



university of
 groningen

faculty of science
 and engineering

mathematics and applied
 mathematics

Verification of resistivemhdFoam as an MHD Solver for the 3D Orszag-Tang Vortex and Hartmann Duct Flow

Bachelor's Project Mathematics & Physics

July 2022

Student: F. P. Pietersen

First supervisor: Prof.dr. ir. R. W. C. P. Verstappen

Second assessor: Prof. dr.ir. P. R. Onck

Abstract

Magnetohydrodynamics (MHD) fluid simulations have the potential to tap into a wealth of knowledge on fusion plasma dynamics at a fraction of the cost of experiments. OpenFOAM is a highly-flexible, open-source CFD software that provides the standard MHD solver `mhdFoam`. An adaptation of this solver, called `resistivemhdFoam`, was used by González et al., 2020 on a toroidal domain and further studied by Jacobberger, 2021 on toroidal and cylindrical domains under fusion conditions. This is the first study that aims to investigate the viability of `resistivemhdFoam` as an MHD solver. `mhdFoam` and `resistivemhdFoam` were studied over the 3D Orszag-Tang Vortex (OTV) benchmark with $\nu = \lambda = 0.02$ and Hartmann duct flows with $Ha = 5, 10, 20, 50, 100$. It was found that `resistivemhdFoam` matches `mhdFoam` in OTV simulations excellently. Both were found to agree well with literature. All Hartmann profiles simulated by `mhdFoam` decently matched literature. `resistivemhdFoam` showed poor profiles for $Ha = 5$, but conformed excellently to theory for $Ha \geq 20$ numbers, outperforming `mhdFoam` for $Ha = 20$.

Contents

1	Introduction	4
2	Magnetohydrodynamics	5
2.1	Governing equations	5
2.2	Parameters of interest	5
3	OpenFOAM	6
3.1	MhdFoam	6
3.2	ResistivemhdFoam	7
4	Orszag-Tang Vortex	8
4.1	Theory	8
4.2	Case setup	9
4.3	Results	9
5	Hartmann flow	11
5.1	Theory	11
5.2	Case setup	12
5.3	Meshing	12
5.4	Results	14
6	Torus	16
6.1	Meshing	16
6.2	Dynamic timestepping	18
7	Discussion	18
8	Conclusions	19
9	Outlook	19
10	Acknowledgements	20

1 Introduction

In a time where renewable energy is increasingly more necessary to dampen climate change (Pedersen et al., 2022), many international efforts are made to investigate nuclear fusion technology. As a clean, on-demand, safe renewable energy source, it is one of the main candidates to replace the fossil fuels. Various concepts of fusion reactors have been proposed and built over the decades (Donné, 2019), but creating a net-positive fusion reactor proves to be a complex challenge due to the extreme conditions that need to be maintained (Linke et al., 2019).

Experimental setups are astronomically expensive (Wendelstein cost an approximate \$1billion Feder, 2011) and the lack of fundamental understanding in magnetohydrodynamics – the physics governing plasma dynamics - makes the research programme difficult to advance. Simulations can aid significantly in this process. With a fraction of the cost and the ability to analyse over various geometries and conditions, it may provide vital insights for the development of nuclear fusion.

This work aims to contribute to the development of simulation software by studying a finite volume solver for magnetohydrodynamics in OpenFOAM. OpenFOAM is a free open-source CFD software for various kinds of fluid dynamics (OpenCFD, 2022). It is renowned for its flexibility, allowing the user to modify existing solvers or develop new ones with ease.

One such custom solver is `resistivemhdFoam`, first implemented by González et al., 2020. It is based on the standard `mhdFoam` solver for magnetohydrodynamics (MHD) provided by OpenFOAM. This work aims to validate `resistivemhdFoam` as a viable MHD solver for fusion conditions ($Ha \geq 10^4$ (Bühler et al., 2007)) and compare it to the existing solver, `mhdFoam`. To achieve this, the solvers are applied to two benchmark MHD cases: the Orszag-Tang Vortex and Hartmann Duct flow. Results of both solvers are compared and are further examined in relation to literature.

2 Magnetohydrodynamics

2.1 Governing equations

First described by Alfvén, magnetohydrodynamics governs the motion of a conducting fluid through a magnetic field. As potential differences arise, electrical currents emerge throughout the fluid. These currents simultaneously induce a magnetic field as a result of Faraday’s law. It is this interplay of conductive flow and (induced) magnetic field that defines magnetohydrodynamics. (Roberts, 1967)

The governing equations of magnetohydrodynamics are derived from Maxwell’s equations of electrodynamics and Ohm’s law, combined with the momentum equation for an incompressible fluid. The resulting system assumes an incompressible conducting fluid in laminar flow. (Cappello and Escande, 2000) (OpenCFD, 2022)

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{U} \times \mathbf{B}) - \frac{\nabla \times (\mathbf{j})}{Rm} \quad (2.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{j} \times \mathbf{B} - (\mathbf{v} \cdot \nabla) \mathbf{u} + \nabla^2 (\nu \mathbf{u}) \quad (2.2)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.3)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2.4)$$

Here, \mathbf{B} is the magnetic field, \mathbf{u} is the velocity field, the current density \mathbf{j} is given by $\mathbf{j} = \nabla \times \mathbf{b}$ and Rm is the magnetic Reynolds number, further explained below.

