Suggestion for re-ordering the velocity unknowns for M3D-C1

Presented by
S. Jardin

Nov 20, 2017
CTTS ZOOM Call
1:00 PM EST

Overview-1—Matrix Elements

- M3D-C1 is a 3D Implicit Magnetohydrodynamics Code that uses high-order finite elements in all 3 directions

- Most of the compute time spent is in either:
  1. Defining the matrix elements
  2. Solving the sparse matrix equations

- We have recently sped up (1.) above by
  - Rearranging loops to make better reuse of data
  - Improved vectorization
  - Use of BLAS intrinsics (e.g. dgemm)

- We have also implemented OpenMP to improve memory footprint in (1.)
  - Remaining bottleneck is lack of thread-safe matrix insertions

Overview-2--Solve

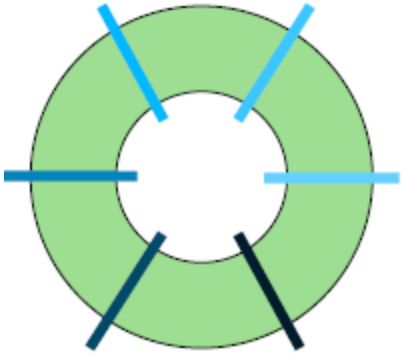
- Solving the sparse matrix equations has 2 elements
  1. Effective preconditioners (SuperLU, STRUMPACK)
  2. Iterative solve using GMRES

- Our algorithm has 3 major linear matrix solves per time-step that can be performed sequentially
  1. The velocity solve …. $\mathbf{V}^{n+1}$
  2. The magnetic field solve…$\mathbf{B}^{n+1}$
  3. The pressure solve….p^{n+1}

- Here, we want to concentrate on the velocity solve as it is normally the most time-consuming and has some unique properties

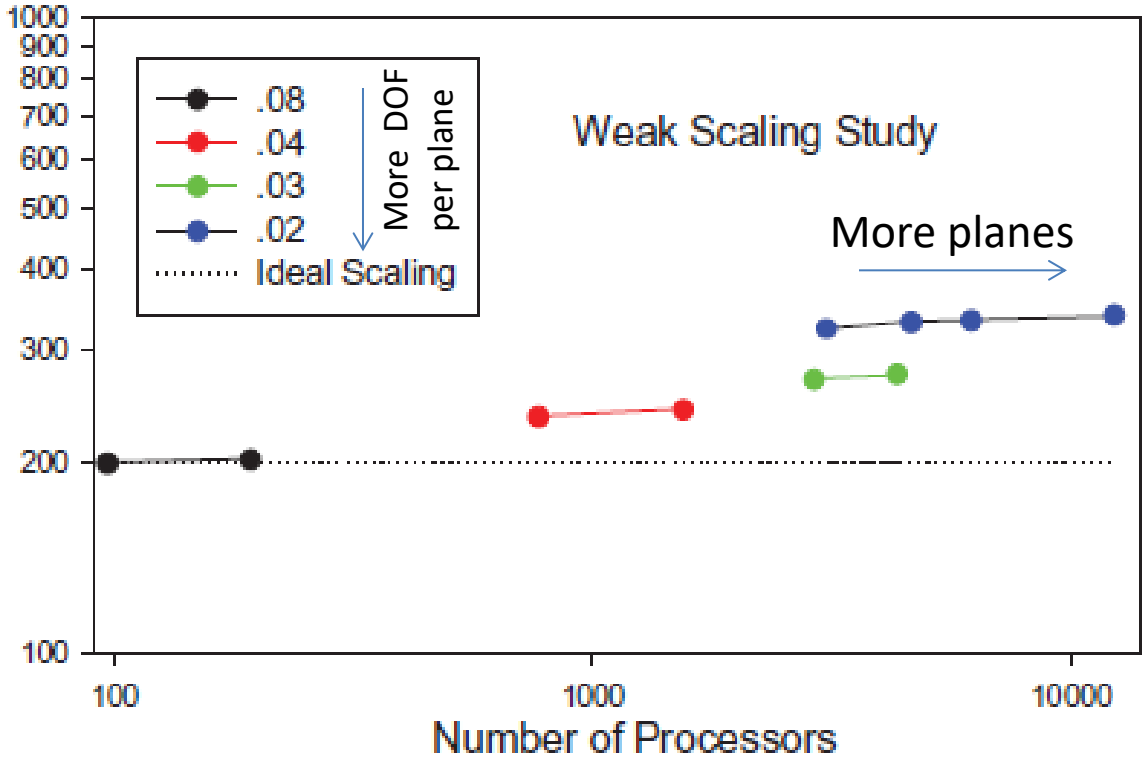
Overview-3—Velocity Solve

- We now make use of the fact that couplings within a toroidal plane are stronger than those across planes.



