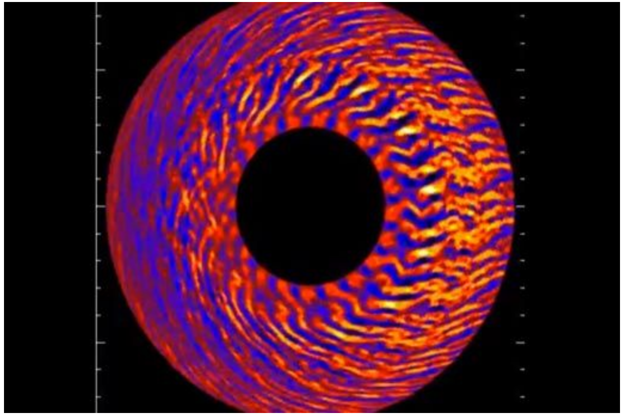
Parallelization and Adaptive Refinement of Velocity Space in Plasma Physics Simulations

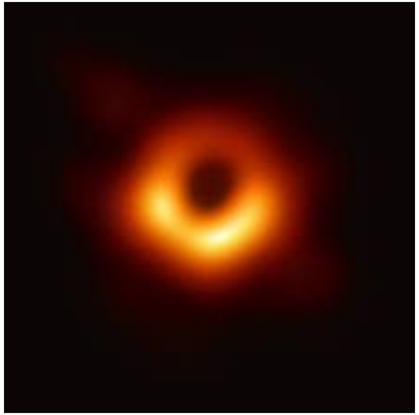
18.337
Vincent Fan, Luka Govedic, Lucas Shoji, Alex Velberg

Overview and Motivation

- Plasma simulations: useful in many ways,



Turbulent transport in fusion reactors



Accretion disks

Predicting space weather

Overview and Motivation

- Plasma simulations: useful in many ways, but hard in most practical scales.
  - Many applications: Fusion energy, Astrophysics, space weather
  - Extremely multiscale (in both space and time), nonlinear physics
- Velocity moments approach to solve the Vlaslov equations
- Existing code: Viriato, semi-implicit scheme
- Proposed improvements to the code: Further parallelization, reduce FFT calls and adaptive Hermite refinement
  - Adaptive resolution of fine-scale physics in velocity space
- Current state of the implementation: Julia and C++ implementations

Plasma Physics Background

- A first principles description of a plasma requires tracking particles in 3D-3V phase space via the Vlasov-Maxwell system of equations:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + q\left(E + \mathbf{v} \times \mathbf{B}\right) \cdot \nabla_v f = C(f)$$

- For realistically large systems of particles, this presents a significant computational hurdle.
- To reduce dimensionality, a common approach is to take velocity moments of the Vlasov eq

$$\int_{-\infty}^{\infty} \mathbf{v}^m F d\mathbf{v}$$

  - In the limit of infinitely many moments, the full physics can be described, but we can truncate the hierarchy by making some physics assumptions
- **By increasing the number of velocity moments used, can achieve better description of kinetic effects**

Plasma Physics Background

- Projecting onto a Hermite polynomial basis allows us to write down generalized equations for the evolution of the m-th moment, while preserving physical properties.
    - $$\partial_t g_m = f(g_{m-1}, g_m, g_{m+1})$$
- Think of higher order Hermite moments as describing the evolution of finer and finer scale structures in velocity space -> "spectral" representation of velocity
- Recent work on plasma turbulence suggests that places where many moments are required to accurately represent physics are highly localized.
    - **May be able to reduce computation by adaptively refining the number of moments**

Existing Code: Viriato

- Fortran code to solve a reduced version of the Vlasov equation by decomposing it in Hermite moments.
- Parallelized by tiling volume (message passing), but not in Hermite moments.
- Pseudospectral: Calculate derivatives in Fourier space for accuracy, but transforms back to real space to avoid convolutions: results in many FFT calls
- Custom semi-implicit time integration scheme

Semi-Implicit Scheme

Predictor Step

$$n_e^{n+1,*} = n_e^n + \Delta t\,\mathcal{N}(n_e^n, A_\parallel^n),$$

$$\begin{aligned} A_\parallel^{n+1,*} &= e^{-D_\eta \Delta t} A_\parallel^n + \left(1 - e^{-D_\eta \Delta t}\right) A_{\parallel,eq} \\ &\quad + \frac{\Delta t}{2}\,\frac{1 + e^{-D_\eta \Delta t}}{1 + k_\perp^2 d_e^2}\,\mathcal{A}(n_e^n, A_\parallel^n, g_2^n), \end{aligned}$$

$$g_2^{n+1,*} = g_2^n + \Delta t\,\mathcal{G}_2(n_e^n, A_\parallel^n, g_2^n, g_3^n),$$

$$\begin{aligned} g_m^{n+1,*} &= e^{-m\nu_{ei}\Delta t} g_m^n \\ &\quad + \frac{\Delta t}{2}\left(1 + e^{-m\nu_{ei}\Delta t}\right) \mathcal{G}_m(n_e^n, A_\parallel^n, g_{m-1}^n, g_m^n, g_{m+1}^n), \end{aligned}$$

