June 20th, 2024
5th IFERC workshop on the usage of GPU based system for fusion applications

Tuning of GKNET Code for GPU Acceleration

Contents

1. Introduction
2. Tuning of GKNET for GPU acceleration
3. Extension of GKNET to outer core region
4. Inference of partial differential equation by PINNs
5. Summary

Kenji IMADERA

Graduate School of Energy Science, Kyoto University, Japan

Collaborators

Akihiro ISHIZAWA, Shuhei OKUDA, Rio NAGANO, Ryoma TAKASU (Kyoto U.)
Masatoshi YAGI, Naoaki MIYATO, Haruki SETO (QST)

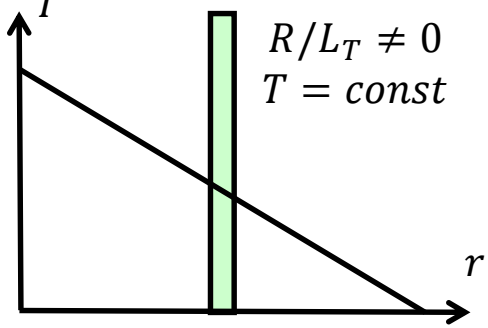
Global/Local Gyrokinetics

Local δf approach

$\partial_t f_{eq} - [H, f_{eq}] = C(f_{eq}) + S$

$\partial_t \delta f - [H, \delta f] - [\delta H, f_{eq}] - [\delta H, \delta f] = C(\delta f)$

Gradient-driven



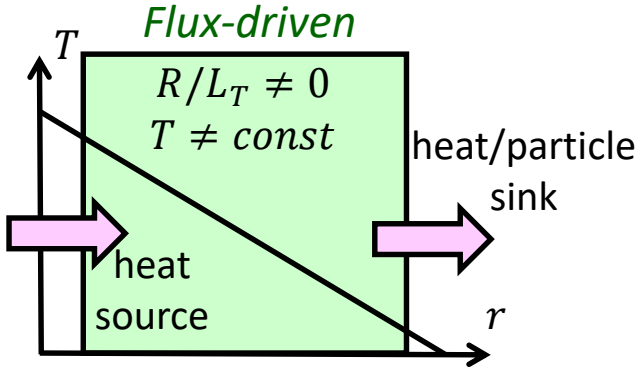
- Very powerful tool to estimate turbulent transport process
- Computationally efficient
  -> multi(ion/electron)-scale simulation

Global full-f approach

$\partial_t f_{eq} - [H, f_{eq}] = C(f_{eq}) + S$

$\partial_t \delta f - [H, \delta f] - [\delta H, f_{eq}] - [\delta H, \delta f] = C(\delta f)$

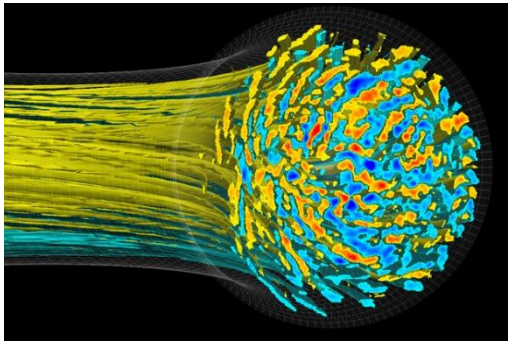
Flux-driven



- Global effects can be treated precisely
- Mean E_r is self-consistently determined
  -> Transport Barriers (ITB & ETB)
- Both neoclassical & turbulent transport process can be traced

GKNET Code and Recent Extension

- ✓ Original GKNET is a full-*f* electrostatic gyrokinetic code with adiabatic electron, which solves the gyrokinetic Vlasov equation by the FDM and the gyrokinetic quasi-neutrality condition by the 2D matrix solver + 1D FFT.
- ✓ GKNET is extend to (1) Toroidal ES version, (2) Toroidal EM version [Next presentation], and (3) Field aligned ES version.



[Imadera+, IAEA-2014]

GKNET
Toroidal
Electrostatic
Adiabatic electron

→

GKNET-KE
Toroidal
Electrostatic
Kinetic electron

→

GKNET-EM
Toroidal
Electromagnetic
Kinetic electron

[Ishizawa+, PoP-2019]

↓

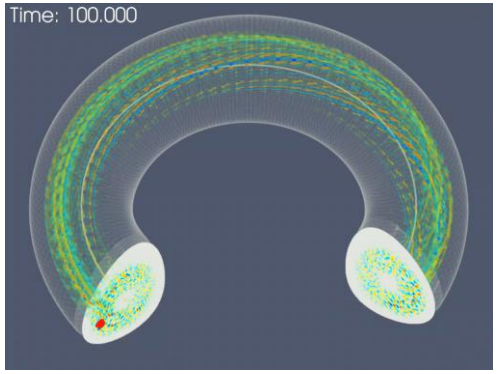
GKNET-FAC
Field aligned
Electrostatic
Adiabatic electron

[Okuda+, PFR-2023]

↓

GKNET-HE
Polar
Electrostatic
(Hybrid) Kinetic electron

[Imadera+, PPCF-2023]



Animation of 3D electrostatic potential and trapped electron obtained by GKNET-HE

Numerical Solvers of GKNET-FAC

Vlasov solver

- ✓ Spatial discretization: 4th-order Morinishi scheme [Idomura, JCP-2007]
- ✓ Time integration: 4th-order explicit Runge-Kutta scheme

