M3D-C1-K Tutorial

M3D-C1-K Tutorial

Compiling M3D-C1-K

When using Makefile to compile the code, make sure PAR=1 is added to the compiling command. For example

```
make ARCH=stellar COM=1 OPT=1 PAR=1
```

When compiling using CMake, make sure to include ENABLE_PARTICLE=1 in CMake switches.

When compiling using Spack, add +particle in m3dc1 spec.

The default version will use CPU to calculate particle pushing. It is recommended to turn on the OpenMP compiler option to get a multi-threaded version, which can accelerate the particle pushing. For Makefile, add OMP=1. For CMake, add ENABLE_OPENMP=1. For Spack, add +openmp in the spec.

To obtain the GPU version, include the OpenACC switch in the compiler commands.

Creating input files

C1input

Here we list the input parameters related to kinetic simulations in C1input

kinetic	0: disable PIC simulation. 1: enable PIC simulation
kinetic_fast_ion	1: enable fast ion PIC simulation
kinetic_thermal_ion	1: enable thermal ion PIC simulation and density coupling between MHD and PIC
igyroaverage	1: enable gyro-averaging for PIC simulation
particle_linear	1: solve linear delta-f equations. 0: include nonlinear terms in delta-f equations
particle_substeps	Number of subcycles for particle pushing in one MHD timestep
fast_ion_mass	Fast ion mass (in units of proton mass). Default value is ion_mass
fast_ion_z	Zeff of fast ion. Default value is z_ion
fast_ion_dist	Type of fast ion distribution function. 0: Read 3D distribution from file. 1: Maxwellian. 2: Slowing-down

fast_ion_max_energy Max energy for the slowing-down distribution. The critical energy profile is set in
profile_tf_rho
num_par_max Maximum number of particle markers
ikinetic_vpar 1: synchronize particle parallel flow to MHD. 0: cancel MHD parallel flow
vpar_reduce Factor of parallel flow reduction for every MHD timestep. Default value is 0.5,
meaning reducing 50% of parallel flow at every timestep
smooth_par Smoothing factor for particle pressure, density and parallel velocity
smooth_pres Smoothing factor for electron pressure, for calculating parallel electric field used in
particle simulation

Particle profiles

For fast ion kinetic simulation (kinetic_fast_ion=1), create the profile files

profile_nf_rho Fast ion density (in the unit of cm^-3) as a function of rho (sqrt(psi_T/psi_lim)),
similar to profile_ne_rho_0
profile_tf_rho For Maxwellian distribution (fast_ion_dist=1), set fast ion temperature (in the unit of
eV) as a function of rho
For slowing-down distribution (fast_ion_dist=2), set the critical energy (in the unit of
eV) as a function of rho

For thermal ion kinetic simulation (kinetic_thermal_ion=1), create the profile files

profile_ni_rho Thermal ion density (in the unit of cm^-3) as a function of rho (sqrt(psi_T/psi_lim)),
similar to profile_ne_rho_0
profile_ti_rho Thermal ion temperature (in the unit of eV) as a function of rho

Note that for stellarator simulation, we use different filenames for EP profile (following the filenames of other quantities)

nf_profile Fast ion density (in the unit of m^-3) as a function of s, similar to n_profile
tf_profile For Maxwellian distribution (fast_ion_dist=1), set fast ion temperature (in the unit of
eV) as a function of s

Visualization

During or after the M3D-C1 run, one can use visualization tools to check the outcome of particle simulation

The following field variables can be plotted using the IDL plot_field routine or Python plot_field function

For fast ion kinetic simulation (kinetic_fast_ion=1)

den_f_0	Fast ion total density
den_f_1	Fast ion perturbed density
p_f_par	Fast ion perturbed parallel pressure
p_f_perp	Fast ion perturbed perpendicular pressure

For thermal ion kinetic simulation (kinetic_thermal_ion=1)

den_i_0	Thermal ion total density
den_i_1	Thermal ion perturbed density
p_i_par	Thermal ion perturbed parallel pressure
p_i_perp	Thermal ion perturbed perpendicular pressure

Examples

You can find the following examples of M3D-C1-K simulation

Fishbone simulation

Stellar: /projects/M3DC1/liuchang/gyfu_fishbone
Flux: /p/m3dc1/cliu/gyfu_fishbone

This is an example of fishbone simulating using the setup from Fu et al., Phys. Plasmas **13**(5), 052517 (2006). To test the simulation, just copy the directory to your own home or scratch directory, copy the compiled m3dc1_2d_complex into it, and run a batch job.

