Baseline for developing a general OpenFOAM solver for magnetohydrodynamic (MHD) flows

Lorenzo Melchiorri

DIAEE - Nuclear Section,
Sapienza University of Rome,
Rome, Italy

21/01/2022

Fusion reactors

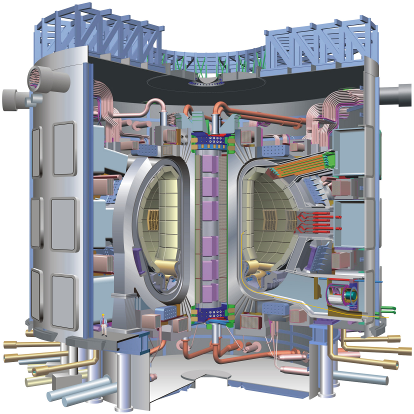


Figure: ITER experiment

Liquid metal blankets are the leading candidate for tritium production in MCF reactors. Interaction between LM and magnetic field cause transition to magnetohydrodynamic flow

MHD-related issues

- High pressure drops
- Enhanced corrosion rates
- Turbulence suppression
- etc.

Codes for MHD

To support the blanket desing a CFD software able to model MHD flows is needed

Required parameters

- $\mathrm{Ha} = 10^4$
- $\mathrm{Re} = 10^4$
- $\mathrm{Gr} = 10^{12}$

No mature MHD code is currently available.

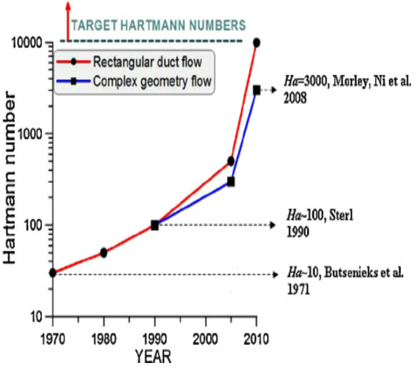


Figure: MHD calculations progress (Smolentsev, 2015)

MHD governing equations

A laminar, isothermal and incompressible flow is assumed

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{D\mathbf{u}}{Dt} = -\nabla(p/\rho) + \nu \nabla^2 \mathbf{u} + (\mathbf{J} \times \mathbf{B})/\rho \quad (2)$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \frac{1}{\mu\,\sigma} \nabla^2 \mathbf{B} \quad (3)$$

$$\mathbf{J} = \frac{1}{\mu} \nabla \times \mathbf{B} \quad (4)$$

This set is called the *B-formulation* of the MHD governing equations

Inductionless approximation

The equation (3) can be simplified, reducing the u-B coupling non-linearity, if the self-induced magnetic field is negligible. This corresponds to the *inductionless* condition

$$R_m \ll 1$$

The parameter $R_m = u_0 \, L / \mu \, \sigma$ is called the magnetic Reynolds number. For the typical values encountered in LM flows the condition is valid and the magnetic field can be uncoupled from the fluid velocity, i.e. it depends just by the boundary conditions.

Electric potential formulation

A Poisson equation for the electric potential and the Ohm's law substitute (3) and (4)

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{D\mathbf{u}}{Dt} = -\nabla(p/\rho) + \nu\nabla^2\mathbf{u} + (\mathbf{J} \times \mathbf{B})/\rho \tag{2}$$

$$\nabla^2(\sigma\phi) = \sigma\nabla \cdot (\mathbf{u} \times \mathbf{B}) \tag{5}$$

$$\mathbf{J} = \sigma(-\nabla\phi + \mathbf{u} \times \mathbf{B}) \tag{6}$$

The new set is called the ϕ-formulation of the MHD equations

Parameters for incompressible LM MHD flow

Hartmann number: adimensional measure of the magnetic field intensity

$$\mathrm{Ha} = BL\sqrt{\frac{\sigma}{\rho\nu}} \tag{7}$$

Wall conductance ratio: measures relative electrical conductivity of the wall compared to the fluid