2.2 Parameters of interest

The ordinary Reynolds number, which indicates the degree of turbulence in a fluid, is defined as:

$$Re = \frac{\mathcal{U} \mathcal{L}}{\nu} \stackrel{(*)}{=} \frac{1}{\nu}$$

Where ν is the kinematic viscosity. This parameter is significant for any CFD simulation, since turbulence has a strong influence on computational and dynamical properties. The OpenFOAM packages used in this study assume laminar flow, so Reynolds numbers in the considered cases must be low enough. In this study, all cases considered have $Re \leq 50$, well below the critical Reynolds number of most geometries, including the ones considered in this study. Motivated by the Reynolds number, magnetohydrodynamics is also characterized by the magnetic Reynolds number. This number is a measure for the magnitude of the induced field relative to the inducing field (Roberts, 1967). It is defined as follows:

$$Rm \equiv \sigma \mu \mathcal{U} \mathcal{L} \stackrel{(*)}{=} \sigma = \frac{1}{\mu \lambda} \stackrel{(*)}{=} \frac{1}{\lambda}$$

Where σ is the electrical conductivity, λ is the magnetic diffusivity, and μ is the magnetic permeability. \mathcal{U} and \mathcal{L} are typical velocity and length scales, respectively. In this study we assume $\mu = 1$ for a plasma, and in the cases studied, we have $\mathcal{U} \cdot \mathcal{L} = 1$, resulting in (*).

Note that the Reynolds number and magnetic Reynolds number are equal to the inverses of ν and λ , respectively.

The energy dissipation per unit mass $\epsilon = \nu \langle \boldsymbol{\omega}^2 \rangle + \lambda \langle \mathbf{j}^2 \rangle$, is the sum of viscous dissipation $\nu \langle \boldsymbol{\omega}^2 \rangle$ and Ohmic dissipation $\lambda \langle \mathbf{j}^2 \rangle$ (Jadhav and Chandy, 2021). Where $\langle \dots \rangle$ is the volumetric average and $\mathbf{j} = \nabla \times \mathbf{b}$. It may be noted that in the limit $\nu \rightarrow 0$ and $\lambda \rightarrow 0$ there is no energy dissipation. This motivates the time evolution of magnetic, kinetic and total energy (their sum) as parameters of interest.

$$E_k = \frac{\langle \mathbf{u}^2 \rangle}{2}, \quad E_m = \frac{\langle \mathbf{B}^2 \rangle}{2}, \quad E_t = E_k + E_m \quad (2.5)$$

The ratio of Lorentz force to viscous force is captured by the Hartmann number Ha , which is defined as:

$$Ha = \mathcal{B}\mathcal{L} \sqrt{\frac{\sigma}{\rho\nu}} \stackrel{(*)}{=} \mathcal{B}\mathcal{L} \frac{1}{\sqrt{\nu\lambda}}$$

With $\mathcal{B}, \mathcal{L}, \sigma, \nu$ and λ defined as above.

3 OpenFOAM

OpenFOAM is free open source CFD software, with a large worldwide user base throughout various fields of science and engineering (OpenCFD, 2022), (Jasak, 2009). The software has the combination of flexibility, computational efficiency and user-friendliness due to its toolkit approach. This allows users to implement custom solvers with relative ease. This study uses the standard OpenFOAM solver `mhdFoam`, and a custom user-made solver `resistivemhdFoam`, which is highly similar to `mhdFoam`. This custom solver was developed by González et al., 2020 and has been reproduced and further studied in this work.

3.1 MhdFoam

The `mhdFoam` solver is a laminar incompressible MHD solver, based on the momentum equation for incompressible fluids and the coupled Maxwell-Navier stokes equations. First the solver sets up the linear algebra for the U equation. The solver then predicts U with the so called 'momentum predictor', $-\nabla p$ at line 12.

The solver then enters a PISO loop (Issa, 1986), that solves the momentum equation "UEqn" up to the desired tolerance specified in `sytem/fvSolution`. In this study, this is always $1e-5$ for all fields - U, B, B_t and B_{fixed} - except for the p field, which has a tolerance of $1e-6$. For further details of the PISO algorithm, the reader is referred to the online documentation provided by OpenFOAM (OpenCFD, 2022).

The B field is computed by a similar method. The code immediately enters the "B-PISO loop" (PISO loop for the magnetic field), loads in the algebra and iteratively solves for the equations (2.1)-(2.4) using the Helmholtz-Hodge decomposition (Maria Denaro, 2003).

```

1   fvVectorMatrix UEqn
2   (
3       fvm::ddt(U)
4       + fvm::div(phi, U)
5       - fvc::div(phiB, 2.0*DBU*B)
6       - fvm::laplacian(nu, U)
7       + fvc::grad(DBU*magSqr(B))
8   );
9
10  if (piso.momentumPredictor())
11  {
12      solve(UEqn == -fvc::grad(p));
13  }

```

Figure 1: Momentum equation for incompressible fluid in mhdFoam with momentum predictor

```

1   while (bviso.correct())
2       fvVectorMatrix BEqn
3       (
4           fvm::ddt(B)
5           + fvm::div(phi, B)
6           - fvc::div(phiB, U)
7           - fvm::laplacian(DB, B)
8       );
9
10      BEqn.solve();
11      ...

```

Figure 2: B-PISO loop in mhdFoam

3.2 ResistivemhdFoam

In order to implement a permanent inducing field, `mhdFoam` is modified and renamed `resistivemhdFoam`. It is the result of replacing the \mathbf{B} field with $\mathbf{B}_{\text{fixed}} + \mathbf{B}_t$ in figure 2. Here, $\mathbf{B}_{\text{fixed}}$ is a permanent magnetic field and \mathbf{B}_t is a time perturbation of the magnetic field. To obtain the `resistivemhdFoam` solver, the linear algebra of the B-PISO loop in the `mhdFoam` solver is modified as in figure 3.

By replacing $\mathbf{B} \rightarrow \mathbf{B}_{\text{fixed}} + \mathbf{B}_t$, all spatial terms are doubled by linearity of the divergence and Laplacian operator. Only the `fvm::ddt` term is not doubled since $\frac{d}{dt} \mathbf{B}_{\text{fixed}} = 0$, which hard codes the time-independence of $\mathbf{B}_{\text{fixed}}$ into the solver. Note that the \mathbf{B}_t terms are discretized into the B-PISO loop, and are solved implicitly (`fvm::`). The $\mathbf{B}_{\text{fixed}}$ terms are calculated explicitly using current values (`fvc::`), because $\mathbf{B}_{\text{fixed}}$ does not depend on time and so does not need to be solved for in the B-PISO loop.