The command line looks like

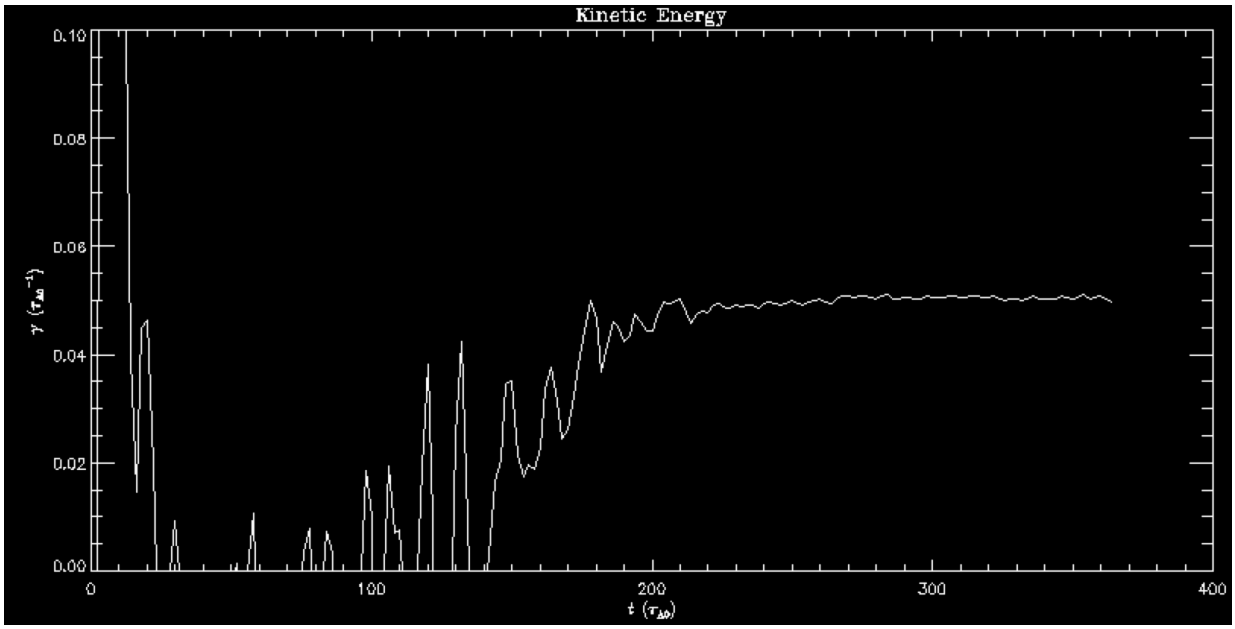
srun -N 1 -n 32 ./m3dc1_2d_complex -ksp_type preonly -pc_type lu -pc_factor_mat_solver_type mumps -mat_mumps_icntl_23 5000 -ksp_converged_reason -ksp_error_if_not_converged true -on_error_abort

For OpenMP version of the code,

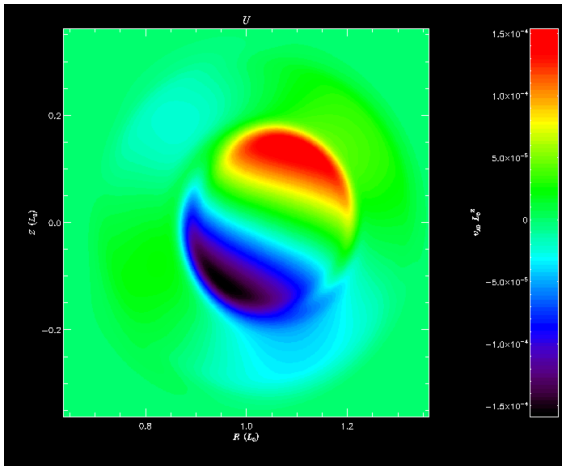
export OMP_NUM_THREADS=3
srun -N 1 -n 32 --cpus-per-task=3 ./m3dc1_2d_complex_omp -ksp_type preonly -pc_type lu -pc_factor_mat_solver_type mumps -mat_mumps_icntl_23 5000 -ksp_converged_reason -ksp_error_if_not_converged true -on_error_abort