$$c = \frac{\sigma_w}{\sigma}\,\frac{t}{L} \tag{8}$$

Fundamental phenomena - insulated

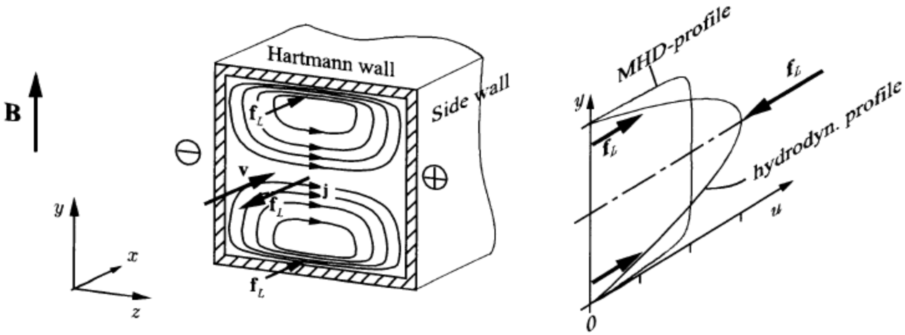


Figure: MHD bounded flow features (Müller and Bühler, 2001)

Fundamental phenomena - other configurations

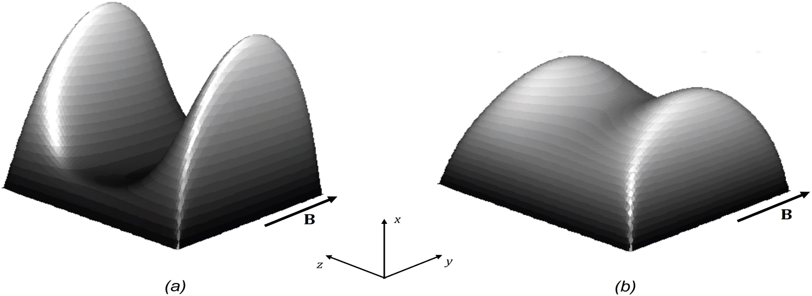


Figure: 3D normalized velocity distribution (Rao and Sankar, 2011). a) Insulated side walls ($c_s = 0$) and perfectly conductive Hartmann ones ($c_h = \infty$). b) Fully conductive channel ($c_s = c_h = \infty$).

In fusion applications $10^{-3} < c < 2 \cdot 10^{-1}$

The built-in solver

The solver `mhdFoam` (PISO-loop based) relies on *B-formulation* of MHD equations. To access the source code:

```
cd $FOAM_SOLVERS/electromagnetics/mhdFoam
```

Upsides

- The solver works properly
- It is able to represent self-induced field

Drawbacks

- Very (computationally) expensive for high Ha
- It can handle only laminar and isothermal problems
- Complex boundary conditions, not well suited to represent conductive walls

Alternative useful tools

Solver `epotFoam` described by Tassone (2017)

Upsides

- It employs the ϕ-formulation
- induction-less

Drawbacks

- It models only perfectly insulating or conducting walls

Solver `Q2DmhdFoam` developed by Iraola (2021)

Upsides

- Temperature field is resolved (buoyancy effects)
- Fast

Drawbacks

- It can only model 2D cases
- It does not solve electric potential and current density

Why employ a solver based on the ϕ-formulation?

Advantages

- Faster
- More stable
- Simpler boundary conditions
- More accurate for coarse mesh

Drawbacks

- Requirement on **J** interpolation
- Nonconservative treatment of Lorentz force
- Constrain on charge conservation $\nabla \cdot \mathbf{J} = 0$

Modified Four Step Projection Method (Ni, 2007) employed for the OpenFOAM implementation.

Coupling solvers strategies

Coupling grid strategy or *monolithic* approach:

- Found in `conjugateHeatFoam` (only foam-extend)
- Direct coupling between fluid/solid domains
- Solving a unique matrix system
- **Disadvantage**: slower at solver level

Segregated method:

- Found in `chtMultiRegionFoam` (in both FE and OpenFOAM)
- Coupling through internal boundary conditions
- Solving a separated matrix system
- **Disadvantage**: iterative process at the interface

The `conjugateHeatFoam` solver has been chosen for the derivation of `MhdPisoFoam`

