Algorithms for High-Performance Global Gyrokinetic PIC Simulations of ITER-size Plasmas

Bei Wang[1] Stephane Ethier[2] William Tang[1,2]
Bruce Scott[3] Khaled Ibrahim[5] Kamesh Madduri[4]
Sam Williams[5] Leonid Oliker [5]

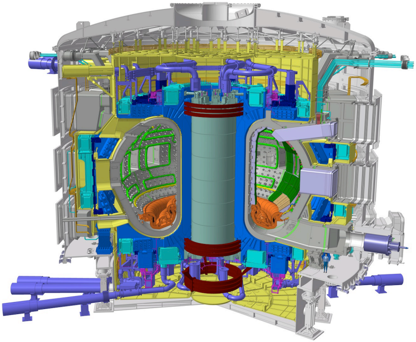
[1]Princeton Institute for Computational Science and Engineering, Princeton University
[2]Princeton Plasma Physics Laboratory, Princeton University
[3]Max-planck Institute for Plasma Physics
[4]Penn State University
[5]Lawrence Berkeley National Laboratory

Workshop on Numerical Methods for the Kinetic Equations of Plasmas

Max-Planck-Institute for Plasma Physics, Sep 3, 2013

Grand Challenge: Global Simulation of ITER

- ✓ ITER is extremely large compared to current experiments, eg., 8x larger in volume compared with JET
- ✓ We need advanced mathematical model, numerical methods and extremely large scale computing to predict its performance



Gyrokinetic Simulations: for studying turbulent transport

- Macroscopic Stability

  What limits the pressure in plasmas?

- Wave-particle interactions

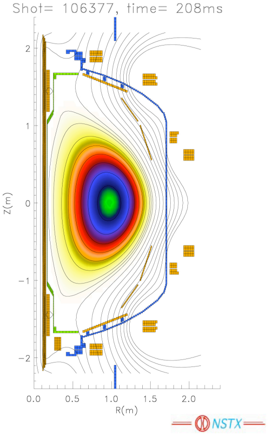
  How do particles and plasma waves interact?

- **Microturbulence and Transport**

  What causes plasma transport?

- Plasma-material Interactions

  How can high-temperature plasma and material surfaces co-exist?



Outline

Gyrokinetic Particle Simulation of Magnetically Confined Plasmas

- Gyrokinetic Vlasov-Poisson equations in toroidal geometry
- Numerical methods for Vlasov Poisson system
- Phase space remapping
- ITG simulation results

High-Performance GTC-P Code for ITER-size Simulation

- GTC-P code
- Parallelization
- Improve single node performance on multicore and manycore systems
- Performance results

Gyrokinetic Vlasov equation (torus)

- We consider kinetic ions and adiabatic electrons.
- The dynamics of the ions is described by the 5D (ψ, θ, ζ, $v_\parallel$, μ) gyrophase-averaged Vlasov equation in toroidal geometry:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{d\mathbf{R}}{dt} \cdot \frac{\partial f}{\partial \mathbf{R}} + \frac{dv_\parallel}{dt} \frac{\partial f}{\partial v_\parallel} = 0$$

$$\frac{d\mathbf{R}}{dt} = v_\parallel \hat{\mathbf{b}} + \mathbf{v}_d + \mathbf{v}_E$$

$$\frac{dv_\parallel}{dt} = -\mathbf{b}^* \cdot \left(\frac{v_\perp^2}{2} \frac{\partial}{\partial \mathbf{R}} \ln B + \frac{\partial \bar{\phi}}{\partial \mathbf{R}}\right), \quad \mathbf{b}^* = \hat{\mathbf{b}} + v_\parallel \hat{\mathbf{b}} \times \left(\hat{\mathbf{b}} \cdot \frac{\partial}{\partial \mathbf{R}}\right)\hat{\mathbf{b}}$$

$$\mathbf{v}_d = v_\parallel^2 \hat{\mathbf{b}} \times \left(\hat{\mathbf{b}} \cdot \frac{\partial}{\partial \mathbf{R}}\right)\hat{\mathbf{b}} + \frac{v_\perp^2}{2} \hat{\mathbf{b}} \times \frac{\partial}{\partial \mathbf{R}} \ln B, \quad \mathbf{v}_E = \mathbf{b} \times \frac{\partial \bar{\phi}}{\partial \mathbf{R}}$$

- The distribution function f is defined on **gyrocenter coordinates**.

