New stream function approach method for Magnetohydrodynamics

Kab Seok Kang
Computational Science Center
Brookhaven National Laboratory

January 12, 2005
Presentation Plan

1. MHD equations and its stream function approach
   - Strauss and Longcope’s stream function approach
   - New stream function approach
2. Finite element formulations
3. Nonlinear and linear solvers
   - Nonlinear Gauss-Seidel method
   - Newton method
   - GMRES and Multigrid preconditioner
   - Implementation
4. Numerical Experiments : Tilt Instability
5. Conclusions
MHD equations and its stream function approach

The incompressible MHD equations:

\[
\frac{\partial B}{\partial t} = \nabla \times (v \times B),
\]

\[
\frac{\partial v}{\partial t} = -v \cdot \nabla v + (\nabla \times B) \times B + \mu \nabla^2 v,
\]

\[
\nabla \cdot v = 0,
\]

\[
\nabla \cdot B = 0.
\]

(1)

\(B\) : the magnetic field, \(v\) : the velocity, \(\mu\) : the viscosity
* Previous stream function approach by Strauss and Longcope

Define two stream functions $\phi$ and $\psi$:

$$v = \left( \frac{\partial \phi}{\partial y}, -\frac{\partial \phi}{\partial x} \right), \quad B = \left( \frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right).$$

$$\frac{\partial}{\partial t} \Omega + [\Omega, \phi] = [C, \psi] + \mu \nabla^2 \Omega,$$

$$\frac{\partial}{\partial t} \psi + [\psi, \phi] = 0,$$

$$\nabla^2 \phi = \Omega,$$

$$C = \nabla^2 \psi,$$

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \quad [a, b] = \frac{\partial a}{\partial x} \frac{\partial b}{\partial y} - \frac{\partial a}{\partial y} \frac{\partial b}{\partial x}. $$
More symmetrical equations

\[
\frac{\partial}{\partial t} \Omega + [\Omega, \phi] = [C, \psi] + \mu \nabla^2 \Omega,
\]

\[
\frac{\partial}{\partial t} C + [C, \phi] = [\Omega, \psi] + 2 \left[ \frac{\partial \phi}{\partial x}, \frac{\partial \psi}{\partial x} \right] + 2 \left[ \frac{\partial \phi}{\partial y}, \frac{\partial \psi}{\partial y} \right]
\]

\[
\nabla^2 \phi = \Omega,
\]

\[
\nabla^2 \psi = C.
\]

with Dirichlet boundary conditions of \( \phi \) and \( \psi \).

- Compute the second derivatives of potentials \( \phi \) and \( \psi \).
* New stream function approach

\[ \mathbf{v} = (v_1, v_2) = \left( \frac{\partial \phi}{\partial y}, -\frac{\partial \phi}{\partial x} \right), \quad B = (B_1, B_2) = \left( \frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \]

\[ \frac{\partial \Omega}{\partial t} + (v_1, v_2) \cdot \nabla \Omega = (B_1, B_2) \cdot \nabla C + \mu \nabla^2 \Omega, \]

\[ \frac{\partial C}{\partial t} + (v_1, v_2) \cdot \nabla C = (B_1, B_2) \cdot \nabla \Omega + 2([v_1, B_1] + [v_2, B_2]), \]

\[ -\nabla^2 v_1 = -\frac{\partial \Omega}{\partial y}, \quad -\nabla^2 v_2 = \frac{\partial \Omega}{\partial x}, \]

\[ -\nabla^2 B_1 = -\frac{\partial C}{\partial y}, \quad -\nabla^2 B_2 = \frac{\partial C}{\partial x} \]

\[ \nabla^2 \phi = \Omega, \quad \nabla^2 \psi = C. \]
Finite element formulations

