Accelerated Simulation of Coulomb Collisions in Plasmas

R.E. Caflisch[1]
In collaboration with: L.F. Ricketson[1], M.S. Rosin[1], A.M. Dimits[2], B.I. Cohen[2]

[1]UCLA Dept. of Mathematics
[2]Lawrence Livermore National Laboratories

Sherwood Fusion Theory Conference
March 25, 2014

Work performed under auspices of U.S. DOE by UCLA under grant DE-FG02-05ER25710, and by LLNL under Contract DE-AC52-07NA27344.

Outline

- Two Monte Carlo Methods for Coulomb Collisions
  - Binary Collision Methods
    - Direct Simulation Monte Carlo (DSMC)
  - Langevin Collision Methods
    - Stochastic Differential Equations (SDEs)
- Hybrid Fluid-Monte Carlo Schemes for Binary Methods
  - Fluid-Particle Representation of Velocity Distribution
  - Collisions and Thermalization/Dethermalization Step
  - Relative Entropy and a New Thermalization Criterion
  - Results for a Test Problem

Outline - Continued

- Higher Order Methods for SDEs
  - Lowest Order: Euler-Murayama
  - Higher Order: Milstein
  - Levy Areas
- Multi-Level MC (MLMC) for SDEs
  - Combine Solutions for Different Time Steps
  - Antithetic Variables
  - Ito Linearization
  - Results for Test Problems
- Conclusions and Outlook

Coulomb Collisions

Kinetic Theory

- Phase space particle number density $f(\vec{x}, \vec{v}, t)$
- Governing Equation (Boltzmann):

$$\partial_t f + \vec{v} \cdot \nabla_x f + \vec{a} \cdot \nabla_v f = C(f, f) \tag{1}$$

- $C(f, f) =$ Bilinear operator representing 'collisions' between particles - Fokker-Planck operator used for Coulomb interactions

Collisionality

In non-dimensional form, kinetic equation is

$$\partial_t f + \vec{v} \cdot \nabla_x f + \vec{a} \cdot \nabla_v f = \frac{1}{\mathrm{Kn}} C(f, f) \tag{2}$$

- $\mathrm{Kn} = \lambda_{mfp}/L$.
- $\mathrm{Kn} \ll 1 \rightarrow$ leading order behavior $C(f, f) = 0$
  - Local equilibrium in velocity
- $\mathrm{Kn} \gg 1 \rightarrow$ leading order behavior purely convective
  - E.g. interstellar medium, plasma in interior of a tokamak

Highly Collisional Case: $\mathrm{Kn} \ll 1$

f satisfies $C(f, f) = 0$, so that it has *local Maxwellian* form:

$$f(\vec{x}, \vec{v}, t) = \frac{\rho}{(2\pi v_t^2)^{d/2}} \exp\left(-\frac{|\vec{v} - \vec{u}|^2}{v_t^2}\right) \tag{3}$$

where $v_t = \sqrt{2T/m}$, and ρ, $\vec{u}$, T may be functions of $(\vec{x}, t)$

- Taking moments of the kinetic equation $\rightarrow$ Euler equations + E-M fields

Collisionless Case: $\mathrm{Kn} \gg 1$

Vlasov equation:

$$\partial_t f + \vec{v} \cdot \nabla_x f + \vec{a} \cdot \nabla_v f = 0 \tag{4}$$

- Standard realm of PIC schemes

Moderately Collisional Case: $\mathrm{Kn} = O(1)$

Nonlinear Landau-Fokker-Planck (LFP) equation:

$$\partial_t f + \vec{v} \cdot \nabla_x f + \vec{a} \cdot \nabla_v f = \frac{1}{\mathrm{Kn}} C(f, f) \tag{5}$$

with LFP collision operator

$$C(f, f) \equiv -\frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{A}(f) f + \frac{\partial^2}{\partial \mathbf{v} \partial \mathbf{v}} : \mathbf{B}(f) f \tag{6}$$

- **A** and **B** are linear integral operators
- Collisions dominate the computational effort

Focus on Collisions

The focus of this talk is only on collisions

- Spatially homogeneous evolution of f

$$\partial_t f = C(f, f) \tag{7}$$

Monte Carlo Simulation Methods

Two methods for Monte Carlo simulation of collisions:

- Binary: Takizuka-Abe '77, Nanbu '97
- Langevin: Jones '96, Manheimer '97

Binary Collision Method

Coulomb interactions involve many grazing "collisions", which are simulated as in DSMC for neutral particles

- $f(v)$ represented by a set of particles $\{v_1, \ldots, v_n\}$
- In each time step Δt
  - Randomly choose $n/2$ pairs of particles for collisions
  - Each numerical collision is an aggregate of grazing collisions over time step Δt, with randomly chosen collision parameters
- Differences from DSMC
  - Every particle collides once in every time step
  - Collisions are aggregates depending on Δt

