values (reconstruct primitive, F)

Flux-Balance – reconstructing f: PFC

- The ‘Positive and Flux-Conservative’ (PFC) method
- Third-order reconstruction using nonlinear slope limiters
- f is reconstructed as a quadratic polynomial

$$f_h(x) = f_i + \varepsilon^+ p_2^+(x)(f_{i+1} - f_i) + \varepsilon^- p_2^-(x)(f_i - f_{i-1})$$

- Preserves positivity of f
- Reconstruction preserves cell averages (mass)

Flux-Balance – reconstructing F: WENO

- ‘PWENO’: Pointwise WENO
  - Same principle as WENO
  - Reconstruction requires the primitive values inside cells (hence the ‘pointwise’ appellation)
  - More expensive than WENO: weights must be tailored depending on location $\rightarrow$ requires linear solve
  - Inherits spurious oscillation-reducing property of non-oscillatory schemes
  - Cannot prove positivity/conservation

Flux-Balance – reconstructing F: WENO

- ‘PWENO’: Pointwise WENO
  - Same principle as WENO
  - Reconstruction requires the primitive values inside cells (hence the ‘pointwise’ appellation)
  - More expensive than WENO: weights must be tailored depending on location $\rightarrow$ requires linear solve
  - Inherits spurious oscillation-reducing property of non-oscillatory schemes
  - Cannot prove positivity/conservation
- Use PWENO to reconstruct $F(x_{j+1/2} - v\Delta t)$

The Semi-Lagrangian Method

- Very similar in spirit to flux-balance method: follow the characteristics
- Degrees of freedom: point-values $f^n(x_i)$

The Semi-Lagrangian Method

- Very similar in spirit to flux-balance method: follow the characteristics
- Degrees of freedom: point-values $f^n(x_i)$
- Recall:

$$\frac{\mathrm{d}X(s)}{\mathrm{d}x} = v(s, X(s))$$
$$X(t^{n+1}) = x$$
$$f(x, t^{n+1}) = f(X(t^n; t^{n+1}, x), t^n)$$

The Semi-Lagrangian Method

- Very similar in spirit to flux-balance method: follow the characteristics
- Degrees of freedom: point-values $f^n(x_i)$
- Recall:

$$\frac{\mathrm{d}X(s)}{\mathrm{d}x} = v(s, X(s))$$
$$X(t^{n+1}) = x$$
$$f(x, t^{n+1}) = f(X(t^n; t^{n+1}, x), t^n)$$

- The semi-Lagrangian method:

$$f^{n+1}(x_i) = f^n(X(t^n; t^{n+1}, x_i)$$

- No time-step restriction!

Semi-Lagrangian Schemes

- We have $f^n(x_i)$ for all i

Semi-Lagrangian Schemes

- We have $f^n(x_i)$ for all i
- Need to interpolate to $X(t^n; t^{n+1}, x_i) = x_i - v\Delta t$ to advance the scheme
  - Lagrange interpolations
    - Implementable to high-order interpolants
    - Conservation: only with a central stencil, linear constant-coefficient advection PDE

Semi-Lagrangian Schemes

- We have $f^n(x_i)$ for all i
- Need to interpolate to $X(t^n; t^{n+1}, x_i) = x_i - v\Delta t$ to advance the scheme
  - Lagrange interpolations
    - Implementable to high-order interpolants
    - Conservation: only with a central stencil, linear constant-coefficient advection PDE
  - Hermite interpolations
    - Match function values and cell interface derivative values
    - Conservation: same as Lagrange interpolants

Semi-Lagrangian Schemes

- We have $f^n(x_i)$ for all i
- Need to interpolate to $X(t^n; t^{n+1}, x_i) = x_i - v\Delta t$ to advance the scheme
  - Lagrange interpolations
    - Implementable to high-order interpolants
    - Conservation: only with a central stencil, linear constant-coefficient advection PDE
  - Hermite interpolations
    - Match function values and cell interface derivative values
    - Conservation: same as Lagrange interpolants
  - PWENO interpolations

Other Solvers

- Spectral methods – computational savings: FFT

Other Solvers

- Spectral methods – computational savings: FFT
- Finite-difference methods – Conservation: mass, energy

Other Solvers

- Spectral methods – computational savings: FFT
- Finite-difference methods – Conservation: mass, energy
- DG methods – positivity

Other Solvers

- Spectral methods – computational savings: FFT
- Finite-difference methods – Conservation: mass, energy
- DG methods – positivity
- Can't have your cake and eat it, too

Questions?

?

Outline

The Collision Operator

$$\frac{\partial f}{\partial t} = \frac{1}{k_n} Q(f,f)$$

$$Q(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B \left[f(v')f(v'_*) - f(v)f(v_*) \right] \mathrm{d}\omega \mathrm{d}v_*$$

The Collision Operator

$$\frac{\partial f}{\partial t} = \frac{1}{k_n} Q(f,f)$$

$$Q(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B \left[f(v')f(v'_*) - f(v)f(v_*) \right] \mathrm{d}\omega \mathrm{d}v_*$$

Properties:

- Structure: Integrate over all post-collision velocities, then over all deflection vectors: 5 integrals at each (x, v).

The Collision Operator

$$\frac{\partial f}{\partial t} = \frac{1}{k_n} Q(f,f)$$

$$Q(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B \left[f(v') f(v'_*) - f(v) f(v_*) \right] \mathrm{d}\omega \mathrm{d}v_*$$

Properties:

- Structure: Integrate over all post-collision velocities, then over all deflection vectors: 5 integrals at each (x, v).
- Q is local in $x \rightarrow$ neglect space dependence in this section.

The Collision Operator

$$\frac{\partial f}{\partial t} = \frac{1}{k_n} Q(f,f)$$

$$Q(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B \left[f(v')f(v'_*) - f(v)f(v_*) \right] \mathrm{d}\omega \mathrm{d}v_*$$

Properties:

- Structure: Integrate over all post-collision velocities, then over all deflection vectors: 5 integrals at each (x, v).
- Q is local in $x \rightarrow$ neglect space dependence in this section.
- Conserves mass, momentum, energy:

$$\int_{\mathbb{R}^3} Q(f,f) \begin{cases} 1 \\ v \\ |v|^2 \end{cases} \mathrm{d}v = 0$$

The Collision Operator–Steady-State

Evolves towards ~~Gaussian in v~~ *Maxwellian*

$$f_\infty(v) := \frac{\rho}{(2\pi T)^{d/2}} \exp\left(-\frac{|V - v|^2}{2T}\right)$$

- Steady-state: $Q(f_\infty, f_\infty) = 0$

The Collision Operator–Steady-State

Evolves towards ~~Gaussian in v~~ *Maxwellian*

$$f_\infty(v) := \frac{\rho}{(2\pi T)^{d/2}} \exp\left(-\frac{|V-v|^2}{2T}\right)$$

- Steady-state: $Q(f_\infty, f_\infty) = 0$
- Does so even for *full* Boltzmann equation.

The Collision Operator–Steady-State

$$f_\infty(v) := \frac{\rho}{(2\pi T)^{d/2}} \exp\left(-\frac{|V-v|^2}{2T}\right).$$

- Density:

$$\rho := \int_{\mathbb{R}^d} f(v)\mathrm{d}v$$

The Collision Operator–Steady-State

$$f_\infty(v) := \frac{\rho}{(2\pi T)^{d/2}} \exp\left(-\frac{|V-v|^2}{2T}\right).$$

- Density:

$$\rho := \int_{\mathbb{R}^d} f(v)\mathrm{d}v$$

- Mean/Bulk Velocity:

$$V := \frac{1}{\rho}\int_{\mathbb{R}^d} v f(v)\mathrm{d}v$$

The Collision Operator–Steady-State

$$f_\infty(v) := \frac{\rho}{(2\pi T)^{d/2}} \exp\left(-\frac{|V-v|^2}{2T}\right).$$

- Density:

$$\rho := \int_{\mathbb{R}^d} f(v)\mathrm{d}v$$

- Mean/Bulk Velocity:

$$V := \frac{1}{\rho}\int_{\mathbb{R}^d} v f(v)\mathrm{d}v$$

- Temperature:

$$T := \frac{1}{3\rho}\int_{\mathbb{R}^d} |V-v|^2 f(v)\mathrm{d}v$$

Simplifying the Collision Operator

$$Q(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B \left[f(v')f(v'_*) - f(v)f(v_*) \right] \mathrm{d}\omega \mathrm{d}v_*$$

Simplifying the Collision Operator

$$Q(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B \left[f(v')f(v'_*) - f(v)f(v_*) \right] \mathrm{d}\omega \mathrm{d}v_*$$

Linearize: $f := f_\infty + g$

Simplifying the Collision Operator

$$Q(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B\left[f(v')f(v'_*) - f(v)f(v_*)\right] \mathrm{d}\omega \mathrm{d}v_*$$

Linearize: $f := f_\infty + g$

$$Q^{\mathrm{lin}}(f,f) \approx \int_{\mathbb{R}^3}\int_{S^2} B[g(v')f_\infty(v'_*) + f_\infty(v')g(v'_*) - g(v)f_\infty(v_*) - f_\infty(v)g(v_*)]\mathrm{d}\omega \mathrm{d}v_*$$

Simplifying the Collision Operator

$$Q(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B\left[f(v')f(v'_*) - f(v)f(v_*)\right] \mathrm{d}\omega \mathrm{d}v_*$$

Linearize: $f := f_\infty + g$

$$\begin{aligned} Q^{\text{lin}}(f,f) \approx \int_{\mathbb{R}^3}\int_{S^2} & B[g(v')f_\infty(v'_*) + f_\infty(v')g(v'_*) \\ & - g(v)f_\infty(v_*) - f_\infty(v)g(v_*)]\mathrm{d}\omega \mathrm{d}v_* \\ \rightsquigarrow \int_{\mathbb{R}^3} & k(v,v')f_\infty(v)f(v') - k(v',v)f_\infty(v')f(v)\mathrm{d}v' \end{aligned}$$