Variational formula for (4):
Find \((\Omega, C, v_1, v_2, B_1, B_2) \in H^{1,\Omega} \times H^1 \times H^{1,v_1} \times H^{1,v_2} \times H^{1,B_1} \times H^{1,B_2}\) such that

\[M_t(\Omega, u) + (v \cdot \nabla \Omega, u) + \mu a(\Omega, u) = (B \cdot \nabla C, u), \quad \forall u \in H^{1,\Omega'},\]
\[M_t(C, v) + (v \cdot \nabla C, w) = (B \cdot \nabla \Omega, w) + 2P(v_1, B_1, w) + 2P(v_2, B_2, w), \quad \forall w \in H^1,\]

\[a(v_1, p_1) = -M \left( \frac{\partial \Omega}{\partial y}, p_1 \right), \quad \forall p_1 \in H^{1,v_1'},\]
\[a(v_2, p_2) = M \left( \frac{\partial \Omega}{\partial x}, p_2 \right), \quad \forall p_2 \in H^{1,v_2'},\]
\[a(B_1, q_1) = -M \left( \frac{\partial C}{\partial y}, q_1 \right), \quad \forall q_1 \in H^{1,B_1'},\]
\[a(B_2, q_2) = M \left( \frac{\partial C}{\partial x}, q_2 \right), \quad \forall q_2 \in H^{1,B_2'},\]

(5)
\[ M_t(u, w) = \int_K \frac{\partial}{\partial t} u wdx, \quad M(u, w) = \int_K u wdx, \quad (u, w) = \int_K u wdx, \]
\[ a(u, w) = \int_K \nabla u \cdot \nabla wdx - \int_{\partial K} \mu \frac{\partial u}{\partial n} wds, \quad P(u, v, w) = \int_K [u, v] wdx. \]

\( H^{1,A} \): the subset of \( H^1(K) \) which satisfy the boundary condition of \( A \)

\( H^{1,A'} \): the subspace of \( H^1(K) \) which have zero values on the Dirichlet boundary of \( A = \Omega, v_1, v_2, B_1, B_2 \).

**Finite element spaces**

\( K_h \): given triangulation of domain \( K \) with the maximum diameter \( h \).

The linear finite element spaces

\[ V_h = \{ v \in L^2(K) : v \text{ is linear on each element triangles of } K_h \text{ and continuous in } K \}. \]

\( V_h^A, A = \Omega, v_1, v_2, B_1, B_2 \): the subsets of \( V_h \) which satisfied the boundary conditions of \( A \) on every boundary points of \( K_h \)

\( V_h^{A'} \): the subspaces of \( V_h \) and \( H^{1,A'} \).
Discretization of time: The first order backward difference scheme, implicit scheme (allowing much larger time steps).

Let \( X = (\Omega, C, v_1, v_2, B_1, B_2) \)

\[
F_1^n(X, u) = \frac{1}{\Delta t} M(\Omega, u) + (\mathbf{v} \cdot \nabla \Omega, u) + \mu a(\Omega, u) - (B \cdot \nabla C, u) - \frac{1}{\Delta t} M(\Omega^{n-1}, u),
\]

\[
F_2^n(X, w) = \frac{1}{\Delta t} M(C, w) + (\mathbf{v} \cdot \nabla C, w)
\]

\[-(B \cdot \nabla \Omega^n, w) - 2P(v_1, B_1, w) - 2P(v_2, B_2, w) - \frac{1}{\Delta t} M(C^{n-1}, w),\]

\[
F_3^n(X, p_1) = a(v_1, p_1) + M \left( \frac{\partial \Omega}{\partial y}, p_1 \right) = F_3((\Omega, v_1), p_1),
\]

\[
F_4^n(X, p_2) = a(v_2, p_2) - M \left( \frac{\partial \Omega}{\partial x}, p_2 \right) = F_4((\Omega, v_2), p_2),
\]

\[
F_5^n(X, q_1) = a(B_1, q_1) + M \left( \frac{\partial C}{\partial y}, q_1 \right) = F_5((\Omega, B_1), q_1),
\]

\[
F_6^n(X, q_2) = a(B_2, q_2) - M \left( \frac{\partial C}{\partial x}, q_2 \right) = F_6((\Omega, B_2), q_2).
\]
Discretized problem