Gyrokinetic Poisson equation (Lee, JCP, 87)

- The quasi-neutrality equation is

$$\cancelto{0}{\nabla^2\phi} + \frac{\tau}{\lambda_D^2}(\phi - \tilde{\phi}) = 4\pi e(\delta\bar{n}_i - \delta n_e)$$

where

$$\tilde{\phi}(\mathbf{x}) = \left\langle \int \bar{\phi}(\mathbf{R}) f_M(\mathbf{R})\delta(\mathbf{R} - \mathbf{x} + \rho)d\mathbf{R}d\mu dv_{\parallel} \right\rangle_{\varphi}$$

$$\bar{\phi}(\mathbf{R}) = \left\langle \int \bar{\phi}(\mathbf{x})\delta(\mathbf{R} - \mathbf{x} + \rho)d\mathbf{x} \right\rangle_{\varphi}$$

$$\frac{\delta\bar{n}_i(\mathbf{x})}{n_0} = \left\langle \int (f(\mathbf{R}) - f_{eq}(\mathbf{R}))\delta(\mathbf{R} - \mathbf{x} + \rho)d\mathbf{R}d\mu dv_{\parallel} \right\rangle_{\varphi}$$

$\tau = \frac{T_e}{T_i}$, λ_D electron Debye length, $\langle\rangle_{\varphi}$ gyroangle average

- Assume adiabatic electrons

$$\frac{\delta n_e}{n_0} = \frac{e}{T_e}(\phi - \langle\phi\rangle)$$

- The Poisson's equation is defined on **particle coordinates**.

Numerical methods for the Vlasov equation

- **Grid-based methods**: spectral methods (Flimas and Farrell, JCP, 94), semi-Lagrangian methods (Cheng and Knorr, JCP, 76; Sonnendrucker et al, JCP, 98; Nakamura and Yabe, CPC, 99), finite volume methods (Fijalkow, CPC, 99; Filbet, Sonnendrucker and Bertrand, JCP, 01; Colella et al, JCP, 11), finite element methods (Zaki, Gardner and Boyd, JCP, 88)
  - ∗ They have drawn much attention in the past decade thanks to increasing processing power
  - ∗ Advantage: Smooth representation of f
  - ∗ Disadvantage: **High dimensions** (up to 6) $\longrightarrow$ high computational cost (specifically memory)
- **Particle methods**,e.g.,the PIC method, usually preferred for high dimension
  - ∗ Advantages: Naturally adaptive, since particles only occupy spaces where the distribution function is not zero; simpler to implement, in particular in high dimensions; good scalability
  - ∗ Disadvantages: **Particle noise** $\longrightarrow$ difficulties to get precise results in some cases, for example, in simulating the problems with large dynamic ranges

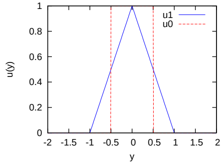
Particle in cell methods

- In particle methods, we approximate the distribution function by a collection of finite-size particles

$$f(\mathbf{x}, \mathbf{v}, t) \approx \sum_k q_k \delta_{\varepsilon_\mathbf{x}}(\mathbf{x} - \tilde{\mathbf{X}}_k(t))\delta_{\varepsilon_\mathbf{v}}(\mathbf{v} - \tilde{\mathbf{V}}_k(t))$$

$$\int_{-\infty}^{\infty} \delta_\varepsilon(y) dy = 1$$

$$\delta_\varepsilon(y) = \frac{1}{\varepsilon} u(\frac{y}{\varepsilon})$$



- At each time step, particles are transported along trajectories described by the equation of motion

$$\frac{df}{dt} = 0 \Rightarrow \frac{dq_k}{dt} = 0$$

$$\frac{d\tilde{\mathbf{X}}_k}{dt} = \tilde{\mathbf{V}}_k, \quad \frac{d\tilde{\mathbf{V}}_k}{dt} = \tilde{\mathbf{F}}_k$$