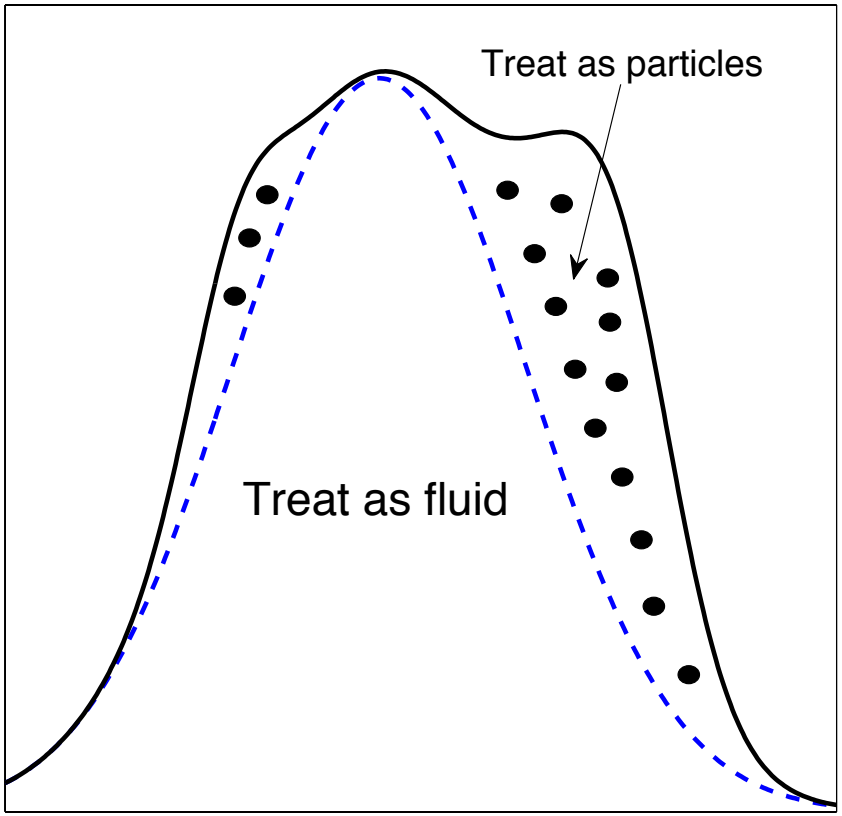
Binary Collision Method: Pros and Cons

- Pros
  - Fully nonlinear and far from equilibirum
- Cons
  - Computationally intensive
  - Especially near the fluid regime $\mathrm{Kn} \ll 1$

Hybrid Schemes for Binary Collision Methods

Hybrid Scheme

Combine fluid and particle simulation methods[1]:



- Separate f into Maxwellian and non-Maxwellian components: $f = m + k$
- Treat m as fluid $\rightarrow$ solves Euler equations
- Simulate k by Monte Carlo algorithm
- Interaction of m and k is the key step

[1] RC et. al, Multiscale Model. Simul. 7 (2008) 865-887.

Interaction of m and k

Two steps

- Collisions
  - Choose a particle from k and sample a particle from M
  - Perform collision as in Monte Carlo algorithm
- Thermalization/dethermalization
  - Collisions drive particles into equilibrium
  - Move particles from k to M when they have collided enough
  - Move sampled particles from M into k if the collision is strong enough
- (De)Thermalization criterion using entropy[2]
  - Alternative criterion based on scattering angle[3]

[2] Ricketson et. al, preprint, 2013

[3] Dimits et. al., private communication

H-Theorem, Relative Entropy

Theorem (Boltzmann H-theorem)

If f solves the kinetic equation, and

$$H = -\int f \log f \, d\vec{v}$$

then $\partial_t H \geq 0$, with equality achieved iff f is Maxwellian.

Lesser known theorem about relative entropy:

Theorem (Relative Entropy Decay)

If f solves $\partial_t f + \vec{v} \cdot \nabla f = C(f, m)$ with m a fixed Maxwellian, and

$$H_{rel}(f, m) = \int f \log\left(\frac{f}{m}\right) \, d\vec{v}$$

then $\partial_t H_{rel} \leq 0$, with equality achieved iff $f = c(\vec{x})m$.

Reinterpreting f

- The velocity space dependence of f is usually interpreted as an ensemble average over many particle velocities.
- We may also assign a velocity space distribution f_p to a single particle, which is interpreted as the probability density of that particle's velocity
- Relative entropy decay theorem ensures that $H_{rel}(f_p, m) \to 0$ through collisions with the fluid component of the scheme
- **Idea**: Track H_{rel} of each simulation particle, thermalize when it falls below some threshold[4]

[4] Ricketson et. al, preprint, 2013

Approximation by Maxwellian

To track H_{rel} exactly, we need to track f_p, which is computationally infeasible

- **Simplifying (but reasonable) assumption**: Approximate f_p by a Maxwellian
- In this case,

$$H_{rel} = \frac{3}{2}\left[\frac{T_p - T_m}{T_m} + \log\left(\frac{T_m}{T_p}\right)\right] + \frac{|\vec{u}_p - \vec{u}_m|^2}{v_{tm}^2} \qquad (8)$$

- So, tracking H_{rel} reduces to tracking T_p and $\vec{u}_p$
- Efficient and accurate method developed for this[5]