- Preconditioner uses SuperLU or STRUMPACK to perform a direct solve within each toroidal plane.
- This is effective, but does not scale well beyond a certain size
- There are additional strong physics coupling that we can take advantage of that would lead to smaller matrices for the SuperLU direct solve. STRUMPACK may also be able to make use of these couplings.

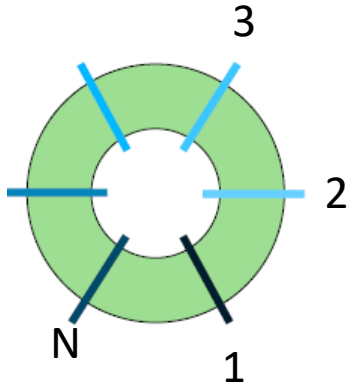
Typical Weak Scaling Result



Different colors correspond to different number of mesh points in a given toroidal plane. Scaling is better when increasing the number of planes than when increasing mesh points within a plane.

The Velocity Matrix

Because each toroidal plane couples only to adjacent toroidal planes, the full velocity matrix is of block-tridiagonal form. Corner elements due to periodicity.



$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{C}_1 & & \mathbf{A}_1 \\ \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & \\ & \ddots & \ddots & \ddots \\ \mathbf{C}_N & & \mathbf{A}_N & \mathbf{B}_N \end{bmatrix} \bullet \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_N \end{bmatrix}$$

$\mathbf{X}_j$ contains all the velocity variables on plane j

$\mathbf{X}_j$ contains all the velocity variables on plane j,

However, there are 3 different kind of velocity variables: U, ω, χ

$$\mathbf{V} = R^2 \nabla U \times \nabla \varphi + \omega R^2 \nabla \varphi + R^{-2} \nabla_\perp \chi$$

$$\mathbf{X}_j = \begin{bmatrix} \mathbf{U}_{1,j} \\ \boldsymbol{\omega}_{1,j} \\ \boldsymbol{\chi}_{1,j} \\ \mathbf{U}_{2,j} \\ \boldsymbol{\omega}_{2,j} \\ \boldsymbol{\chi}_{2,j} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{U}_{M,j} \\ \boldsymbol{\omega}_{M,j} \\ \boldsymbol{\chi}_{M,j} \end{bmatrix}$$

$\mathbf{U}_{i,j}$ contains all 12 DOF for the variable U at node i of plane j $\Rightarrow$

$$\mathbf{U}_{i,j} = \begin{bmatrix} U \\ U_R \\ U_Z \\ U_{RR} \\ U_{RZ} \\ U_{ZZ} \\ U_\varphi \\ U_{R\varphi} \\ U_{Z\varphi} \\ U_{RR\varphi} \\ U_{RZ\varphi} \\ U_{ZZ\varphi} \end{bmatrix}_{i,j}$$

We presently do not take into account that the 3 velocity variables $\mathbf{U}, \omega, \chi$ are very different.