Implementation - variables

createFields.H and createSolidField.H additions

```
Info<< "Reading field sigma\n" << endl;
 volScalarField sigma
 (
     IOobject
     (
         "sigma",
         runTime.timeName(),
         mesh,
         IOobject::MUST_READ,
         IOobject::AUTO_WRITE
     ),
     mesh
 );

 Info<< "Reading field PotE\n" << endl;
 volScalarField PotE
 (
     IOobject
     (
         "PotE",
         runTime.timeName(),
         mesh,
         IOobject::MUST_READ,
         IOobject::AUTO_WRITE
     ),
     mesh
 );
```

```
//mhd modelling
Info<< "Reading field PotEsolid\n" << endl;
volScalarField PotEsolid
(
    IOobject
    (
        "PotE",
        runTime.timeName(),
        solidMesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    solidMesh
);

Info<< "Reading solid electrical conductivity sigma_w\n" << endl;
volScalarField sigmasolid
(
    IOobject
    (
        "sigma_w",
        runTime.timeName(),
        solidMesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    solidMesh
);
```

readTransportProperties.H addition

```

    //applied magnetic field
    dimensionedVector B0(laminarTransport.lookup("B0"));
```

Implementation - Lorentz force term

In MhdPisoFoam.C initialize Lorentz force

```
Info<< "\nStarting time loop\n" << endl;

// Lorentz force estimate
volVectorField lorentz = sigma * (-fvc::grad(PotE) ^ B0) + sigma * ((U ^ B0) ^ B0);
```

In UEqn.H modify the momentum equation

```
// Solve the momentum equation

fvVectorMatrix UEqn
(
    fvm::ddt(U)
  + fvm::div(phi, U)
  + turbulence->divDevReff()
  - (1.0/rho) * lorentz
);

UEqn.relax();
```

Implementation - Electric potential

In `solvePotE.H` rearrange Eq. 5 $\nabla^2(\sigma\phi) = \sigma\nabla\cdot(\mathbf{u}\times\mathbf{B})$:

```
{

    //calculate cross-product of velocity and magnetic field
    surfaceScalarField psiub = fvc::interpolate(U ^ B0) & mesh.Sf();
    surfaceScalarField sigmasolidF = fvc::interpolate(sigmasolid);

    while (simpleSolid.correctNonOrthogonal())
    {
        coupledFvScalarMatrix PotEqns(2);

        // Add fluid equation
        PotEqns.set
        (
            0,
            new fvScalarMatrix
            (
               fvm::laplacian(sigma, PotE) - sigma * fvc::div(psiub)
            )
        );

        // Add solid equation
        PotEqns.set
        (
            1,
            new fvScalarMatrix
            (
               fvm::laplacian(sigmasolidF, PotEsolid)
            )
        );

        PotEqns.solve();
    }
```

$$\psi = (\mathbf{u}\times\mathbf{B})\cdot\mathbf{S}_f \quad (9)$$

$$\nabla^2(\sigma\phi) - \sigma\nabla\cdot\psi = 0 \quad (10)$$

$$\nabla^2(\sigma_w\phi) = 0 \quad (11)$$

Implementation - current density

Interpolation scheme for current density

```
    //consistent and conservative scheme for current density (Ni, 2007)
    surfaceScalarField jn = -(fvc::snGrad(PotE) * mesh.magSf()) + psiub;

    surfaceVectorField jnv = jn * mesh.Cf();

    volVectorField jfinal = fvc::surfaceIntegrate(jnv) - fvc::surfaceIntegrate(jn) * mesh.C();

    jfinal.correctBoundaryConditions();

    //update lorentz field, maybe here we should just use lorentz.correct();
    lorentz = sigma * (jfinal ^ B0);
}
```

$$j_f = -\nabla_{sn}\phi \cdot S_f + \psi \tag{12}$$

$$J_c = \frac{1}{\Omega_p}\sum_{f=1}^{nf} j_f(\mathbf{r}_f - \mathbf{r}_p)\cdot \mathbf{S}_f \tag{13}$$

$$\texttt{lorentz} = \sigma(\mathbf{J}_c \times \mathbf{B}) \tag{14}$$

Tutorial set-up

Test cases: 2 MHD flows 1 hydrodynamic

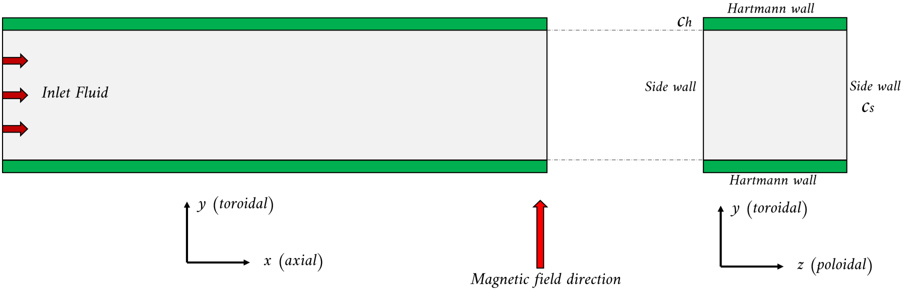


Figure: 2D Sketch

- Characteristic length $L = 0.5$m (Half-length y-direction)
- Axial length $20L$