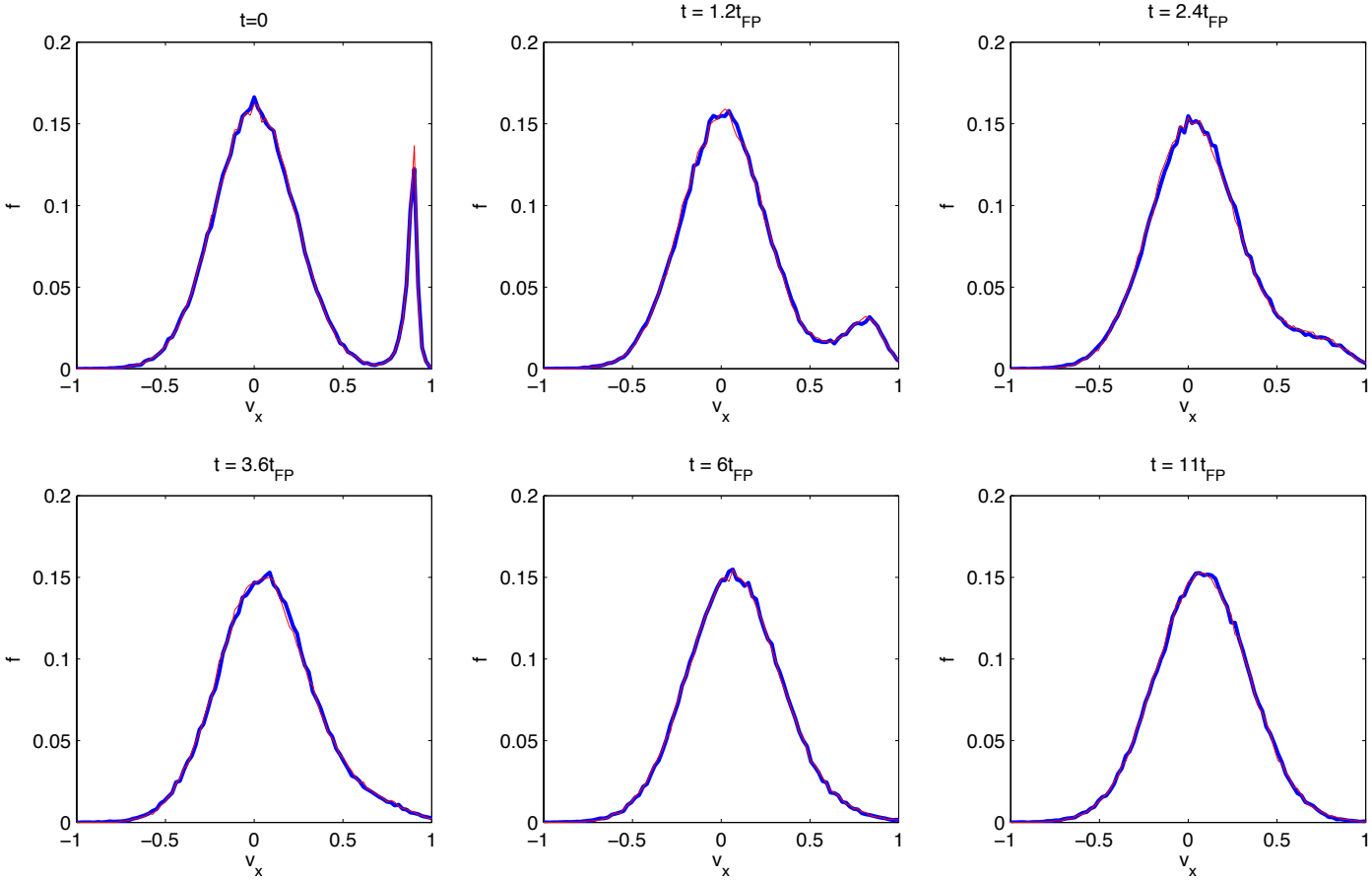
[5] Ricketson et. al, preprint, 2013

Algorithm Summary

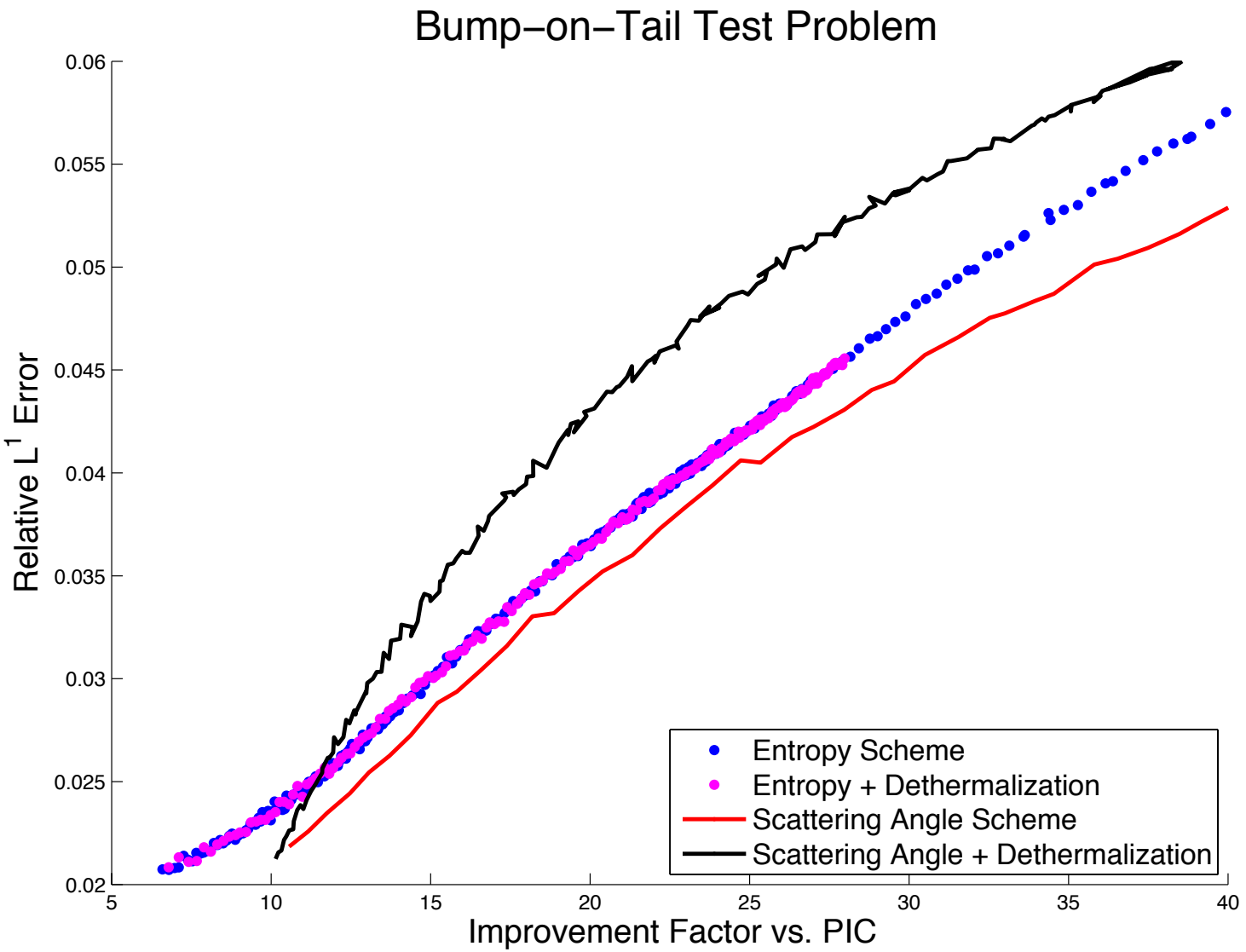
First, fix a value $H_c > 0$

- Simulate collisions using Monte-Carlo algorithm for Fokker-Planck equation (see Takizuka-Abe or Nanbu). Sample particles from fluid portion of scheme where necessary, assigning them $T_p = T_m$, $\vec{u}_p = \vec{u}_m$
- Evolve $\vec{u}_p$ and T_p according to relevant ODEs, using parameters of collision partner as input
- Loop over all kinetic particles: thermalize if $H_{rel} < H_c$
- Loop over all sampled fluid particles: dethermalize if $H_{rel} \geq H_c$

Bump-on-Tail


t=0
t = 1.2t_FP
t = 2.4t_FP
t = 3.6t_FP
t = 6t_FP
t = 11t_FP
f
v_x
0.2
0.15
0.1
0.05
0
−1
−0.5
0
0.5
1

Scheme Comparison



Hybrid Scheme: Pros and Cons

- Pros
  - Efficient and accurate
  - Captures what's really meant by "thermalization"
- Cons
  - Extra computational load
  - Generalization to spatial inhomogeneities is nontrivial

