WASP: Weighted Algorithms for Stochastic Particles

Ian Schreiber

University of Texas at Dallas

2021

Table of Contents

What is WASP?

WASP is a code created to implement homogeneous and inhomogeneous stochastic weighted particle methods developed at the University of Texas at Dallas. These methods were primarily explored by Sonam Lama and Brian Jensen respectively, both under Drs. Matthew Goeckner and John Zweck.

Intro to Fluid Dynamics

Fluids, at their most primitive level, are a collection of interacting particles. Simulating the dynamics of a collection of particles can be very difficult, so simplifications are necessary for non-trivial computation of fluid dynamics. A very commonly applicable simplification is that of *continuum mechanics*, which approximates the collection as a single deformable body. A measure of the applicability of continuum mechanics is the *Knudsen number*, defined as the ratio of the mean free path of a particle λ to a representative length L of a problem in question:

$$Kn = \frac{\lambda}{L}.$$

Continuum models are applicable for $Kn \approx< 0.1$, while statistical (also known as molecular) methods are needed for $Kn \approx> 0.1$.

Continuum Models of Fluid Dynamics

- Euler equations: most simplified system. Assume adiabatic and inviscid flow. $Kn = 0$.
- Navier-Stokes Equations: applicable to many everyday fluids. $Kn < 0.1$.
- Burnett Equations: an extension of the NS Equations utilizing Chapman-Enskog theory. $Kn < 1$

Molecular Models of Fluid Dynamics

- Boltzmann Equation: nonlinear integro-differential equation describing a set of particles utilizing a probability distribution function (pdf) for an average single particle. Assumes that most particles spend most of their time evolving freely, with only occasional collisional perturbations. $Kn > 0.1$.
- BBGKY Hierarchy: Hamiltonian mechanics plus pdfs.
- Hamiltonian mechanics: directly simulating each particle and all collisions with no simplifying assumptions. Mathematically intractable for any more than a trivial number of particles.

Model Applicabilities vs. Knudsen Number

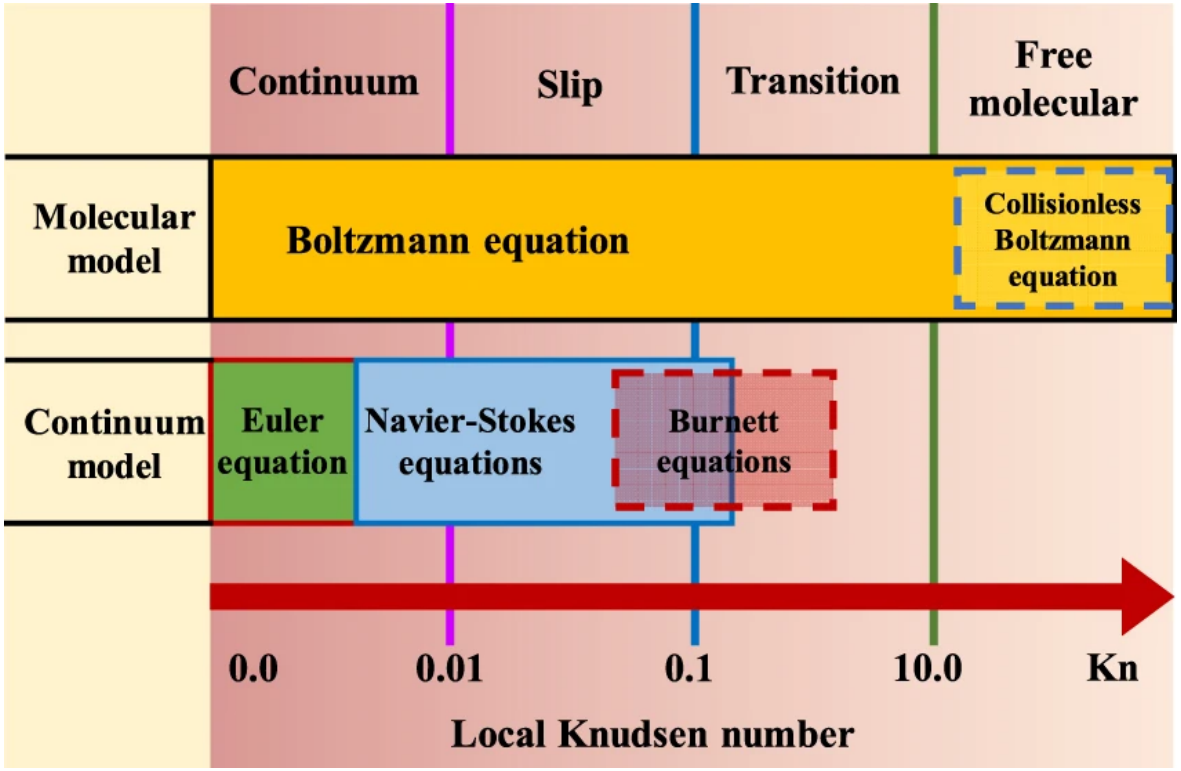

Figure: Table of applicabilities for different gaseous models. Note that while the Boltzmann equation is technically applicable for $Kn < 0.1$, it becomes mathematically impossible in practice.