Simplifying the Collision Operator

$$Q(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B\left[f(v')f(v'_*) - f(v)f(v_*)\right] \mathrm{d}\omega \mathrm{d}v_*$$

Linearize: $f := f_\infty + g$

$$\begin{aligned} Q^{\text{lin}}(f,f) \approx \int_{\mathbb{R}^3}\int_{S^2} & B[g(v')f_\infty(v'_*) + f_\infty(v')g(v'_*) \\ & - g(v)f_\infty(v_*) - f_\infty(v)g(v_*)]\mathrm{d}\omega\mathrm{d}v_* \\ \rightsquigarrow \int_{\mathbb{R}^3} & k(v,v')f_\infty(v)f(v') - k(v',v)f_\infty(v')f(v)\mathrm{d}v' \end{aligned}$$

$\rightarrow$*Linear Vlasov-Boltzmann Equation*

Simplifying the Collision Operator

$$Q(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B \left[f(v')f(v'_*) - f(v)f(v_*) \right] \mathrm{d}\omega \mathrm{d}v_*$$

Even simpler: *Bhatnagar-Gross-Krook (BGK) Approximation*

Simplifying the Collision Operator

$$Q(f,f) = \int_{\mathbb{R}^3}\int_{S^2} B\left[f(v')f(v'_*) - f(v)f(v_*)\right] \mathrm{d}\omega \mathrm{d}v_*$$

Even simpler: *Bhatnagar-Gross-Krook (BGK) Approximation*

$$Q^{\mathrm{BGK}}(f) = \frac{1}{\tau}(f_\infty - f)$$

Simplifying the Collision Operator

$$Q(f,f) = \int_{\mathbb{R}^3} \int_{S^2} B \left[f(v')f(v'_*) - f(v)f(v_*) \right] \mathrm{d}\omega \mathrm{d}v_*$$

Even simpler: *Bhatnagar-Gross-Krook (BGK) Approximation*

$$Q^{\mathrm{BGK}}(f) = \frac{1}{\tau}(f_\infty - f)$$

Still stronger than (i.e. implies) Navier-Stokes.

Discretizing Velocity Space

v-space: infinite.

Discretizing Velocity Space

v-space: infinite. Computers: finite.

Discretizing Velocity Space

v-space: infinite. Computers: finite. →Something's got to go.

Discretizing Velocity Space

v-space: infinite. Computers: finite. →Something's got to go.

Lemma (Compact support for Q)

Suppose $\operatorname{supp}_v(f) \subset \mathcal{B}(0, R)$. *Then*

Discretizing Velocity Space

v-space: infinite. Computers: finite. →Something's got to go.

Lemma (Compact support for Q)

Suppose $\text{supp}_v(f) \subset \mathcal{B}(0, R)$. *Then*

1. $\text{supp}_v(Q(f, f)(v)) \subset \mathcal{B}(0, R\sqrt{2})$.

Discretizing Velocity Space

v-space: infinite. Computers: finite. →Something's got to go.

Lemma (Compact support for Q)

Suppose $\operatorname{supp}_v(f) \subset \mathcal{B}(0, R)$. *Then*

1. $\operatorname{supp}_v(Q(f,f)(v)) \subset \mathcal{B}(0, R\sqrt{2})$.
2. *Change of variables* $g := v - v_*$. *We can restrict* $g \in \mathcal{B}(0, 2R)$ *with no change:*

$$Q = \int_{\mathcal{B}(2R)\times S^2} B(|g|, \theta) \left[f'(v') f(v'_*) - f(v) f(v-g) \right] \mathrm{d}\omega \mathrm{d}g$$

Discretizing Velocity Space

v-space: infinite. Computers: finite. $\rightarrow$Something's got to go.

Lemma (Compact support for Q)

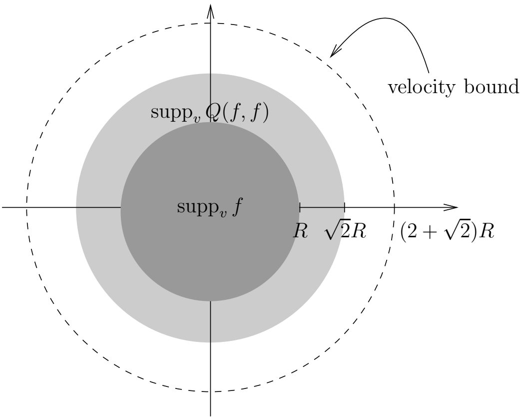
Suppose $\operatorname{supp}_v(f) \subset \mathcal{B}(0,R)$. *Then*

1. $\operatorname{supp}_v(Q(f,f)(v)) \subset \mathcal{B}(0, R\sqrt{2})$.
2. *Change of variables* $g := v - v_*$. *We can restrict* $g \in \mathcal{B}(0,2R)$ *with no change:*

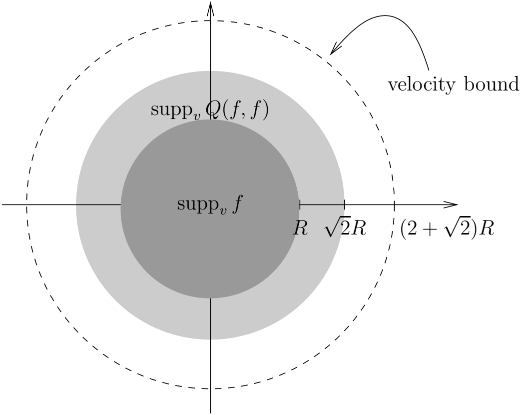
$$Q = \int_{\mathcal{B}(2R)\times S^2} B(|g|,\theta)\left[f'(v')f(v'_*) - f(v)f(v-g)\right] \mathrm{d}\omega \mathrm{d}g$$

3. *With this limitation, we automatically also have* $v_* = v - g, v', v'_* \in \mathcal{B}(0,(2+\sqrt{2})R)$.

Compact support for Q

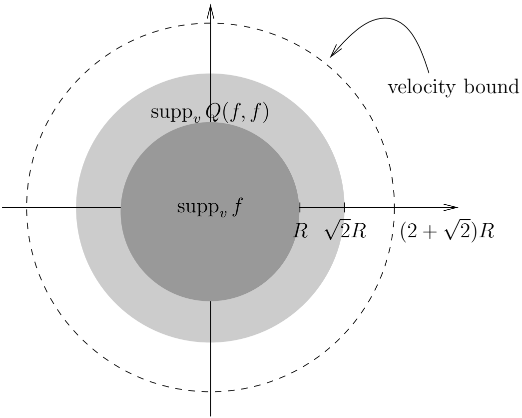


Compact support for Q



$\mathrm{supp}_v(f)$ grows by a factor of $\sqrt{2}$ each timestep

Compact support for Q



$\mathrm{supp}_v(f)$ grows by a factor of $\sqrt{2}$ each timestep—for any $\Delta t > 0$!

Fourier Transforms in v-Space: Why?

Despite negative result:

Fourier Transforms in v-Space: Why?

Despite negative result:
Keep fixed, bounded discretization in v-space.
Hope for the best.

Fourier Transforms in v-Space: Why?

Despite negative result:
Keep fixed, bounded discretization in v-space.
Hope for the best. (i.e. that f is small where we truncate)

Fourier Transforms in v-Space: Why?

Despite negative result:
Keep fixed, bounded discretization in v-space.
Hope for the best. (i.e. that f is small where we truncate)

Need to overcome N^8 complexity.

Fourier Transforms in v-Space: Why?

Despite negative result:
Keep fixed, bounded discretization in v-space.
Hope for the best. (i.e. that f is small where we truncate)

Need to overcome N^8 complexity.
Strategy: Look for convolution structure,

Fourier Transforms in v-Space: Why?

Despite negative result:
Keep fixed, bounded discretization in v-space.
Hope for the best. (i.e. that f is small where we truncate)

Need to overcome N^8 complexity.
Strategy: Look for convolution structure, carry out using FFT.

Fourier Transforms in v-Space: Why?

Despite negative result:
Keep fixed, bounded discretization in v-space.
Hope for the best. (i.e. that f is small where we truncate)

Need to overcome N^8 complexity.
Strategy: Look for convolution structure, carry out using FFT.

- Convolution carried out directly: $O(N^2)$

Fourier Transforms in v-Space: Why?

Despite negative result:
Keep fixed, bounded discretization in v-space.
Hope for the best. (i.e. that f is small where we truncate)

Need to overcome N^8 complexity.
Strategy: Look for convolution structure, carry out using FFT.

- Convolution carried out directly: $O(N^2)$
- Convolution carried out using FFT: $O(N \log N)$

Fourier Transforms in v-Space: Why?

Despite negative result:
Keep fixed, bounded discretization in v-space.
Hope for the best. (i.e. that f is small where we truncate)

Need to overcome N^8 complexity.
Strategy: Look for convolution structure, carry out using FFT.

- Convolution carried out directly: $O(N^2)$
- Convolution carried out using FFT: $O(N \log N)$

Hope: Similar savings!

Fourier Transforms in v-Space: Why?

Despite negative result:
Keep fixed, bounded discretization in v-space.
Hope for the best. (i.e. that f is small where we truncate)

Need to overcome N^8 complexity.
Strategy: Look for convolution structure, carry out using FFT.

- Convolution carried out directly: $O(N^2)$
- Convolution carried out using FFT: $O(N \log N)$

Hope: Similar savings!
$\rightarrow$Investigate Fourier Transforms in v-Space.

