closures-2d

C. Kristopher Garrett, Tim Shaffer

August 4, 2015

What Are Kinetic Equations?

Macroscopic

- $\rho(\mathbf{x}, t)$ – Density
- $\mathbf{u}(\mathbf{x}, t)$ – Velocity
- $E(\mathbf{x}, t)$ – Kinetic Energy

Discretize $\mathbf{x}$, t into 100 values: 4GB memory requirement

Mesoscopic

- $f(\mathbf{x}, \mathbf{v}, t)$ – Density with respect to space *and velocity*

Discretize $\mathbf{x}$, $\mathbf{v}$, t into 100 pieces: 800TB memory requirement

Macroscopic can be derived from mesoscopic

- $\rho(\mathbf{x}, t) = \int_{\mathbb{R}^3} f \, d\mathbf{v}$
- $\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho} \int_{\mathbb{R}^3} \mathbf{v} f \, d\mathbf{v}$
- $E(\mathbf{x}, t) = \frac{1}{2} \int_{\mathbb{R}^3} \|\mathbf{v} - \mathbf{u}\|^2 \, d\mathbf{v}$

What Are Kinetic Equations?

Form of Kinetic Equation – Neutral Particles

$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} = C(f)$

Left side is the transport equation

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} = C(f)$$

Right side governs collisions

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} = C(f)$$

Collisions change the direction of particles
The collision operator is problem dependent

What Are Kinetic Equations?

First used for rarefied gas dynamics (e.g. high altitude gases where collisions do not dominate the physics)

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} = C(f)$$

where $\int C(f)\,\mathrm{d}\mathbf{v} = 0$.

Integrate against $\mathbf{v}$ to get First Euler/Navier-Stokes equation

$$\partial_t \rho + \nabla_{\mathbf{x}} \cdot (\rho \mathbf{u}) = 0$$

Other areas:

- Radiation transport
- Plasma simulations

What Are Spherical Harmonics?

(Real) Spherical Harmonics

$$Y_{\ell m}(\mu, \phi) = \begin{cases} \sqrt{2} N_\ell^{|m|} P_\ell^{|m|}(\mu) \sin(|m|\phi), & m < 0 \\ N_\ell^0 P_\ell(\mu), & m = 0 \\ \sqrt{2} N_\ell^m P_\ell^m(\mu) \cos(m\phi), & m > 0 \end{cases}$$

where $N_\ell^m = \sqrt{\frac{2\ell+1}{4\pi} \frac{(\ell-m)!}{(\ell+m)!}}$.

- Spherical harmonics are an orthonormal basis of $L^2(S^d)$
- They are usually considered for the unit sphere S^2
- For $d = 1$, you get Fourier series
- For $d = 2$, you get $Y_{\ell m}(\mu, \phi)$ where ℓ is the degree and m is the order

steal pic

Kinetic Problem

Unit Speed, Isotropic Scattering

$$\partial_t f + \Omega \cdot \nabla_{\mathbf{x}} = \frac{\sigma}{4\pi} \langle f \rangle - \sigma f$$

where $x \in \mathbb{R}^3$, $\Omega \in S^2$, σ is the scattering cross section, and $\langle \cdot \rangle = \int_{S^2} \cdot \, \mathrm{d}\Omega$.

- Let $\mathbf{m}(\Omega) = \left(Y_{0,0}, Y_{1,-1}, Y_{1,0}, Y_{1,1}, \ldots, Y_{N,-N}, \ldots, Y N, N\right)^T$ be a vector of spherical harmonics up to and including degree N
- Take moments with respect to $\mathbf{m}$
- $\mathbf{u}(\mathbf{x}, t) = \left\langle \mathbf{m}(\Omega) f(\mathbf{x}, \Omega, t) \right\rangle$

Why? Because $u_{\ell m} = f_{\ell m}$ and the collision operator is diagonalized

Moment Closures

Exact Moment System

$\partial_t \mathbf{u} + \nabla_x \cdot \langle \Omega \mathbf{m} f \rangle = \mathrm{D}\mathbf{u}$

System is not closed
Think of 1D case with $\mathbf{m}(\mu) = (1, \mu, \mu^2, \ldots, \mu^n)$:

$$
\begin{aligned}
\partial_t u_0 + \partial_x u_1 &= \mathrm{D}_{00}\, u_0 \\
\partial_t u_1 + \partial_x u_2 &= \mathrm{D}_{11}\, u_1 \\
\partial_t u_2 + \partial_x u_3 &= \mathrm{D}_{22}\, u_2 \\
&\vdots \\
\partial_t u_n + \partial_x u_{n+1} &= \mathrm{D}_{nn}\, u_n
\end{aligned}
$$

To close the system: Replace f with $\mathcal{E}(\mathbf{u})$ such that $\left\langle \mathbf{m}\mathcal{E}(\mathbf{u}) \right\rangle = \mathbf{u}$

Moment Closures

$\mathrm{P_N}$ Moment Closure

$$\mathcal{E}(\mathbf{u}) = \mathbf{u}^T \mathbf{m}$$

$\mathrm{M_N}$ Moment Closure

$$\mathcal{E}(\mathbf{u}) = \exp\left(\alpha^{\mathsf{T}} \mathbf{m}\right)$$

where α solves $\min_{\hat{\alpha}} \left\langle \exp\left(\hat{\alpha}^{\mathsf{T}} \mathbf{m}\right) \right\rangle - \hat{\alpha}^{\mathsf{T}} \mathbf{u}$