Field solver

Gyrokinetic quasi-neutrality condition: $(L_0 + L_1)\phi(x,y,z) = s(x,y,z)$

$$L_0 = c_1(x,z)\frac{\partial^2}{\partial x^2} + c_2(x,z)\frac{\partial^2}{\partial y^2} + c_3(x,z)\frac{\partial}{\partial x}\frac{\partial}{\partial y} + c_4(x,z)\frac{\partial}{\partial x} + c_5(x,z)\frac{\partial}{\partial y} + c_6(x)$$

$$L_1 = l_1(x,z)\frac{\partial}{\partial x}\frac{\partial}{\partial z} + l_2(x,z)\frac{\partial}{\partial y}\frac{\partial}{\partial z} + l_3(x,z)\frac{\partial^2}{\partial z^2} + l_4(x,z)\frac{\partial}{\partial z}$$

Step-1 : FFT along the y direction ← because all the coefficients are independent to y

Step-2 : Set the initial guess $\hat{\phi}_n^{(0)}(x,z)$, and then solve $\hat{L}_0\hat{\phi}_n^{(1)}(x,z) + \hat{L}_{1,D}\hat{\phi}_n^{(1)}(x,z) = \hat{s}_n(x,z) - \hat{L}_{1,ND}\hat{\phi}_n^{(0)}(x,z)$ by using the 1D matrix solver

Step-3 : By repeating Step-2 (=Jacobi method), get the conveged solution $\hat{\phi}_n$

← because $\frac{\partial\phi}{\partial z}$ is higher order, a few iterations are enough for the convergence

Targets of This Talk

(A) Tuning of GKNET-FAC code for GPU acceleration

- ✓ By means of OpenACC directives, we have GPU-accelerated GKNET-FAC. Especially, by utilizing multi-GPUs & direct GPU-GPU data transfer, We have achieved 83 times speed-up for Vlasov solver and 9.1 times speed-up for Field solver. [K. Imadera, 3rd IFERC GPU workshop (2023).]
- ✓ In this year, we have tuned FFT part by using batched cuFFT.

(B) Extension of GKNET-FAC to outer core region

- ✓ We have extended GKNET-FAC to outer core region by allocating numerical mesh to SOL and divertor region. [S. Okuda *et al*., 40th JSPF meeting (2023).]
- ✓ We have verified the validity of our extended code (GKNET-X) through linear ITG simulations.

(C) Inference of partial differential equation by PINN

- ✓ By using Physics Informed Neural Network (PINN) model, we have inferred the partial differential equations which dominates turbulent transport process.

Contents

Flow chart of GKENT-FAC code

✓ The flow chart of GKNET-FAC is shown below. It is GPU-accelerated based on three concepts.

A) multi-GPU

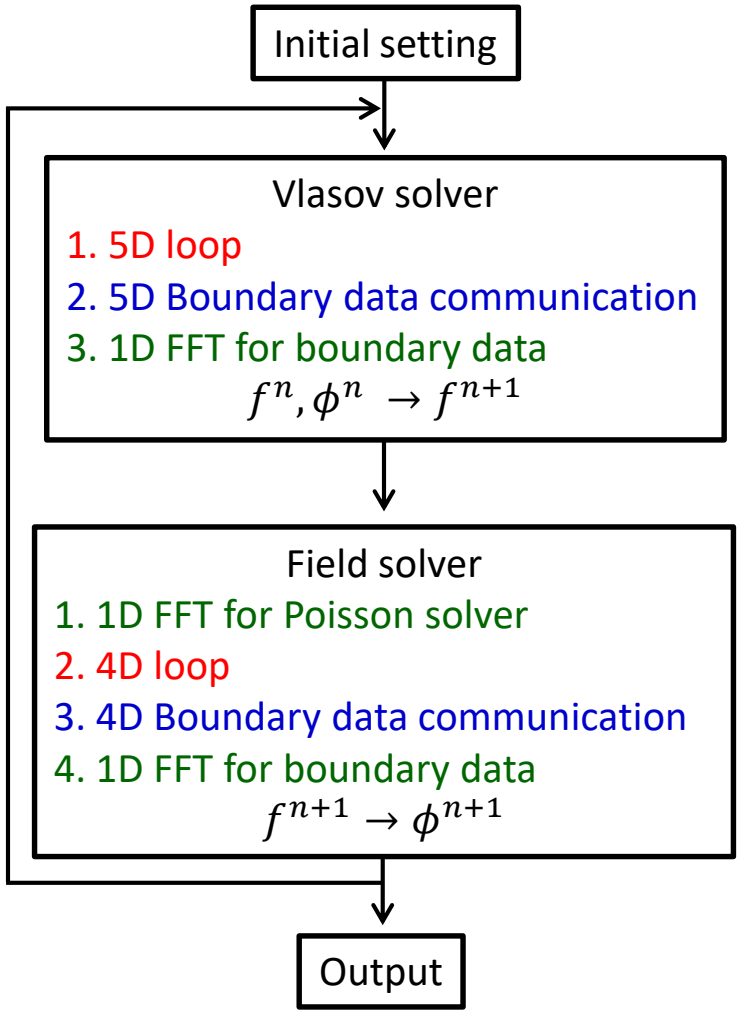
✓ By utilizing multi-GPU on each node, we accelerated 5D/4D loops in the Vlasov/Field solvers.

B) Direct GPU-GPU data transfer

✓ Instead of CPU-GPU data transfer, direct GPU-GPU data transfer is introduced for 5D/4D boundary data exchange in the Vlasov/Poisson solvers.