As \mathbf{B}_t is now the field that is solved for in the B-PISO loop, the rest of the B-PISO code

```

1 while (bviso.correct())
2 {
3     fvVectorMatrix BtEqn
4     (
5         fvm::ddt(Bt)
6         + fvc::div(phi, Bfixed) // for Bfixed field use current value (fvc)
7         + fvm::div(phi, Bt)
8         - fvc::div(phiBt, U)
9         - fvc::div(phiBfixed, U)
10        - fvc::laplacian(DB, Bfixed)
11        - fvm::laplacian(DB, Bt)
12    );
13    ...

```

Figure 3: Modified B-PISO loop in resistivemhdFoam

needs a slight adjustment: `mhdFoam` reduces the divergence of \mathbf{B} by solving for the potential field \mathbf{pB} using the Helmholtz-Hodge decomposition (further information can be found in (Maria Denaro, 2003)). The B-PISO loop in `resistivemhdFoam` needs to solve iteratively for \mathbf{Bt} , which is implemented by replacing "`phiB = fvc::flux(B)`" \rightarrow "`phiBt = fvc::flux(Bt)`" and replacing `phiB` accordingly.

While it has been suggested by (Jacobberger, 2021) that `resistivemhdFoam` may lead to a \mathbf{B} field with $\nabla B \neq 0$ (forbidden by 2.3), this is actually not the case as long as $B_{fixed} = 0$, since this implies $\nabla B = \nabla B_t + \nabla B_{fixed} = 0$. This should therefore not form a problem in applications, as $\nabla B_{fixed} = 0$ for any physical magnetic field.

4 Orszag-Tang Vortex

4.1 Theory

This vortex was first studied by Orszag and Tang in 1979 (Orszag and Tang, 1979). Since then, it has been extensively studied and by now it is a well-understood test case. The original Orszag-Tang vortex was specified in two dimensions, but a three dimensional extension of this test has been studied by numerous papers over the decades (Mininni et al., 2006), (Mena, 2013), (Jadhav and Chandy, 2021).

The three dimensional test consists of a periodic cube with sides of length 2π , with the initial profiles for the velocity and magnetic field as:

$$v_0 = (-2 \sin y, 2 \sin x, 0) \tag{4.1}$$

$$B_0 = \beta(-2 \sin y + \sin z, 2 \sin x + \sin z, \sin x + \sin y) \tag{4.2}$$

Where $\beta = \frac{4}{5}$ is chosen such that $E_k = E_m = 2$.

4.2 Case setup

In this study, the 3D Orszag-Tang vortex was examined using timestep $\Delta t = 0.001$ and grids of $N = 40^3, 80^3, 160^3, 200^3$ homogeneously distributed cells. The mesh was set up using OpenFOAM's integrated `blockMesh` utility, with `simpleGrading (1 1 1)`, i.e. with an expansion ratio of 1 in x, y and z direction. The initial profiles for velocity u (4.1) and magnetic field (4.2) are set using the `setExprFields` utility. Transport variables are set equal to $\mu = \rho = 1$, while ν and λ are set equal to 0.02, implying $Re = Rm = 50$.

With the initial conditions and mesh as above, both `mhdFoam` and `resistivemhdFoam` were run. In order to run `mhdFoam`, the B field is initialized as in equation (4.2), and the U field as equation (4.1). For running `resistivemhdFoam`, the `Bfixed` field was set to 0, and the `Bt` field was set to the initial magnetic field profile (4.2).

4.3 Results

Maximal current density is plotted for various mesh sizes and compared to previous work in figure 4. The data from previous work was retrieved by a visual data extractor.

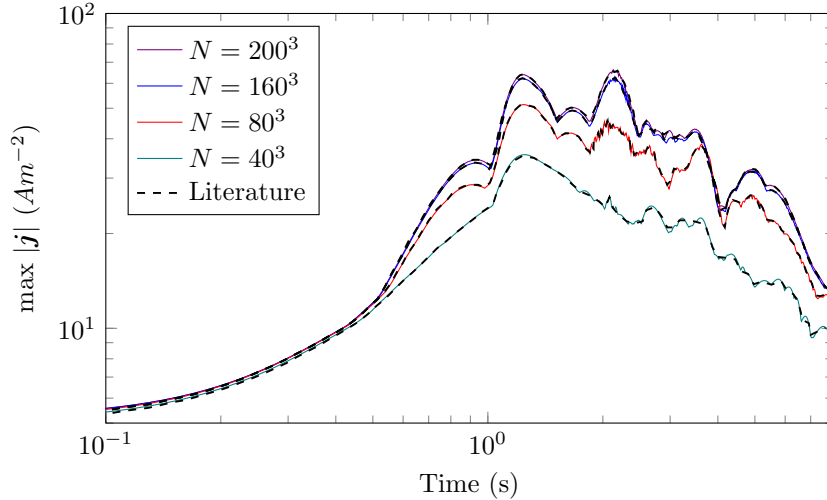


Figure 4: Maximal current density over time for various mesh sizes. The solver `mhdFoam` is used with $\nu = \lambda = 0.02$ and compared to the findings of González et al., 2020 (dashed lines)

Due to the high degree of consistency in both `mhdFoam` and `resistivemhdFoam` - as can be seen in figure 5 - data was compared more quantitatively by extracting the maximal relative deviation of `resistivemhdFoam` relative to `mhdFoam`. Relative deviation was bounded by 0.61%, with the average deviation being just 0.23%. This was also done for the $N = 160^3$ and $N = 200^3$ maximal current density plot. The $N = 160^3$ showed a maximal deviation of 10.7%, with an average of 1.7%. The $N = 200^3$ showed a maximal deviation of 5.5% and an average of 1.7%. The maximal errors arise around $t = 2.9$ and $t = 3.05$, respectively. These differences appear small in the figures 5a and 5b due to the choice of logarithmic axes.