For each discretized times $n$, find the solutions

$$(\Omega^n_h, C^n_h, v^n_{1,h}, v^n_{2,h}, B^n_{1,h}, B^n_{2,h}) \in V_h^\Omega \times V_h \times V_h^{v_1} \times V_h^{v_2} \times V_h^{B_1} \times V_h^{B_2}$$

which are satisfied the following equations

$$F^n_1((\Omega^n_h, C^n_h, v^n_{1,h}, v^n_{2,h}, B^n_{1,h}, B^n_{2,h}), u) = 0.0,$$
$$F^n_2((\Omega^n_h, C^n_h, v^n_{1,h}, v^n_{2,h}, B^n_{1,h}, B^n_{2,h}), w) = 0.0,$$
$$F^n_3((\Omega^n_h, v^n_{1,h}), p) = 0.0,$$
$$F^n_4((\Omega^n_h, v^n_{2,h}), p) = 0.0,$$
$$F^n_5((C^n_h, B^n_{1,h}), q) = 0.0,$$
$$F^n_6((C^n_h, B^n_{2,h}), q) = 0.0,$$

for all $u \in V_h^\Omega'$, $w \in V_h$, $p_1 \in V_h^{v_1'}$, $p_2 \in V_h^{v_2'}$, $q_1 \in V_h^{B_1'}$, and $q_2 \in V_h^{B_2'}$. 
Nonlinear and linear solvers

The first nonlinear Gauss-Seidel (GS1) : For each time $n$, let any initial solution $(\Omega_{h}^{n,0}, C_{h}^{n,0}, v_{1,h}^{n,0}, v_{2,h}^{n,0}, B_{1,h}^{n,0}, B_{2,h}^{n,0})$. For $k = 1, 2, \ldots$,

1. Get $\Omega_{h}^{n,k} = \Omega_{h}^{n,k-1} + \delta \Omega$ by solving the equation
   \[
   \frac{\partial F_{1}^{n}}{\partial \Omega} (\delta \Omega, v_{1,h}^{n,k-1}, v_{2,h}^{n,k-1}, u) = -F_{1}^{n}((\Omega_{h}^{n,k-1}, C_{h}^{n,k-1}, v_{1,h}^{n,k-1}, v_{2,h}^{n,k-1}, B_{1,h}^{n,k-1} B_{2,h}^{n,k-1}), u).
   \]

2. Get $C_{h}^{n,k} = C_{h}^{n,k-1} + \delta C$ by solving the equation
   \[
   \frac{\partial F_{2}^{n}}{\partial C} (\delta C, v_{1,h}^{n,k-1}, v_{2,h}^{n,k-1}, w) = -F_{2}^{n}((\Omega_{h}^{n,k}, C_{h}^{n,k-1}, v_{1,h}^{n,k-1}, v_{2,h}^{n,k-1}, B_{1,h}^{n,k-1} B_{2,h}^{n,k-1}), w).
   \]

3. Get $v_{i,h}^{n,k} = v_{i,h}^{n,k-1} + \delta v_{i}$ and $B_{i,h}^{n,k} = B_{i,h}^{n,k-1} + \delta B_{i}$, for $i = 1, 2$, by solving the equations
   \[
   a(\delta v_{1}, p_{1}) = -F_{3}((\Omega_{h}^{n,k}, v_{1,h}^{n,k-1}), p_{1}), \quad a(\delta v_{2}, p_{2}) = -F_{4}((\Omega_{h}^{n,k}, v_{2,h}^{n,k-1}), p_{2}),
   \]
   \[
   a(\delta B_{1}, q_{1}) = -F_{5}((C_{h}^{n,k}, B_{1,h}^{n,k-1}), q_{1}), \quad a(\delta B_{2}, q_{2}) = -F_{6}((C_{h}^{n,k}, B_{2,h}^{n,k-1}), q_{2}).
   \]
   until
   \[
   \| F((\Omega_{h}^{n,k}, C_{h}^{n,k}, v_{1,h}^{n,k}, v_{2,h}^{n,k}, B_{1,h}^{n,k}, B_{2,h}^{n,k}), (u, w, p_{1}, p_{2}, q_{1}, q_{2})) \| \leq \gamma_{n},
   \]
The second nonlinear Gauss-Seidel (GS2): For each time \( n \), let any initial solution \((\Omega^{n,0}_h, C^{n,0}_h, v^{n,0}_{1,h}, v^{n,0}_{2,h}, B^{n,0}_{1,h}, B^{n,0}_{2,h})\). For \( k = 1, 2, \ldots \),