- the long range forces are usually solved on a grid

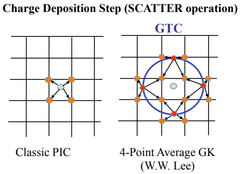
Charge assignment and field interpolation in gyrokinetic PIC method

- In 5D gyrokinetic Vlasov-Poisson system, the Vlasov equation is defined on guiding center coordinates and the Poisson's equations is defined on particle coordinates
- The coordinate transformation is approximated by 4-point average (Lee, JCP, 87)

Gyrokinetic PIC

$$\tilde{\rho}(\mathbf{x_j}) = \sum_k \sum_{p=1}^{p=4} \frac{q_k}{4\varepsilon_x} \mathbf{u}_1\left(\frac{\mathbf{x_j} - \tilde{\mathbf{X}}_p}{\varepsilon_x}\right)\delta(\tilde{\mathbf{R}}_k - \tilde{\mathbf{X}}_p + \rho_k)$$

$$\tilde{\mathbf{E}}(\tilde{\mathbf{R}}_k) = \sum_{p=1}^{p=4} \sum_{\mathbf{j}} \mathbf{E_j}\mathbf{u}_1\left(\frac{\mathbf{x_j} - \tilde{\mathbf{X}}_p}{\varepsilon_{\mathbf{x}}}\right)\delta(\tilde{\mathbf{R}}_k - \tilde{\mathbf{X}}_p + \rho_k))$$



What is particle noise? - The approach in vortex method

The numerical error introduced when evaluating the moments of the distribution function using particles in phase space and **the particle disorder induced by the numerical error**

Error analysis:

- consistency error + stability error (Cottet and Raviart, SIAM J. Numer. Anal., 84; Wang, Miller and Colella, SIAM J. Sci. Comput., 11):

$$\textit{error} \propto \text{consistency error} \times (\exp(at) - 1)$$

where $a = \|\frac{\partial E}{\partial x}\|_{L^\infty(\mathbb{R})}$ is a physical parameter.

Error Analysis of the PIC method for the VP system - The approach in vortex method

The charge density error is

$$
\begin{aligned}
|\rho(x,t)-\tilde{\rho}(x,t)| &= |\rho(x,t)-\sum_k q_k\delta_{\varepsilon_x}(x-\tilde{X}_k(t))| \\
&\le \underbrace{\left|\rho(x,t)-\int_{\mathbb{R}}\rho(y,t)\delta_{\varepsilon_x}(x-y)dy\right|}_{\textbf{moment error}:\, e_m(x,t)\propto\varepsilon_x^2} \\
&+ \underbrace{\left|\int_{\mathbb{R}}\rho(y,t)\delta_{\varepsilon_x}(x-y)dy-\sum_k q_k\delta_{\varepsilon_x}(x-X_k(t))\right|}_{\textbf{discretization error}:\, e_d(x,t)\propto{\varepsilon_x}^2\left(\frac{h_x}{\varepsilon_x}\right)^2} \\
&+ \underbrace{\left|\sum_k q_k\delta_{\varepsilon_x}(x-X_k(t))-\sum_k q_k\delta_{\varepsilon_x}(x-\tilde{X}_k(t))\right|}_{\textbf{stability error}:\, e_s(x,t)\propto\frac{1}{\varepsilon_x}\max_k|X_k-\tilde{X}_k|}
\end{aligned}
$$

$$
|E(x,t)-\tilde{E}(x,t)|\propto\left({\varepsilon_x}^2+{\varepsilon_x}^2\left(\frac{h_x}{\varepsilon_x}\right)^2+\max_k|\tilde{X}_k(t)-X_k(t)|\right)
$$