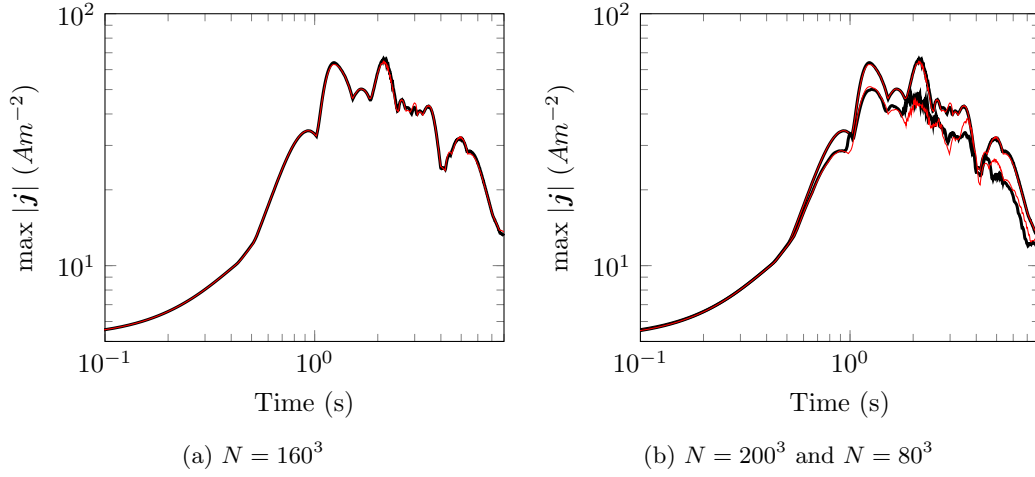


Figure 5: Maximal current density of `mhdFoam` (black) and `resistivemhdFoam` (red) over grids of $N = 160^3$ (left), $N = 80^3$ (right, bottom) and $N = 200^3$ (right, top)

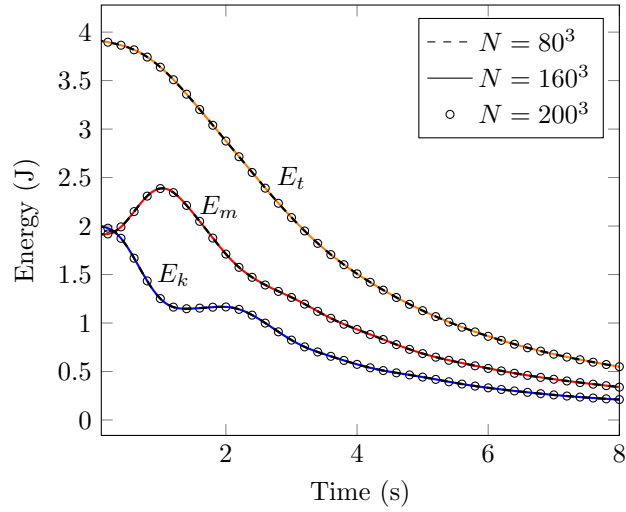


Figure 6: Total energy (yellow), magnetic energy (red) and kinetic energy (blue) over time as computed by `mhdFoam` on a mesh of $N = 160^3$, compared to meshes of $N = 80^3$ (dashed black line) and $N = 200^3$ (black circles). Energy data of `resistivemhdFoam` at $N = 200^3$ had a maximal relative deviation of 0.61%, with an average of 0.23%.

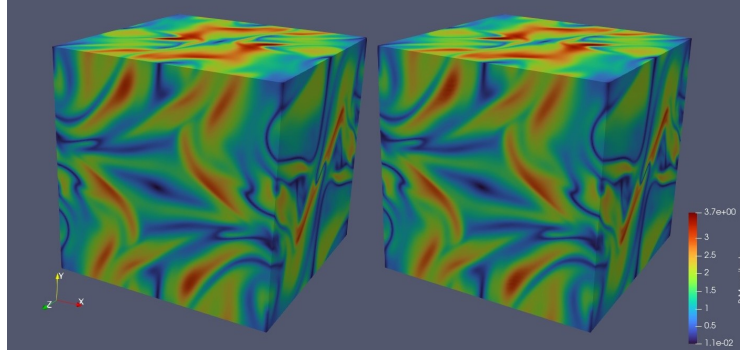


Figure 7: B magnitude plots of mhdFoam (left) and resistivemhdFoam (right) at $t = 2$ for a grid of $N = 200^3$.

5 Hartmann flow

5.1 Theory

Magnetohydrodynamics in a duct was first studied experimentally and analytically by Hartmann in 1937 (Hartmann and Lazarus, 1937). This early test case considers a duct with an opening of size $2a$ and length L , oriented along the y and x direction, respectively. A magnetic field B is applied in the z -direction. This setup, contrary to the Orszag-Tang vortex, has an analytic solution. The equilibrium velocity profile can be derived as (Davidson, 2002):

$$u(y) = u_0 \left(1 - \frac{\cosh[(Ha)y/a]}{\cosh(Ha)} \right) \quad (5.1)$$

$$\delta = \left(\frac{\rho\nu}{\sigma B^2} \right)^{1/2} \quad (5.2)$$

Where δ is the thickness of the so-called 'Hartmann layer', and $u_0 = u(y = 0)$ is the equilibrium velocity in the centre of the duct. The Hartmann number in this setting is equal to $Ha = \frac{a}{\delta}$.