Fourier Transform in v-Space

Introduce a lattice:

$$v_k := h_v k, \quad h_v > 0, \quad k \in \mathbb{Z}^3$$

Fourier Transform in v-Space

Introduce a lattice:

$$v_k := h_v k, \quad h_v > 0, \quad k \in \mathbb{Z}^3$$

Approximate the Fourier Integral

$$\varphi(\xi) = \int_{\mathbb{R}^3} f(v) e^{i(v \cdot \xi)} \mathrm{d}v$$

Fourier Transform in v-Space

Introduce a lattice:

$$v_k := h_v k, \quad h_v > 0, \quad k \in \mathbb{Z}^3$$

Approximate the Fourier Integral

$$\varphi(\xi) = \int_{\mathbb{R}^3} f(v) e^{i(v \cdot \xi)} \mathrm{d}v$$

using the midpoint rule

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k \in \mathbb{Z}^3} f(v_k) e^{i(v \cdot \xi)}.$$

Fourier Transform in v-Space

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k \in \mathbb{Z}^3} f(v_k) e^{i(v \cdot \xi)}.$$

But for which ξ do we need to evaluate this?

Fourier Transform in v-Space

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k\in\mathbb{Z}^3} f(v_k)e^{i(v\cdot\xi)}.$$

But for which ξ do we need to evaluate this?
Pick a grid in ξ-space, too:

Fourier Transform in v-Space

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k \in \mathbb{Z}^3} f(v_k) e^{i(v \cdot \xi)}.$$

But for which ξ do we need to evaluate this?
Pick a grid in ξ-space, too: $\xi_j := h_\xi j$

Fourier Transform in v-Space

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k \in \mathbb{Z}^3} f(v_k) e^{i(v \cdot \xi)}.$$

But for which ξ do we need to evaluate this?

Pick a grid in ξ-space, too: $\xi_j := h_\xi j$ with $h_\xi > 0$ and $j \in \mathbb{Z}^3$.

Fourier Transform in v-Space

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i(v\cdot\xi)}.$$

But for which ξ do we need to evaluate this?
Pick a grid in ξ-space, too: $\xi_j := h_\xi j$ with $h_\xi > 0$ and $j \in \mathbb{Z}^3$.
FFT requires: $h_v h_\xi = 2\pi/n$ for $n \in \mathbb{N}$.

Fourier Transform in v-Space

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k \in \mathbb{Z}^3} f(v_k) e^{i(v \cdot \xi)}.$$

But for which ξ do we need to evaluate this?
Pick a grid in ξ-space, too: $\xi_j := h_\xi j$ with $h_\xi > 0$ and $j \in \mathbb{Z}^3$.
FFT requires: $h_v h_\xi = 2\pi/n$ for $n \in \mathbb{N}$.
Get n-periodicity in k and j:

Fourier Transform in v-Space

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i(v\cdot\xi)}.$$

But for which ξ do we need to evaluate this?
Pick a grid in ξ-space, too: $\xi_j := h_\xi j$ with $h_\xi > 0$ and $j \in \mathbb{Z}^3$.
FFT requires: $h_v h_\xi = 2\pi/n$ for $n \in \mathbb{N}$.
Get n-periodicity in k and j:

$$\tilde{\varphi}(\xi_j) = \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i\frac{2\pi}{n}(k\cdot j)}$$

Fourier Transform in v-Space

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i(v\cdot\xi)}.$$

But for which ξ do we need to evaluate this?
Pick a grid in ξ-space, too: $\xi_j := h_\xi j$ with $h_\xi > 0$ and $j \in \mathbb{Z}^3$.
FFT requires: $h_v h_\xi = 2\pi/n$ for $n \in \mathbb{N}$.
Get n-periodicity in k and j:

$$\tilde{\varphi}(\xi_j) = \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i\frac{2\pi}{n}(k\cdot j)}$$

$$= \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i\frac{2\pi}{n}((k+e_i n)\cdot j)}$$

Fourier Transform in v-Space

$$\tilde{\varphi}(\xi) := h_v^3 \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i(v\cdot\xi)}.$$

But for which ξ do we need to evaluate this?
Pick a grid in ξ-space, too: $\xi_j := h_\xi j$ with $h_\xi > 0$ and $j \in \mathbb{Z}^3$.
FFT requires: $h_v h_\xi = 2\pi/n$ for $n \in \mathbb{N}$.
Get n-periodicity in k and j:

$$\tilde{\varphi}(\xi_j) = \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i\frac{2\pi}{n}(k\cdot j)}$$

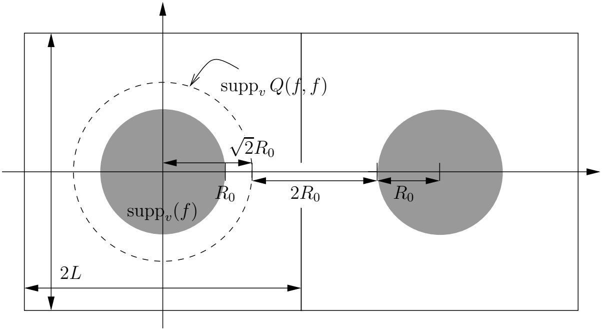
$$= \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i\frac{2\pi}{n}((k+e_i n)\cdot j)} = \sum_{k\in\mathbb{Z}^3} f(v_k) e^{i\frac{2\pi}{n}(k\cdot(j+e_i n))}$$

Faking Periodicity

So v-space becomes (artificially) $2L$-periodic, with $L = nh_v/2$.

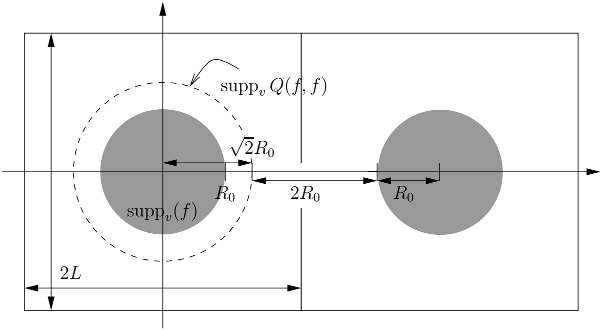
Faking Periodicity

So v-space becomes (artificially) $2L$-periodic, with $L = nh_v/2$.



Faking Periodicity

So v-space becomes (artificially) $2L$-periodic, with $L = nh_v/2$.



Avoid aliasing: choose $L \geqslant \frac{1}{2}(\sqrt{2}R_0 + 2R_0 + R_0) = \frac{3+\sqrt{2}}{2}R_0$.

Overview

	M	H	V	O	Order	Ops	Ms	Mm	En
DSMC	✓	✓	✓	✓	$N^{-1/2}$	N^3	✓	✓	✓
IbrR	✓	✓	✓	–	N^{-2}	N^6	✓	○	○
gen.	–	–	–	✓	N^{-2}	$N^6 \log N$	✓	○	○
BobR	–	3	–	–	N^{-2}	$N^6 \log N$	✓	○	○
ParRu	✓	✓	✓	–	$N^{-\infty}$	N^6	✓	○	○
MouP	2	3	–	–	$N^{-\infty}$	$N^5 \log N$	✓	○	○

Bobylev-Rjasanow's Integral Transform Method

Idea:

Rewrite $Q(f, f)$ into convolution integrals.
Exploit the Fast Fourier Transform to evaluate these integrals quickly.

skip the details

Bobylev-Rjasanow's Method

Lemma (Symmetric Representation)

The collision operator can be represented as

$$Q(f,f) = 4\int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \tilde{B}(y,z)\delta(z\cdot y)[f(v+z)(f(v+y) - f(v)f(v+y+z)]\mathrm{d}y\mathrm{d}z,$$

with

$$\tilde{B}(y,z) := \frac{1}{|y+z|}B\left(|y+z|, \frac{y\cdot(y+z)}{|y||y+z|}\right).$$

skip the rest of this method

Bobylev-Rjasanow's Method

Theorem (Integral Transform Representation)

The collision operator for hard spheres *can also be represented as*

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2,$$

(note the implicit v-dependency)

Bobylev-Rjasanow's Method

Theorem (Integral Transform Representation)

The collision operator for hard spheres *can also be represented as*

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2,$$

(note the implicit v-dependency) where

$$\Phi(v,\omega) := \int_{-\infty}^{\infty} |\rho| f(v+\rho\omega)\mathrm{d}\rho$$

Bobylev-Rjasanow's Method

Theorem (Integral Transform Representation)

The collision operator for hard spheres *can also be represented as*

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1\cdot\omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right]\mathrm{d}\omega_1\mathrm{d}\omega_2,$$

(note the implicit v-dependency) where

$$\Phi(v,\omega) := \int_{-\infty}^{\infty} |\rho| f(v+\rho\omega)\mathrm{d}\rho$$

$$\Psi(v,\omega_1,\omega_2) := \int_{-\infty}^{\infty}\int_{\infty}^{\infty} |\rho_1||\rho_2| f(v+\rho_1\omega_1+\rho_2\omega_2)\mathrm{d}\rho_1\mathrm{d}\rho_2.$$

Remarks on Bobylev-Rjasanow

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1\cdot\omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2,$$

- We've grown an extra integral. (but also a δ-function)

Remarks on Bobylev-Rjasanow

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2,$$

- We've grown an extra integral. (but also a δ-function)
- Integrating over all pairs of orthogonal unit vectors: Apply cleverness to exploit structure.

Remarks on Bobylev-Rjasanow

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\,[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)]\,\mathrm{d}\omega_1\mathrm{d}\omega_2,$$

- We've grown an extra integral. (but also a δ-function)
- Integrating over all pairs of orthogonal unit vectors: Apply cleverness to exploit structure.
- Φ: "*Generalized X-ray Transform*"

Remarks on Bobylev-Rjasanow

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2,$$

- We've grown an extra integral. (but also a δ-function)
- Integrating over all pairs of orthogonal unit vectors: Apply cleverness to exploit structure.
- Φ: "*Generalized X-ray Transform*"
- Ψ: "*Generalized Radon Transform*"

Remarks on Bobylev-Rjasanow

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2,$$

- We've grown an extra integral. (but also a δ-function)
- Integrating over all pairs of orthogonal unit vectors:
  Apply cleverness to exploit structure.
- Φ: "*Generalized X-ray Transform*"
- Ψ: "*Generalized Radon Transform*"
- Φ and Ψ of convolution type: good for FFT.