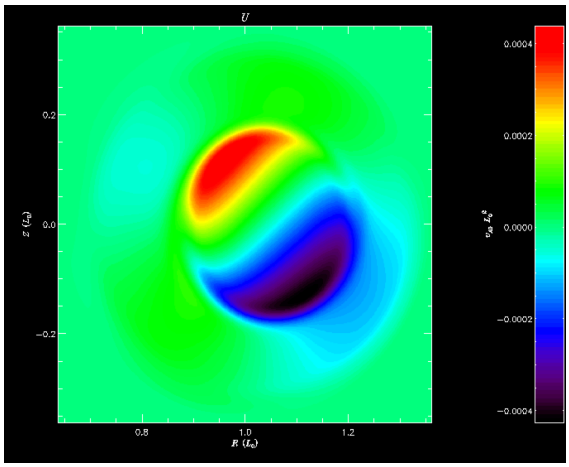
Expected result:

Kinetic energy grows with growth rate around 0.05



(1,1) mode structure rotates with time





AE simulation in W-7X

Stellar: /projects/M3DC1/liuchang/handihuang_w7x_ae
Perlmutter: /global/cfs/cdirs/mp288/changliu/handihuang_w7x_ae

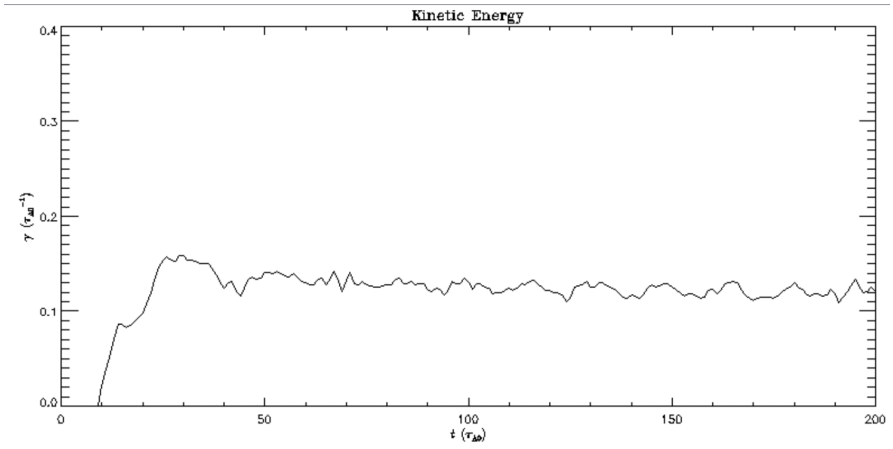
This is an example of AE simulating using the equilibrium and EP profile from Handi Huang. To test the simulation, just copy the directory to your own home or scratch directory, copy the compiled m3dc1_3d_st into it, and run a batch job.

The command line looks like

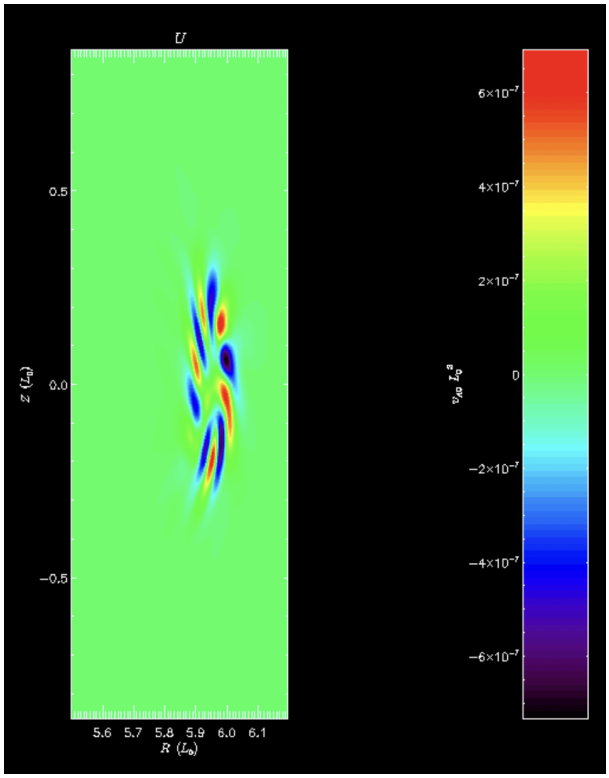
srun -N 4 -n $((32*8)) ./m3dc1_3d_st -pc_type bjacobi -pc_bjacobi_blocks 8 -sub_pc_type lu -sub_pc_factor_mat_solver_type superlu_dist -mat_mumps_icntl_14 5000 -sub_ksp_type preonly -ksp_type fgmres -ksp_max_it 2000 -ksp_rtol 1.e-9 -ksp_atol 1.e-20 -hard_pc_type bjacobi -hard_pc_bjacobi_blocks 8 -hard_sub_pc_type lu -hard_sub_pc_factor_mat_solver_type superlu_dist -hard_mat_mumps_icntl_14 5000 -hard_sub_ksp_type preonly -hard_ksp_type fgmres -hard_ksp_max_it 2000 -hard_ksp_rtol 1.e-9 -hard_ksp_atol 1.e-20 -ksp_converged_reason

