Simulation of Incompressible magneto-hydrodynamic flows (MHD) using mhdFoam solver.

Anant Talasikar.

Introduction.

This case-study describes the theoretical background and the implementation of OpenFOAM solver suitable for the simulation of incompressible magneto-hydrodynamic (MHD) flows, called as mhdFoam. This class is a topic of interest for many research activities and industrial applications including nuclear fusion reactors, materials engineering and metallurgy. The case that is being simulated is, the modified version of the tutorial case (in opt/ OpenFOAM-4.x/ tutorials/ electromagnetics/ mhdFoam/ hartmann) where the 2D MHD flows that arise in a rectangular duct with walls of arbitrary electrical conductivity when an electric conductive fluid moves in the presence of a transverse magnetic field.

Theoretical background.

A magneto-hydrodynamic flow occurs whenever the motion of an electric conductive fluid happens in the presence of an imposed magnetic field which induces currents and the velocity distribution of the flow is significantly modified by the arising of Lorentz forces which oppose the movement in directions perpendicular to the field lines. However, the influence exerted by the magnetic field is not unidirectional i.e. the same currents generate a secondary magnetic field which, abiding to the Faraday's law, lessen the imposed one. Therefore, the velocity and magnetic field are coupled and the flow features can no longer be described by the ordinary hydrodynamics laws but a new set of governing equations have been developed.

For a laminar and incompressible MHD flow, the governing differential equations are obtained by combining the Navier-Stokes and Maxwell equations. Considering an externally applied magnetic field B_0, the total field acting on the fluid can be represented by the sum of the external and the induced one.

$$B = B_0 + b \tag{1}$$

The magnetic field influence on the flow could be represented by proper source terms added to the following equations.

Conservation of mass

$$\nabla . u = 0 \tag{2}$$

Transport of linear momentum

$$\frac{\partial u}{\partial t} + (u . \nabla) u = -\frac{\nabla . p}{\rho} + \nu \nabla^2 u + \frac{j \times B}{\rho} \tag{3}$$

Transport of magnetic induction

$$\frac{\partial B}{\partial t} + (u . \nabla) B = \frac{1}{\mu_m \sigma} \nabla^2 B + (B . \nabla) u \tag{4}$$

Here μ_m is the magnetic permeability. The lorrentz for term in equation (3) has not been explicitly defined, therefore by recalling fundamental laws of electromagnetism such as,

Ampere's law,

$$\nabla \times B = \mu_m j \tag{5}$$

Therefore the lorrentz force term can be represented as,

$$\frac{j\times B}{\rho} = \frac{1}{\rho\mu}(\nabla\times B\times B) = -\frac{1}{2\rho\mu}\nabla B^2 + \frac{1}{\rho\mu}(B.\nabla)B \quad (6)$$

which has also been depicted in the source code as,

```
fvVectorMatrix UEqn
(
fvm::ddt(U)
+ fvm::div(phi, U)
- fvc::div(phiB, 2.0*DBU*B)
- fvm::laplacian(nu, U)
+ fvc::grad(DBU*magSqr(B))
);
if (piso.momentumPredictor())
        {
           solve(UEqn == -fvc::grad(p));
        }
dimensionedScalar DB = 1.0/(mu*sigma);
DB.name() = "DB";
dimensionedScalar DBU = 1.0/(2.0*mu*rho);
DBU.name() = "DBU";

// --- B-PISO loop
while (bpiso.correct())
{
fvVectorMatrix BEqn
(
fvm::ddt(B)
+ fvm::div(phi, B)
- fvc::div(phiB, U)
- fvm::laplacian(DB, B)
);
```

Therefore to sum it up, according to the Faraday's law of electromagnetic induction the interaction of electrically conducting fluid with imposed magnetic field, generates an electromagnetic force. Simultaneously, electric currents are generated as per ohm's law ($j = \sigma(E + u\times B)$), which give rise to an induced magnetic field (b) which in combination with imposed magnetic field interacts with current density (j) to generate Lorentz force (j × B) . This force tends to retard the flow in core region and thus resulting in flatter velocity profile compared to pure hydrodynamic flow, where we obtain a parabolic flow profile.

Simulation set-up.

The standard tutorial for mhdFoam has been chosen, and the geometry has slightly been modified such that we obtain a domain of parallel plates of infinite flow length, by implementing the periodic boundary condition by making changes to the following files,

1. blockMeshDict.
   boundary
   (
      inlet
      {

```
        type                mappedPatch;
        offset              (2 0 0);
        sampleRegion        region0;
        sampleMode          nearestCell;
        samplePatch         none;
        faces
        (
            (0 4 7 3)
        );
```

2. Velocity (U).

```
inlet
    {
        type                        mapped;
        value                       uniform (1 0 0);
        interpolationScheme         cell;
        setAverage                  true;
        average                     (1 0 0);
    }
outlet
    {
        type            inletOutlet;
        inletValue      uniform (0 0 0);
        value           uniform (0 0 0);
     }
```

This basically maps the values all the computed values at the outlet on to the inlet patch, which allows us to have a smaller duct, coarser mesh, which saves us a lot of computational time. The modified domain is depicted below.