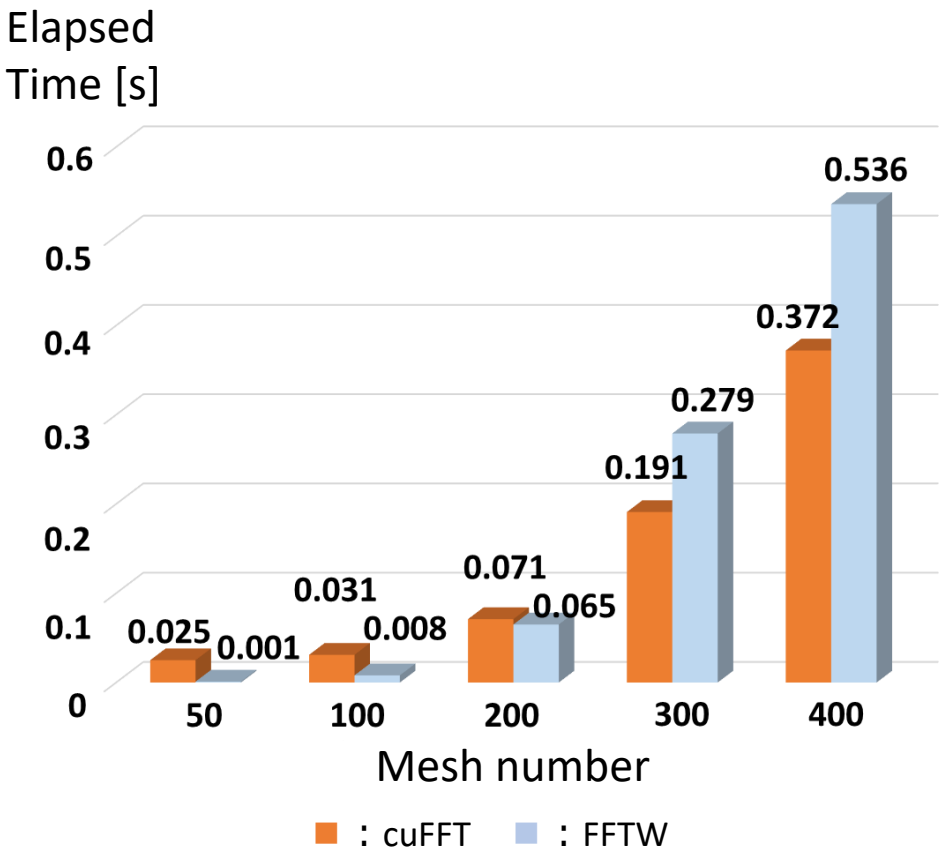
C) Batched CuFFT

✓ For boundary data in the Vlasov/Poisson solvers & for solving the quasi-neutrality condition in the Poisson solver, the batched cuFFT is installed.



Tuning of cuFFT - 1

- ✓ In the original GKNET, FFTW is used for FFT calculation on CPUs. To accelerate this part, cuFFT is installed.
- ✓ As is shown by the simple 3D FFT test below, the efficiency of cuFFT is confirmed only in the large-size problem.
- ✓ But, taking into account for the CPU-GPU data transfer for using FFTW, cuFFT is expected to be faster than FFTW even in the small-size problem.



Tuning of cuFFT - 2

Call of cuFFT

```
…

ierr1  = cufftPlanMany(plan, 1, …, CUFFT_Z2Z, Nbatch)
   if (ierr1 /= CUFFT_SUCCESS) then
     print *, 'cufftPlanMany: error', ierr1
   end if

…

!$acc host_data use_device(in, out)
ierr1 = cufftExecZ2Z(plan, in, out, CUFFT_FORWARD)
if (ierr1 /= CUFFT_SUCCESS) then
print *, 'cufftExecC2C: error', ierr1
end if
!$acc end host_data

…

ierr1 = cufftDestroy(plan)
   if (ierr1 /= CUFFT_SUCCESS) then
      print *, 'cufftDestroy: error', ierr1
   end if
```

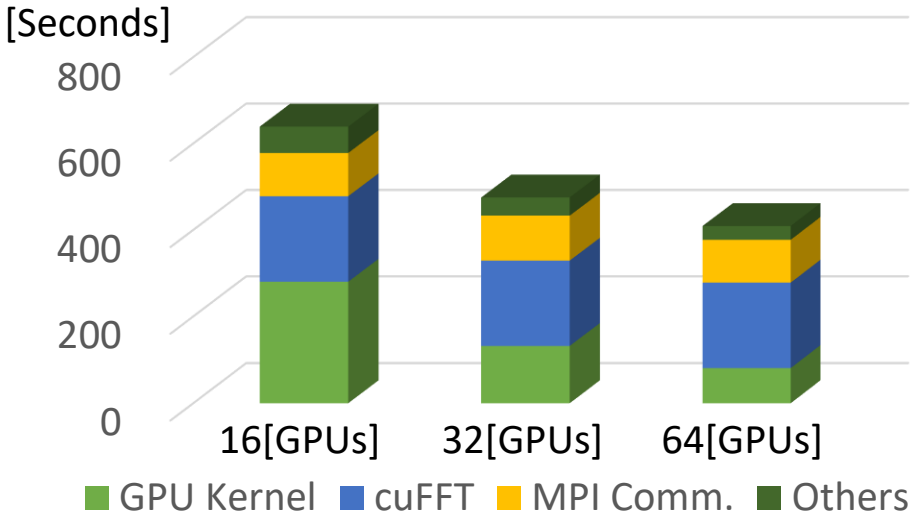
- ✓ Since we need 1D FFT for 3D data as follows, batched cuFFT is utilized.

$$f(x, k_y, z) = \sum_i f(x, y_i, z) \exp(i k_y y_i)$$

- ✓ In this case, Nbatch=N_x*N_z is set.
- ✓ In addition, direct GPU-GPU data transfer is also utilized.