Corrector Step

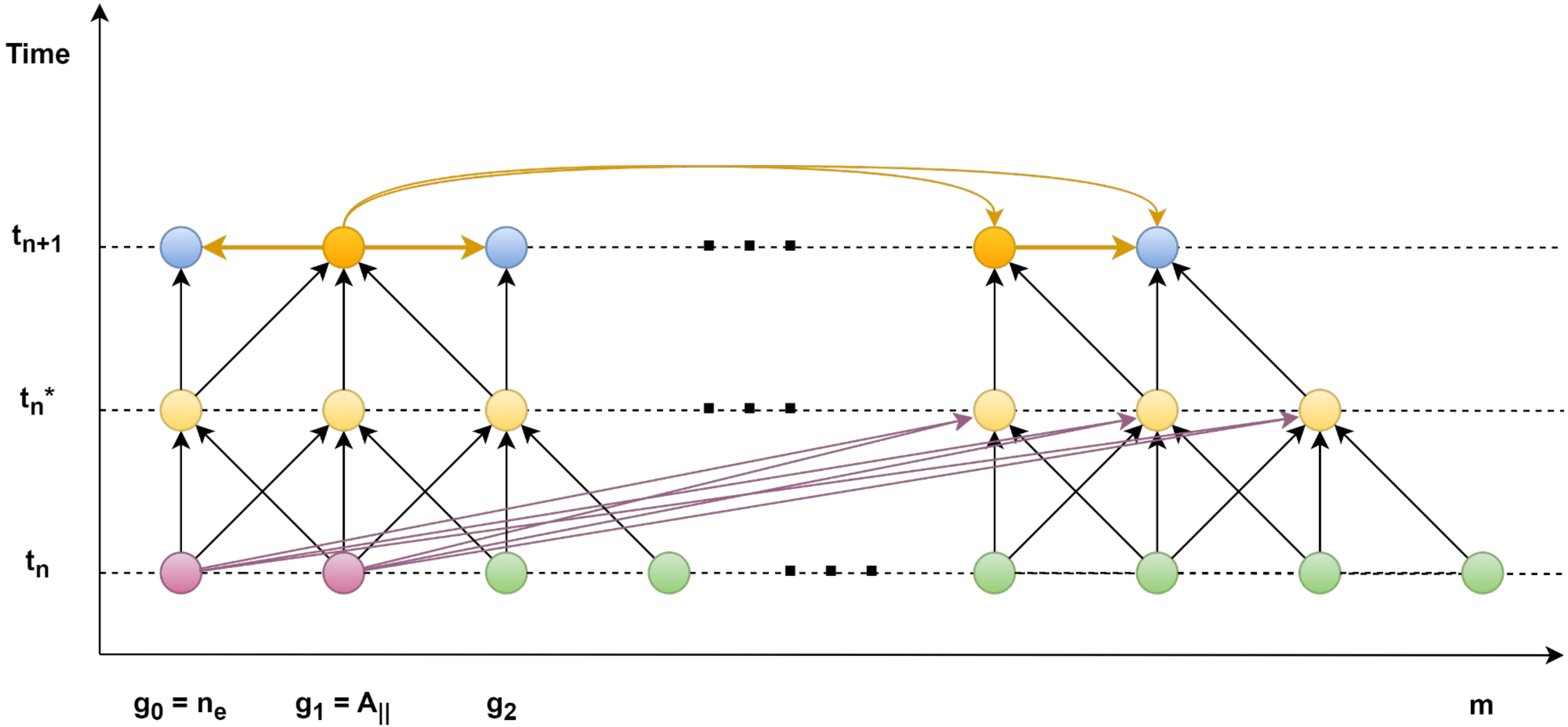
$$\begin{aligned} A_\parallel^{n+1,p+1} &= e^{-D_\eta \Delta t} A_\parallel^n + \left(1 - e^{-D_\eta \Delta t}\right) A_{\parallel,eq} \\ &\quad + \frac{\Delta t}{2}\,\frac{e^{-D_\eta \Delta t}}{1 + k_\perp^2 d_e^2}\,\mathcal{A}(n_e^n, A_\parallel^n, g_2^n) \\ &\quad + \frac{\Delta t}{2}\,\frac{1}{1 + k_\perp^2 d_e^2}\,\mathcal{A}(n_e^{n+1,p}, A_\parallel^{n+1,p}, g_2^{n+1,p}), \end{aligned} \tag{52}$$