Rewriting X-Ray Transform→Fourier Transform

$$\Phi(v,\omega) = \mathcal{F}^{-1}[\mathcal{F}[\Phi](\xi,\omega)](v,\omega)$$

Rewriting X-Ray Transform→Fourier Transform

$$\begin{aligned}\Phi(v,\omega) &= \mathcal{F}^{-1}[\mathcal{F}[\Phi](\xi,\omega)](v,\omega)\\ &= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3}\Phi(v,\omega)e^{i(v\cdot\xi)}\mathrm{d}v\right](v,\omega)\end{aligned}$$

Rewriting X-Ray Transform→Fourier Transform

$$\begin{aligned}
\Phi(v,\omega) &= \mathcal{F}^{-1}[\mathcal{F}[\Phi](\xi,\omega)](v,\omega) \\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3} \Phi(v,\omega) e^{i(v\cdot\xi)} \mathrm{d}v\right](v,\omega) \\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3} \int_{-\infty}^{\infty} |\rho| f(v+\rho\omega) \mathrm{d}\rho e^{i(v\cdot\xi)} \mathrm{d}v\right](v,\omega)
\end{aligned}$$

Rewriting X-Ray Transform→Fourier Transform

$$
\begin{aligned}
\Phi(v,\omega) &= \mathcal{F}^{-1}[\mathcal{F}[\Phi](\xi,\omega)](v,\omega) \\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3} \Phi(v,\omega) e^{i(v\cdot\xi)} \mathrm{d}v\right](v,\omega) \\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3} \int_{-\infty}^{\infty} |\rho| f(v+\rho\omega) \mathrm{d}\rho e^{i(v\cdot\xi)} \mathrm{d}v\right](v,\omega) \\
(\tilde{v} := v+\rho\omega) &= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3} \int_{-\infty}^{\infty} |\rho| f(\tilde{v}) e^{i((\tilde{v}-\rho\omega)\cdot\xi)} \mathrm{d}\rho \mathrm{d}v\right](v,\omega)
\end{aligned}
$$

Rewriting X-Ray Transform→Fourier Transform

$$\begin{aligned}
\Phi(v,\omega) &= \mathcal{F}^{-1}[\mathcal{F}[\Phi](\xi,\omega)](v,\omega) \\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3}\Phi(v,\omega)e^{i(v\cdot\xi)}\mathrm{d}v\right](v,\omega) \\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3}\int_{-\infty}^{\infty}|\rho|f(v+\rho\omega)\mathrm{d}\rho e^{i(v\cdot\xi)}\mathrm{d}v\right](v,\omega) \\
(\tilde{v} := v+\rho\omega) \quad &= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3}\int_{-\infty}^{\infty}|\rho|f(\tilde{v})e^{i((\tilde{v}-\rho\omega)\cdot\xi)}\mathrm{d}\rho\mathrm{d}v\right](v,\omega) \\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3}f(\tilde{v})e^{i(\tilde{v}\cdot\xi)}\mathrm{d}v\underbrace{\int_{-\infty}^{\infty}|\rho|e^{-i(\rho\omega\cdot\xi)}\mathrm{d}\rho}_{d(\omega\cdot\xi):=\mathcal{F}[|\rho|](\omega\cdot\xi):=}\right](v,\omega)
\end{aligned}$$

Rewriting X-Ray Transform→Fourier Transform

$$\begin{aligned}
\Phi(v,\omega) &= \mathcal{F}^{-1}[\mathcal{F}[\Phi](\xi,\omega)](v,\omega)\\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3}\Phi(v,\omega)e^{i(v\cdot\xi)}\mathrm{d}v\right](v,\omega)\\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3}\int_{-\infty}^{\infty}|\rho|f(v+\rho\omega)\mathrm{d}\rho e^{i(v\cdot\xi)}\mathrm{d}v\right](v,\omega)\\
(\tilde{v}:=v+\rho\omega) \quad &= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3}\int_{-\infty}^{\infty}|\rho|f(\tilde{v})e^{i((\tilde{v}-\rho\omega)\cdot\xi)}\mathrm{d}\rho\mathrm{d}v\right](v,\omega)\\
&= \mathcal{F}^{-1}\left[\int_{\mathbb{R}^3}f(\tilde{v})e^{i(\tilde{v}\cdot\xi)}\mathrm{d}v\underbrace{\int_{-\infty}^{\infty}|\rho|e^{-i(\rho\omega\cdot\xi)}\mathrm{d}\rho}_{d(\omega\cdot\xi):=\mathcal{F}[|\rho|](\omega\cdot\xi):=}\right](v,\omega)\\
&= \mathcal{F}^{-1}\left[\varphi(\xi)d(\xi\cdot\omega)\right](v,\omega)
\end{aligned}$$

The Algorithm

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2$$

$$\Phi(v,\omega) = \mathcal{F}^{-1}\left[\varphi(\xi) d(\xi \cdot \omega)\right](v,\omega)]$$

$$\Psi(v,\omega) = \mathcal{F}^{-1}[\varphi(\xi) d(\xi \cdot \omega_1) d(\xi \cdot \omega_2)](v,\omega_1,\omega_2)]$$

1. Fourier transform $f \to \varphi$

The Algorithm

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2$$
$$\Phi(v,\omega) = \mathcal{F}^{-1}\left[\varphi(\xi)d(\xi\cdot\omega)\right](v,\omega)]$$
$$\Psi(v,\omega) = \mathcal{F}^{-1}[\varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)](v,\omega_1,\omega_2)]$$

1. Fourier transform $f \to \varphi$
2. For each node ω_1 of a quadrature rule on S^2:

The Algorithm

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1\cdot\omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right]\mathrm{d}\omega_1\mathrm{d}\omega_2$$

$$\Phi(v,\omega) = \mathcal{F}^{-1}\left[\varphi(\xi)d(\xi\cdot\omega)\right](v,\omega)]$$

$$\Psi(v,\omega) = \mathcal{F}^{-1}[\varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)](v,\omega_1,\omega_2)]$$

1. Fourier transform $f \to \varphi$
2. For each node ω_1 of a quadrature rule on S^2:
    1. For each node ω_2 of a quadrature rule on $\omega_1^\perp$:

The Algorithm

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2$$

$$\Phi(v,\omega) = \mathcal{F}^{-1}\left[\varphi(\xi)d(\xi\cdot\omega)\right](v,\omega)]$$

$$\Psi(v,\omega) = \mathcal{F}^{-1}[\varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)](v,\omega_1,\omega_2)]$$

1. Fourier transform $f \to \varphi$
2. For each node ω_1 of a quadrature rule on S^2:
    1. For each node ω_2 of a quadrature rule on $\omega_1^\perp$:
        1. Compute $F_\Phi(\xi) := \varphi(\xi)d(\xi\cdot\omega)$.

The Algorithm

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1\cdot\omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right]\mathrm{d}\omega_1\mathrm{d}\omega_2$$

$$\Phi(v,\omega) = \mathcal{F}^{-1}\left[\varphi(\xi)d(\xi\cdot\omega)\right](v,\omega)]$$

$$\Psi(v,\omega) = \mathcal{F}^{-1}[\varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)](v,\omega_1,\omega_2)]$$

1. Fourier transform $f \to \varphi$
2. For each node ω_1 of a quadrature rule on S^2:
    1. For each node ω_2 of a quadrature rule on $\omega_1^\perp$:
        1. Compute $F_\Phi(\xi) := \varphi(\xi)d(\xi\cdot\omega)$.
        2. Compute $F_\Psi(\xi) := \varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)$.

The Algorithm

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1\cdot\omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right]\mathrm{d}\omega_1\mathrm{d}\omega_2$$

$$\Phi(v,\omega) = \mathcal{F}^{-1}\left[\varphi(\xi)d(\xi\cdot\omega)\right](v,\omega)]$$

$$\Psi(v,\omega) = \mathcal{F}^{-1}[\varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)](v,\omega_1,\omega_2)]$$

1. Fourier transform $f \to \varphi$
2. For each node ω_1 of a quadrature rule on S^2:
    1. For each node ω_2 of a quadrature rule on $\omega_1^\perp$:
        1. Compute $F_\Phi(\xi) := \varphi(\xi)d(\xi\cdot\omega)$.
        2. Compute $F_\Psi(\xi) := \varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)$.
        3. Compute the inverse Fourier transform of F_Φ and F_Ψ.

The Algorithm

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1 \cdot \omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right] \mathrm{d}\omega_1 \mathrm{d}\omega_2$$

$$\Phi(v,\omega) = \mathcal{F}^{-1}\left[\varphi(\xi)d(\xi\cdot\omega)\right](v,\omega)]$$

$$\Psi(v,\omega) = \mathcal{F}^{-1}[\varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)](v,\omega_1,\omega_2)]$$

1. Fourier transform $f \to \varphi$
2. For each node ω_1 of a quadrature rule on S^2:
    1. For each node ω_2 of a quadrature rule on $\omega_1^\perp$:
        1. Compute $F_\Phi(\xi) := \varphi(\xi)d(\xi\cdot\omega)$.
        2. Compute $F_\Psi(\xi) := \varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)$.
        3. Compute the inverse Fourier transform of F_Φ and F_Ψ.
3. Add up contributions, giving $Q(f,f)$ for all v at once.