Tuning of cuFFT - 3

Benchmark test



Number of GPU	GPU Kernel	cuFFT	MPI Comm.	Others
16	280	198	100	61
32	132	197	104	42
64	81	197	99	32

- ✓ Though the simple benchmark test by NVIDIA A100, we found that the cost of cuFFT part does not change while that of GPU kernel part clearly decreases.
- ✓ Since we proportionally increase the number of MPI in this test, it implies that cuFFT part is not accelerated by either MPI or OpenACC parallelization.
- ✓ We found that it originates from the planning of batched cuFFT because all the intermediate buffer allocations on CPU/GPU memory take place During the planning.

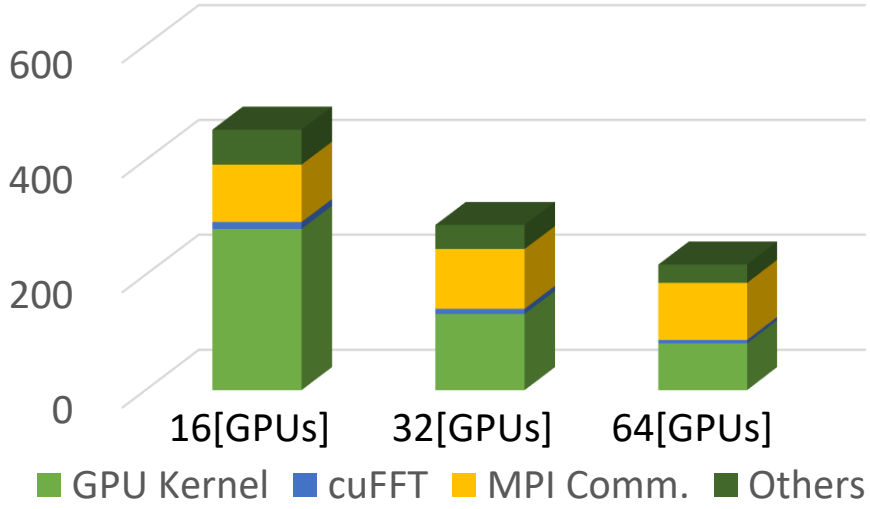
Tuning of cuFFT - 4

Call of cuFFT (modified)

```
…

ierr1  = cufftPlanMany(plan, 1, …, CUFFT_Z2Z, Nbatch)

DO timestep = 0, N_t
    Main part
END DO

ierr1 = cufftDestroy(Plan)

…
```

- ✓ To avoid this issue, we did the planning of cuFFT before the main loop.
- ✓ As the result, the cost of cuFFT becomes less than 1/10.

Benchmark result (modified)



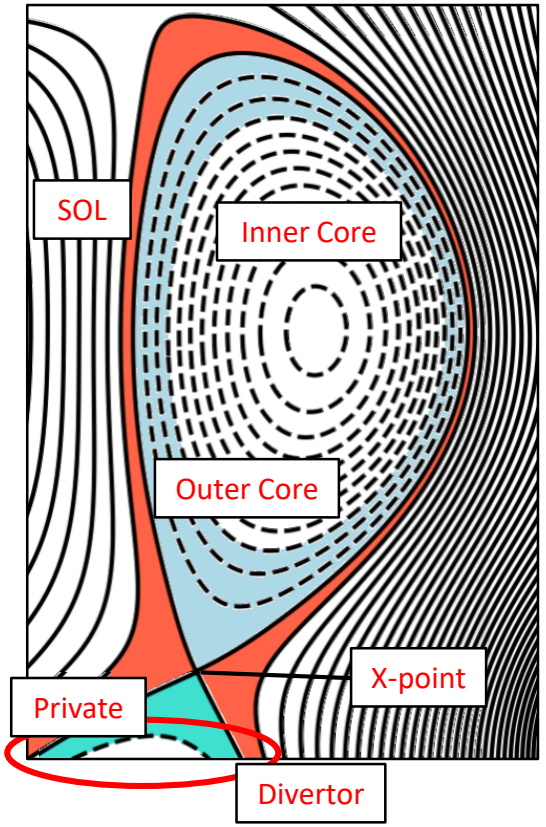
Number of GPU	GPU Kernel	cuFFT	MPI Comm.	Others
16	280	14	100	61
32	132	11	104	42
64	81	8	99	32

Contents

Background

- ✓ Plasma dynamic in Tokamak edge region is directly related to
  - Fuel supply/Impurity exhaust
  - Diverter heat load
  - L-H transition
- ✓ Gyrokinetic simulation is considered to be an essential tool to study the above physics based on the first-principles.
- ✓ But, it is still challenging to apply it to the edge region due to higher *q* values and complex magnetic surface geometries that are not present in the core region.

Some gyrokinetic codes to study edge region are recently developed in the world.



Strategy for Code Extension

[Imadera+, IAEA-2014]

GKNET
Electrostatic
Inner Core
Toroidal
Circular equil.

[Okuda+, PFR-2023]

GKNET-FAC
Electrostatic
Inner core
Field aligned
Realistic equil.