Low noise particle methods - The approach in vortex method

- **phase space remapping (coarse graining)**: remap the distorted charge distribution on regularized grid(s) in phase space and then create a new set of particle charges from the grids with regularized distribution
  $\Rightarrow$ **control exponential term**
  - Vortex methods: Cottet and Raviart, SIAM J. Numer. Anal., 84
  - Smoothed particle hydrodynamics: Koumoutsakos, JCP, 97
  - PIC for plasma physics: Denavit, JCP, 72; Vadlamani et al., CPC, 04; Parker and Chen, PoP, 07; Wang, Miller and Collela, SIAM J. Sci. Comp., 11, 12
- **W-stat or Krook-like operator** (Krommes, PoP, 99; Jolliet et al., PoP, 09, Villard et al, Plasm Phys and Contr Fusion, 10)

Phase space remapping (coarse graining): control noise in long runs

- In plasma physics, simple geometry:
  - Denavit, JCP, 72: pioneer work, uniform loading (on lattice)
  - Vadlamani et al., CPC, 04: first apply on δf method, uniform loading (on lattice)
  - Chen and Parker, PoP, 07: δf, random loading
  - Wang, Miller and Colella, SIAM J. Sci. Comput., 11,12: high order with AMR, uniform loading (on lattice)
- Apply remapping to global ITG simulations in 3D torus
- Start from the algorithm by Chen and Parker, PoP, 07 since it is easy to implement with randomized initial loading in GTC-P code
- On-going research: uniform loading (on lattice) with high order remapping on 3D torus geometry

What is particle noise? - The approach in Monte Carlo method

- **Monte Carlo estimate** (Aydemir, PoP, 93)

$$error \propto \frac{\sigma}{\sqrt{N}}$$

where σ is the variance of $g = \frac{f(z)}{p(z)}$, depending on the distribution function and the particle sampling.

- **Perturbative methods, such as the δf method** (Dimits and Lee, JCP, 93; Parker and Lee, PoP 93): discretize the perturbation with respect to a (local) Maxwellian in velocity space using particles

$$f = f_0 + \delta f, \quad \frac{df}{dt} = \frac{df_0}{dt} + \frac{d\delta f}{dt} = 0, \quad \frac{d\delta f}{dt} = -\frac{df_0}{dt}$$

$\Rightarrow$ **reduce the variance of g**, where $g = \frac{\delta f(z)}{p(z)}$.

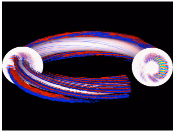
ITG Simulation, No remapping

- Apply remapping to global ITG simulations in 3D torus
- Start from the algorithm by Chen and Parker, PoP, 07 since it is easy to implement with randomized initial loading in GTC-P code

ITG Simulation, With remapping

Gyrokinetic Toroidal Code @ Princeton (GTC-P) (1)

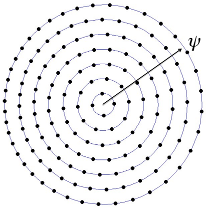
- δf PIC code solves 5D gyrokinetic equation in full global torus geometry



- The equilibrium magnetic geometry is described by a large aspect ratio analytical model of simplified toroidal magnetic field with a circular cross-section
- Kinetic ions and adiabatic electrons
- Takes into account all the toroidicity effects such as the curvature drift and multiple rational surfaces, but not the non-circular cross-section effects or the fully electromagnetic, non-adiabatic electron dynamics
- Uses magnetic coordinates and field line following grid on toroidal geometry

GTCP (2)

- The poloidal plane is discretized by unstructured grid



- The gradient operator is approximated by the second order finite difference method
- The Poisson equation is solved by a damped Jacobi iterative solver in which the damping parameter is chosen to favor the desired range of wavelengths for the fastest growing modes in the simulation of plasma turbulence (Lin and Lee, Physical Review E, 95)

History of Gyrokinetic Toroidal Code @ Princeton (GTC-P)