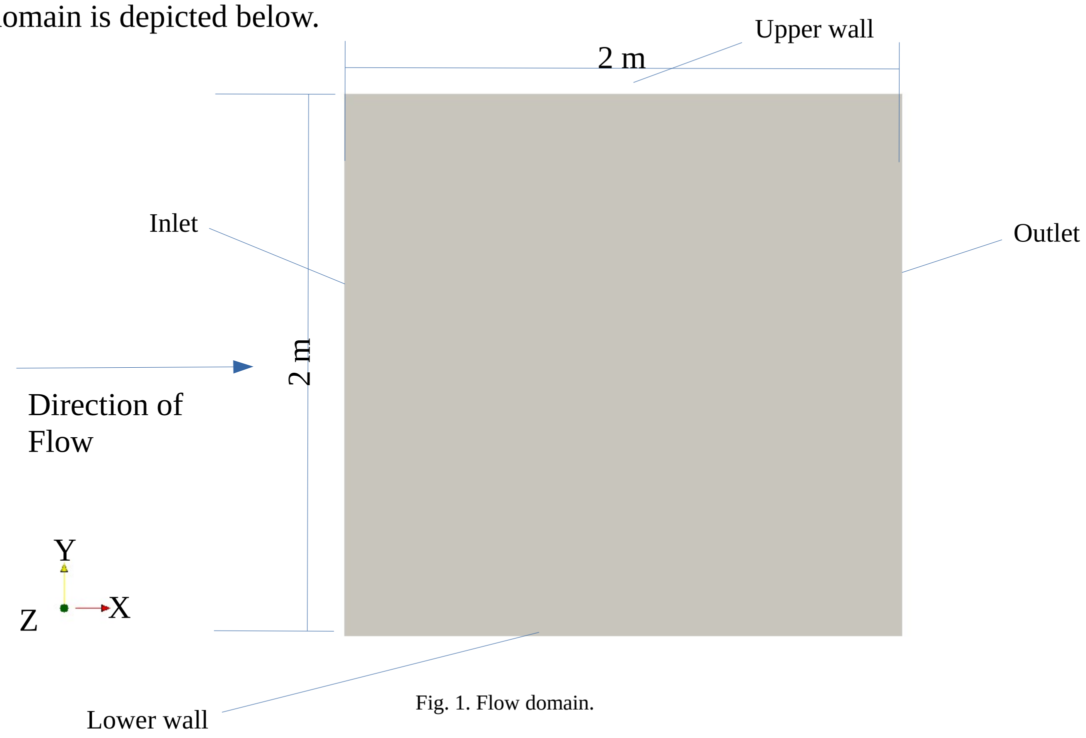

Fig. 1. Flow domain.

The magnetic field is imposed perpendicular to the direction of the flow i.e. on the lower and upper walls, which generates electromagnetic forces within the fluid. The parameter that expresses the relation between, the electromagnetic and viscous forces is called as the hartmann number. Mathematically,

$$Hartmann\ \ number(M)\ =\ \left(\frac{Electromagnetic\ forces}{Viscous\ forces}\right)^{1/2}\ =\ BL\sqrt{\frac{\sigma}{\rho\,\nu}} \tag{7}$$

Here σ is the electrical conductivity, L is half the characteristic length, and B is the applied magnetic field. The greater the hartmann number, higher are the electromagnetic forces and more will be the devation from regular flow. The magnetic field greatly affects the boundary layer. The modified boundary layer is called as the Hartmann layer,

$$Hartmann\ \ layer\ \ thickness(\delta_H)\ =\ \frac{1}{M} \tag{8}$$

It is clear that by increasing the hartmann number the, the hartmann layer thickness reduces, therefore during simulations, five cell points have been kept in the hartmann layer. The flow profile/ velocity distribution completely depends on the Hartmann number and therefore a hypothetical fluid (arbitrary fluid properties) has been chosen for simulations. The velocity profile is given by,

$$U\ =\ U_i\frac{M\,.\cosh M}{M\,.\cosh M-\sinh M}.[1-\frac{\cosh M\,.\,x}{\cosh M}] \tag{9}$$

Here U_i is the inlet velocity (U_i=1 m/s) and x is the y-cordinate of the point along the duct. In this case-study hartmann number is varied by varying, the magnetic field and simulations have been performed Hartmann numbers 10, 50, 100, 500. The post- processing has been done by writing codes in Python 3.6. The y co-ordinates of the cell centres (mesh) have been used to plot the analytical functions.