Langevin Method for Collisions

Near Equilibrium

Linear Landau-Fokker-Planck (LFP) equation:

$$\partial_t f = \frac{1}{\mathrm{Kn}} C(M, f) \tag{9}$$

with linear LFP collision operator

$$C(M, f) \equiv -\frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{F} f + \frac{\partial^2}{\partial \mathbf{v} \partial \mathbf{v}} : \mathbf{D}^2 f \tag{10}$$

in which $F = \mathbf{A}(M)$ and $D^2 = \mathbf{B}(M)$.

Langevin Formulation

Linear LFP equation for $f(v,t)$ is in exact correspondence with the Langevin equation (SDE) for $v(t)$

$$dv_i = F_i\,dt + D_{ij}\,dW_j, \tag{11}$$

where f is probability density of $\mathbf{v}$ and i, j are component indices

- $W = W(t)$ is Brownian motion in velocity
- dW is white noise in velocity
- Direct extension to spatial dependence
- Valid for nonlinear LFP, if $\mathbf{F}$ and $\mathbf{D}$ are updated as needed

Objective is an average of f:

$$\frac{1}{\rho}\int P(\mathbf{v})f(\mathbf{v},t)\,d\mathbf{v} \equiv \mathbb{E}[P(\mathbf{v}(t))] \tag{12}$$