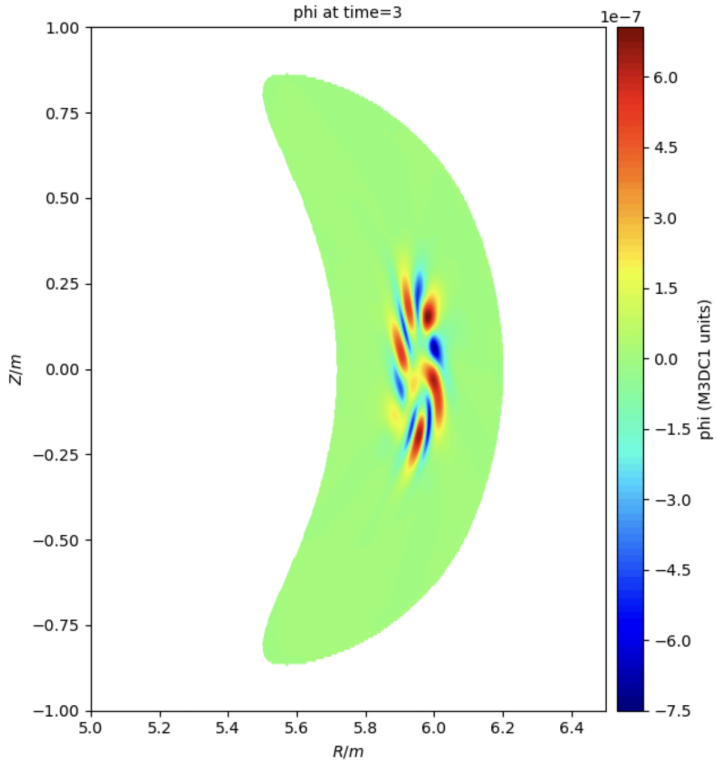
Expected result:

Kinetic energy grows with growth rate around 0.11



Excitation of (5,6) mode, with frequency around 67kHz


U
Z (L0)
R (L0)
5.6 5.7 5.8 5.9 6.0 6.1
0.5
0.0
−0.5
6×10⁻⁷
4×10⁻⁷
2×10⁻⁷
0
−2×10⁻⁷
−4×10⁻⁷
−6×10⁻⁷


phi at time=3
1e−7
Z/m
R/m
phi (M3DC1 units)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
5.0 5.2 5.4 5.6 5.8 6.0 6.2 6.4
6.0
4.5
3.0
1.5
0.0
−1.5
−3.0
−4.5
−6.0
−7.5

Thermal ion kinetic simulation in DIII-D

Stellar: /projects/M3DC1/liuchang/d3d_178631_lfm
Perlmutter: /global/cfs/cdirs/mp288/changliu/d3d_178631_lfm

This is an example of thermal ion kinetic simulation using a DIII-D equilibrium. The equilibrium pressure was artificially amplified to make the mode more unstable. The objective is to show an unstable KBM corresponding to the low frequency mode (LFM) observed in experiments. However, current M3D-C1-K simulation shows an unstable BAE which is different from the KBM found in GTC simulation, and the growth rate is much smaller.