1. Get \( \Omega^{n,k}_h = \Omega^{n,k-1}_h + \delta \Omega \) and \( C^{n,k}_h = C^{n,k-1}_h + \delta C \) by solving the equation

\[
\frac{\partial (F^1_n, F^2_n)}{\partial (\Omega, C)} (\delta \Omega, \delta C, v^{n,k-1}_{1,h}, v^{n,k-1}_{2,h}, B^{n,k-1}_{1,h} B^{n,k-1}_{2,h}, u, w) = \begin{pmatrix}
-F^1_n ((\Omega^{n,k-1}_h, C^{n,k-1}_h, v^{n,k-1}_{1,h}, v^{n,k-1}_{2,h}, B^{n,k-1}_{1,h} B^{n,k-1}_{2,h}), u) \\
-F^2_n ((\Omega^{n,k}_h, C^{n,k-1}_h, v^{n,k-1}_{1,h}, v^{n,k-1}_{2,h}, B^{n,k-1}_{1,h} B^{n,k-1}_{2,h}), w)
\end{pmatrix}.
\]

2. Get \( v^{n,k}_{i,h} = v^{n,k-1}_{i,h} + \delta v_i \) and \( B^{n,k}_{i,h} = B^{n,k-1}_{i,h} + \delta B_i \), for \( i = 1, 2 \), by solving the equations

\[
a(\delta v_1, p_1) = -F_3 ((\Omega^{n,k}_h, v^{n,k-1}_{1,h}), p_1),
a(\delta v_2, p_2) = -F_4 ((\Omega^{n,k}_h, v^{n,k-1}_{2,h}), p_2),
a(\delta B_1, q_1) = -F_5 ((C^{n,k}_h, B^{n,k-1}_{1,h}), q_1),
a(\delta B_2, q_2) = -F_6 ((C^{n,k}_h, B^{n,k-1}_{2,h}), q_2).
\]

until

\[\|F((\Omega^{n,k}_h, C^{n,k}_h, v^{n,k}_{1,h}, v^{n,k}_{2,h}, B^{n,k}_{1,h}, B^{n,k}_{2,h}), (u, w, p_1, p_2, q_1, q_2))\| \leq \gamma_n.\]
\[ \| F((\Omega, C, v_1, v_2, B_1, B_2), (u, w, p_1, p_2, q_1, q_2)) \| ^2 = \| F_1^n((\Omega, C, v_1, v_2, B_1, B_2), u) \| ^2 \\
+ \| F_2^n((\Omega, C, v_1, v_2, B_1, B_2), w) \| ^2 + \| F_3((\Omega, v_1), p_1) \| ^2 \\
+ \| F_4((\Omega, v_2), p_2) \| ^2 + \| F_5((C, B_1), q_1) \| ^2 + \| F_6((C, B_2), q_2) \| ^2, \]

\( \gamma_n \) : given convergence tolerance.

\[ \frac{\partial F_1}{\partial \Omega}(\delta \Omega, v_1, v_2, u) = \frac{1}{\Delta t} M(\delta \Omega, u) + ((v_1, v_2) \cdot \nabla \delta \Omega, u) + \mu a(\delta \Omega, u). \]

\[ \frac{\partial F_2}{\partial C}(\delta C, v_1, v_2, w) = \frac{1}{\Delta t} M(\delta C, w) + ((v_1, v_2) \cdot \nabla \delta C, w). \]

\[ \frac{\partial (F_1^n, F_2^n)}{\partial (\Omega, C)}(\delta \Omega, \delta C, v_1, v_2, B_1, B_2, u, v) = \begin{pmatrix} \frac{\partial F_1^n}{\partial \Omega} & \frac{\partial F_1^n}{\partial C} \\ \frac{\partial F_2^n}{\partial \Omega} & \frac{\partial F_2^n}{\partial C} \end{pmatrix} (\delta \Omega, \delta C, v_1, v_2, B_1, B_2, u, v) \]

\[ = \begin{pmatrix} \frac{\partial F_1}{\partial \Omega}(\delta \Omega, v_1, v_2, u) & -(B_1, B_2) \cdot \nabla \delta C, u) \\ -(B_1, B_2) \cdot \nabla \delta \Omega, w) & \frac{\partial F_2}{\partial C}(\delta C, v_1, v_2, w) \end{pmatrix} \]
**Newton Method (NM)** : For each time $n$, let be any initial solution $(\Omega_h^n, C_h^n, v_{1,h}^n, v_{2,h}^n, B_{1,h}^n, B_{2,h}^n)$. Get