- Developed from Gyrokinetic Toroidal Code (GTC, Lin et al., Science, 98)
- Improve the parallel scalability with additional level of domain decomposition (Adams, Ethier and Wichmann, JoP Conf., 07; Ethier et al, VECPAR, 10)
- Benefit from computer science advances in deploying multi-threading capabilities to facilitate large-scale simulations on modern low memory per core systems (Madduri et al., SC11; Wang et al., SC13 (accepted))
- Included modern diagnosis tools by B. Scott, Max-Planck Institute of Plasma Physics
- Included coarse graining

GTC-P: Parallelization (1)

- 3 levels of parallelism in the original GTC:

  1d domain decomposition in toroidal dimensional
  particle decomposition in each toroidal domain
  loop-level parallelization with OpenMP

- Almost perfect scaling in terms of number of particles (Ethier, Tang and Lin, JoP Conf, 05)
- However, Massive grid memory footprint in system size scaling: difficulty for simulating large scale plasmas such as ITER (assume 100 particles per cell)

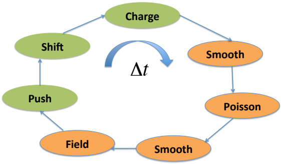
Problem Size	grid size	num. of particles in one toro. dom.
A ($a/\rho = 125$)	32,449 (2M)	3,235,896 (0.3G)
B ($a/\rho = 250$)	128,893 (8M)	12,943,584 (1.2G)
C ($a/\rho = 500$)	513,785 (32M)	51,774,336 (4.8G)
D ($a/\rho = 1000$)	2,051,567 (128M)	207,097,344 (19.2G)

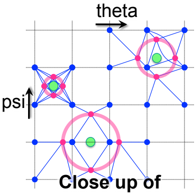
GTC-P: Parallelization (2)

- Introduce the key additional level of domain decomposition in the radial dimension-which is essential for efficiently carrying out ITER size simulations (Adams, Ethier and Wichmann, JoP Conf., 07; Ethier et al, VECPAR, 10)
- GTC-P includes 4 levels of parallelism, thus can easily scale to the largest supercomputer systems world wide

Computational Kernels in GTC-P

- Charge: Deposited charge from particles to the grid using the 4-point gyro-averaging method
- Poisson/Field/Smooth: solves the gyrokinetic Poisson equation, computes an electric field and smooths the charge and potential with a filter on the grids
- Push: interpolates the electric field onto particles and advances particle phase space position
- Shift: In distributed memory environment, moves particles between processes





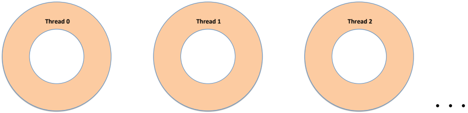
Close up of Poloidal Plane

Multi-core Optimization Challenges

- Particle related subroutines (Charge, Push and Shift) dominates execution time (num. of particles is usually 100x than num. of grid points)
- **low computation intensity** for charge, push and shift have very
- **Random access nature (gatter/scatter)** of charge and push
- In addition to data locality, charge involves **data dependency challenge** in multithreading environment

Optimization Strategies

- **Managing data hazard**: Static replication of grid for charge deposition: no costly synchronization required



$\rightarrow$ Large number of grid replicas is only affordable with radial domain decomposition for ITER-size simulation

- **Improving locality**: Particle Binning (Kamesh et al, SC, 11)
- **Further improving locality**: A multilevel binning algorithm to improve locality and reduce data conflict: Binning the gyro-center of the particles periodically; Binning the four points of the gyro-particles at every time step (Wang et al, SC, 12)

Miscellaneous optimizations

- A structure of array (SOA) data structure for particle array
- Aligned memory allocation to facilitate use of SIMD intrinsics
- Explicit SIMDization (via intrinsics)
- Process pinning: NUMA-aware memory allocation relying on first-touch policy
- Loop fusion to improve computing intensity
- Processor placement in the toroidal dimension first