The Algorithm

$$Q(f,f) = \int_{S^2}\int_{S^2} \delta(\omega_1\cdot\omega_2)\left[\Phi(\omega_1)\Phi(\omega_2) - f\Psi(\omega_1,\omega_2)\right]\mathrm{d}\omega_1\mathrm{d}\omega_2$$
$$\Phi(v,\omega) = \mathcal{F}^{-1}\left[\varphi(\xi)d(\xi\cdot\omega)\right](v,\omega)]$$
$$\Psi(v,\omega) = \mathcal{F}^{-1}[\varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)](v,\omega_1,\omega_2)]$$

1. Fourier transform $f \to \varphi$
2. For each node ω_1 of a quadrature rule on S^2:
    1. For each node ω_2 of a quadrature rule on $\omega_1^\perp$:
        1. Compute $F_\Phi(\xi) := \varphi(\xi)d(\xi\cdot\omega)$.
        2. Compute $F_\Psi(\xi) := \varphi(\xi)d(\xi\cdot\omega_1)d(\xi\cdot\omega_2)$.
        3. Compute the inverse Fourier transform of F_Φ and F_Ψ.
3. Add up contributions, giving $Q(f,f)$ for all v at once.

Altogether: $N^6 \log N$.

Pareschi-Russo's Spectral Method

Idea:

Do Fourier-Galerkin on $\partial_t f = Q(f, f)$.
Demand that the residual of $\partial_t f_N = Q(f_N, f_N)$ is orthogonal to all trigonometric polynomials, where f_N is a Fourier expansion.

skip the details

Pareschi-Russo's Spectral Method

Expand

$$f_N(v) := \sum_{k \in K_N^3} \hat{f}_k e^{i(k \cdot v)}$$

Pareschi-Russo's Spectral Method

Expand

$$f_N(v) := \sum_{k \in K_N^3} \hat{f}_k e^{i(k \cdot v)}$$

with Fourier coefficients

$$\hat{f}_k = \frac{1}{(2\pi)^3} \int_{[-\pi,\pi]^3} f(v) e^{-i(k \cdot v)} \mathrm{d}v.$$

Pareschi-Russo's Spectral Method

Expand

$$f_N(v) := \sum_{k \in K_N^3} \hat{f}_k e^{i(k \cdot v)}$$

with Fourier coefficients

$$\hat{f}_k = \frac{1}{(2\pi)^3} \int_{[-\pi,\pi]^3} f(v) e^{-i(k \cdot v)} \mathrm{d}v.$$

Assume $\mathrm{supp}_v(f) \subset \mathcal{B}(0, R_0)$.

Pareschi-Russo's Spectral Method

Expand

$$f_N(v) := \sum_{k \in K_N^3} \hat{f}_k e^{i(k \cdot v)}$$

with Fourier coefficients

$$\hat{f}_k = \frac{1}{(2\pi)^3} \int_{[-\pi,\pi]^3} f(v) e^{-i(k \cdot v)} \mathrm{d}v.$$

Assume $\mathrm{supp}_v(f) \subset \mathcal{B}(0, R_0)$. Take $K_N := \{-N, \ldots, N\}$.

Pareschi-Russo's Spectral Method

Expand

$$f_N(v) := \sum_{k \in K_N^3} \hat{f}_k e^{i(k \cdot v)}$$

with Fourier coefficients

$$\hat{f}_k = \frac{1}{(2\pi)^3} \int_{[-\pi,\pi]^3} f(v) e^{-i(k \cdot v)} \mathrm{d}v.$$

Assume $\operatorname{supp}_v(f) \subset \mathcal{B}(0, R_0)$. Take $K_N := \{-N, \dots, N\}$.
As before, work on $[-L, L]^3$.

Pareschi-Russo's Spectral Method

Expand

$$f_N(v) := \sum_{k \in K_N^3} \hat{f}_k e^{i(k \cdot v)}$$

with Fourier coefficients

$$\hat{f}_k = \frac{1}{(2\pi)^3} \int_{[-\pi,\pi]^3} f(v) e^{-i(k \cdot v)} \mathrm{d}v.$$

Assume $\text{supp}_v(f) \subset \mathcal{B}(0, R_0)$. Take $K_N := \{-N, \ldots, N\}$.
As before, work on $[-L, L]^3$. WLOG $\pi = L$.

Pareschi-Russo's Spectral Method

Expand

$$f_N(v) := \sum_{k \in K_N^3} \hat{f}_k e^{i(k\cdot v)}$$

with Fourier coefficients

$$\hat{f}_k = \frac{1}{(2\pi)^3} \int_{[-\pi,\pi]^3} f(v) e^{-i(k\cdot v)} \mathrm{d}v.$$

Assume $\operatorname{supp}_v(f) \subset \mathcal{B}(0, R_0)$. Take $K_N := \{-N, \dots, N\}$.
As before, work on $[-L, L]^3$. WLOG $\pi = L$.

$$\text{Avoid aliasing:} \quad \pi = L \geqslant \frac{3+\sqrt{2}}{2} R_0.$$

Pareschi-Russo's Spectral Method

Residual of $\partial_t f_N = Q(f_N, f_N)$ is L^2-orthogonal to all trigonometric polynomials:

Pareschi-Russo's Spectral Method

Residual of $\partial_t f_N = Q(f_N, f_N)$ is L^2-orthogonal to all trigonometric polynomials:

$$\int_{[-\pi,\pi]^3} \left(\frac{\partial f_N}{\partial t} - Q(f_N, f_N) \right) e^{-ik\cdot v} \mathrm{d}v \stackrel{!}{=} 0 \qquad k \in K_N^3$$

Pareschi-Russo's Spectral Method

Residual of $\partial_t f_N = Q(f_N, f_N)$ is L^2-orthogonal to all trigonometric polynomials:

$$\int_{[-\pi,\pi]^3} \left(\frac{\partial f_N}{\partial t} - Q(f_N, f_N) \right) e^{-ik\cdot v} \mathrm{d}v \stackrel{!}{=} 0 \qquad k \in K_N^3$$

$\longrightarrow$ get the scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(k-m, m) - \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(m, m)$$

Pareschi-Russo's Spectral Method

Residual of $\partial_t f_N = Q(f_N, f_N)$ is L^2-orthogonal to all trigonometric polynomials:

$$\int_{[-\pi,\pi]^3} \left(\frac{\partial f_N}{\partial t} - Q(f_N, f_N) \right) e^{-ik\cdot v} \mathrm{d}v \overset{!}{=} 0 \qquad k \in K_N^3$$

$\longrightarrow$ get the scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(k-m, m) - \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(m, m)$$

with the *kernel modes*

$$\hat{B}(l, m) = \int_{\mathcal{B}(0,2R_0)} \int_{S^2} B(|g|, \cos\theta) e^{-ig\cdot\frac{(l+m)}{2} - i|g|\omega\cdot\frac{(m-l)}{2}} \mathrm{d}\omega \mathrm{d}g$$

Performance of the Pareschi-Russo Spectral Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(k-m, m) - \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(m, m)$$

- Evaluation of the scheme takes $O(N^6)$ operations.

Performance of the Pareschi-Russo Spectral Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(k-m, m) - \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(m, m)$$

- Evaluation of the scheme takes $O(N^6)$ operations.
  - Because we can pre-evaluate $\hat{B}$.

Performance of the Pareschi-Russo Spectral Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(k-m, m) - \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(m, m)$$

- Evaluation of the scheme takes $O(N^6)$ operations.
  - Because we can pre-evaluate $\hat{B}$.
  - Recall: naive method would take $O(N^6 N_a)$.

Performance of the Pareschi-Russo Spectral Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(k-m, m) - \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(m, m)$$

- Evaluation of the scheme takes $O(N^6)$ operations.
  - Because we can pre-evaluate $\hat{B}$.
  - Recall: naive method would take $O(N^6 N_a)$.
- Big plus: Rolled evaluation of problematic sphere integral into pre-computation.

Performance of the Pareschi-Russo Spectral Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(k-m, m) - \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{B}(m, m)$$

- Evaluation of the scheme takes $O(N^6)$ operations.
  - Because we can pre-evaluate $\hat{B}$.
  - Recall: naive method would take $O(N^6 N_a)$.
- Big plus: Rolled evaluation of problematic sphere integral into pre-computation.
- Loss term is sum of convolutions.
  $\rightarrow$ Can evaluate in $O(N^3 \log N)$.

Kernel modes of the Pareschi-Russo Spectral Scheme

$$\hat{B}(l, m) = \int_{\mathcal{B}(0,2R_0)} \int_{S^2} B(|g|, \cos\theta) e^{-ig\cdot\frac{(l+m)}{2} - i|g|\omega\cdot\frac{(m-l)}{2}} \mathrm{d}\omega \mathrm{d}g$$

- The pre-evaluated $\hat{B}$ takes up only limited storage.

Kernel modes of the Pareschi-Russo Spectral Scheme

$$\hat{B}(l,m) = \int_{\mathcal{B}(0,2R_0)} \int_{S^2} B(|g|, \cos\theta) e^{-ig\cdot\frac{(l+m)}{2} - i|g|\omega\cdot\frac{(m-l)}{2}} \mathrm{d}\omega \mathrm{d}g$$

- The pre-evaluated $\hat{B}$ takes up only limited storage.
  - In the general case, $\hat{B}$ depends on $|l-m|$, $|l+m|$, $(l-m)\cdot(l+m)$.

Kernel modes of the Pareschi-Russo Spectral Scheme

$$\hat{B}(l, m) = \int_{\mathcal{B}(0,2R_0)} \int_{S^2} B(|g|, \cos\theta) e^{-ig\cdot\frac{(l+m)}{2} - i|g|\omega\cdot\frac{(m-l)}{2}} \mathrm{d}\omega \mathrm{d}g$$

- The pre-evaluated $\hat{B}$ takes up only limited storage.
  - In the general case, $\hat{B}$ depends on $|l-m|$, $|l+m|$, $(l-m)\cdot(l+m)$.
  - For VHS, $\hat{B}$ depends only on $|l-m|$, $|l+m|$.
    $\rightarrow$ Two-dimensional array suffices.