$$
\begin{pmatrix}
\Omega_h^{n,k} \\
C_h^{n,k} \\
v_{1,h}^{n,k} \\
v_{2,h}^{n,k} \\
B_{1,h}^{n,k} \\
B_{2,h}^{n,k}
\end{pmatrix} =
\begin{pmatrix}
\Omega_h^{n,k} \\
C_h^{n,k-1} \\
v_{1,h}^{n,k-1} \\
v_{2,h}^{n,k-1} \\
B_{1,h}^{n,k-1} \\
B_{2,h}^{n,k-1}
\end{pmatrix}
+ \begin{pmatrix}
\delta \Omega \\
\delta C \\
\delta v_1 \\
\delta v_2 \\
\delta B_1 \\
\delta B_2
\end{pmatrix},
$$

for $k = 1, 2, \ldots$, until satisfies

$$
\| F((\Omega_h^{n,k}, C_h^{n,k}, v_{1,h}^{n,k}, v_{2,h}^{n,k}, B_{1,h}^{n,k}, B_{2,h}^{n,k}), (u, w, p_1, p_2, q_1, q_2)) \| \leq \gamma_n,
$$
where \((\delta \Omega, \delta C, \delta v_1, \delta v_2, \delta B_1, \delta B_2)\) is from the linear equation

\[
\frac{\partial (F_1^n, F_2^n, F_3, F_4, F_5, F_6)}{\partial (\Omega, C, v_1, v_2, B_1, B_2)} (\delta \Omega, \delta C, \delta v_1, \delta v_2, \delta B_1, \delta B_2, u, w, p_1, p_2, q_1, q_2) =
\begin{bmatrix}
-F_1^n ((\Omega_{h}^{n,k-1}, C_h^{n,k-1}, v_1^{n,k-1}, v_2^{n,k-1}, B_{1,h}^{n,k-1} B_{2,h}^{n,k-1}), u) \\
-F_2^n ((\Omega_{h}^{n,k-1}, C_h^{n,k-1}, v_1^{n,k-1}, v_2^{n,k-1}, B_{1,h}^{n,k-1} B_{2,h}^{n,k-1}), w) \\
-F_3 ((\Omega_{h}^{n,k-1}, v_1^{n,k-1}, p_1) \\
-F_4 ((\Omega_{h}^{n,k-1}, v_2^{n,k-1}, p_2) \\
-F_5 ((C_h^{n,k-1}, B_{1,h}^{n,k-1}), q_1) \\
-F_6 ((C_h^{n,k-1}, B_{2,h}^{n,k-1}), q_2)
\end{bmatrix},
\]

where \(\frac{\partial (F_1^n, F_2^n, F_3, F_4, F_5, F_6)}{\partial (\Omega, C, v_1, v_2, B_1, B_2)}\) be the Jacobian, i.e.,

\[
J_k = 
\begin{bmatrix}
\frac{\partial F_1^n}{\partial \Omega} & \frac{\partial F_1^n}{\partial C} & \frac{\partial F_1^n}{\partial v_1} & \frac{\partial F_1^n}{\partial v_2} & \frac{\partial F_1^n}{\partial B_1} & \frac{\partial F_1^n}{\partial B_2} \\
\frac{\partial F_2^n}{\partial \Omega} & \frac{\partial F_2^n}{\partial C} & \frac{\partial F_2^n}{\partial v_1} & \frac{\partial F_2^n}{\partial v_2} & \frac{\partial F_2^n}{\partial B_1} & \frac{\partial F_2^n}{\partial B_2} \\
\frac{\partial F_3}{\partial \Omega} & 0 & \frac{\partial F_3}{\partial v_1} & 0 & 0 & 0 \\
\frac{\partial F_4}{\partial \Omega} & 0 & 0 & \frac{\partial F_4}{\partial v_2} & 0 & 0 \\
0 & \frac{\partial F_5}{\partial C} & 0 & 0 & \frac{\partial F_5}{\partial B_1} & 0 \\
0 & \frac{\partial F_6}{\partial C} & 0 & 0 & 0 & \frac{\partial F_6}{\partial B_2}
\end{bmatrix}.
\]
Krylov space method and Multigrid preconditioners