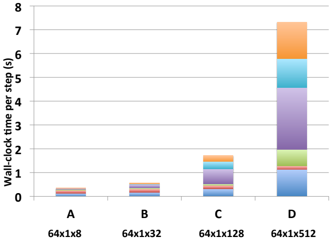
GTC-P numerical settings for different plasma sizes

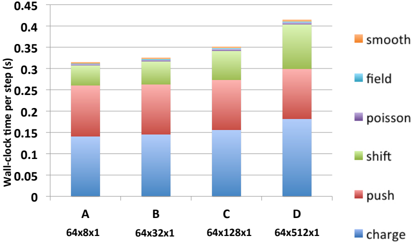
Grid Size	A	B	C	D
mpsi	90	180	360	720
mthetamax	640	1280	2560	5120
mgrid (grid points per plane)	32449	128893	513785	2051567
chargei grid (MB)†	0.5	1.97	7.84	31.30
evector grid (MB)†	1.49	5.90	23.52	93.91
Total particles *micell*=100 (GB)	0.29	1.16	4.64	18.56

B: D3D size, C: JET size, **D: ITER size**

- The grid and particle memory usage is for one toroidal section
- A 3D torus usually consists of 32 or 64 toroidal sections
- Moving to a plasma of one size larger, the grid size and the number of particles increase 4x

Weak scaling comparison of GTC and GTC-P



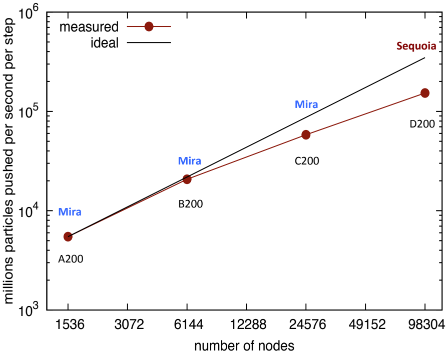


Left: Weak scaling of GTC

Right: Weak scaling of GTC-P

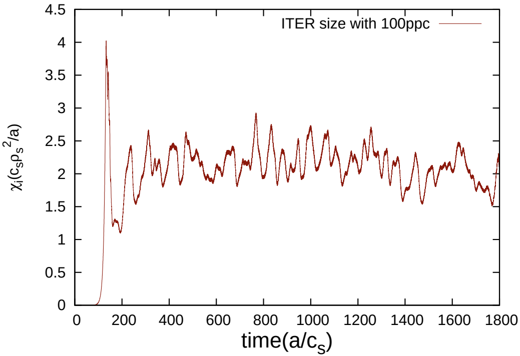
- ntoroidald x nradiald x npe_radiald
- Without radial domain decomposition, the time spent on grid based subroutines increases dramatically
- The performance boost is **18x** by turning on the additional domain decomposition

Compute power plot of GTC-P on BG/Q



An award of computer time was provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program. This research used resources of the ALCF. We would also like to thank the NNSA for access to the Sequoia system at LLNL.

Study ITG driven turbulence spreading with the ultrafast GTCP code on BG/Q



- Now we can simulate ITER size (130 million grid points) plasma with 100 ppc (13 billion particles) for 30k time steps in 6.5 hours on 8192 Mira nodes.

Conclusion

- Control the stability error in particle method is necessary for long time simulation
- Radial domain decomposition is important for large size plasma simulations, eg., ITER
- HPC can lead to a significant return in terms of "time to solution "

  For example, the new optimized GTC-P code can deliver up to 5x speed up compared with the previous version on BG/Q system

Current and future work

- Study turbulence spreading and size scaling up to ITER size with high resolution simulations
- Develop advanced phase space remapping for long time simulations with gyrokinetic δf particle in cell method
- Develop full-f, electron dynamic capabilities in GTC-P
- Develop efficient diagnosis tool in GTC-P (collaborated with B. Scott)
- Develop efficient parallel I/O in GTC-P with ADIOS (collaborated with S. Klasky)

Thank you!
Questions?