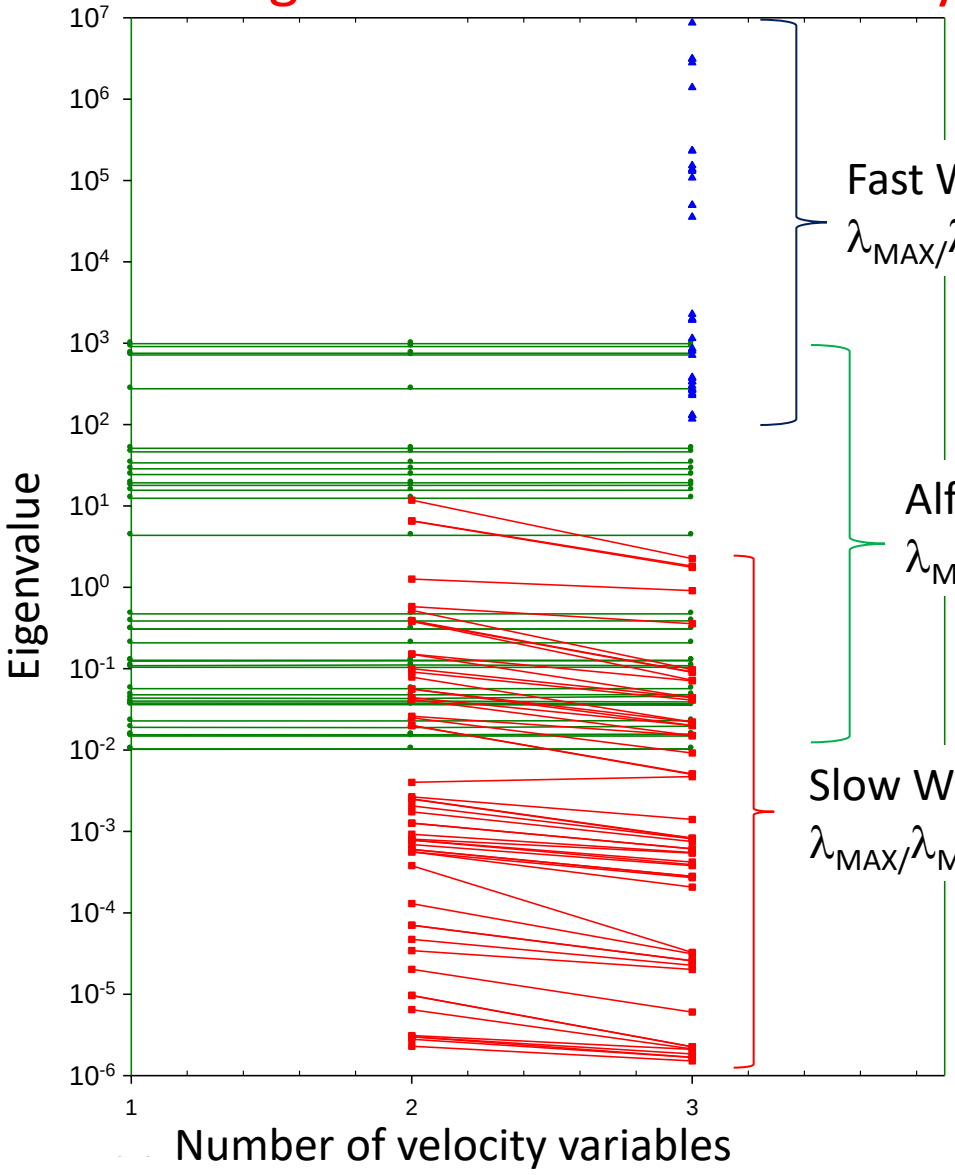
We propose rearranging the locations of the unknowns in the DOF vector, putting all similar velocity components together.

$$\mathbf{X}_j = \begin{bmatrix} \mathbf{U}_{1,j} \\ \boldsymbol{\omega}_{1,j} \\ \boldsymbol{\chi}_{1,j} \\ \mathbf{U}_{2,j} \\ \boldsymbol{\omega}_{2,j} \\ \boldsymbol{\chi}_{2,j} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{U}_{M,j} \\ \boldsymbol{\omega}_{M,j} \\ \boldsymbol{\chi}_{M,j} \end{bmatrix} \Rightarrow \mathbf{X}_j = \begin{bmatrix} \mathbf{U}_{1,j} \\ \mathbf{U}_{2,j} \\ \vdots \\ \mathbf{U}_{M,j} \\ \boldsymbol{\omega}_{1,j} \\ \boldsymbol{\omega}_{2,j} \\ \vdots \\ \boldsymbol{\omega}_{M,j} \\ \boldsymbol{\chi}_{1,j} \\ \boldsymbol{\chi}_{2,j} \\ \vdots \\ \boldsymbol{\chi}_{M,j} \end{bmatrix}$$

Within each plane (with M elements), the unknown vector will first have all the $\mathbf{U}$ variables, next all the $\boldsymbol{\omega}$ variables, and finally all the $\boldsymbol{\chi}$ variables.