[Okuda+, JSPF meeting-2023]

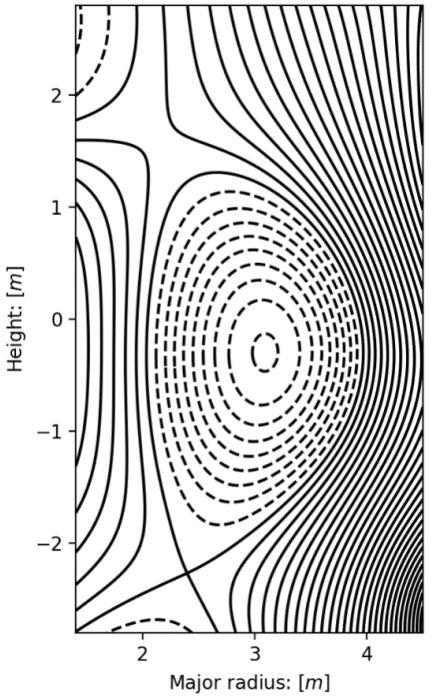
GKNET-X
Electrostatic
Outer core
Field aligned
Realistic equil.

Field aligned coordinate (+Shifted Metric)

$$x = \psi$$
$$y = y_{\mathrm{shift},j} - \zeta \quad [0, 2\pi/N_w]$$
$$z = \theta - \theta_j \qquad [-\pi/N_s, \pi/N_s]$$
$$y_{\mathrm{shift},j} = \int_{\theta_j}^{\theta} \frac{\boldsymbol{B}\cdot\boldsymbol{\nabla}\zeta}{\boldsymbol{B}\cdot\boldsymbol{\nabla}\theta} d\theta \quad (j = 0, 1, \ldots N_s - 1)$$

ψ : Poloidal magnetic flux function
θ : Label of poloidal angle
ζ : Geometric toroidal angle
N_s : Division number of domain
N_w: Toroidal wedge number

Tokamak Equilibrium
(JT-60SA ITER like case)



Numerical Condition of Benchmark Test

Simulation Parameters

$R_0/L_{T_i} = 6.92, \quad R_0/L_{T_e} = 6.92$

$R_0/L_n = 2.22, \quad 1/\rho_* = 150$

Magnetic equilibrium : JT-60SA ITER-like case

Boundary condition :

Radial : Fixed boundary $\delta f(\psi_{min}) = \delta f(\psi_{max}) = 0,$
$\phi(\psi_{min}) = \phi(\psi_{max}) = 0$

Divertor : Fixed boundary $\delta f(\theta_{div}) = 0, \phi(\theta_{div}) = 0$

Numerical domain : $0.1 < \psi < 1.075$ (GKNET-X)

$0.1 < \psi < 1$ (GKNET-FAC)

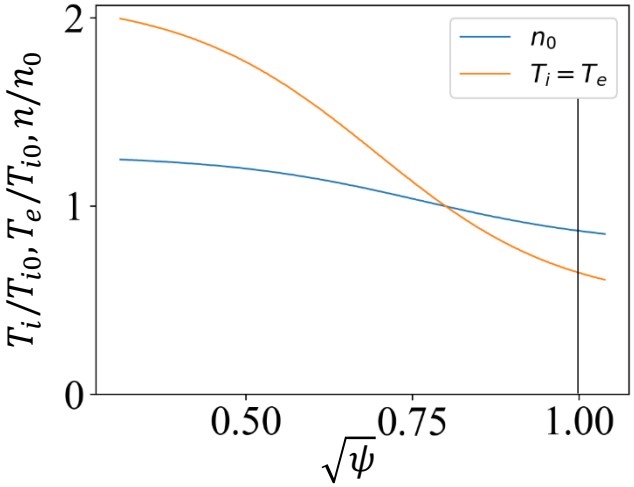
1/4 wedge torus

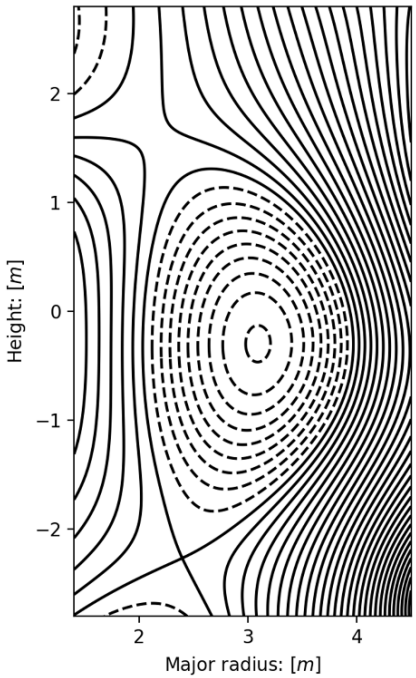
Mesh number : $(N_x, N_y, N_z, N_{v_\parallel}, N_\mu)$

$= (384 + 32, 104, 6 + 64 + 9, 80, 16)$

Core SOL Right div. Left div.