The Boltzmann Equation

We are modelling plasmas with the Boltzmann equation, in order to capture as much physical detail as is possible with currently available computing technology. For a plasma with m species, for any given species i described by a pdf $f_i = f_i(\mathbf{x}, \mathbf{v}, t)$, we have

$$\frac{\partial f_i}{\partial t} = \left(\frac{\partial f_i}{\partial t}\right)_{coll} - \mathbf{v} \cdot \nabla f_i - \mathbf{a} \cdot \frac{\partial f_i}{\partial v} \tag{1}$$

$$\left(\frac{\partial f_i}{\partial t}\right)_{coll} = \sum_{j=1}^{m} \int (f_i' f_j' - f_i f_j) B(\mathbf{v}_i, \mathbf{v}_j, \mathbf{\Theta}) d\mathbf{\Theta} d\mathbf{v}_j \tag{2}$$

where B is the *collision kernel*, $\mathbf{\Theta}$ is the *scattering angle*, and $f_i' = f_i(\mathbf{x}, \mathbf{v}', t)$, and $\mathbf{v}'$ is the postcollisional velocity.

Collision Kernel

A collision kernel (sometimes referred to as the *differential cross section* or just *cross section*) is a function that relates incoming particles to their probabilities of ejection, depending on the outgoing direction in question. In more specificity, consider a probability distribution $\sigma(\theta, \varphi)$ of incoming particles to a collision event, and a small given section of that, $d\sigma$. Consider an outgoing solid angle $d\Omega$. Then, the differential cross section is simply $d\sigma/d\Omega$. In the isotropic, hard sphere case, we assume no angle of ejection is any more probable than others, so the collision kernel is assumed to be

$$B = \frac{1}{4\pi}.$$

Solving the BE: determinism.

There are two main methods of solving the BE: deterministic and stochastic approaches.
Deterministic approaches utilize simplifying assumptions to allow direct application of numerical methods of solution.

- Kinetic models: employs assumptions about the smoothness of the velocity pdf.
- Fluid model: assumes the shape of the velocity pdf is known, allowing highly accurate, low computational cost statistics to be computed. This assumption is invalid when the plasma is not in thermal equilibrium or the species are not uniformly distributed in space.

Solving the BE: stochastics.

Rather than apply simplifying assumptions, one may apply Monte Carlo methods to approximate the solution with a random amount of error. The most fundamental method in this discipline is the *direct simulation Monte Carlo* (DSMC) method, developed by Dr. Graeme Bird at the University of Sydney. Used for rarefied gas dynamics, it employs the fundamental assumption that collision and motion can be perfectly decoupled and computed serially, and the Boltzmann equation can be solved through the dynamic simulation of *stochastic particles* and *stochastic collisions*.

DSMC.

We are attempting to solve the Boltzmann Equation utilizing an algorithm called the *stochastic weighted particle method for low probability tails* (SWPM-LP), developed by Sonam Lama and extended to the inhomogeneous case by Brian Jensen. SWPM is an extension of DSMC methods, which we will describe next.

DSMC Structure, I

DSMC blends together molecular dynamics (MD) and probabilistic methods to solve the BE. Specifically, rather than calculating exact collisions for exact particles, equal sized groups of particles positioned closely in phase space (that is, particles with momenta of similar direction and magnitude) are lumped together in a mathematical object called a *stochastic particle*, reducing computational complexity. Collisions between stochastic particles are also similarly simplified by substituting exact, expensive dynamical collision computations with *stochastic collisions*, where postcollision velocities and scattering angles are not truly computed, but instead stochastically selected from theoretical models of kinetic collision.

DSMC Structure, II

SPARTA, a parallelized implementation of DSMC methods, uses the following steps:

1. Create: particles are created according to situational needs.
2. Move: particles are advected based on the forces and momenta applied.
3. Communicate: in parallel methods, approximately equal numbers of particles are desired in each parallel thread. So, this step may migrate mathematical "possession" of some stochastic particles to other threads.
4. Sort: specific to SPARTA, this improves calculation time by pre-sorting a list.
5. Collide: collisions and chemistry routines are calculated.
6. Stats: statistics are calculated, either for particles, surface elements, or volume elements.

SWPM

Rjasanow and Wagner created the original SWPM. As a modification of the DSMC, it simply allowed each stochastic particle to have a different weight, representing differing numbers of particles per stochastic particle. This modification forces the number of stochastic particles to change over time, so regrouping of particles is needed to keep the number of SP's constant. This process is called *particle reduction and clustering.*

SWPM-LP

Lama's modifications of the SWPM consist mainly of changes to the particle reduction and clustering methods. The particle reduction methods proposed in the original SWPM did not conserve the heat flux moment across the entire system. Additionally, the clustering algorithms are altered, forcing stochastic particles to contain particles closer in phase space than the original methods in SWPM, increasing physical accuracy. Brian Jensen extended Lama's work to include plasmas that have different velocity pdfs over space, called *inhomogeneous* plasmas.