The rational for this is that these variables couple most strongly to themselves.

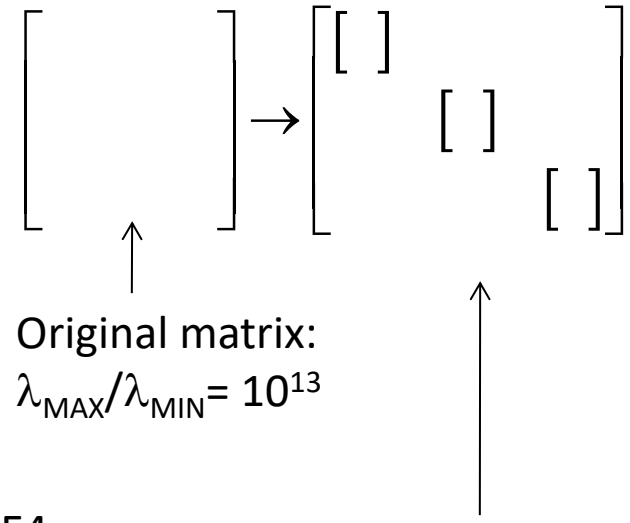
M3D-C^1 can be run with 1, 2, or 3 velocity variables. Tracking the eigenvalues shows how they separate into 3 groups



Fast Wave
$\lambda_{MAX}/\lambda_{MIN}$ =7.3 E4

Alfven Wave
$\lambda_{MAX}/\lambda_{MIN}$ =9.9 E4

Slow Wave
$\lambda_{MAX}/\lambda_{MIN}$ =1.46 E6



Original matrix:
$\lambda_{MAX}/\lambda_{MIN}= 10^{13}$

Annihilation operators split this into 3 loosely connected matrices, each with a condition number much less than original matrix

New form for velocity matrix

$$\begin{bmatrix}
\begin{bmatrix} [b_{11}] & b_{12} & b_{13} \\ b_{21} & [b_{22}] & b_{23} \\ b_{31} & b_{32} & [b_{33}] \end{bmatrix}_1 & \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}_1 & & \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}_1 \\
\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}_2 & \begin{bmatrix} [b_{11}] & b_{12} & b_{13} \\ b_{21} & [b_{22}] & b_{23} \\ b_{31} & b_{32} & [b_{33}] \end{bmatrix}_2 & \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}_2 & \\
& \begin{bmatrix} \ddots & & \\ & \ddots & \\ & & \ddots \end{bmatrix} & \begin{bmatrix} [\ddots] & & \\ & [\ddots] & \\ & & [\ddots] \end{bmatrix} & \begin{bmatrix} \ddots & & \\ & \ddots & \\ & & \ddots \end{bmatrix} \\
\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}_N & & \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}_N & \begin{bmatrix} [b_{11}] & b_{12} & b_{13} \\ b_{21} & [b_{22}] & b_{23} \\ b_{31} & b_{32} & [b_{33}] \end{bmatrix}_N
\end{bmatrix} \bullet \begin{bmatrix} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\omega} \\ \boldsymbol{\chi} \end{bmatrix} \\ \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\omega} \\ \boldsymbol{\chi} \end{bmatrix} \\ \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix} \\ \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\omega} \\ \boldsymbol{\chi} \end{bmatrix} \end{bmatrix} = RHS$$

Instead of a single block for each plane, we now will have 3 blocks for each plane. This should be more efficient and more parallel.

Summary

- We propose reordering the unknowns for the velocity vector so that all similar velocity types ($\mathbf{U}$, ω, χ) are ordered to be adjacent to one another on each plane.

- This will result in 3 times the number of blocks in the block-Jacobi solve…increased concurrency and smaller direct solves.

- This will require help from SCOREC to reorder the unknowns on each plane.

- STRUMPACK may also be able to take advantage of this reordering.

- This is a generalization of using knowledge of the “physics couplings” to produce diagonally dominant block matrix.