$N_s = 32$

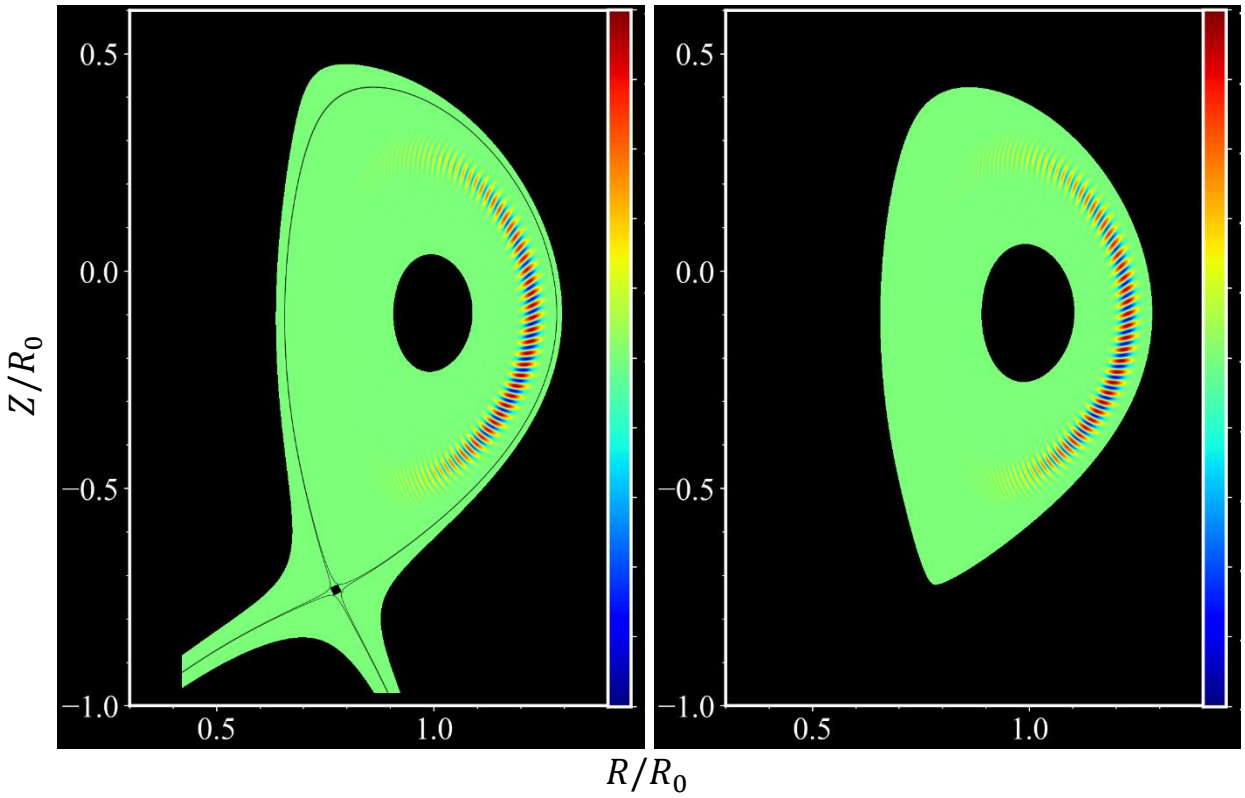




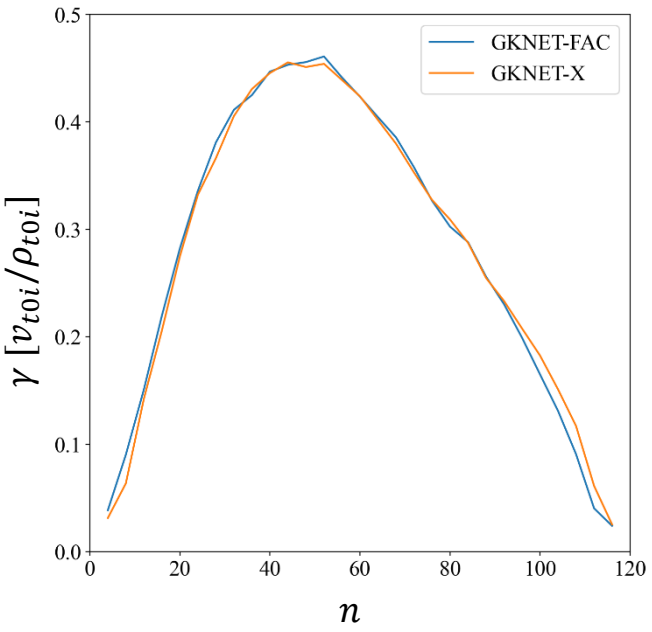
Results of Benchmark Test

$Re(\hat{\phi}_{n=60})$
calculated by GKNET-X

$Re(\hat{\phi}_{n=60})$
calculated by GKNET-FAC

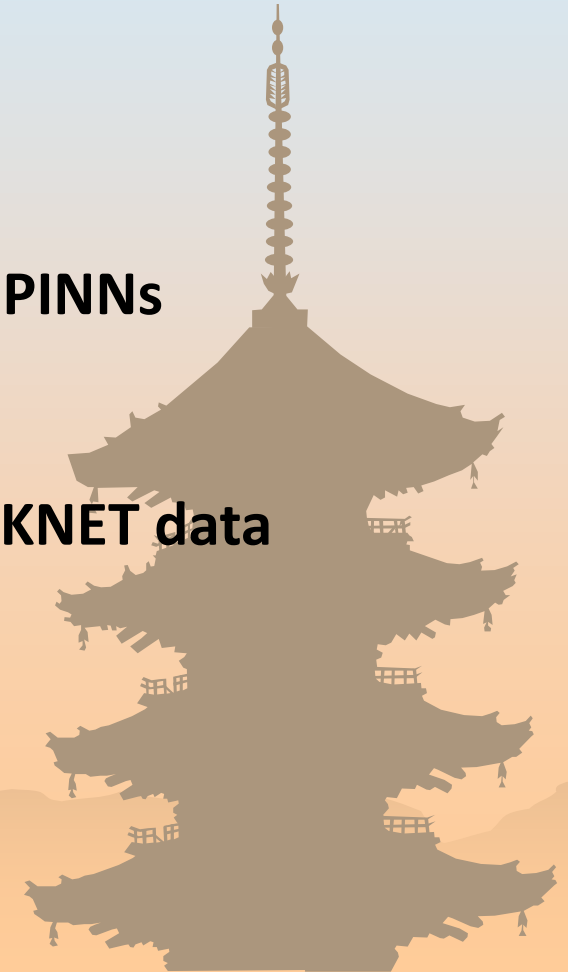


Comparison of dispersion relation



- ✓ GKNET-X and GKNET-FAC shows almost same linear results of ITG instability.
- ✓ This result demonstrates that GKNET-X precisely work to handle inner core region.