Parameters and boundary conditions

A "dummy" materials are assumed:

Cases parameters

- $\mathrm{Ha} = 50$
- $\mathrm{Re} = 4$ (laminar)
- $c_s = 0$ (insulated)
- $c_h = 0.1$ or 0.4
- Isothermal (273 K)

Velocity BC

- Inlet = `fixedValue`
- Outlet = `zeroGradient`
- Walls = `noSlip`

Remember Eq.8:

$$c = \frac{\sigma_w}{\sigma} \frac{t}{L}$$

Electric potential BC

- Hartmann-walls = `regionCoupling` (uniform value)
- Side-walls = `zeroGradient`
- Inlet = `zeroGradient`
- Outlet = `zeroGradient`

Mesh grid

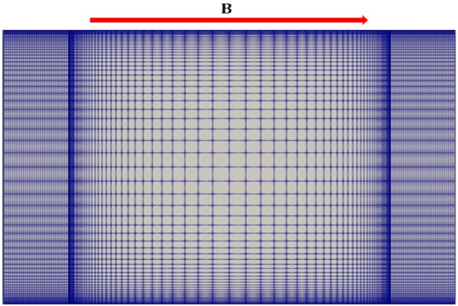


Figure: 2D view (outlet) of the mesh grid

Non-uniform structured mesh:

$$\delta_{\mathrm{Ha}} = \frac{1}{\mathrm{Ha}} \quad (15)$$

$$\delta_{\mathrm{Side}} = \frac{1}{\sqrt{\mathrm{Ha}}} \quad (16)$$

```
hex (0 1 2 3 4 5 6 7) (60 70 70)
simpleGrading (1 ((50 50 36.17701855) (50 50 0.02764))
((50 50 36.17701855)(50 50 0.02764)))
```

Mesh grid

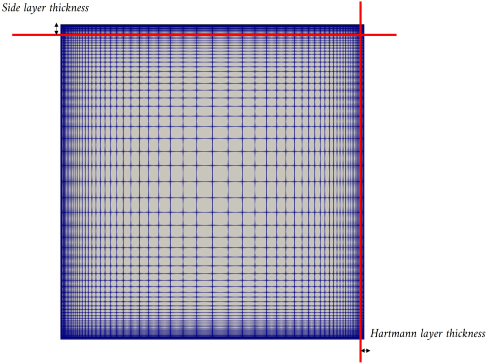


Figure: Layers refinement

- $\approx$ 7 elements δ_{Ha}
- $\approx$ 35 elements δ_{Side}
- Total mesh grid $\approx 5.5 \cdot 10^5$ (!!)

Running the tutorial case

Run the TopoMesh script in the tutorial folder (NOT with foam-extend):

```
Tut="Ha50c01 Ha50c04 Hydro"
for folder in $Tut;
 do
   cd $folder;
   blockMesh -region solid;
   setSet -region solid -batch solid.setSet;
   setsToZones -region solid -noFlipMap;
   blockMesh;
   setSet -batch fluid.setSet;
   setsToZones -noFlipMap;
   cd ..;
done
```

Running the tutorial case

Source foam-extend now and run the ParaRun script in the tutorial folder:

```
Tut="
Ha50c01
Ha50c04
"
decomposePar -case Hydro;
decomposePar -case Hydro -region solid;
mpirun -np 2 MhdPisoFoam -case Hydro -parallel > Hydro/log &
for folder in $Tut;
 do
   decomposePar -case $folder;
   decomposePar -case $folder -region solid;
   mpirun -np 3 MhdPisoFoam -case $folder -parallel > $folder/log &
done
```

Running the tutorial case - simulation control

Simulation convergence must be controlled by the user changing the time-step from the `controlDict`, paying attention that this will not cause a divergent trend.