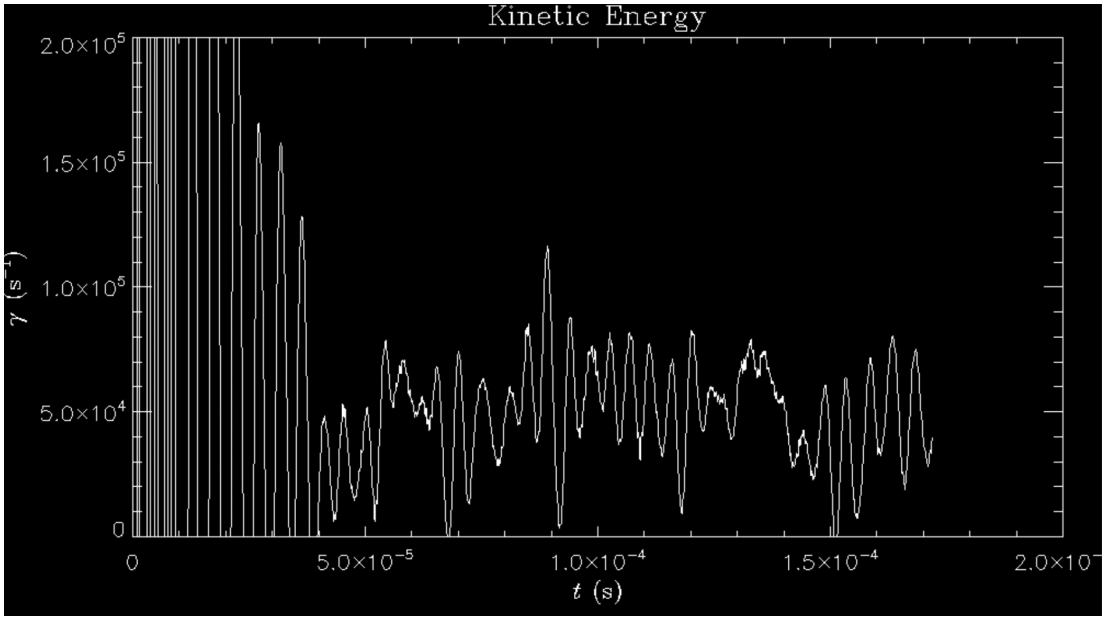
To test the simulation, just copy the directory to your own home or scratch directory, copy the compiled m3dc1_2d_complex into it, and run a batch job.

The command line looks like

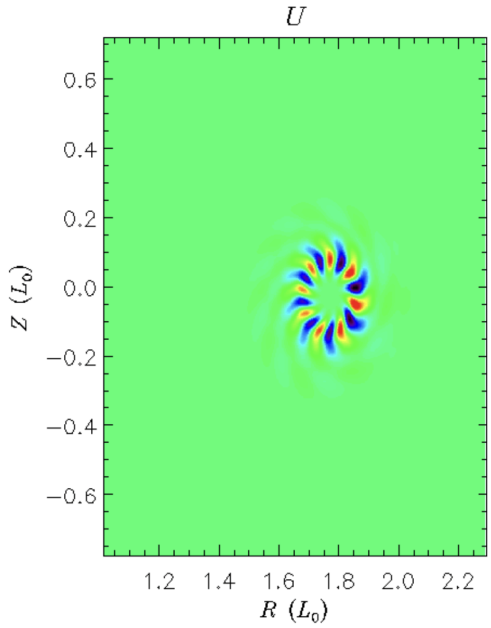
srun -N 1 -n 64 ./m3dc1_2d_complex -ksp_type preonly -pc_type lu -pc_factor_mat_solver_type mumps -mat_mumps_icntl_23 5000 -ksp_converged_reason -ksp_error_if_not_converged true -on_error_abort

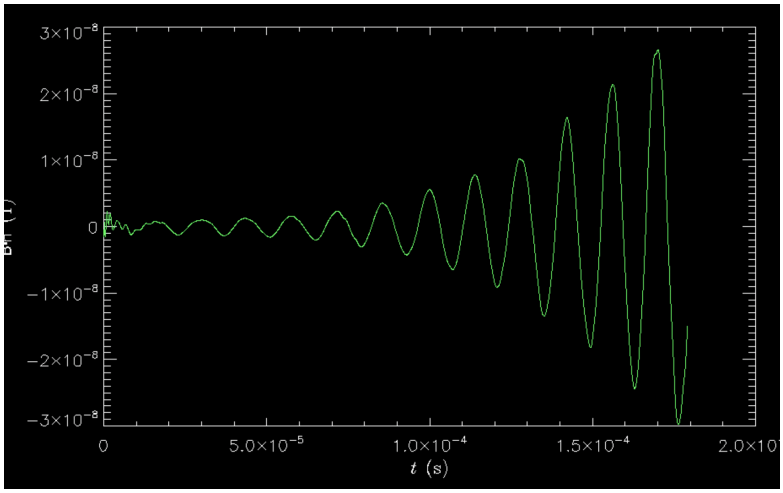
Expected result:

Kinetic energy grows with growth rate around 5*10^4 s^-1



Excitation of (6,8) mode, with frequency around 69.5kHz


U
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Z (L0)
1.2
1.4
1.6
1.8
2.0
2.2
R (L0)


3×10−8
2×10−8
1×10−8
0
−1×10−8
−2×10−8
−3×10−8
0
5.0×10−5
1.0×10−4
1.5×10−4
t (s)