Explicit method for solving Alfven wave in M3D-C1

Matrix assembling and factoring is time consuming

- Currently we use implicit or semi-implicit method to solve MHD equations.
- For nonlinear simulation, this requires recalculating and refactoring the MHD equation matrix at every timestep.
- For Alfven wave simulation ($\omega * \tau_A \sim 0.1$), the time step must be small ($\Delta t \sim \tau_A$), leading to long simulation time.

Linear solver avoids assembling but miss nonlinear physics

- We can solve linear MHD equation, whose matrix only depends on equilibrium quantities (B0, p0) and is fixed.
- The Alfven mode will still saturated at a small level ($\delta B/B \sim 0.001$) because of particle nonlinearity.
- Mode-mode coupling will be missing in the simulation.

Treat nonlinear terms explicitly

- To include the nonlinear MHD effect, we can treat the nonlinear MHD terms as explicit terms on the right-hand-side of equations.

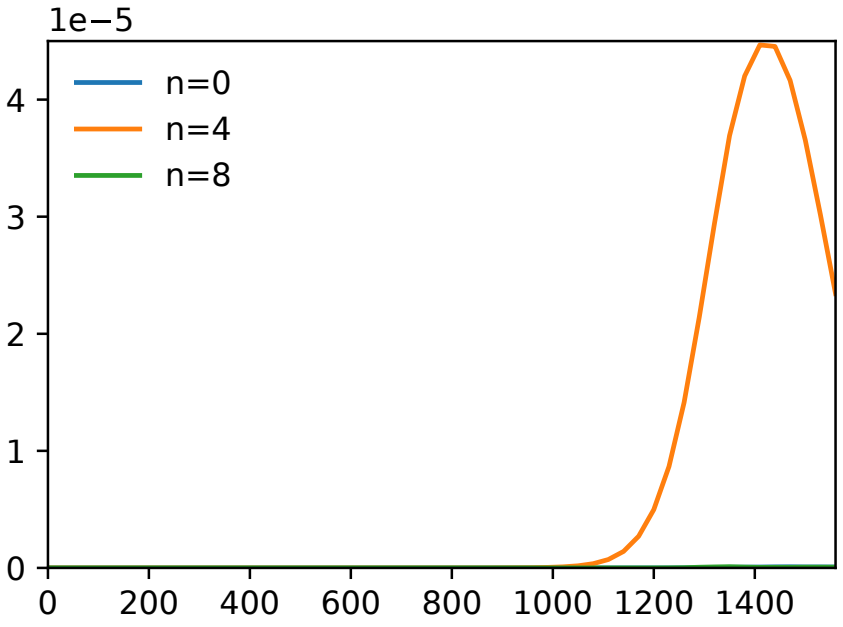
$$Sx_{n+1}=Dx_n+b$$

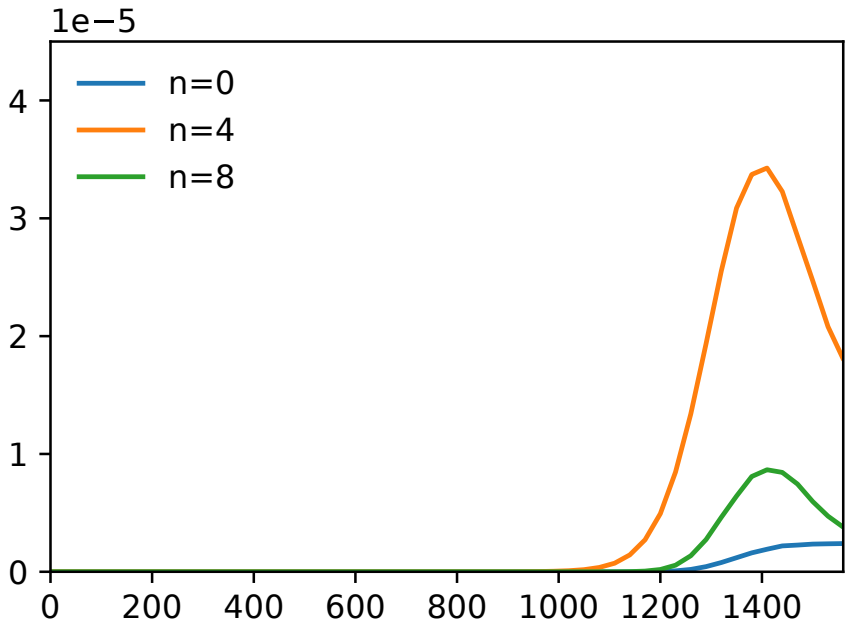
- The matrices S,D are unchanged and only need to be factorized once.
- The nonlinear terms only needs to be calculated as a vector instead of matrix. The time complexity is O(dofs*pts) instead of O(dofs*dofs*pts).
- The CFL condition will be determined by δv, δB and not by v_A, B0
  - To avoid numerical instability, we can use a smaller timestep (~0.2 τ_A) and subcycles.

Computation time comparison

- Implicit method
  - 70s (matrix assembling) + 15s (solver)
- Explicit method
  - [1.5s (nonlinear vector calculation) + 1.5s (solver)] *5

Results


1e−5
n=0
n=4
n=8
0
1
2
3
4
0
200
400
600
800
1000
1200
1400


1e−5
n=0
n=4
n=8
0
1
2
3
4
0
200
400
600
800
1000
1200
1400

Acceleration using GPU?

- Calculation of nonlinear vector can be accelerated using GPU.
- I wonder if there is a good way to accelerate the solving of a factorized matrix using GPU.