jh190065-NAHI

Yuuichi ASAHI (National Institutes for Quantum and Radiological Science and Technology)

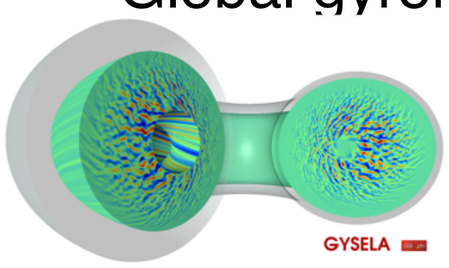
Modernizing and accelerating fusion plasma turbulence codes targeting exa-scale systems

Background: Plasma turbulence simulation

- First principle model for plasma turbulence transport → Predict the fusion reactor performance
- Resolving **machine scale (m)** with **particle scale mesh (cm)** → **5D** (3D space + **2D velocity space**)
  Velocity space structure due to collision-less features of plasma $100^5 \sim 10^{10}$ stencil computation
- Concerning kinetic electrons, more computational power needed → **GPU** porting completed in jh170020

Physical research plan: Kinetic electron model and particle transport

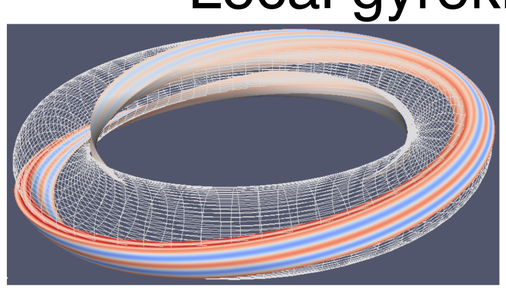
Global gyrokinetic code: GYSELA



- **Plasma profile evolution**
- **Simplified** physical models (hybrid electron model)
- Computational cost **High**

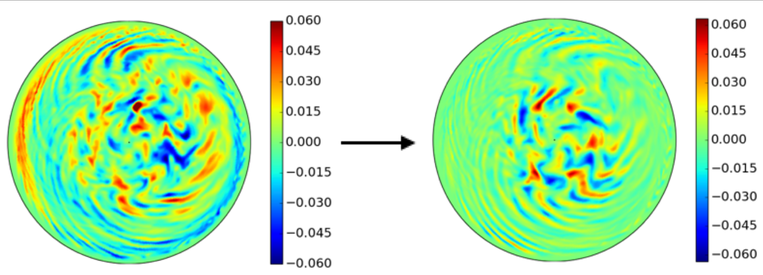
[V. Grandirard et al., CPC(2016)]

Local gyrokinetic code: GKV

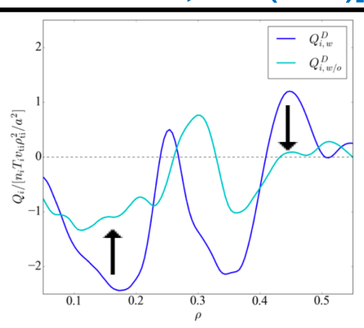


- **Fixed** plasma profile
- **Advanced** physical models (full kinetic electron)
- Computational cost **Low**

[T.-H. Watanabe et al., NF(2006)]

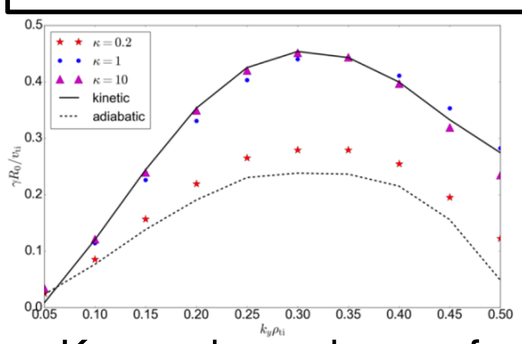


Filter out poloidal asymmetry



[Y. Asahi et al., PPCF (2019)]

→ Collisional energy transport affected by **asymmetry**



Kappa dependence of Linear growth rate

- Larger growth rate with non-adiabatic electron $\kappa > 0$
- Small kappa ~ adiabatic
  Large kappa ~ kinetic