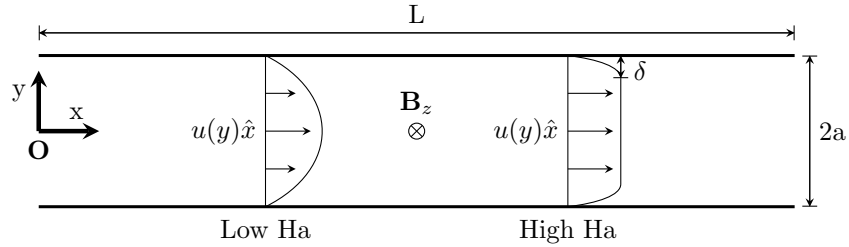


Figure 8: Schematic of the Hartmann duct flow for low and high Hartmann number

As the forced flow at the inlet in this study is $u|_{x=0}(y) = 1$, this means $\int_{-1}^1 u(y) \cdot y \, dy = 2$,

and u_0 may be computed as

$$\int_{-1}^1 u_0 \left(1 - \frac{\cosh(Ha \cdot y/a)}{\cosh(Ha)} \right) dy = 2 \Rightarrow u_0 = \frac{2}{\int \dots dy}$$

5.2 Case setup

In this study, Hartmann flow is considered in a 2D duct with $a = 1$, $L = 20$ and $Re = Rm = 1$. The flow is forced by $U(y) = 1 \cdot \hat{x}$ at $x = 0$. Note the useful identities $Ha = B = \frac{1}{\delta}$ for this case. Hartmann flows of Hartmann number 5, 10, 20, 50 and 100 are studied by varying the B field.

OpenFOAM has already implemented the Hartmann case with `mhdFoam` as a tutorial case. The permanent magnetic field is introduced as a Dirichlet boundary condition of $B(y = \pm 1) = 20\hat{z}$, and the internal field is initialized as $B_{t=0}(z) = 20\hat{z}$.

Running the Hartmann case with `resistivemhdFoam` is done in two ways. 1) The `Bfixed` field is set to $20\hat{z}$, and `Bt` = 0. This method resembled physical intuition most closely, as it represents the permanent field by `Bfixed` while an additional induced field can arise in `Bt`. This method is called `resistivemhdFoam` from now on. 2) The `Bt` field is set to $20\hat{z}$ and `Bfixed` = 0. This is similar to how `mhdFoam` is run, and serves as a basic comparison between the two solvers. This method is called `BtFoam` from now on. Note that `resistivemhdFoam` fixes `Bfixed` and solves the dynamics as perturbations in `Bt`, while `BtFoam` eliminates `Bfixed` and sets Dirichlet conditions on `Bt`, bearing strong resemblance to `mhdFoam`.

Preliminary 'slice' scans in Paraview are done to verify that the flow profile reaches an equilibrium at $x = 15$ and $t = 2$. Then, `mhdFoam` and `resistivemhdFoam` y-velocity profiles are taken at $x = 15$ and $t = 2$, and plotted against the analytical equilibrium profile. Note that only the profiles are depicted starting from the centre of the duct at $y = 0$ to the upper wall at $y = 1$. The δ values are also extracted by taking the y-value of the mesh point \vec{p} with

$$U(\vec{p}) \approx u(1 - \delta) = \left(1 - \frac{\cosh(B - 1)}{\cosh B} \right) u_0$$

closest to the analytical value. Here, U indicates the velocity field computed by OpenFOAM, and $u(1 - \delta)$ is the analytical velocity value at the edge of the Hartmann boundary layer.

5.3 Meshing

The mesh for the Hartmann flow is set up using `blockMesh` with `simpleGrading` options, set up as follows:

More cells are assigned towards the upper and lower walls of the duct mesh, to allow a more precise determination of δ . For higher Hartmann number a finer mesh is required to resolve the value of δ , due to the $\delta = \frac{1}{Ha}$ relationship. As such, the number of cells in the y-direction is increased for higher Hartmann numbers, and the timestep is decreased accordingly. The precise mesh configurations per simulation, and the resolution can be found

```

1 blocks
2 (
3   hex (0 1 2 3 4 5 6 7) (100 80 1)
4   simpleGrading
5     (
6       1           // x-direction expansion ratio
7       (
8         (0.2 0.3 4) // 20\% y-dir, 30\% cells, expansion = 4
9         (0.6 0.4 1) // 60\% y-dir, 40\% cells, expansion = 1
10        (0.2 0.3 0.25) // 20\% y-dir, 30\% cells, expansion = 0.25 (1/4)
11       )
12     ) 3           // z-direction expansion ratio
13   )
14 );

```

Figure 9: The used simpleGrading options in blockMeshDict

in table 1. The resolution is the y-dimension of the cells at $y = 1 - \delta$.

Ha	#y-cells	resolution	Δt
5	80*	2.5e-2	0.001
10	80	1.3e-2	0.001
20	80	1.0e-2	0.001
50	80	8.6e-3	0.001
100	120	5.1e-3	0.0005

Table 1: Parameters for the various cases of Hartmann flow. *The mesh was setup with simpleGrading (1 1 1) for $Ha = 5$

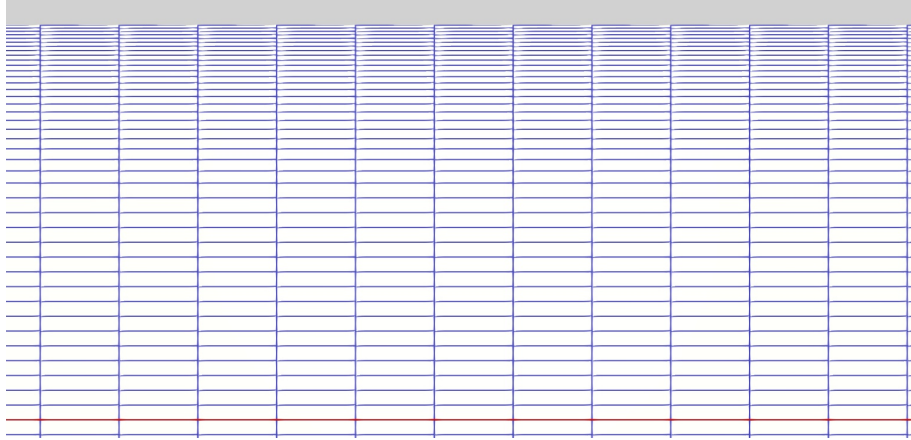


Figure 10: Part of the rectangular mesh with 80 y-cells, used for the runs with Hartmann number 10,20 and 50 (see figure 1). The red line indicates the centre of the duct. The settings in blockMeshDict are as described in figure 9