By substituting $\cosh x\ =\ \frac{e^x+e^{-x}}{2}$ and $\sinh x\ =\ \frac{e^x-e^{-x}}{2}$ in (9) and by further simplifications, we obtain the analytical relation in two seperate parts, for the positive and negative points,

$$U_{positive}\ =\ \frac{M}{M-1}.[\frac{1+e^{-2M}}{1+(\frac{M+1}{M-1}).e^{-2M}}].[1-(\frac{1+e^{-2Mx}}{1+e^{-2M}})].e^{M(x-1)} \tag{10}$$

$$U_{negative}\ =\ \frac{M}{M-1}.[\frac{1+e^{-2M}}{1+(\frac{M+1}{M-1}).e^{-2M}}].[1-(\frac{1+e^{2Mx}}{1+e^{-2M}})].e^{-M(x+1)} \tag{11}$$

The main purpose of developing this expression is to covert the e^x form in (9) to e^{2x} form, which helps us in error minimization. The resulting plots from the two expressions above have been concatenated and have been displayed in the following sections.

Post-processing.

The set of points/ values over which the analytical expression has been evaluated, are the cell centre/ mid point co-ordinates. The plots obtained from simulation have also been developed from the values at these cell centre co-ordinates. This has been done to avoid any kind of interpolation or extrapolation which other post-processing softwares usually do.

These cell centre values have been obtained by utilising the sampling dictionary from the *postProcess* utility. This sampling dictionary, is located in the system folder, and has to be executed using the following command in the main folder.

“postProcess -func sample”

This will create a folder called as postProcessing in the main folder, which would contain postProcessed data for every time step. However, some changes have been made to the original sampling dictionary and the modified dictionary is shown below,

```
type sets;
libs ("libsampling.so");

interpolationScheme          cell;

setFormat                    raw;

sets
(
    line_centreProfile
    {
        type    midPoint;
        axis    distance;
        start   (1 -1 0.01);
        end     (1 1 0.01);
    }
);

fields        ( Ux );
```

The start and the end location cordinates inside the geometry have to be mentioned inside *start* and *end* paranthesis. The field/entities whose data is desired has to be mentioned in the field paranthesis.

Results.

The simulations results agree with the theoretical studies i.e. velocity profile keeps on getting flatter by increasing the Hartmann number. The figure below shows comparison the velocity profiles for flows with various hartmann numbers.

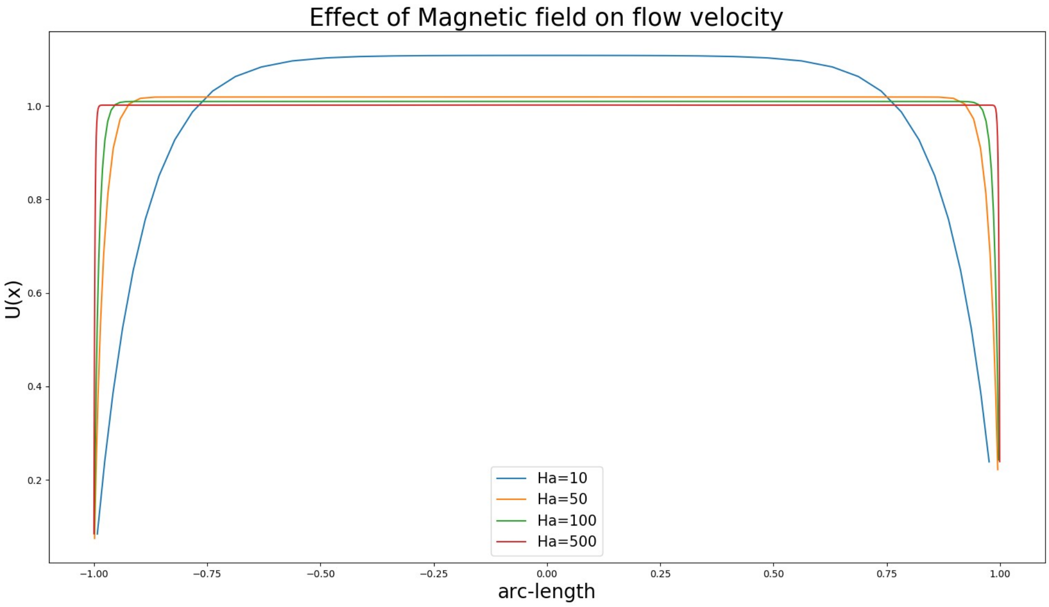

Fig. 2. Comparison of various flow profiles

The simulation results agree very much with the analytical results and the error is below 1% as shown below,

