GPU optimization and Kinetic-MHD simulation with thermal ions and Landau damping with M3D-C1

Chang Liu

Princeton Plasma Physics Laboratory

CTTS Meeting

Apr 3, 2023

GPU Optimization in M3D-C1

- We have two ongoing efforts for optimizing M3D-C1 code for GPUs
  - Optimizing the calculation of finite-element matrix terms

    Conducted by Chang Liu and Jin Chen. We participated the GPU Hackathon at Princeton University in 2021.

  - Optimizing the preconditioner and iterative matrix solver.

    Collaborative work between PPPL and LBL groups.
- In addition, we have a particle-pushing code optimized for GPUs, which can be used in the kinetic-MHD module and runaway electron module.

Matrix assembling using GPUs in M3D-C1

- In Galerkin method, matrix element is calculated by integrating over each test and basis functions,

$$A_{ij} = \left(\nu_i, \hat{A}u_j\right) = \int \nu_i \hat{A} u_j d\mathbf{x}$$

- The integral can be calculated by sum over the contribution of quadrature points.
- There are 4 nested loops here: terms, test functions (i), basis functions (j), and quadrature points. These nested loops are independent and can run parallelly on GPUs.
  - The loop over mesh element runs parallelly using MPI on CPU, which is not changed. We employ Multi-Process Service (MPS) to utilize GPU with multiple MPI processes.
  - However, some of the calculation are repetitive for test and basis functions. The code can run even slower if we do the calculation of all the loops parallelly.
- Split the physics part and the numerical integral part
  - In the physics part, $\hat{A}$ of all terms are calculated for all the quadrature points. In the numerical integration part, integral overall all ν_i and u_i are calculated.
  - Physics part is most complicated, while numerical part is the most time consuming.

Matrix assembling loop in M3D-C1: Step 1

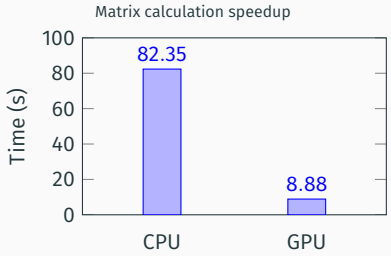
```
!$acc parallel loop gang collapse(2) private(tempss,tempdd) async(igpu)
do iterm=1,nterm
   do j=1,dofs_per_element
      tempss=ssarray(:,iterm)*nu79(j,:,op2(iterm))
      tempdd=ddarray(:,iterm)*nu79(j,:,op2(iterm))
      !$acc loop vector
      do i=1,dofs_per_element
         ssterm(i,j,iterm)=sum(mu79(i,:,op1(iterm))*tempss)
         ddterm(i,j,iterm)=sum(mu79(i,:,op1(iterm))*tempdd)
      end do
   end do
end do
```

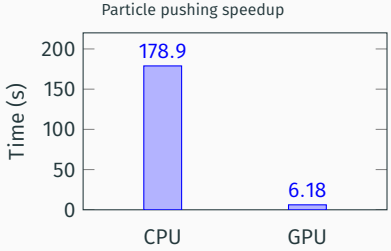
Matrix assembling loop in M3D-C1: Step 2

```
!$acc parallel loop gang collapse(2) async(igpu)
do i=1,dofs_per_element
   do j=1,dofs_per_element
      !$acc loop seq
      do iterm=1,nterm
            ss(i,j,term(iterm))=ss(i,j,term(iterm))+ssterm(i,j,iterm)
            dd(i,j,term(iterm))=dd(i,j,term(iterm))+ddterm(i,j,iterm)
      end do
   end do
end do
```

Speedup of matrix element calculation using GPUs

- Significant speedup was obtained using GPU on Traverse cluster at Princeton University.
- We have about 9x speedup for the whole matrix calculation part. If only counting the loop over all the ν, u, the speedup is about 30x.





Motivation of thermal ion kinetic simulation

- Kinetic effect of thermal ions is important for future fusion devices
  - For $T_i > 10$keV, the thermal ions themselves can drive Alfven eigenmodes which can lead to minor disruptions (Du et al., PRL 2021)
  - For lower frequency EP driven modes (fishbones, BAAE), the Landau damping effect from thermal ions can suppress the mode growth.
- In the classical kinetic-MHD approach like M3D-K, the coupling scheme of EP and MHD are based on the assumption that EPs only takes a small portion of ion density.
  - The ion motion is still dominated by the thermal ions, which can be treated as a fluid component in MHD equations.