5.4 Results

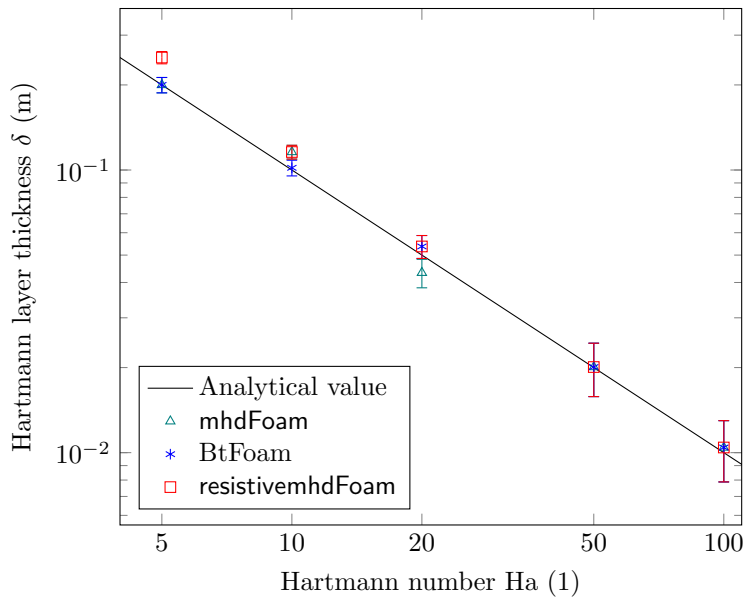


Figure 11: Log-log plot of Hartmann layer thickness δ over Hartmann number Ha . Error bars are taken as resolution/2

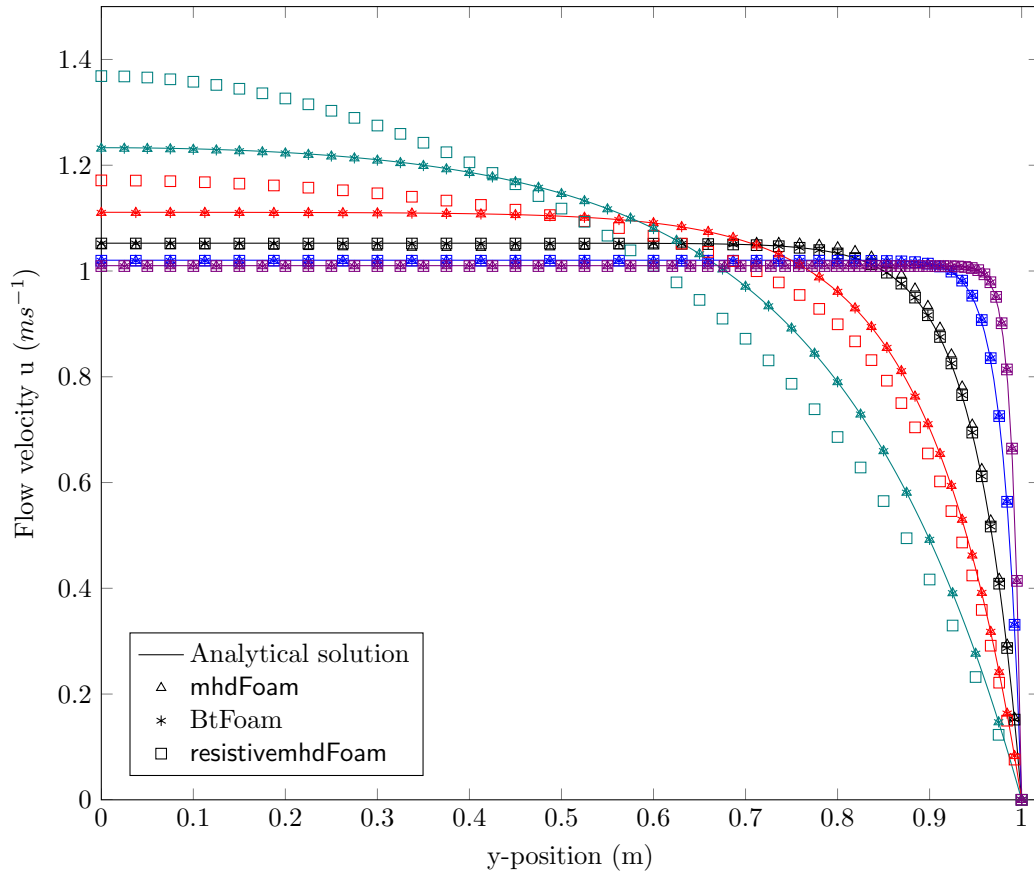


Figure 12: Analytical vs. simulated Hartmann flow profiles for Hartmann numbers 5 (green), 10 (red), 20 (black), 50 (blue) and 100 (purple).

6 Torus

This paper set out to study MHD inside a toroidal configuration with $R_0 = 0.55\pi$ and $r_0 = 0.3\pi$ using `resistivemhdFoam`, and build on the work of González et al., 2020. Due to time constraints and the relative complexity of the geometry, results could not be obtained for discussion in the current study. In this section, important hurdles and complications are laid out in order to expedite future work.

6.1 Meshing

Having a good mesh set up is an important part of any CFD simulation. It not only reduces the computational effort, but also improves accuracy. On top of this, poor meshing may lead to unphysical (as seen in the Orszag-Tang case for low number of cells) or even diverging results. An example of diverging results on a torus is demonstrated in González et al., 2020.

As a starting point, parameters of the discussed Hartmann and Orszag-Tang test cases were applied to the toroidal confinement. The mesh was set up using OpenFOAM’s `snappyHexMesh` utility. This utility takes an STL file and a background mesh (usually generated with `blockMesh`) and sets up a mesh in three steps. A variety of `snappyHexMesh` settings was explored, but none yielded a non-diverging mesh. In what follows, the `snappyHexMesh` settings as in figure 13 and their output will be discussed. The `addLayersControls` dictionary is excluded since this mesh operation is not deemed the source of problems.

```
1  castellatedMeshControls
2  {
3      refinementSurfaces
4      {
5          torus
6          {
7              level (2 3);
8          }
9      }
10     locationInMesh (-0.01 1 0.01); // Offset from (0 0 0) to avoid
11     nCellsBetweenLevels 3;
12 }
13
14 snapControls
15 {
16     nSolveIter          30;
17     nFeatureSnapIter    10;
18 }
19 ...
```

Figure 13: Settings for `snappyHexMesh` that resulted in poor mesh quality.