- Default $\Delta t = 1 \cdot 10^{-3}$
- Might be increased to $\Delta t = 7 \cdot 10^{-3}$

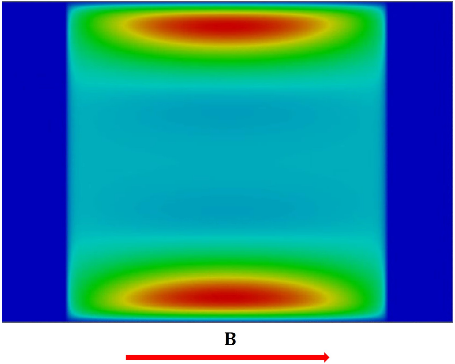
Post-process

Run the `graph` script in the tutorial folder:

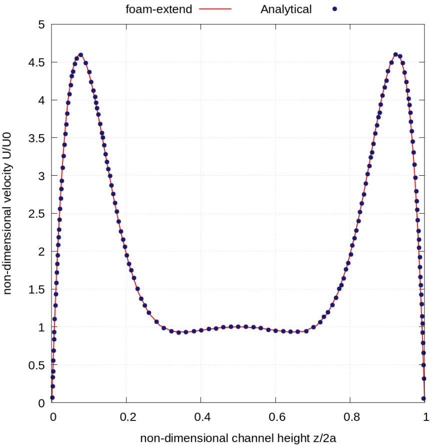
```
cases="Ha50c01 Ha50c04 Hydro"
mkdir Results
for folder in $cases;
 do
    reconstructPar -case $folder;
    reconstructPar -case $folder -region solid;
    latestTime=$(ls $folder -1 | sort -n | tail -n 1);
    sample -case $folder;
    Uref=$(sort -nk 1 $folder/postProcessing/sets/$latestTime/CentralVelocity_U.xy  \
    | head -n 1 | awk '{print $2}');
    GraphName="Velocity Pofiles_$folder.png"
      gnuplot<<Velocity
        set terminal pngcairo font "helvetica,20" size 1000, 1000
        set output 'Results/$GraphName'
        set key center top outside
        set key horizontal
        set xlabel "non-dimensional channel height z/2a"
        set ylabel "non-dimensional velocity U/U0"
        set grid
        plot \
         "$folder/postProcessing/sets/$latestTime/SideLayerProfiles_U.xy" u 1:(\$2/$Uref) w lines
          lw 2 lc rgb "red" title "foam-extend", \
         "SloanSol$folder" u 1:2 w points pt 7 ps 1.5 lc rgb "midnight-blue" title "Analytical"
     Velocity
done
```

Outcomes

(a) Velocity outlet contour

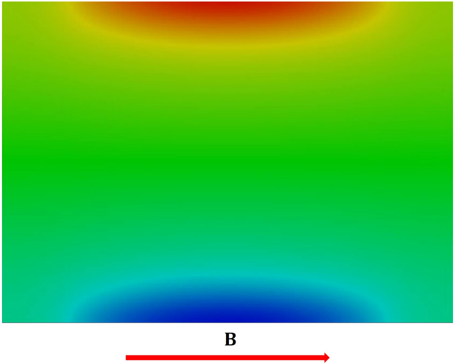


(b) MHD case $\mathrm{Ha} = 50$ and $c = 0.4$

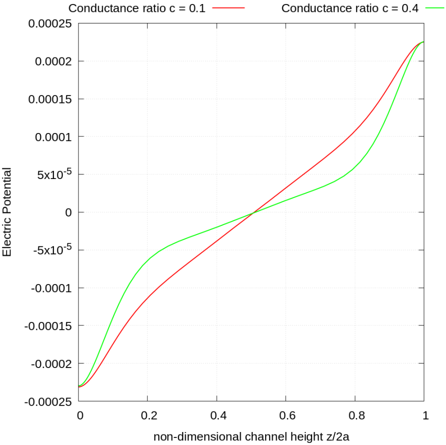


Outcomes

(a) Potential outlet contour



(b) Electric potential between the side walls in Volt.



Follow-up activities

An MHD coupled solver has been developed for foam-extend. The new `MhdPisoFoam` it is able to model the influence of finite electrical conductive walls.

- Simplify the tutorial case through symmetry and periodic patches
- Optimization of the current density conservation scheme (Ni second method)
- Extend the V&V procedure of the solver with experimental results and further analytical solutions
- Far far future: MHD turbulence modelling

Thank you for your attention

Questions?