Kernel modes of the Pareschi-Russo Spectral Scheme

$$\hat{B}(l,m) = \int_{\mathcal{B}(0,2R_0)} \int_{S^2} B(|g|, \cos\theta) e^{-ig\cdot\frac{(l+m)}{2} - i|g|\omega\cdot\frac{(m-l)}{2}} \mathrm{d}\omega \mathrm{d}g$$

- The pre-evaluated $\hat{B}$ takes up only limited storage.
  - In the general case, $\hat{B}$ depends on $|l-m|$, $|l+m|$, $(l-m)\cdot(l+m)$.
  - For VHS, $\hat{B}$ depends only on $|l-m|$, $|l+m|$.
    $\rightarrow$ Two-dimensional array suffices.
- A far more explicit expression for $\hat{B}$ is available.

Analysis of the Pareschi-Russo Spectral Scheme

Theorem (Spectral Accuracy)

Let $f \in L^2([-\pi,\pi]^3)$. Then

$$\left\| Q^{\mathrm{tr}}(f,f) - \mathcal{P}_N Q^{\mathrm{tr}}(f,f) \right\|_2 \leqslant C \left(\|f - f_N\|_2 + \frac{\|Q^{\mathrm{tr}}(f_N,f_N)\|_{H^r_{\mathrm{per}}}}{N^r} \right)$$

for all $r \geqslant 0$.

Analysis of the Pareschi-Russo Spectral Scheme

Theorem (Spectral Accuracy)

Let $f \in L^2([-\pi, \pi]^3)$. Then

$$\left\| Q^{\mathrm{tr}}(f, f) - \mathcal{P}_N Q^{\mathrm{tr}}(f, f) \right\|_2 \leqslant C \left(\|f - f_N\|_2 + \frac{\|Q^{\mathrm{tr}}(f_N, f_N)\|_{H^r_{\mathrm{per}}}}{N^r} \right)$$

for all $r \geqslant 0$.

Can show stability after including some filtering in the scheme.

Mouhot-Pareschi's Spectral Method

Idea:

Combine the symmetric representation from Bobylev-Rjasanow with the Fourier-Galerkin approach from Pareschi-Russo.

skip the details

Symmetric Representation

Recall

Lemma (Symmetric Representation)

The collision operator can be represented as

$$Q(f,f) = \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \tilde{B}(y,z)\delta(z\cdot y)[f(v+z)f(v+y) - f(v)f(v+y+z)]\mathrm{d}y\mathrm{d}z,$$

with

$$\tilde{B}(y,z) := \frac{2^{d-1}}{|y+z|^{d-2}} B\left(|y+z|, \frac{y\cdot(y+z)}{|y||y+z|}\right).$$

Symmetric Representation

Recall

Lemma (Symmetric Representation)

The collision operator can be represented as

$$Q(f,f) = \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \tilde{B}(y,z)\delta(z\cdot y)[f(v+z)f(v+y) - f(v)f(v+y+z)]\mathrm{d}y\mathrm{d}z,$$

with

$$\tilde{B}(y,z) := \frac{2^{d-1}}{|y+z|^{d-2}} B\left(|y+z|, \frac{y\cdot(y+z)}{|y||y+z|}\right).$$

(slight generalization for multi-d)

Truncating the Symmetric Representation

As before, assume $\text{supp}_v(f) \subset \mathcal{B}(0, R_0)$.

Truncating the Symmetric Representation

As before, assume $\text{supp}_v(f) \subset \mathcal{B}(0, R_0)$. Work on $[-L, L]^3$.

Truncating the Symmetric Representation

As before, assume $\text{supp}_v(f) \subset \mathcal{B}(0, R_0)$. Work on $[-L, L]^3$. WLOG $\pi = L$.

Truncating the Symmetric Representation

As before, assume $\mathrm{supp}_v(f) \subset \mathcal{B}(0, R_0)$. Work on $[-L, L]^3$.
WLOG $\pi = L$.

$$\text{Avoid aliasing:} \quad \pi = L \geqslant \frac{3+\sqrt{2}}{2} R_0.$$

Truncating the Symmetric Representation

As before, assume $\mathrm{supp}_v(f) \subset \mathcal{B}(0, R_0)$. Work on $[-L, L]^3$.
WLOG $\pi = L$.

$$\text{Avoid aliasing:} \quad \pi = L \geqslant \frac{3+\sqrt{2}}{2} R_0.$$

Then y and z may be limited to $\mathcal{B}(0, R)$, with $R := \sqrt{2} R_0$:

Truncating the Symmetric Representation

As before, assume $\mathrm{supp}_v(f) \subset \mathcal{B}(0, R_0)$. Work on $[-L, L]^3$. WLOG $\pi = L$.

$$\text{Avoid aliasing:} \quad \pi = L \geqslant \frac{3+\sqrt{2}}{2} R_0.$$

Then y and z may be limited to $\mathcal{B}(0, R)$, with $R := \sqrt{2} R_0$:

$$Q^R(f, f) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y, z) \delta(z \cdot y) [f(v+z) f(v+y) \\ - f(v) f(v+y+z)] \mathrm{d}y \mathrm{d}z$$

Fourier-Galerkin Derivation

▸ skip the details

$$0 \stackrel{!}{=} \int_{[-\pi,\pi]^d} \left(\frac{\partial f_N}{\partial t} - \left[Q^+(f_N, f_N) - L(f_N) f_N \right] \right) e^{-ik\cdot v} \mathrm{d}v$$

Fourier-Galerkin Derivation

skip the details

$$0 \stackrel{!}{=} \int_{[-\pi,\pi]^d} \left(\frac{\partial f_N}{\partial t} - \left[Q^+(f_N, f_N) - L(f_N) f_N \right] \right) e^{-ik\cdot v} \mathrm{d}v$$

$$= \int_{[-\pi,\pi]^d} \left(\sum_{k' \in K_N^d} \frac{\partial \hat{f}_{k'}}{\partial t} e^{i(k'\cdot v)} - Q^R(f_N, f_N) \right) e^{-ik\cdot v} \mathrm{d}v$$

Fourier-Galerkin Derivation

skip the details

$$0 = \int_{[-\pi,\pi]^d} \left(\sum_{k' \in K_N^d} \frac{\partial \hat{f}_{k'}}{\partial t} e^{i(k' \cdot v)} - Q^R(f_N, f_N) \right) e^{-ik\cdot v} \mathrm{d}v$$

Fourier-Galerkin Derivation

skip the details

$$0 = \int_{[-\pi,\pi]^d} \left(\sum_{k' \in K_N^d} \frac{\partial \hat{f}_{k'}}{\partial t} e^{i(k' \cdot v)} - Q^R(f_N, f_N) \right) e^{-ik\cdot v} \mathrm{d}v$$

$$= \frac{\partial \hat{f}_k}{\partial t} - \int_{[-\pi,\pi]^d} \left[Q^R(f_N, f_N) \right] e^{-ik\cdot v} \mathrm{d}v$$

Fourier-Galerkin Derivation

▸ skip the details

$$0 = \frac{\partial \hat{f}_k}{\partial t} - \int_{[-\pi,\pi]^d} \left[Q^R(f_N, f_N) \right] e^{-ik\cdot v} \mathrm{d}v$$

Fourier-Galerkin Derivation

skip the details

$$
\begin{aligned}
0 =& \frac{\partial \hat{f}_k}{\partial t} - \int_{[-\pi,\pi]^d} \left[Q^R(f_N, f_N) \right] e^{-ik\cdot v} \mathrm{d}v \\
=& \frac{\partial \hat{f}_k}{\partial t} - \int_{[-\pi,\pi]^d} \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y)[f_N(v+z)f_N(v+y) \\
&- f_N(v)f_N(v+y+z)]e^{-ik\cdot v} \mathrm{d}y\mathrm{d}z\mathrm{d}v
\end{aligned}
$$

Fourier-Galerkin Derivation

skip the details

$$0 = \frac{\partial \hat{f}_k}{\partial t} - \int_{[-\pi,\pi]^d} \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z \cdot y)[f_N(v+z)f_N(v+y) - f_N(v)f_N(v+y+z)]e^{-ik\cdot v}\mathrm{d}y\mathrm{d}z\mathrm{d}v$$

Fourier-Galerkin Derivation

skip the details

$$
\begin{aligned}
0 = & \frac{\partial \hat{f}_k}{\partial t} - \int_{[-\pi,\pi]^d} \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y)[f_N(v+z)f_N(v+y) \\
& - f_N(v)f_N(v+y+z)]e^{-ik\cdot v}\mathrm{d}y\mathrm{d}z\mathrm{d}v \\
= & \frac{\partial \hat{f}_k}{\partial t} - \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y) \int_{[-\pi,\pi]^d} \\
& \sum_{l,m\in K_N^3} \hat{f}_l \hat{f}_m e^{i[l\cdot(v+z)+m\cdot(v+y)-k\cdot v]} \\
& - \sum_{l,m\in K_N^3} \hat{f}_l \hat{f}_m e^{i[l\cdot v+m\cdot(v+y+z)-k\cdot v]}\mathrm{d}v\mathrm{d}y\mathrm{d}z
\end{aligned}
$$

Fourier-Galerkin Derivation

skip the details

$$0 = \frac{\partial \hat{f}_k}{\partial t} - \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z \cdot y) \int_{[-\pi,\pi]^d} \sum_{l,m \in K_N^3} \hat{f}_l \hat{f}_m e^{i[l\cdot(v+z)+m\cdot(v+y)-k\cdot v]} - \sum_{l,m \in K_N^3} \hat{f}_l \hat{f}_m e^{i[l\cdot v+m\cdot(v+y+z)-k\cdot v]} \mathrm{d}v\mathrm{d}y\mathrm{d}z$$