Discretization of SDEs

Euler-Maruyama discretization in time:

$$v_{i,n+1} = v_{i,n} + F_{i,n}\Delta t + D_{ij,n}\,\Delta W_{j,n}, \tag{13}$$

$$\Delta \mathbf{W}_n = \mathbf{W}_{n+1} - \mathbf{W}_n \tag{14}$$

in which $v_{i,n} = v_i(t_n)$ and $\mathbf{F}_n = \mathbf{F}(\mathbf{v}_n)$

- Choose N Brownian paths to get N values of $P(\mathbf{v}(T))$
- Average to approximate $\mathbb{E}[P(\mathbf{v}(T))]$

Computational cost vs. Error ε:

- Statistical error is $O(N^{-1/2})$
- Δt error is $O(\Delta t)$, since $\Delta W = O(\sqrt{\Delta t})$ and random
- Optimal choice is $\varepsilon = N^{-1/2} = \Delta t$
- Cost $= N\Delta t^{-1} = \varepsilon^{-3}$

Higher Order Methods for SDEs

Higher Order Discretization

Milstein discretization in time:

$$v_{i,n+1} = v_{i,n} + F_{i,n}\Delta t + D_{ij,n}\,\Delta W_{j,n} + G_{ijk,n} I_{jk,n} \quad (15)$$

$$I_{jk,n} = \int_0^{\Delta t} \int_0^{s} dW_j(t_n + s')dW_k(t_n + s) \quad (16)$$

in which $v_{i,n} = v_i(t_n)$ and $\mathbf{F}_n = \mathbf{F}(\mathbf{v}_n)$

- **G** depends on **D** and its derivatives

- Milstein is tractable in 1D
  - Only requires diagonal term $I_{11} = ((\Delta W_1)^2 - \Delta t)/2$
- Milstein is intractable in 3D and higher
  - Off-diagonal I_{jk}'s involve "Levy areas" which are intractable
- Milstein is tractable in 2D
  - Special methods for calculating a single Levy area[6]

[6] Dimits et. al., JCP, 2013

Approximation of Milstein in 2D

Off-diagonal Milstein term includes Levy area L_{12}:

$$I_{12} = \frac{1}{2}\Delta W_1 \Delta W_2 + \frac{1}{2} L_{12} \quad (17)$$

$$L_{12} = \int_0^{\Delta t} \int_0^s dW_1(s')dW_2(s) - dW_2(s')dW_1(s) \quad (18)$$

Requires conditional probability distribution function

$$P(L_{12}|\Delta W_1, \Delta W_2) = \hat{P}(L_{12}|\sqrt{\Delta W_1^2 + \Delta W_2^2})^7 \quad (19)$$

- Dimits found a simple approximation of $\hat{P}(L_{12}|R_{12})$[8]
  - Numerical values given through a 1D lookup table
  - Related to earlier work[9]
- Generalization to $d > 2$ is possible but difficult

[7] Levy, 2nd Berkeley Symp Pob Stat, 1951
[8] Dimits et. al., JCP, 2013
[9] Gaines & Lyons , SIAM J Appl Math, 1997

Weak and Strong Convergence

Weak convergence of time discretization:

$$|\mathbb{E}[P(\mathbf{v})] - \mathbb{E}[P(\mathbf{v}_{\Delta t})]| = O(\Delta t) \quad \text{for Euler-Maruyama} \tag{20}$$
$$|\mathbb{E}[P(\mathbf{v})] - \mathbb{E}[P(\mathbf{v}_{\Delta t})]| = O(\Delta t) \quad \text{for Milstein} \tag{21}$$

- Weak convergence implies convergence of distributions
- Milstein is no better than Euler-Maruyama in weak sense

Strong convergence of time discretization:

$$\mathbb{E}[|\mathbf{v} - \mathbf{v}_{\Delta t}|] = O(\sqrt{\Delta t}) \quad \text{for Euler-Maruyama} \tag{22}$$
$$\mathbb{E}[|\mathbf{v} - \mathbf{v}_{\Delta t}|] = O(\Delta t) \quad \text{for Milstein} \tag{23}$$

- Strong convergence implies convergence for each realization
- Milstein is better than Euler-Maruyama in strong sense

Usefulness of Milstein

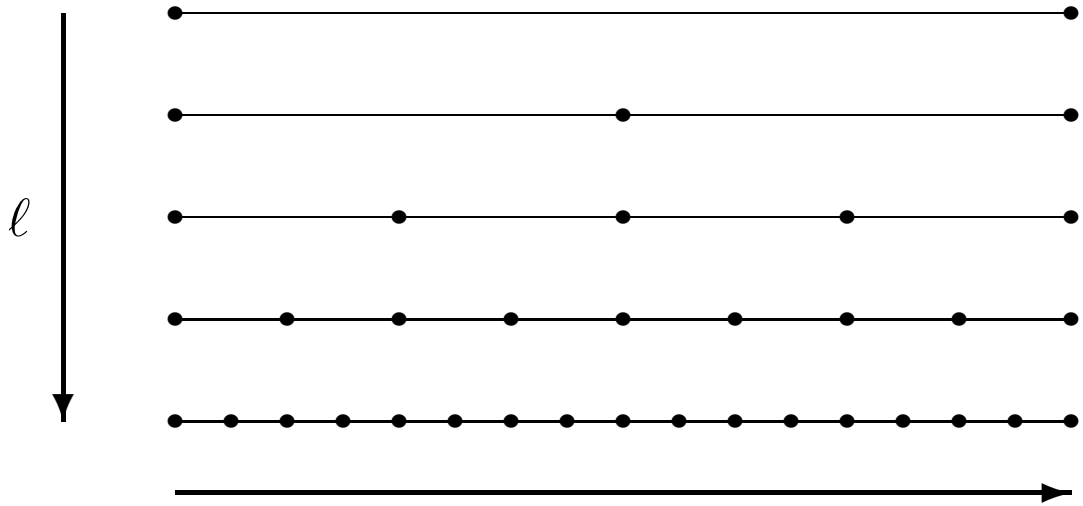
- Monte Carlo mostly aimed at computation of $\mathbb{E}[P(\mathbf{v})]$
  - Milstein offers no advantage over Euler-Maruyama
- Multilevel Monte Carlo (MLMC) leverages strong convergence to accelerate computation of $\mathbb{E}[P(\mathbf{v})]$ [10]
  - Milstein superior to Euler-Maruyama
  - Previous uses of MLMC mostly confined to finance
  - Our application of MLMC to plasma simulation is its first use for SDEs from physics