* Krylove method:
  - Can be preconditioned for efficiency.
  - GMRES: Guarantee convergence with nonsymmetric, nonpositive definite systems, memory intensive, expensive.
  - Restarted GMRES: Does not memory intensive and inexpensive, lack a theory of convergence

* Preconditioning:

\[ J_k \delta x_k = -F(x_k), \quad P_k^{-1} J_k \delta x_k = -P_k^{-1} F(x_k) \]  \hspace{1cm} (6)

- Straight forward implementation.
- Two systems are equivalent for any nonsingular operator \( P_k^{-1} \).
- Preconditioner determines the rate of convergence of GMRES, and hence the efficiency of the algorithm.

* Multigrid preconditioner:
- Apply one time multigrid iteration on Jacobian or reduced system.
- Well-known scalable preconditioner for many problems.
- Multigrid preconditioner may cause more ill-conditioned system if original system are nonsymmetric and highly ill-conditioned.

- To get numerically stable multigrid preconditioner, use reduced system or symmetrized reduced system instead of original one for preconditioner.

\[
J_{R,k} = \begin{pmatrix}
\frac{\partial F_1^n}{\partial \Omega} & 0 \\
0 & \frac{\partial F_2^n}{\partial C}
\end{pmatrix}, \quad J_{R,k} = \begin{pmatrix}
\frac{\partial F_1^n}{\partial \Omega} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\partial F_2^n}{\partial C} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{\partial F_3}{\partial v_1} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\partial F_4}{\partial v_2} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{\partial F_5}{\partial B_1} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{\partial F_6}{\partial B_2}
\end{pmatrix}.
\]

\[
J_{S,k} = \frac{1}{2} \left( J_{R,k} + J_{R,k}^T \right).
\]

- May need more iterations to solve linear system.
**Implementation**

* PETSc library

- well developed library to solve nonlinear PDE problems
- easily implemented multigrid preconditioner on GMRES.
- use the index sets for parallelization of finite element discretization on unstructured meshes.

* Mesh generation

- Generate finer grids by recursively subdividing the coarsest grid.
- Use two global numbers, one is from coarsest meshes (does not depend on the number of processors) and another is according to processors and used for numbering of vectors.
- Use index sets to match numberings the vertices which are included in many processors.
Numerical Experiments : Tilt Instability

* Initial condition \((k = 3.831705970 \ (J_1(k) = 0))\): From

\[
\phi(0) = 10^{-3} e^{-(x^2+y^2)}, \psi(0) = \begin{cases} 
-1.295961618J_1(kr)y/r & \text{if } r < 1 \\
-(\frac{1}{r} - r)y/r & \text{if } r > 1
\end{cases}
\]

\[\Omega(0) = 0.0, \quad C(0) = \begin{cases} 
19.0272743J_1(kr)y/r & \text{if } r < 1 \\
0.0 & \text{if } r > 1
\end{cases}\]

\[v_1(0) = -2y10^{-3} e^{-(x^2+y^2)}, \quad v_2(0) = 2x10^{-3} e^{-(x^2+y^2)},\]

\[B_1(0) = \frac{\partial \psi(0)}{\partial y} = \begin{cases} 
-1.295961618 \left( \frac{ky^2}{r^2} J_0(kr) + \frac{x^2-y^2}{r^3} J_1(kr) \right), & \text{if } r < 1 \\
1 - \frac{x^2-y^2}{r^4}, & \text{if } r > 1
\end{cases}\]

\[B_2(0) = -\frac{\partial \psi(0)}{\partial x} = \begin{cases} 
1.295961618 \left( \frac{kxy}{r^2} J_0(kr) - \frac{2xy}{r^3} J_1(kr) \right), & \text{if } r < 1 \\
-\frac{2xy}{r^4}, & \text{if } r > 1
\end{cases}\]
* Boundary condition for $[-R, R] \times [-R, R]$ : Dirichlet boundary conditions on $\phi$ and $\psi$ for conservation of energy.