Fourier-Galerkin Derivation

▸ skip the details

$$
\begin{aligned}
0 =& \frac{\partial \hat{f}_k}{\partial t} - \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y) \int_{[-\pi,\pi]^d} \\
& \sum_{l,m\in K_N^3} \hat{f}_l \hat{f}_m e^{i[l\cdot(v+z)+m\cdot(v+y)-k\cdot v]} \\
& \quad - \sum_{l,m\in K_N^3} \hat{f}_l \hat{f}_m e^{i[l\cdot v+m\cdot(v+y+z)-k\cdot v]} \mathrm{d}v\mathrm{d}y\mathrm{d}z \\
=& \frac{\partial \hat{f}_k}{\partial t} - \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y) \\
& \sum_{l,m\in K_N^3} \hat{f}_l \hat{f}_m \delta_{l+m,k} \left[e^{i[l\cdot z+m\cdot y]} - e^{im\cdot(y+z)} \right] \mathrm{d}y\mathrm{d}z
\end{aligned}
$$

Fourier-Galerkin Derivation

skip the details

$$0 = \frac{\partial \hat{f}_k}{\partial t} - \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z \cdot y) \sum_{l,m \in K_N^3} \hat{f}_l \hat{f}_m \delta_{l+m,k} \left[e^{i[l \cdot z + m \cdot y]} - e^{im \cdot (y+z)} \right] \mathrm{d}y\mathrm{d}z$$

Fourier-Galerkin Derivation

skip the details

$$
\begin{aligned}
0 =& \frac{\partial \hat{f}_k}{\partial t} - \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y) \\
& \sum_{l,m \in K_N^3} \hat{f}_l \hat{f}_m \delta_{l+m,k} \left[e^{i[l\cdot z + m\cdot y]} - e^{im\cdot(y+z)} \right] \mathrm{d}y\mathrm{d}z \\
=& \frac{\partial \hat{f}_k}{\partial t} - \sum_{l,m \in K_N^3, l+m=k} \hat{f}_l \hat{f}_m \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y) \\
& \left[e^{i[l\cdot z + m\cdot y]} - e^{im\cdot(y+z)} \right] \mathrm{d}y\mathrm{d}z
\end{aligned}
$$

Fourier-Galerkin Derivation

skip the details

$$0 = \frac{\partial \hat{f}_k}{\partial t} - \sum_{l,m \in K_N^3, l+m=k} \hat{f}_l \hat{f}_m \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y) \left[e^{i[l\cdot z + m\cdot y]} - e^{im\cdot(y+z)} \right] \mathrm{d}y\mathrm{d}z$$

Fourier-Galerkin Derivation

skip the details

$$0 = \frac{\partial \hat{f}_k}{\partial t} - \sum_{l,m \in K_N^3, l+m=k} \hat{f}_l \hat{f}_m \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z \cdot y) \left[e^{i[l \cdot z + m \cdot y]} - e^{im \cdot (y+z)} \right] \mathrm{d}y \mathrm{d}z$$

$$= \frac{\partial \hat{f}_k}{\partial t} - \sum_{l,m \in K_N^3, l+m=k} \hat{f}_l \hat{f}_m \hat{\beta}(l,m)$$

Fourier-Galerkin Derivation

skip the details

$$0 = \frac{\partial \hat{f}_k}{\partial t} - \sum_{l,m \in K_N^3, l+m=k} \hat{f}_l \hat{f}_m \hat{\beta}(l, m)$$

Fourier-Galerkin Derivation

skip the details

$$
\begin{aligned}
0 &= \frac{\partial \hat{f}_k}{\partial t} - \sum_{l,m\in K_N^3, l+m=k} \hat{f}_l \hat{f}_m \hat{\beta}(l,m) \\
&= \frac{\partial \hat{f}_k}{\partial t} - \sum_{m\in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{\beta}(k-m,m)
\end{aligned}
$$

Fourier-Galerkin Summary

We obtained

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{\beta}(k-m, m)$$

Fourier-Galerkin Summary

We obtained

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{\beta}(k-m, m)$$

with

$$\hat{\beta}(l, m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y, z) \delta(z \cdot y) \left[e^{i[l \cdot z + m \cdot y]} - e^{im \cdot (y+z)} \right] \mathrm{d}y \mathrm{d}z.$$

Fourier-Galerkin Summary

We obtained

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{\beta}(k-m, m)$$

with

$$\hat{\beta}(l, m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y, z) \delta(z \cdot y) \left[e^{i[l \cdot z + m \cdot y]} - e^{im \cdot (y+z)} \right] \mathrm{d}y \mathrm{d}z.$$

Simplify by setting

$$\hat{\beta}(l, m) = \beta(l, m) - \beta(m, m)$$

Fourier-Galerkin Summary

We obtained

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \hat{\beta}(k-m, m)$$

with

$$\hat{\beta}(l, m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y, z) \delta(z \cdot y) \left[e^{i[l \cdot z + m \cdot y]} - e^{im \cdot (y+z)} \right] \mathrm{d}y \mathrm{d}z.$$

Simplify by setting

$$\hat{\beta}(l, m) = \beta(l, m) - \beta(m, m)$$

with

$$\beta(l, m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y, z) \delta(z \cdot y) \left[e^{i[l \cdot y + m \cdot z]} \right] \mathrm{d}y \mathrm{d}z = \beta(m, l).$$

Remarks on the Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m [\beta(k-m, m) - \beta(m, m)]$$

Pareschi-Russo scheme looked exactly the same.

Remarks on the Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m [\beta(k-m, m) - \beta(m, m)]$$

Pareschi-Russo scheme looked exactly the same.
(but with different β)

Remarks on the Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m [\beta(k-m, m) - \beta(m, m)]$$

Pareschi-Russo scheme looked exactly the same.
(but with different β)
$\rightarrow$ How are we going to save time here?

Remarks on the Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m [\beta(k-m, m) - \beta(m, m)]$$

Pareschi-Russo scheme looked exactly the same.
(but with different β)
$\longrightarrow$ How are we going to save time here?
Recall: Pareschi-Russo loss term could be evaluated in $O(N^3 \log N)$ because of special structure.

Remarks on the Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m [\beta(k-m, m) - \beta(m, m)]$$

Pareschi-Russo scheme looked exactly the same.
(but with different β)
$\longrightarrow$ How are we going to save time here?
Recall: Pareschi-Russo loss term could be evaluated in $O(N^3 \log N)$ because of special structure.
$\longrightarrow$ Try for the same here.

Special Structure of the Kernel Modes

For Hard Spheres in 3D, we obtain the special structure

Special Structure of the Kernel Modes

For Hard Spheres in 3D, we obtain the special structure

$$\beta(l, m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y, z)\delta(z \cdot y) \left[e^{i[l \cdot y + m \cdot z]} \right] \mathrm{d}y \mathrm{d}z$$

Special Structure of the Kernel Modes

For Hard Spheres in 3D, we obtain the special structure

$$\beta(l,m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y)\left[e^{i[l\cdot y+m\cdot z]}\right] \mathrm{d}y\mathrm{d}z$$

$$\approx \tilde{\beta}(l,m) := \frac{\pi^2}{M^2}\sum_{p,q=0}^{M} \alpha_{p,q}(l)\alpha'_{p,q}(m)$$

Special Structure of the Kernel Modes

For Hard Spheres in 3D, we obtain the special structure

$$\beta(l,m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y)\left[e^{i[l\cdot y + m\cdot z]}\right] \mathrm{d}y\mathrm{d}z$$

$$\approx \tilde{\beta}(l,m) := \frac{\pi^2}{M^2}\sum_{p,q=0}^{M} \alpha_{p,q}(l)\alpha'_{p,q}(m)$$

How? skip

Special Structure of the Kernel Modes

For Hard Spheres in 3D, we obtain the special structure

$$\beta(l,m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y)\left[e^{i[l\cdot y+m\cdot z]}\right] \mathrm{d}y\mathrm{d}z$$

$$\approx \tilde{\beta}(l,m) := \frac{\pi^2}{M^2}\sum_{p,q=0}^{M} \alpha_{p,q}(l)\alpha'_{p,q}(m)$$

How? ▸ skip (roughly)

- Introduce spherical coordinates.

Special Structure of the Kernel Modes

For Hard Spheres in 3D, we obtain the special structure

$$\begin{aligned}\beta(l,m) &:= \int_{\mathcal{B}(0,R)}\int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y)\left[e^{i[l\cdot y+m\cdot z]}\right]\mathrm{d}y\mathrm{d}z\\ &\approx \tilde{\beta}(l,m) := \frac{\pi^2}{M^2}\sum_{p,q=0}^{M}\alpha_{p,q}(l)\alpha'_{p,q}(m)\end{aligned}$$

How? skip (roughly)

- Introduce spherical coordinates.
- Assume decoupling $\tilde{B}(y,z) = a(|y|)b(|z|)$ for $y \perp z$. (Holds for Hard Spheres in 3D, Maxwell molecules in 2D, ...)

Special Structure of the Kernel Modes

For Hard Spheres in 3D, we obtain the special structure

$$\beta(l,m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y)\left[e^{i[l\cdot y + m\cdot z]}\right] \mathrm{d}y\mathrm{d}z$$
$$\approx \tilde{\beta}(l,m) := \frac{\pi^2}{M^2}\sum_{p,q=0}^{M} \alpha_{p,q}(l)\alpha'_{p,q}(m)$$

How? ▸ skip (roughly)

- Introduce spherical coordinates.
- Assume decoupling $\tilde{B}(y,z) = a(|y|)b(|z|)$ for $y \perp z$. (Holds for Hard Spheres in 3D, Maxwell molecules in 2D, ...)
- Periodicity reduces spherical integration to hemisphere S^2_+.