In the first step, `snappyHexMesh` identifies all cubes of the background mesh that intersect

with surfaces in the STL geometry. It removes all cells that are not in the region of interest (specified by the `locationInMesh` keyword), and refines the mesh close to the boundary by halving the cells a number of times, specified by `levels`.

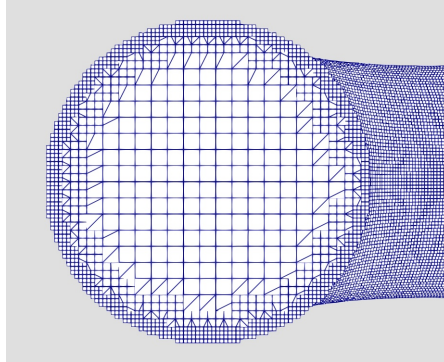


Figure 14: The toroidal mesh after cell removal and refinement. Note the castellated boundary.

In the second step, `snap`, the vertex points of the outer cells are moved onto the STL surface in order to smoothen the boundary. The resulting mesh is shown in figure 15.

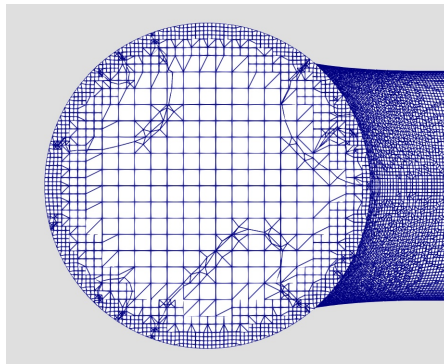


Figure 15: The toroidal mesh after smoothening by `snap`. Note the diagonal cuts through the inner mesh, as well as the tiny cells in the outer regions.

The `snap` feature results in a well-smoothened boundary. As a downside, it has created some very small cells in the outer layer of the mesh, as well as a diagonal cut through the inner regions of the mesh. The reader may also consider using OpenFOAM's `refineMesh` utility instead (Gisen, 2014).

For completeness, the final mesh after the third stage of meshing, `addLayers`, is also added in figure 16. It is not further elaborated on, since the poor meshing of `snap` makes it hard to assess the meshing quality of `addLayers`.

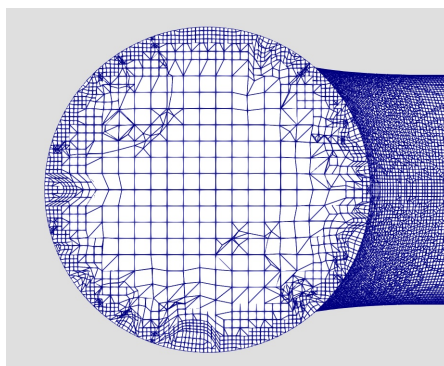


Figure 16: The toroidal mesh after the final stage, `addLayers`. The mesh has only gotten worse

6.2 Dynamic timestepping

An important parameter for implicit solvers is the Courant number $C = u \frac{\Delta t}{\Delta x}$, where Δt is the time step and Δx is the cell size. If $C > 1$, information in one cell can flow past the neighbouring cell in only one timestep. This causes problems for implicit solution schemes (which solve the equations locally), of which several are used in both `resistivemhdFoam` and `mhdFoam`. After running `resistivemhdFoam` and `mhdFoam` on meshes the same as, or similar to, the mesh described above, simulations tended to terminate after approximately 20-120 iterations. As may be expected, termination was strongly correlated to Courant number of value greater than 1. Though this may likely be attributed to poor meshing, future work may benefit from the dynamic timestepping functionalities in OpenFOAM. These enable the user to set a maximal Courant number, after which OpenFOAM will adjust the timestep when needed.

7 Discussion

To verify that the Orszag-Tang vortex results obtained using `mhdFoam` are valid, maximal current density data was compared to the findings of González et al., 2020 in figure 4. The results of this study agree almost perfectly with González et al., 2020 over all four meshes. Upon close inspection, deviations can be seen for $N = 40^3$, but these are only the result of under sampling when extracting the data. Meshes of size $N \geq 160^3$ seem to have converged to physically accurate results. Comparing the maximal current density plots of `mhdFoam` to `resistivemhdFoam` in figure 5, it is clear that `resistivemhdFoam` shows highly similar behaviour. For the rough mesh of $N = 80^3$, there is some discrepancy between the two, but the `resistivemhdFoam` results quickly converge as the mesh is refined to $N \geq 160^3$ (see figure 5a and figure 5b top). This is confirmed by looking at the 3D plots of current density in figure 7.

Energy plots agree excellently with the findings of Orszag and Tang, 1979 (figure 5), Jadhav and Chandy, 2021 (figure 2) and Politano et al., 1995 (figure 1, run A). Figure ?? shows excellent agreement of energy plots between runs of `mhdFoam`. A data comparison for `mhdFoam` and `resistivemhdFoam` show a high degree of precision, with relative deviations

of at most 0.61%.

For Hartmann duct flow at low Hartmann numbers ($Ha = 5, 10$), `mhdFoam` and `BtFoam` significantly outperform `resistivemhdFoam`, as seen in figure 12. This can also be seen in the determination of δ in figure 11. Note, however, that the delta measurement in essence is the extraction of the duct position at the value of $u(1 - \delta)$ from figure 12. As such, consideration of figure 12 should be given more weight.

For $Ha = 20$, velocity profile results of both applications of `resistivemhdFoam` agree excellently with theory and outperform the profiles of the conventional `mhdFoam` solver. Results of `mhdFoam`, show a slight deviations for $y \geq 0.7$. This is also seen in the determination of the Hartmann layer thickness δ (figure 11), where both `BtFoam` and `resistivemhdFoam` choose a cell containing the analytic value, while `mhdFoam` is significantly off.