\[
\Omega(x, y, t) = 0.0, \quad \frac{\partial C}{\partial n}(x, y, t) = 0.0,
\]

\[
\begin{align*}
&v_1(x, y, t) = 0.0, \quad \text{on } x = \pm R, \\
&\frac{\partial v_1}{\partial n}(x, y, t) = 0.0, \quad \text{on } y = \pm R ,
\end{align*}
\]

\[
\begin{align*}
&v_2(x, y, t) = 0.0, \quad \text{on } y = \pm R, \\
&\frac{\partial v_2}{\partial n}(x, y, t) = 0.0, \quad \text{on } x = \pm R ,
\end{align*}
\]

\[
\begin{align*}
&B_1(x, y, t) = 1 - \frac{x^2-y^2}{r^4}, \quad \text{on } x = \pm R, \\
&\frac{\partial B_1}{\partial n}(x, y, t) = \begin{cases} 
-\frac{2y(3x^2-y^2)}{(x^2+y^2)^3}, & \text{on } y = -R \\
\frac{2y(3x^2-y^2)}{(x^2+y^2)^3}, & \text{on } y = R
\end{cases}, \\
&B_2(x, y, t) = -\frac{2xy}{r^4}, \quad \text{on } y = \pm R, \\
&\frac{\partial B_2}{\partial n}(x, y, t) = \begin{cases} 
-\frac{2y(3x^2-y^2)}{(x^2+y^2)^3}, & \text{on } x = -R \\
\frac{2y(3x^2-y^2)}{(x^2+y^2)^3}, & \text{on } x = R
\end{cases}.
\end{align*}
\]

If need $\phi$ and $\psi$, use $\phi(x, y, t) = 0.0, \quad \psi(x, y, t) = y - \frac{y}{x^2+y^2}$
Preview of highlights

- Effect of domain resolution: The solutions of two approaches are closer when the domain are increased.

- GS2 and Newton method are more numerically stable than GS1 in solving nonlinear

- When the original system are ill-conditioned, i.e., time step size is large and/or the value of velocity and magnetic field are big, we have to use multigrid preconditioner applying on the symmetrized reduced system to get good numerical stability.

- Newton method with multigrid preconditioner has a better scalability
Fig. 1. Triangulations.
\[ 0 \mathcal{V} = \mathcal{T}(q) \]
\[ 0 = t'(c) \]
0.2 = \mathcal{L}(p)
Effect of domain resolution

(a) Initial contours $R = 3$

(b) Initial contours $R = 2$

*Effect of domain resolution*
Previous scheme (use $\phi$, $\psi$ and its derivatives)

- Contours at $t = 7.0$ and $R = 3$. 

(d) Contours at $t = 7.0$ and $R = 3$. 

(e) Contours at $t = 7.0$ and $R = 3$. 

27
Fig. 3. Contours of $\psi$.

New scheme (use $(v_1, v_2, B_1, B_2)$).

$\tilde{z} = H$ and $\tilde{z} = t$ contours at $\tilde{z} = 3$.

$\tilde{z} = 0$ and $\tilde{z} = t$ contours at $\tilde{z} = 3$. 
Table 1. Average growth rate $\gamma$ of kinetic energy from $t = 0.0$ to $t = 6.0$

<table>
<thead>
<tr>
<th></th>
<th>previous, $R = 2$</th>
<th>previous, $R = 3$</th>
<th>new, $R = 2$</th>
<th>new, $R = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma$</td>
<td>2.167</td>
<td>2.152</td>
<td>1.744</td>
<td>2.102</td>
</tr>
</tbody>
</table>

- The solutions of two approaches are closer when the domain are increased previous approach from the high growth rate and new approach from the low growth rate.
Fig. 4. Kinetic Energy.
* Convergence behavior
- level = 5, $R = 3$
- nonlinear problem tolerance : $10^{-8}$ (absolute)
- linear problem tolerance : $10^{-6}$ (relative)
- at $t = 0.0$ and $t = 6.0$. 
Table 2. The average number of nonlinear iterations of one time step according to time step sizes $dt$.