Two notes

- The moment closure occurs on every spatial cell and every time point
- All the moment closures are **independent**

Discrete Ordinates (S_N)

- Textbook method used in many applications
- Simple, direct approximation of the kinetic equation using quadrature sets on the unit sphere
- Strong ray effects at low orders

$$\partial_t f_q + \Omega_q \cdot \nabla_x f_q = \frac{\sigma}{4\pi} \sum_{q'=1}^{Q} w_{q'} f_{q'} - \sigma f_q$$

where $f_q(x,t) \approx f(x, \Omega_q, t)$ for $q = 1, \ldots, Q$

Initial Conditions

Gaussian Initial Condition

Two dimensional Gaussian function centered at the origin

$$f(x, y, \Omega, t = 0) = \max\left(\frac{1}{2\pi\sigma_g^2} e^{-(x^2+y^2)/(2\sigma_g^2)}, \texttt{floor}\right),$$

where σ_g is the configurable Gaussian sigma and `floor` is the floor value for the grid.

Delta Initial Condition

Limiting case of Gaussian I.C. with $\sigma_g \to 0$.
Simulates an initial pulse of particles distributed isotropically along an infinite line in space.
Centermost cell is given a high initial density.

Initial Conditions

Lattice Initial Condition

Checker board pattern of highly scattering and highly absorbing regions with empty initial grid.
Reminiscent of a small section of a nuclear reactor core.

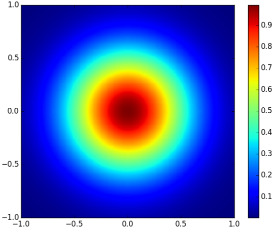
Smooth Initial Condition

Primarily intended for testing convergence
Initialize the grid points with a periodic boundary given by

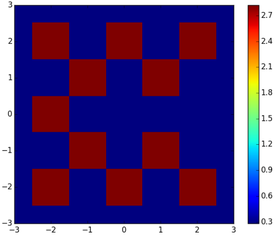
$$f(x, y, \Omega, t = 0) = 1 + \sin(2\pi x)\cos(2\pi y).$$

Initial Conditions

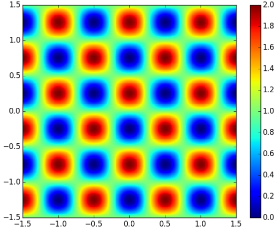
Gaussian Initial Condition



σ_T for Lattice Initial Condition



Smooth Initial Condition



Implementation Notes

Spherical Harmonics: $Y_{\ell m}(\Omega)$

- $\ell = 0, \ldots, N$ is the degree
- $m = -\ell, \ldots, \ell$ is the order
- Total number of moments: $M = (N+1)^2$

$$
\begin{array}{ccccc}
 & & Y_{0,0} & & \\
 & Y_{1,-1} & Y_{1,0} & Y_{1,1} & \\
Y_{2,-2} & Y_{2,-1} & Y_{2,0} & Y_{2,1} & Y_{2,2}
\end{array}
$$

Quadrature in angle: Ω_q, w_q

- $\int_{S^2} F(\Omega)\,\mathrm{d}\Omega \approx \sum_q w_q F(\Omega_q)$
- Product quadrature: $\int_{S^2} \mathrm{d}\Omega = \int_{-1}^{1} \int_0^{2\pi} \mathrm{d}\phi\,\mathrm{d}\mu$
- n_g Gaussian nodes on μ axis
- $2n_g$ equally spaced nodes on latitudinal circles (for ϕ)
- Total number of quadrature points $Q = 2n_g^2$
- Integrates spherical harmonics of degree $2n_g - 1$ exactly!

Spatial Discretization

- Use a 2D Cartesian mesh with constant cell size to break problem space into an m x n grid
- Store density with respect to direction for each cell
- Also need a halo of ghost cells for boundary conditions, MPI communication

Upwinding

Since f depends on position, **direction**, and time, we *must* use information from the correct side

steal pictures?

Time Stepping

CFL Condition

A small enough time step is necessary for convergence. In this case,

$$\frac{\Delta t}{\Delta x} + \frac{\Delta t}{\Delta y} \leq 1$$

Heun's Method

To compute a first estimate, carry out two Euler steps. Now use the average of the initial state and the estimate.

- Explicit, two-stage Runge-Kutta method
- Second order accurate
- Strong stability preserving

Results

Tabulate some runtimes, convergence rates, etc.

Show some plots

Release

We are releasing this code as open source software.

Built using SCons

Depends on

- GSL
- BLAS
- LAPACK
- OpenMP (optional, provided by many compilers)
- Open MPI (optional)

Tests

We implemented automated testing for the following aspects. The testing code carries out

- Regression tests – compare output to reference data included with the software
- Convergence tests – measure the effect of decreasing the cell size on the precision of the output
- Mass Conservation tests – check that total density remains the same

The testing code is written in Python and integrated with the build system

Improvements

Algorithmic changes:

- Added experimental Lebedev quadrature
- Removed δ factor for ansatz correction in `momopt`
- Changed `momopt`'s regularization in case of bad condition number

Software-related improvements:

- Cross-platform build system
- Automated testing
- Better MPI communication
- Improved documentation
- Improved interface
- Bugfixes
- Profiling and optimization

Bibliography

cite that presentation, linesource paper