Coupling scheme with kinetic thermal ions and fluid electrons from Park 1999

$$\frac{\partial(\rho \mathbf{V})}{\partial t} = -\nabla p_e - \nabla \cdot \mathbf{P}_i - \nabla \cdot \mathbf{P}_f + \mathbf{J} \times \mathbf{B}$$
$$\partial \mathbf{B}/\partial t = -\nabla \times \mathbf{E}$$
$$\mathbf{E} = -\mathbf{V} \times \mathbf{B} - \frac{1}{en} \nabla p_e$$
$$\partial p_e/\partial t = -\nabla \cdot (p_e \mathbf{V}) - (\gamma - 1)\, p_e \nabla \cdot \mathbf{V} + \ldots$$
$$\mathbf{P}_{i,f} = \int \mathbf{v}\mathbf{v} f d^3\mathbf{v}$$

- Both EPs and thermal ions are coupled to MHD equations through pressure terms in the momentum equation.
- This coupling scheme works mostly fine in M3D-C1 when excluding ∇p_e term in Ohm's law. However, when including this term, the code can give nonphysical instabilities with large pressure perturbation.
  - This scheme does not guarantee the quasineutrality condition as it has two independent equations to calculate the ion parallel motion, the fluid momentum equation and kinetic equations. Two equations are solved with staggered advance.
  - It is possible to avoid the issue using a fully implicit scheme for both MHD and particles, which is computationally difficult.

Coupling scheme with kinetic thermal ions and fluid electrons from Sato 2020

$$\frac{\partial\left(\rho \mathbf{V}_{\perp}\right)}{\partial t}=-\nabla_{\perp} p_{e}-\nabla_{\perp} \cdot \mathbf{P}_{i}-\nabla_{\perp} \cdot \mathbf{P}_{f}+\mathbf{J} \times \mathbf{B}$$

$$\partial \mathbf{B} / \partial t=-\nabla \times \mathbf{E}$$

$$\mathbf{E}=-\mathbf{V}_{\perp} \times \mathbf{B}-\frac{1}{en} \nabla_{\|} p_{e}$$

$$\partial p_{e} / \partial t=-\nabla \cdot\left[p_{e}\left(\mathbf{V}_{\perp}+V_{\|} \mathbf{b}\right)\right]-\left(\gamma-1\right) p_{e} \nabla \cdot\left(\mathbf{V}+V_{\|} \mathbf{b}\right)+\ldots$$

$$\mathbf{P}_{i,f}=\int \mathbf{v} \mathbf{v} f_{i,f} d^{3} \mathbf{v}, \qquad V_{\|}=\frac{1}{n_{i}+n_{f}}\left[\int v_{\|} f_{i} d^{3} \mathbf{v}+\int v_{\|} f_{f} d^{3} \mathbf{v}\right]$$

- In addition to pressure, this scheme also uses the parallel velocity from kinetic species to replace the parallel component of **V** in MHD equations.
- With the synchronization of parallel velocity between kinetic and MHD equations, the scheme enforce quasineutrality and avoid nonphysical modes.
- By including the ∇p_e term in Ohm's law, one can use the scheme to study the parallel kinetic effects and Landau damping of acoustic modes.

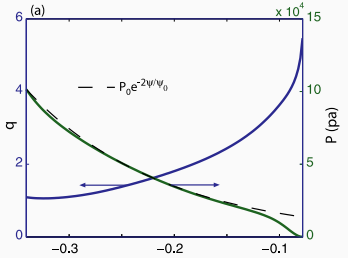
Synchronization of parallel velocity in M3D-C1

1. Calculate the parallel velocity from kinetic ion particles, store it in a scalar field.
2. After each MHD timestep, remove the parallel velocity from the 3 velocity fields in M3D-C1 (ϕ, v_ϕ, χ).
3. Add the particle parallel velocity to the M3D-C1 velocity fields.