$$n_e^{n+1,p+1} = n_e^n + \frac{\Delta t}{2}\mathcal{N}\left(n_e^n, A_\parallel^n\right) + \frac{\Delta t}{2}\mathcal{N}\left(n_e^{n+1,p}, A_\parallel^{n+1,p+1}\right), \tag{53}$$

$$\begin{aligned} g_2^{n+1,p+1} &= g_2^n + \frac{\Delta t}{2}\mathcal{G}_2\left(n_e^n, A_\parallel^n, g_2^n, g_3^n\right) \\ &\quad + \frac{\Delta t}{2}\mathcal{G}_2\left(n_e^{n+1,p+1}, A_\parallel^{n+1,p+1}, g_2^{n+1,p}, g_3^{n+1,p}\right), \end{aligned} \tag{54}$$

$$\begin{aligned} g_m^{n+1,p+1} &= e^{-m\nu_{ei}\Delta t} g_m^n + \frac{\Delta t}{2} e^{-m\nu_{ei}\Delta t}\mathcal{G}_m\left(n_e^n, A_\parallel^n, g_{m-1}^n, g_m^n, g_{m+1}^n\right) \\ &\quad + \frac{\Delta t}{2}\mathcal{G}_m\left(n_e^{n+1,p+1}, A_\parallel^{n+1,p+1}, g_{m-1}^{n+1,p+1}, g_m^{n+1,p}, g_{m+1}^{n+1,p}\right). \end{aligned} \tag{55}$$

Semi-Implicit Scheme Dependencies

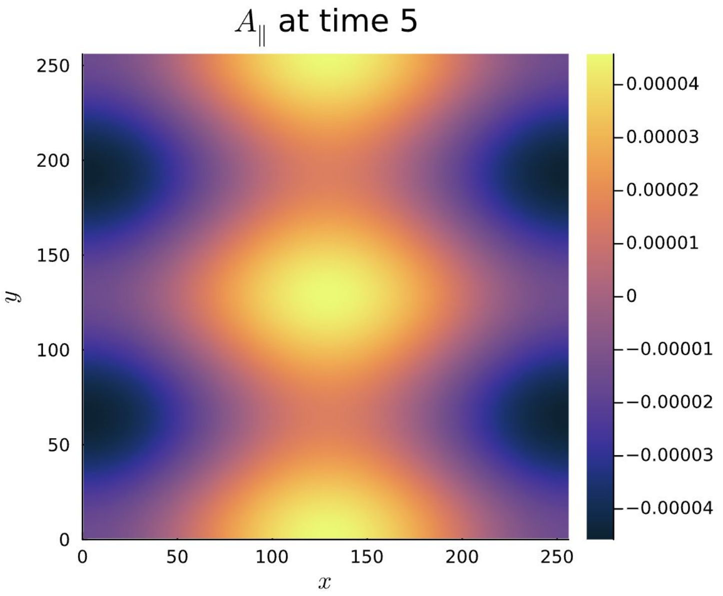


Proposed Improvements

- Most of the time spent performing FFT and inverse FFT operations
  - Reducing the number of FFTs should yield the most significant performance gains
  - Proposed improvement : reduce number of FFT calls by avoiding recomputing iFFTs by reusing subexpressions
- Viriato is parallelized using MPI by tiling the physical volume, but there is no parallelization in velocity space
  - Introduce fork-join shared-memory parallelism for better scalability (improvement on MPI)
  - Leverage sparse dependency graph to parallelize in velocity space
  - Leverage locality of moment equations to improve cache locality
- Adaptive moment scheme
  - Reduce computation by only computing high number of moments in real-space locations where higher physics fidelity is strictly necessary

Current state of project

- 2D implementation of Viriato in Julia from scratch completed
  - Code runs, and produces sensible but incorrect output
  - Debugging in progress…
- C++ implementation
  - Use efficient work-stealing from OpenCilk
  - Improve cache-locality
- Next Steps
  - Benchmark against Viriato.F90
  - Profile
  - Implement optimizations



Code output for Orszag-Tang Vortex Equilibrium after a few timesteps.

References

N. F. Loureiro *et al.*, “Viriato: a Fourier-Hermite spectral code for strongly magnetised fluid-kinetic plasma dynamics,” *Computer Physics Communications*, vol. 206, pp. 45–63, Sep. 2016, doi: 10.1016/j.cpc.2016.05.004.

Zhou et al, 2022, arXiv:2208.02441