Contents



Background

Globality of turbulent transport process

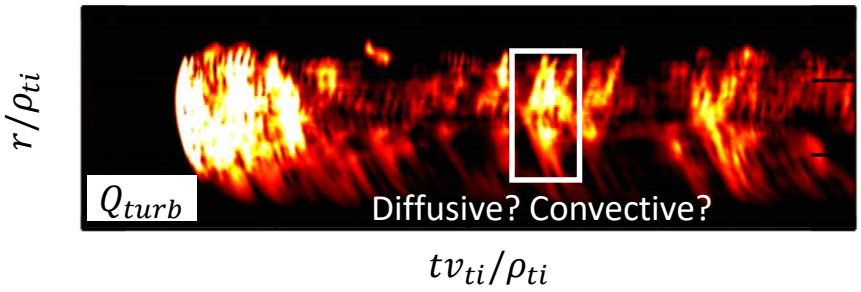
- ✓ Global intermittent heat transport phenomena are often observed in flux-driven turbulence simulations (see figure below right).
- ✓ By means of data-driven approach, can we infer the partial differential equations governing such physical phenomena from simulation data?
- ✓ Is it possible to understand that transport phenomena is diffusive or convective by inferring transport equation?

Governing equations

$$\frac{\partial T}{\partial t} + c\frac{\partial T}{\partial r} + \cdots = \nu\frac{\delta^2 T}{\delta r^2}$$

Inference of governing equations from simulation data

Understanding of plasma dynamics



Physics Informed Neural Network (PINN)

Applied loss function of PINNs (Ex. Advection-diffusion equation)

$$L = \underbrace{\frac{1}{N_x}\sum_{i=1}^{N_x}|f_i^n - \hat{f}_i^{n=0}|^2}_{\text{Initial condition}} + \underbrace{\frac{1}{N_t}\sum_{n=1}^{N_t}|f_i^n - \hat{f}_{i=1}^n|^2 + \frac{1}{N_t}\sum_{n=1}^{N_t}|f_i^n - \hat{f}_{i=N_x}^n|^2}_{\text{Boundary condition}}$$

$$+\text{Max}\left\{\underbrace{\frac{\alpha}{N_x N_t}\sum_{n=1}^{N_t}\sum_{i=1}^{N_x}|f_i^n - \bar{f}_i^n|^2}_{\text{Constraint to converge to teacher data}}, \underbrace{\frac{1}{N_x N_t}\sum_{n=1}^{N_t}\sum_{i=1}^{N_x}|F_i^n|^2}_{\text{Constraint to satisfy PDEs}}\right\}$$

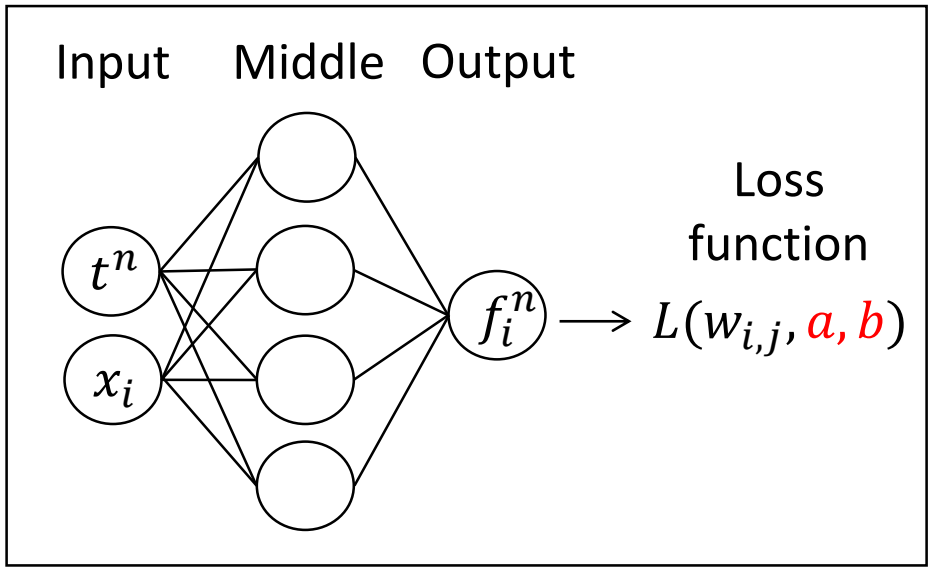
$$F_i^n = \left.\frac{\partial f}{\partial t}\right|_{x=x_i}^{t=t^n} + a\left.\frac{\partial f}{\partial x}\right|_{x=x_i}^{t=t^n} - b\left.\frac{\partial^2 f}{\partial x^2}\right|_{x=x_i}^{t=t^n}$$