[10] Giles, *Operations Research*, 56(3):607, 2008

Multilevel Monte Carlo for Langevin Method

MLMC Basics

- Introduce time step levels, $\Delta t_\ell = T2^{-\ell}$, for $\ell = 0, ..., L$



- Let $P_\ell = P(\mathbf{v}_{\Delta t_\ell})$. Then

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{\ell=1}^{L} \mathbb{E}\left[P_\ell - P_{\ell-1}\right] \tag{24}$$

- When computed using same Brownian path, the variance of $(P_\ell - P_{\ell-1})$ is $O(\text{strong error})^2$

MLMC Scaling

Optimal number of samples used to compute each $\mathbb{E}[P_\ell - P_{\ell-1}]$, constrained by RMSE $< \varepsilon$. The complexity now scales like[11]

$$\text{Cost} = \begin{cases} O\left(\varepsilon^{-2}(\log \varepsilon)^2\right) & \text{for Euler-Maruyama} \\ O\left(\varepsilon^{-2}\right) & \text{for Milstein} \end{cases} \tag{25}$$

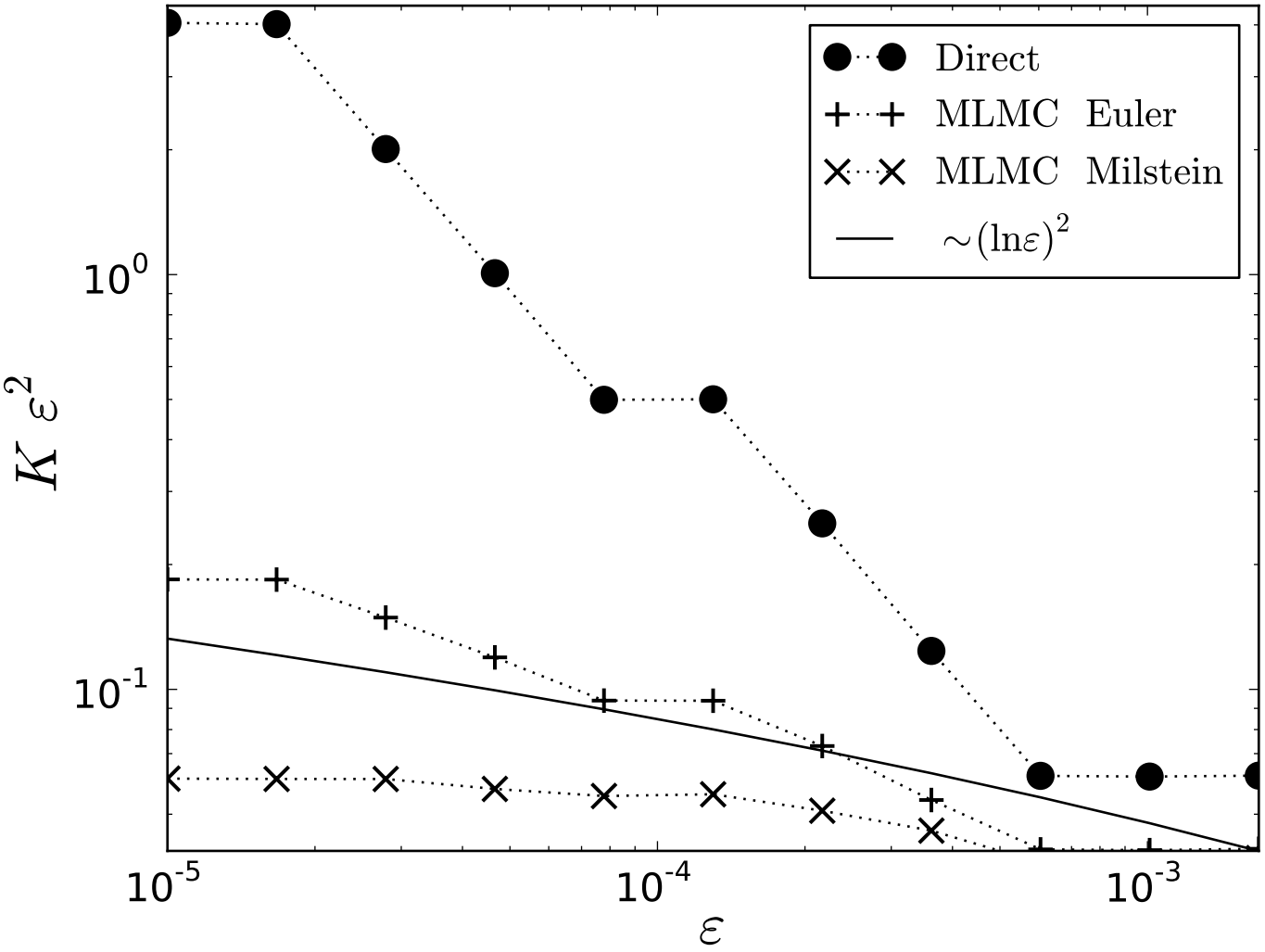
Notes:

- MLMC-Euler-Maruyama scales better than standard MC
- MLMC-Milstein is even better
  - Restricted to $d = 1, 2$ due to difficulty with Levy areas
- $O\left(\varepsilon^{-2}\right)$ scaling is possible without Milstein, using antithetic sampling method [12]

[11] Giles, *Operations Research*, 56(3):607, 2008

[12] Giles & Szpruch, *arXiv:1202.6283*, 2012

A Sample Plasma Problem



Rosin, Ricketson, et. al., submitted to JCP, 2013

Improving on MLMC through Antithetic Sampling

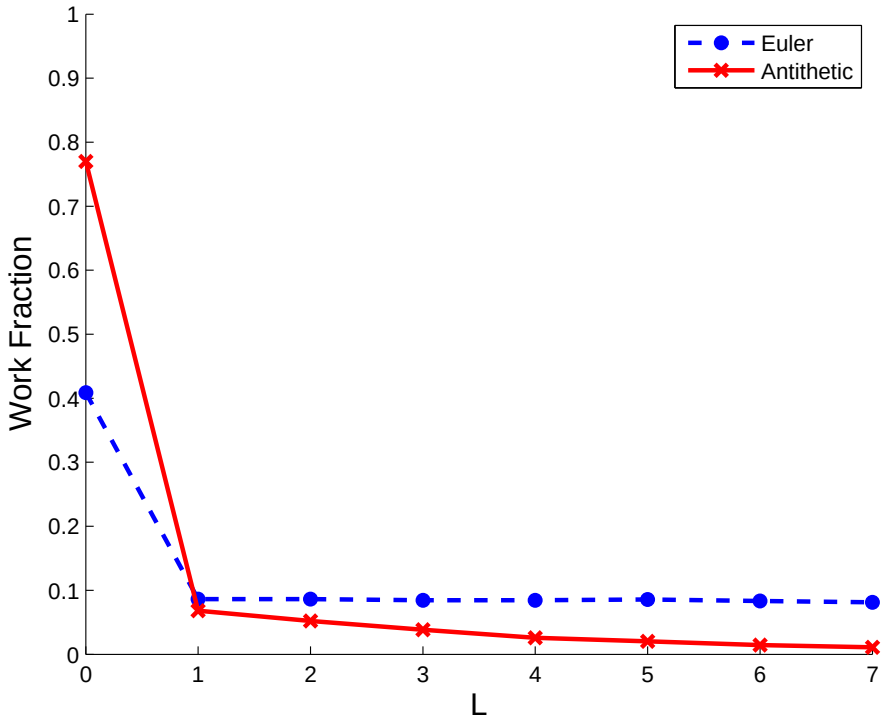
Antithetic sampling is a Monte Carlo variance reduction method

- For MC estimation of $E[f(x)]$ with normal random variable x
  - Standard estimator is $N^{-1}\sum f(x_i)$
  - Antithetic estimator is $(2N)^{-1}\sum (f(x_i)+f(-x_i))$
- Antithetic sampling for Milstein does not eliminate Levy areas
- Antithetic sampling for MLMC-Milstein achieves $O\left(\varepsilon^{-2}\right)$ without Levy areas! [13]

[13] Giles & Szpruch, *arXiv:1202.6283*, 2012

MLMC with Antithetic Sampling

Avoid use of Levy areas through antithetic sampling



Dominant cost is in computing $\mathbb{E}[P_0]$, since there is no variance reduction at this level.