Special Structure of the Kernel Modes

For Hard Spheres in 3D, we obtain the special structure

$$\begin{aligned}\beta(l,m) &:= \int_{\mathcal{B}(0,R)}\int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y)\left[e^{i[l\cdot y+m\cdot z]}\right]\mathrm{d}y\mathrm{d}z\\ &\approx \tilde{\beta}(l,m) := \frac{\pi^2}{M^2}\sum_{p,q=0}^{M}\alpha_{p,q}(l)\alpha'_{p,q}(m)\end{aligned}$$

How? skip (roughly)

- Introduce spherical coordinates.
- Assume decoupling $\tilde{B}(y,z) = a(|y|)b(|z|)$ for $y \perp z$. (Holds for Hard Spheres in 3D, Maxwell molecules in 2D, ...)
- Periodicity reduces spherical integration to hemisphere S^2_+.
- Parameterize S^2_+ by $(\theta,\varphi) \in [0,\pi)^2$.

Special Structure of the Kernel Modes

For Hard Spheres in 3D, we obtain the special structure

$$\beta(l,m) := \int_{\mathcal{B}(0,R)} \int_{\mathcal{B}(0,R)} \tilde{B}(y,z)\delta(z\cdot y)\left[e^{i[l\cdot y+m\cdot z]}\right] \mathrm{d}y\mathrm{d}z$$

$$\approx \tilde{\beta}(l,m) := \frac{\pi^2}{M^2}\sum_{p,q=0}^{M} \alpha_{p,q}(l)\alpha'_{p,q}(m)$$

How? skip (roughly)

- Introduce spherical coordinates.
- Assume decoupling $\tilde{B}(y,z) = a(|y|)b(|z|)$ for $y \perp z$. (Holds for Hard Spheres in 3D, Maxwell molecules in 2D, ...)
- Periodicity reduces spherical integration to hemisphere S^2_+.
- Parameterize S^2_+ by $(\theta,\varphi) \in [0,\pi)^2$.
- Discretize

$$\theta := p\pi/M, \quad \varphi := q\pi/M.$$

Fast Evaluation of the Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \frac{\pi^2}{M^2} \sum_{l,m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \beta(k-m, m) \quad - \quad \text{loss}$$

Fast Evaluation of the Scheme

$$\frac{\partial \hat{f}_k}{\partial t} = \frac{\pi^2}{M^2} \sum_{l,m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \beta(k-m, m) \quad - \quad \text{loss}$$

$$= \frac{\pi^2}{M^2} \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \sum_{p,q=0}^{M} \alpha_{p,q}(k-m) \alpha'_{p,q}(m) \quad - \quad \text{loss}$$

Fast Evaluation of the Scheme

$$\begin{aligned}
\frac{\partial \hat{f}_k}{\partial t} &= \frac{\pi^2}{M^2} \sum_{l,m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \beta(k-m, m) \quad - \quad \text{loss} \\
&= \frac{\pi^2}{M^2} \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \sum_{p,q=0}^{M} \alpha_{p,q}(k-m) \alpha'_{p,q}(m) \quad - \quad \text{loss} \\
&= \frac{\pi^2}{M^2} \sum_{p,q=0}^{M} \sum_{m \in K_N^3} \hat{f}_{k-m} \alpha_{p,q}(k-m) \hat{f}_m \alpha'_{p,q}(m) \quad - \quad \text{loss.}
\end{aligned}$$

Fast Evaluation of the Scheme

$$\begin{aligned}
\frac{\partial \hat{f}_k}{\partial t} &= \frac{\pi^2}{M^2} \sum_{l,m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \beta(k-m, m) \quad - \quad \text{loss} \\
&= \frac{\pi^2}{M^2} \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \sum_{p,q=0}^{M} \alpha_{p,q}(k-m) \alpha'_{p,q}(m) \quad - \quad \text{loss} \\
&= \frac{\pi^2}{M^2} \sum_{p,q=0}^{M} \sum_{m \in K_N^3} \hat{f}_{k-m} \alpha_{p,q}(k-m) \hat{f}_m \alpha'_{p,q}(m) \quad - \quad \text{loss.}
\end{aligned}$$

$\rightarrow$ Sum of M^2 convolutions.

Fast Evaluation of the Scheme

$$\begin{aligned}
\frac{\partial \hat{f}_k}{\partial t} &= \frac{\pi^2}{M^2} \sum_{l,m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \beta(k-m, m) \quad - \quad \text{loss} \\
&= \frac{\pi^2}{M^2} \sum_{m \in K_N^3} \hat{f}_{k-m} \hat{f}_m \sum_{p,q=0}^{M} \alpha_{p,q}(k-m) \alpha'_{p,q}(m) \quad - \quad \text{loss} \\
&= \frac{\pi^2}{M^2} \sum_{p,q=0}^{M} \sum_{m \in K_N^3} \hat{f}_{k-m} \alpha_{p,q}(k-m) \hat{f}_m \alpha'_{p,q}(m) \quad - \quad \text{loss.}
\end{aligned}$$

$\rightarrow$ Sum of M^2 convolutions. Use FFT: $M^2 N^3 \log N$ operations.

Final Remarks on Mouhot-Pareschi

$$\frac{\partial \hat{f}_k}{\partial t} = \frac{\pi^2}{M^2} \sum_{p,q=0}^{M} \sum_{m \in K_N^3} \hat{f}_{k-m} \alpha_{p,q}(k-m) \hat{f}_m \alpha'_{p,q}(m) \quad - \quad \text{loss.}$$

- Fastest method around.

Final Remarks on Mouhot-Pareschi

$$\frac{\partial \hat{f}_k}{\partial t} = \frac{\pi^2}{M^2} \sum_{p,q=0}^{M} \sum_{m \in K_N^3} \hat{f}_{k-m} \alpha_{p,q}(k-m) \hat{f}_m \alpha'_{p,q}(m) \quad - \quad \text{loss.}$$

- Fastest method around.
- Spectrally accurate. (Similar result as for Pareschi-Russo)

Final Remarks on Mouhot-Pareschi

$$\frac{\partial \hat{f}_k}{\partial t} = \frac{\pi^2}{M^2} \sum_{p,q=0}^{M} \sum_{m \in K_N^3} \hat{f}_{k-m} \alpha_{p,q}(k-m) \hat{f}_m \alpha'_{p,q}(m) \quad - \quad \text{loss.}$$

- Fastest method around.
- Spectrally accurate. (Similar result as for Pareschi-Russo)
- Only "sophisticated" numerical technique is FFT. Probably the easiest to implement of all three methods discussed.

Final Remarks on Mouhot-Pareschi

$$\frac{\partial \hat{f}_k}{\partial t} = \frac{\pi^2}{M^2} \sum_{p,q=0}^{M} \sum_{m \in K_N^3} \hat{f}_{k-m} \alpha_{p,q}(k-m) \hat{f}_m \alpha'_{p,q}(m) \quad - \quad \text{loss.}$$

- Fastest method around.
- Spectrally accurate. (Similar result as for Pareschi-Russo)
- Only "sophisticated" numerical technique is FFT. Probably the easiest to implement of all three methods discussed.
- Some care needs to be taken to retain proper conservation.

Conclusion

	M	H	V	O	Order	Ops	Ms	Mm	En
DSMC	✓	✓	✓	✓	$N^{-1/2}$	N^3	✓	✓	✓
IbrR	✓	✓	✓	–	N^{-2}	N^6	✓	◦	◦
gen.	–	–	–	✓	N^{-2}	$N^6 \log N$	✓	◦	◦
BobR	–	3	–	–	N^{-2}	$N^6 \log N$	✓	◦	◦
ParRu	✓	✓	✓	–	$N^{-\infty}$	N^6	✓	◦	◦
MouP	2	3	–	–	$N^{-\infty}$	$N^5 \log N$	✓	◦	◦

Conclusion

	M	H	V	O	Order	Ops	Ms	Mm	En
DSMC	✓	✓	✓	✓	$N^{-1/2}$	N^3	✓	✓	✓
IbrR	✓	✓	✓	—	N^{-2}	N^6	✓	○	○
gen.	—	—	—	✓	N^{-2}	$N^6 \log N$	✓	○	○
BobR	—	3	—	—	N^{-2}	$N^6 \log N$	✓	○	○
ParRu	✓	✓	✓	—	$N^{-\infty}$	N^6	✓	○	○
MouP	2	3	—	—	$N^{-\infty}$	$N^5 \log N$	✓	○	○

Additional topic: Wild expansion timestepping for the collision operator.

Conclusion

	M	H	V	O	Order	Ops	Ms	Mm	En
DSMC	✓	✓	✓	✓	$N^{-1/2}$	N^3	✓	✓	✓
IbrR	✓	✓	✓	–	N^{-2}	N^6	✓	○	○
gen.	–	–	–	✓	N^{-2}	$N^6 \log N$	✓	○	○
BobR	–	3	–	–	N^{-2}	$N^6 \log N$	✓	○	○
ParRu	✓	✓	✓	–	$N^{-\infty}$	N^6	✓	○	○
MouP	2	3	–	–	$N^{-\infty}$	$N^5 \log N$	✓	○	○

Additional topic: Wild expansion timestepping for the collision operator.

- maintains positivity

Conclusion

	M	H	V	O	Order	Ops	Ms	Mm	En
DSMC	✓	✓	✓	✓	$N^{-1/2}$	N^3	✓	✓	✓
IbrR	✓	✓	✓	–	N^{-2}	N^6	✓	∘	∘
gen.	–	–	–	✓	N^{-2}	$N^6 \log N$	✓	∘	∘
BobR	–	3	–	–	N^{-2}	$N^6 \log N$	✓	∘	∘
ParRu	✓	✓	✓	–	$N^{-\infty}$	N^6	✓	∘	∘
MouP	2	3	–	–	$N^{-\infty}$	$N^5 \log N$	✓	∘	∘

Additional topic: Wild expansion timestepping for the collision operator.

- maintains positivity
- see notes

Questions?

?