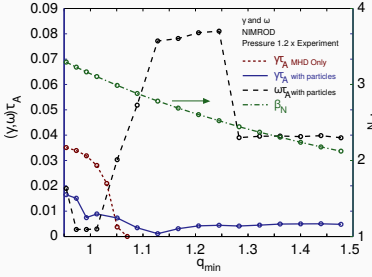
Test of the new coupling scheme on DIII-D kink/fishbone simulation

- We use the new code to redo the $n = 1$ linear kink/fishbone simulation for DIII-D #125476.
- The equilibrium was studied in Brennan NF 2012 using NIMROD for the transition from low-frequency fishbone mode to high frequency BAE-like mode.
  - The mode has strong parallel motion and pressure perturbation.

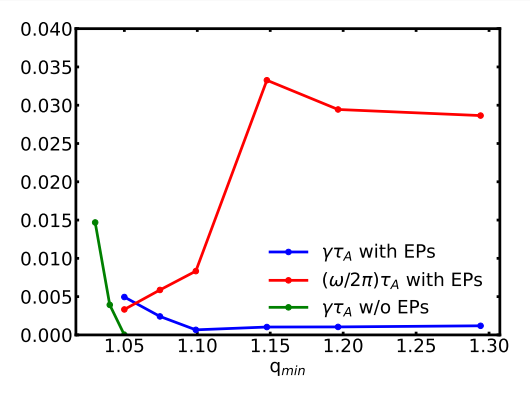
Equilibrium profiles of q and pressure for DIII-D # 125476



Growth rates and frequencies for different q_{min} from NIMROD

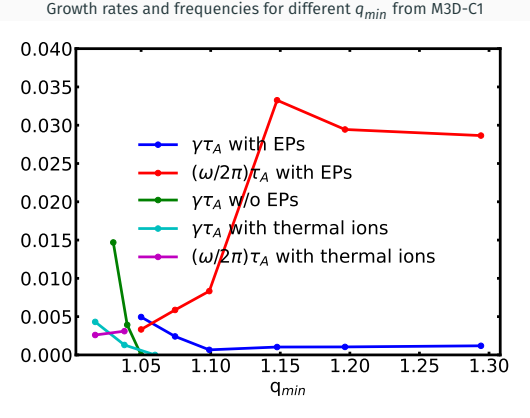


M3D-C1 simulation without thermal ions or parallel E fields



Using the kinetic-MHD module with only EPs, we successfully reproduce both the low frequency fishbone modes and high frequency BAE modes, with some differences on mode frequency.

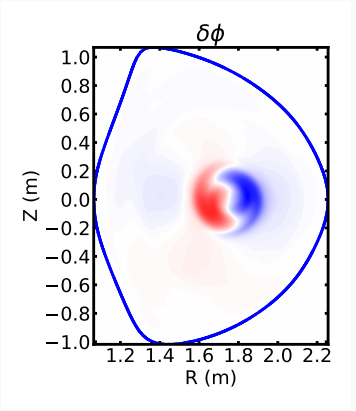
M3D-C1 simulation with thermal ions and parallel E fields



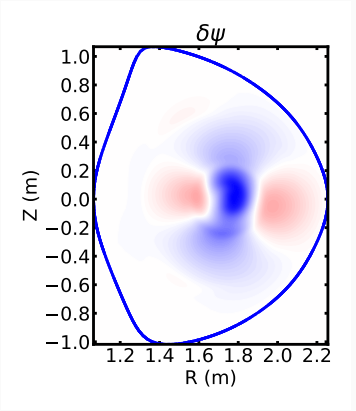
After including the thermal ion kinetics and $E_{\|}$, we don't see mode excited at all for qmin>1.05, which is similar to the simulation without EPs. The mode has a finite frequency about 4kHz.

Mode structure of fishbone

Perturbed ϕ structure



Perturbed ψ structure



Summary

- To include the thermal ion kinetic effect and parallel *E* field physics, we need to synchronize the parallel velocity between kinetic species and MHD, to ensure quasineutrality and avoid nonphysical modes.
- Landau damping of thermal ions can affect the fishbone mode and high frequency BAE mode, making the mode barely unstable or stable for qmin>1.