Improving on MLMC through Ito Linearization

Add an SDE for P

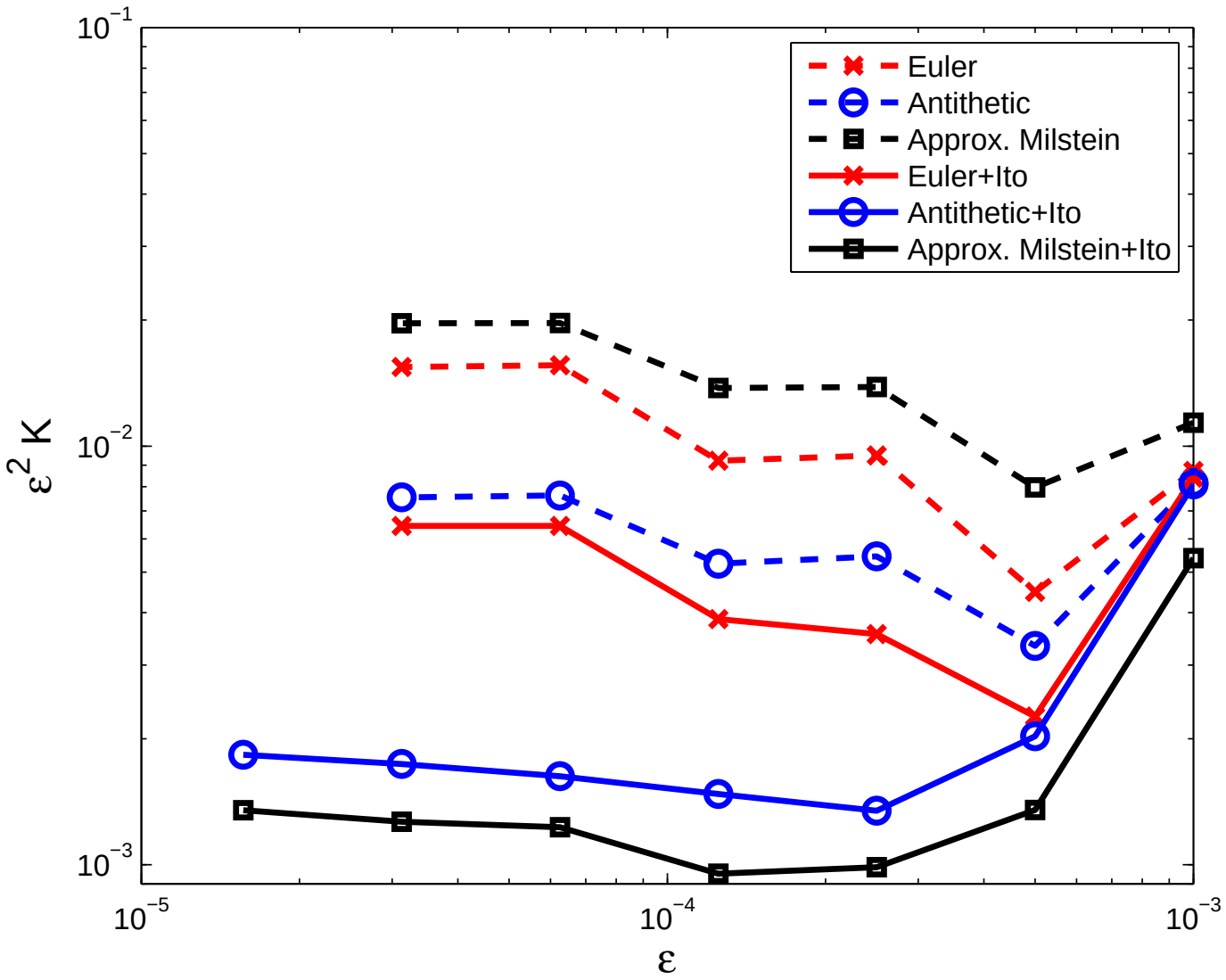
$$dv_i = F_i\,dt + D_{ij}\,dW_j, \quad i \le d \tag{26}$$

$$d\tilde{P} = \left(F_i P_{v_i} + \frac{1}{2} D_{ij} D_{kj} P_{v_i v_k}\right) dt + D_{ij} P_{v_i}\,dW_j \tag{27}$$

- Benefits:
  - Dominant term $\mathbb{E}[\tilde{P}_0]$ is easy to compute without error![14]
    - $\mathbb{E}[\tilde{P}_0]$ is linear in $\tilde{P}_0$, so that:
    - dW contribution to $\mathbb{E}[\tilde{P}_0]$ is 0
    - dt contribution to $\mathbb{E}[\tilde{P}_0]$ depends only on initial data
- Drawbacks:
  - One additional dependent variable
  - Limited to $P \in C^2$

[14] Ricketson, preprint, 2013

Computational Results



Conclusions for Langevin Methods

- Higher order SDE solvers - Milstein or antithetic sampling - are useful for MLMC
- MLMC accelerates collisional simulation in the Langevin formulation, up to 2 orders of magnitude
- Additional order of magnitude speed-up possible when Ito Linearization is applicable
- Generalization needed for applications with spatial dependence

Summary: Computational Complexity of Monte Carlo

Two methods for Monte Carlo simulation of collisions:

- Binary: Takizuka-Abe '77, Nanbu '97
- Langevin: Jones '96, Manheimer '97

Computation cost to achieve RMS error of size ε:

- Binary: $O(\varepsilon^{-3})$ *at best*, $O(\varepsilon^{-4})$ *at worst*[15]
- Langevin: $O(\varepsilon^{-3})$

The **Hybrid Method** reduces the computational cost of the binary collision method.

Multilevel Monte Carlo reduces the computational cost of the Langevin formulation to $O(\varepsilon^{-2}(\log \varepsilon)^2)$ or even $O(\varepsilon^{-2})$.

[15] Bobylev & Potapenko, J Comp Phys, 2013