<table>
<thead>
<tr>
<th>starting time</th>
<th>$dt$</th>
<th>GS1</th>
<th>GS2</th>
<th>NM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0.0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0005</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.002</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.005</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>*</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td>*</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>*</td>
<td>18</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>$t = 6.0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0005</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.002</td>
<td>5</td>
<td>5</td>
<td>4(5)</td>
<td></td>
</tr>
<tr>
<td>0.005</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>*</td>
<td>8</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td>*</td>
<td>13</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>*</td>
<td>39</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3. *The average number of linear iterations in one time step according to time step sizes*

<table>
<thead>
<tr>
<th>starting time</th>
<th>$dt$</th>
<th>GS1</th>
<th>GS2</th>
<th>GS2(R)</th>
<th>GS2(S)</th>
<th>NM(R)</th>
<th>NM(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0.0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0005</td>
<td>4,4</td>
<td>4</td>
<td>4.3</td>
<td>4.3</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>4,4</td>
<td>4</td>
<td>5.3</td>
<td>5.3</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>0.002</td>
<td>4,4</td>
<td>4.3</td>
<td>6.6</td>
<td>6.6</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>0.005</td>
<td>3,4</td>
<td>*</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>*</td>
<td>*</td>
<td>18</td>
<td>18</td>
<td>18.5</td>
<td>18.5</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td>*</td>
<td>*</td>
<td>28.8</td>
<td>28.8</td>
<td>31.6</td>
<td>31.6</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>*</td>
<td>*</td>
<td>53</td>
<td>53</td>
<td>62.5</td>
<td>62.5</td>
<td></td>
</tr>
<tr>
<td>$t = 6.0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0005</td>
<td>4,4</td>
<td>4.4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>4,4</td>
<td>4</td>
<td>4.5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>0.002</td>
<td>4,4</td>
<td>4</td>
<td>5</td>
<td>5.2</td>
<td>6.8</td>
<td>6.25</td>
<td></td>
</tr>
<tr>
<td>0.005</td>
<td>3.9,4.2</td>
<td>*</td>
<td>7.8</td>
<td>8.8</td>
<td>*</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>15.2</td>
<td>*</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>27.3</td>
<td>*</td>
<td>33.3</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>61</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4. *Average number of iterations from* \( t = 0.0 \) *to* \( t = 0.05 \) *with* \( dt = 0.005 \).

<table>
<thead>
<tr>
<th>Solvers</th>
<th>level</th>
<th>nonlinear</th>
<th>linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS2(S)</td>
<td>4</td>
<td>4</td>
<td>7.9</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.1</td>
<td>11.2</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3</td>
<td>16.1</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>3.4</td>
<td>19.1</td>
</tr>
<tr>
<td>NM(S)</td>
<td>4</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>11.7</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3</td>
<td>16.4</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>3.4</td>
<td>20.1</td>
</tr>
</tbody>
</table>
Table 5. *Average time of one time step according to level and number of processors at $t = 0.0$ and $dt = 0.005$ on the Brookhaven Galaxy Cluster (BGC) - 696 MHz*

<table>
<thead>
<tr>
<th>Solvers</th>
<th>level</th>
<th>Number of processor</th>
<th>solving time of one time step</th>
<th>solving time of linear system</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS2(S)</td>
<td>4</td>
<td>1</td>
<td>13.7</td>
<td>2.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>13.3</td>
<td>2.74</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>42.5</td>
<td>11.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>29.4</td>
<td>7.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>38.6</td>
<td>11.12</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>8</td>
<td>120.5</td>
<td>35.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>64.5</td>
<td>19.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32</td>
<td>118.1</td>
<td>36.95</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>32</td>
<td>226.3</td>
<td>61.8</td>
</tr>
<tr>
<td>NM(S)</td>
<td>4</td>
<td>1</td>
<td>12.3</td>
<td>2.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>7.76</td>
<td>1.40</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>33.5</td>
<td>6.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>21.0</td>
<td>4.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>19.9</td>
<td>4.89</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>8</td>
<td>59.9</td>
<td>14.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>39.0</td>
<td>9.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32</td>
<td>61.6</td>
<td>17.8</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>32</td>
<td>142.9</td>
<td>36.2</td>
</tr>
</tbody>
</table>
Conclusions

* We study the new stream function approach method for Magnetohydrodynamics with tilt instability example.

* Nonlinear Gauss-Seidel (GS2) and Newton method have a similar numerical behaviors and numerical stability.

* Have to use the Multigrid Preconditioner applying on symmetrized reduced system for GS2 and Newton method to get numerical stability.

* Newton method has a better scalability than nonlinear Gauss-Seidel method.
Acknowledgements

Thank to David E. Keyes (Columbia University and Brookhaven National Laboratory).