For the highest Hartmann numbers studied, $Ha = 50, 100$, all three solving methods yield flow profiles that agree excellently with theory. This is confirmed in the excellent agreement of the determined values of the Hartmann layer thickness.

8 Conclusions

This work shows the viability of `resistivemhdFoam` as an MHD solver for the Orszag-Tang and Hartmann duct flow conditions.

Overall, `BtFoam` (`resistivemhdFoam` with `Bfixed=0`) outperforms `mhdFoam` for Hartmann duct flow, and shows excellent agreement with `mhdFoam` and previously obtained results by Orszag and Tang, 1979, Jadhav and Chandy, 2021, Politano et al., 1995 on the Orszag-Tang Vortex. This support the validity of `resistivemhdFoam` as an MHD solver. On top of this, `BtFoam` matches or outperforms `mhdFoam` in all velocity profiles for Hartmann duct flow.

Additionally, `resistivemhdFoam` deviates from theory in Hartmann velocity flow profiles with low Hartmann number ($Ha = 5, 10$), but the results converge quickly as Hartmann number increases. As Hartmann number This makes `resistivemhdFoam` a viable option simulations of fusion conditions.

9 Outlook

Further research may investigate the validation of `resistivemhdFoam` at higher Hartmann numbers. Due to time constraints, Hartmann numbers relevant to fusion conditions ($Ha \approx 10^4$ (Bühler et al., 2007)) were not investigated. As Hartmann number increases, carefully constructed meshes and timestep are necessary to resolve $\delta = 1/(Ha)^{1/2}$ and the flow profile details, which are at the same order of magnitude.

¹For paramaters as in this study.

10 Acknowledgements

My thanks go out to my supervisors professor Verstappen and professor Onck. Without them this thesis would not have been possible. I want to thank Julian Koellermeier for his constructive feedback and support. Thanks to Jing Sun for always being ready to help. My thanks go out to professor Futatani, for his detailed and swift responses to my questions. Finally, I want to thank the team of the Peregrine High-Performance Computing cluster who facilitated this work, especially Fokke Dijkstra, for his useful comments and tips on using the Peregrine cluster.

References

- Bühler, L., Horanyi, S., & Arbogast, E. (2007). Experimental investigation of liquid-metal flows through a sudden expansion at fusion-relevant hartmann numbers. *Fusion engineering and Design*, 82(15-24), 2239–2245.
- Cappello, S., & Escande, D. (2000). Bifurcation in viscoresistive mhd: The hartmann number and the reversed field pinch. *Physical review letters*, 85(18), 3838.
- Davidson, P. A. (2002). *An introduction to magnetohydrodynamics*. American Association of Physics Teachers.
- Donné, A. (2019). The european roadmap towards fusion electricity. *Philosophical Transactions of the Royal Society A*, 377(2141), 20170432.
- Feder, T. (2011). Us narrows fusion research focus, joins german stellarator. *Physics Today*, 64(9), 30.
- Gisen, D. (2014). Generation of a 3d mesh using snappyhexmesh featuring anisotropic refinement and near-wall layers. *ICHE 2014. Proceedings of the 11th International Conference on Hydrosience & Engineering*, 983–990.
- González, D., Suárez Cambra, D., & Futatani, S. (2020). Non-linear mhd simulations of magnetically confined plasma using openfoam.
- Hartmann, J., & Lazarus, F. (1937). *Hg-dynamics*. Levin & Munksgaard Copenhagen.
- Issa, R. I. (1986). Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of computational physics*, 62(1), 40–65.
- Jacoberger, V. (2021). *Numerical simulations of passive scalar advection in mhd turbulence using openfoam* (B.S. thesis). Universitat Politècnica de Catalunya.
- Jadhav, K., & Chandy, A. J. (2021). Analysis of energy transfer through direct numerical simulations of magnetohydrodynamic orszag–tang vortex. *Physics of Fluids*, 33(6), 065112.
- Jasak, H. (2009). Openfoam: Open source cfd in research and industry. *International Journal of Naval Architecture and Ocean Engineering*, 1(2), 89–94.
- Linke, J., Du, J., Loewenhoff, T., Pintsuk, G., Spilker, B., Steudel, I., & Wirtz, M. (2019). Challenges for plasma-facing components in nuclear fusion. *Matter and Radiation at Extremes*, 4(5), 056201.
- Maria Denaro, F. (2003). On the application of the helmholtz–hodge decomposition in projection methods for incompressible flows with general boundary conditions. *International Journal for Numerical Methods in Fluids*, 43(1), 43–69.
- Mena, J. M. (2013). *Magnétohydrodynamique confinée appliquée aux plasmas de fusion magnétiques* (Doctoral dissertation). Ecole Centrale de Lyon.

- Mininni, P., Pouquet, A., & Montgomery, D. (2006). Small-scale structures in three-dimensional magnetohydrodynamic turbulence. *Physical review letters*, *97*(24), 244503.
- OpenCFD. (2022). *Openfoam documentation*. <https://www.openfoam.com/documentation/overview>
- Orszag, S. A., & Tang, C.-M. (1979). Small-scale structure of two-dimensional magnetohydrodynamic turbulence. *Journal of Fluid Mechanics*, *90*(1), 129–143.
- Pedersen, J. T. S., van Vuuren, D., Gupta, J., Santos, F. D., Edmonds, J., & Swart, R. (2022). Iccc emission scenarios: How did critiques affect their quality and relevance 1990–2022? *Global Environmental Change*, *75*, 102538.
- Politano, H., Pouquet, A., & Sulem, P. (1995). Current and vorticity dynamics in three-dimensional magnetohydrodynamic turbulence. *Physics of Plasmas*, *2*(8), 2931–2939.
- Roberts, P. H. (1967). *An introduction to magnetohydrodynamics* (Vol. 6). Longmans London.