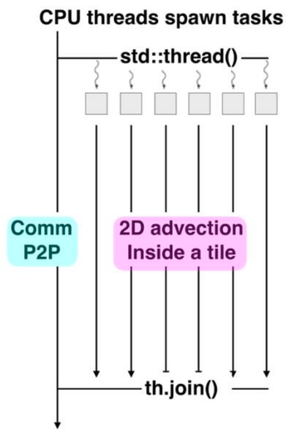
→ **Hybrid electron model** works appropriately

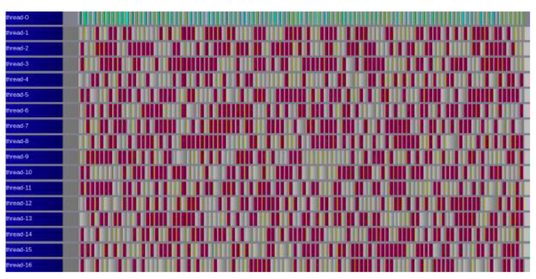
- Complete verification tests of hybrid kinetic electron model in GYSELA
- Analyze the impact of poloidal asymmetry on particle transport with GYSELA

HPC research plan: Task level parallelization + Portable framework

2D advection (1 task: comp/comm on 1 tile)

```
for (it=start; it<=end; it++)
  // Perform the advection for internal points
  threads[it-start]
    = std::thread(
        [&, it](){
          tile_internal(&(tiles[it]), it);
        });
for (it=start; it<=end; it++)
  // [HOST] Post send/recv from master thread
  ghost_sendrecv(tiles,ghosts, &mpi,it,nt);
// Task synchronization
for (std::thread &th : threads) th.join();
```

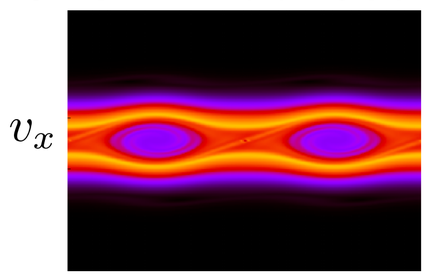




: tile_internal : idle

- Gantt chart on 32 KNL nodes.
  (monitor: komp †, visualize: vite††)
- Intuitive understanding of task execution order
  (e.g. idle time is the current issue)
- Large amounts of tasks are not ideal for GPUs (fine for CPUs)

FY2019: 4D advection (space 2D + velocity 2D)



- Vlasov: Semi-Lagrangian, Strang splitting
- Poisson: 2D FFT (Periodic boundary)

Porting cost with **Kokkos**†††

Task	Hours
Original (1000 lines, OpenMP) to Kokkos (GPU)	7-8 hours
Debugging	1 Week
cufft wrapper (200 lines, GPU) to fftw wrapper (OpenMP)	3-4 hours
Kokkos (GPU) to Kokkos (CPU)	**10 mins**

Performance

P100, Kokkos	0.155 [s]
BDW, Kokkos:	4.51 [s]
BDW, OpenMP:	1.81 [s]

- Same code works on CPU/GPU
- Test CPU suited optimizations

komp†: https://gitlab.inria.fr/openmp/libkomp
vite††: http://vite.gforge.inria.fr/index.php
kokkos†††:https://github.com/kokkos/kokkos

Collaborations: Kinetic electron + Task level Parallelization + AMR

French Group

Physics
- Turbulence profile interaction
- Impact of symmetry breaking

HPC
- **Task-level parallelization** in GYSELA prototype
- GYSELA prototype on GPUs

Japanese Group

Physics
- Develop kinetic **electron model**
- Characterize local transport

HPC
- **Task-level parallelization** in GKV code
- Testing AMR in mini-app

- Visiting CEA to discuss HPC and physical topics (28/June-5/July)
- Submitting physics (Q2 or Q3) and HPC (Q2) papers

Research group and roles

Representative:	**Y. ASAHI (QST)**	**Code development**
Deputy Representative:	**S. Maeyama (Nagoya Univ.)**	**Plasma turbulence**
Deputy Representative:	**G. Latu (CEA)**	**Task-level parallelization**
Collaborating researcher:	**X. Garbet (CEA)**	**Global plasma turbulence**
Collaborating researcher:	**T.-H. Watanabe (Nagoya Univ.)**	**Local plasma turbulence**
Collaborating researcher:	**T. Aoki (Tokyo Tech.)**	**Optimization on GPU**
Collaborating researcher:	**M. Ogino (Daido Univ.)**	**Optimization on CPU**