f_i^n : Solution at $x = x_i,\ t = t^n$ estimated by PINN

$\hat{f}_i^n$: Initial and boundary conditions

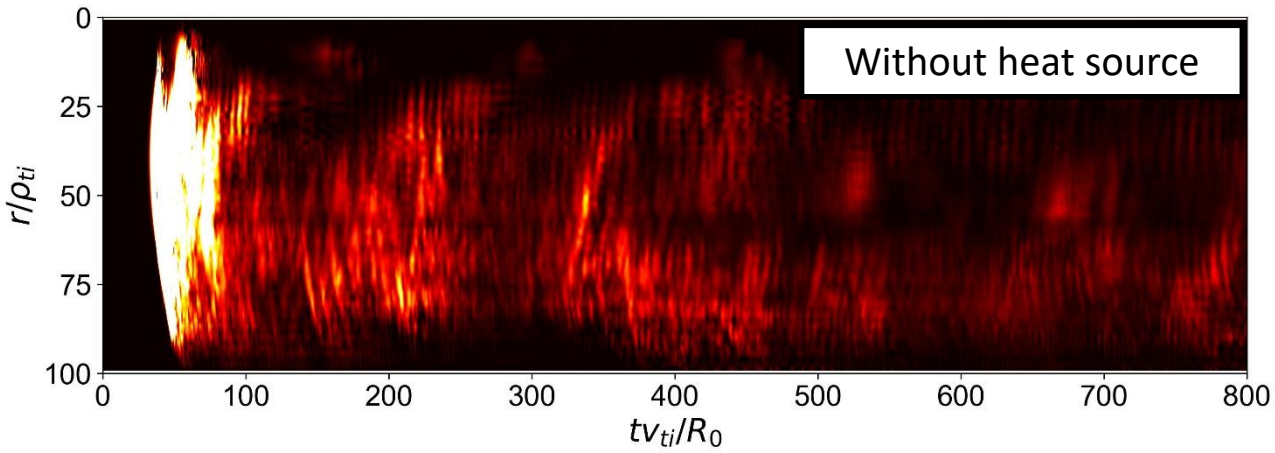
$\bar{f}_i^n$: Teacher data

✓ The coefficients are updated in the direction to which the loss function decreases.



Inference of Transport Equations from GKNET Data

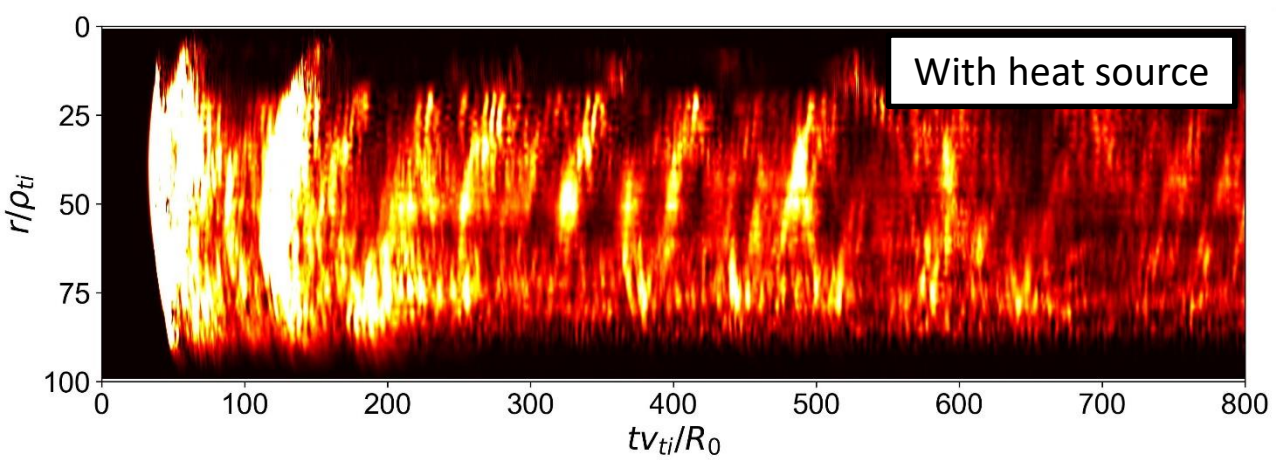
Target: Advection-diffusion equation -> $\frac{\partial T}{\partial t} + a\frac{\partial T}{\partial r} + b\frac{\partial^2 T}{\partial r^2} = 0$



$$\sum_{i=1}^{N} a\left(\frac{\partial T}{\partial r}\right)_i = 111$$

$$\sum_{i=1}^{N} b\left(\frac{\partial^2 T}{\partial r^2}\right)_i = -105$$

$$\rightarrow \frac{|\text{Convective}|}{|\text{Diffusive}|} = \mathbf{1.05}$$



$$\sum_{i=1}^{N} a\left(\frac{\partial T}{\partial r}\right)_i = 203$$

$$\sum_{i=1}^{N} b\left(\frac{\partial^2 T}{\partial r^2}\right)_i = -55.6$$

$$\rightarrow \frac{|\text{Convective}|}{|\text{Diffusive}|} = \mathbf{3.66}$$

Summary & Future Plans

(A) Tuning of GKNET-FAC code for GPU acceleration

✓ By introducing batched cuFFT and planning it before main loop, we have much reduced the computational cost for FFT part.

(B) Extension of GKNET-FAC to outer core region

✓ In addition to appropriate mesh allocation for SOL/divertor region, we have confirmed the validity of GKNET-X through global linear ITG simulations.

(C) Inference of partial differential equation by PINN

✓ By inferring the governing transport equation from GKNET data, we quantitatively verified that convective heat transport becomes dominant by the onset of heat source.

Future plans

(A) Reduction of MPI communication part

(B) Development of electromagnetic version of GKNET-FAC/GKNET-X

(C) Extension of target equation to nonlinear simultaneous equations