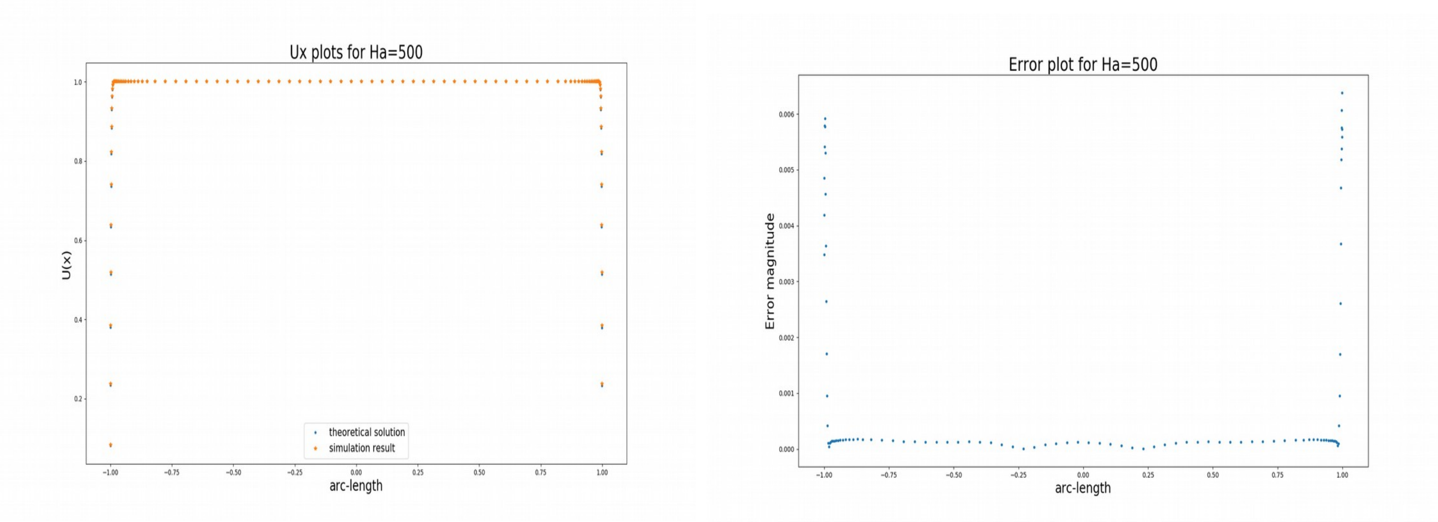

Fig. 3. Comaprison of analytical and simulation flow profiles.

The cellcentre values of the mesh have been plotted in the two graphs above. The number of cells keep on increasing along the transverse direction with the increase in the Hartmann number as the Hartmann layer thickness reduces. The details of the simulation have been tabulated below,

Hartmann number	Number of cells	Simulation time
10	40 x 40 =1600	3.01s
50	40 x 40 =1600	12.89s
100	40 x 60 =2400	35.34s
500	40 x 100 =4000	17 mins 10s

The simulation time is observed to drastically rise with the increase in Hartmann number, because of the Courant number criterion below,

$$C + 2L + \frac{\nu \Delta t}{\Delta x^2} \leq 1 \tag{12}$$

where C is the Courant-Friedrichs-Lewy condition, L is the Von Neumann stability parameter,

$$L = \frac{\sigma B_o^2 \Delta t}{\rho} \tag{13}$$

This is a critical parameter to keep the simulation running, as by increasing the Magnetic field, L increases even more, and therefore to keep the condition satisfied, Δt has to be reduced a lot, which increases simualtion time, which makes higher Hartmann number simulations even more complex.

Conclusion and future work.

The simulation and analytical results have been found to be in a very good agreement, as well as the gradients in the Hartmann layer have been captured. However, some fluctuations in the flow velocity are observed if the the number of cells along the x direction are kept below 40 during simulations. This can be attributed to the numerical instabilities in obtaining the solution of the induction equation.

A more stable solver is needed to be developed to overcome these instabilities for higher Hartmann number simulations as well as to reduce the time complexity.

References.

1. H.G. Weller et al. A tensorial approach to computational continuum mechanics using object-oriented techniques. Computers in physics, 12(6):620–631, 1998.
2. P.A. Davidson. An introduction to magnetohydrodynamics, volume 25. Cambridge university press, 2001.
3. U. Müller and L. Bühler. Magnetofluiddynamics in channels and containers. Springer Science & Business Media, 2013.
4. J.C.R. Hunt. Magnetohydrodynamic flow in rectangular ducts. Journal of Fluid Mechanics, 21(04):577–590, 1965.