Code

SWPM-LP methods are focused simply on atomistic simulation. At their most narrow, they do not take external chemistry, complex boundary conditions, or other multiphysics considerations into concern. Additionally, the stochastic nature of SWPM-LP methods is innately incompatible with deterministic methods such as FEM. To extend SWPM-LP methods and integrate them with different simulation paradigms, we integrate it with a hierarchical set of codes.

Hierarchy

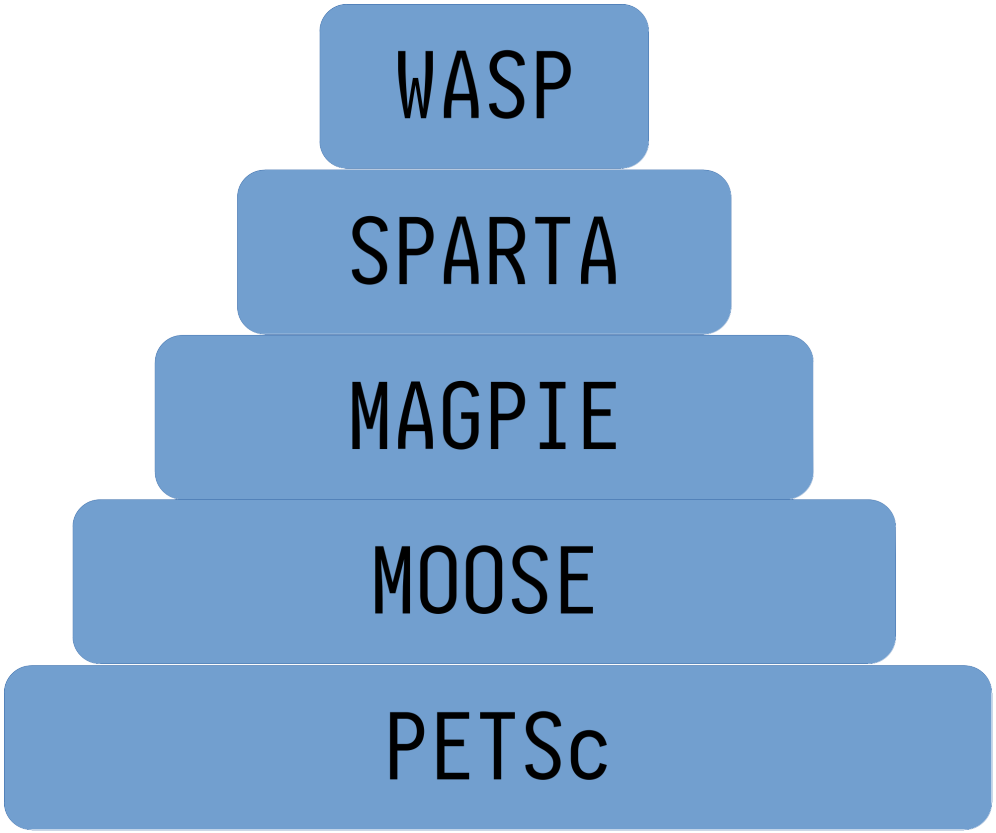

Figure: Hierarchy of programs.

PETSc

- Created September 1991 at Argonne National Laboratories
- Numerical software for large-scale computation
- Focused on non/linear PDE and sparse matrix calculations
- Parallelized with Message Passing Interface (MPI)
- Very low-level; written in C.

MOOSE

- Initially released June 2008, developed at Idaho National Laboratories
- Built on PETSc and libmesh
- Parallel and written in C++; includes scripting language to set up problems
- Includes many common weak form PDEs, called kernels, increasing readability and ease-of-use
- Handles finite element analysis automatically
- Highly extensible; designed with "hooks" to integrate with external programs
- Large suite of community produced modules: ZAPDOS, MARMOT, BISON, etc.

MAGPIE

- Mesoscale-Atomistics Glue Program for Integrated Execution
- Developed 2015 at Idaho National Laboratories
- Built to integrate with the MOOSE Framework.
- Glue application used to link various atomistic codes to MOOSE. Currently linked to:
  - SPPARKS: kinetic MC by Steve Plimpton at Sandia NL
  - MyTRIM: binary collision MC by INL

SPARTA

- Stochastic PArallel Rarefied-gas Time-accurate Analyzer
- Started 2014 at Sandia National Laboratory, authored by Steve Plimpton et al.
- Implements DSMC methods in 2D, 2D axisymmetric, or 3D domains.
- Parallelized with MPI
- Capable of simulating gas-phase and surface collisions and chemistry.
- Built with "hooks" to integrate into other codes, similarly to SPPARKS.
- Currently not integrated with MAGPIE or MOOSE; integration is a goal.

WASP

- Weighted Algorithms for Stochastic Particles
- Developed 2021 at UT Dallas by Ian Schreiber under Drs. Goeckner and Zweck.
- Modifies SPARTA, augmenting the DSMC methods with (in?)homogeneous SWPM-LP methods.