PMSM 2D Model using FEniCSx

Abhinav T K

Dec 2024

1 Induction Motor

Reference: Github Wells-Group/TEAM30

I began my literature review by studying the implementation of the TEAM 30 - Induction Motor Analysis problem which simulates an induction motor in 2D and calculates the magnetic flux density (**B**) using the A-V formulation.

1.1 Electromagnetic Field Quantities

Table 1: List of notations

Symbol	Quantity	Unit
$\mathbf{B}$	magnetic flux density	T or Wb/m^2
$\mathbf{H}$	magnetic field intensity	A/m
$\mathbf{E}$	electric field intensity	V/m
$\mathbf{D}$	electric displacement field	C/m^2
$\mathbf{J}$	eddy current density	A/m^2
$\mathbf{J_0}$	source current density	A/m^2
μ	magnetic permeability	N/A^2
μ_0	permeability of free space ($4\pi \times 10^{-7}$)	N/A^2
μ_r	relative permeability (material-specific) ($\mu = \mu_0\mu_r$)	dimensionless
ν	reluctivity ($\nu = 1/\mu$)	A^2/N
ϵ	electric permittivity	F/m
σ	electrical conductivity	S/m
ρ	resistivity ($\rho = 1/\sigma$)	$\Omega \cdot m$
$\mathbf{A}$	magnetic vector potential	Wb/m
V	electric scalar potential	V
$\mathbf{n}$	unit normal vector	
$\mathbf{\Gamma_B}$	external boundary	
$\mathbf{\Gamma_{nc}}$	interface b/w conducting & non-conducting region	
$\mathbf{n}$	unit normal vector	
∇	del operator	
∇V	gradient of V	
$\nabla \cdot \mathbf{A}$	divergence of A	
$\nabla \times \mathbf{A}$	curl of A	
$\omega_\mathbf{r}$	angular velocity of the rotor	rad/s
$\mathbf{u}$	velocity of the rotor	m/s

1.2 Governing equations

1.2.1 Maxwell's equations

$$\left.\begin{array}{c}\nabla \times \mathbf{H} = \mathbf{J_0} \\ \nabla \cdot \mathbf{B} = 0\end{array}\right\} \quad \text{in } \Omega_n \tag{1a}$$

$$\left.\begin{array}{c}\nabla \times \mathbf{H} = \mathbf{J} \\ \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \mathbf{B} = 0 \\ \nabla \cdot \mathbf{J} = 0\end{array}\right\} \quad \text{in } \Omega_c \tag{1b}$$

Ω_c is the conducting region and Ω_n is the non-conducting region.

1.2.2 Constitutive relations

$$\mathbf{B} = \mu_0 \mu_r \mathbf{H} \tag{2a}$$

$$\mathbf{H} = \nu_0 \nu_r \mathbf{B} \tag{2b}$$

$$\mathbf{J} = \sigma \mathbf{E} \tag{2c}$$

$$\mathbf{E} = \rho \mathbf{J} \tag{2d}$$

$$\nu = \frac{1}{\mu} \tag{2e}$$

$$\rho = \frac{1}{\sigma} \tag{2f}$$

1.2.3 Boundary conditions

$$\mathbf{B} \cdot \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_B \tag{3a}$$

$$\mathbf{H_c} \times \mathbf{n_c} + \mathbf{H_n} \times \mathbf{n_n} = \mathbf{0} \quad \text{on } \Gamma_{nc} \tag{3b}$$

$$\mathbf{B_c} \cdot \mathbf{n_c} + \mathbf{B_n} \cdot \mathbf{n_n} = \mathbf{0} \quad \text{on } \Gamma_{nc} \tag{3c}$$

$$\mathbf{J} \cdot \mathbf{n_c} = \mathbf{0} \quad \text{on } \Gamma_{nc} \tag{3d}$$

1.3 A-V formulation

$$\mathbf{B} = \nabla \times \mathbf{A}. \tag{4}$$

$$\mathbf{E} = -\nabla V - \frac{\partial \mathbf{A}}{\partial t}. \tag{5}$$

Final set of equations inserting the identity of $\mathbf{B}$ and $\mathbf{E}$ and using the gauge conditions $\nabla \cdot \mathbf{A} = 0$.

$$\nabla \times \left(\mu_R^{-1} \nabla \times \mathbf{A}\right) = \mu_0 \mathbf{J_0} \quad \text{in } \Omega_n \tag{6a}$$

$$\nabla \times \left(\mu_R^{-1} \nabla \times \mathbf{A}\right) + \mu_0 \sigma \frac{\partial \mathbf{A}}{\partial t} + \mu_0 \sigma \nabla V = \mathbf{0} \quad \text{in } \Omega_c \tag{6b}$$

$$\mu_0 \nabla \cdot (\sigma \nabla V) = 0 \quad \text{in } \Omega_c \tag{6c}$$

Motion voltage term

$$\mathbf{E}' = \mathbf{E} + \mathbf{u} \times \mathbf{B}$$

$$\mathbf{J}' = \mathbf{J} = \sigma \mathbf{E}' = \sigma(\mathbf{E} + \mathbf{u} \times \mathbf{B})$$

$$\mathbf{J} = \sigma \left(-\frac{\partial \mathbf{A}}{\partial t} - \nabla V + \mathbf{u} \times \mathbf{B}\right)$$

Adding motion voltage term in Ω_c

$$\nabla \times \left(\mu_R^{-1} \nabla \times \mathbf{A}\right) + \mu_0 \sigma \frac{\partial \mathbf{A}}{\partial t} + \mu_0 \sigma \nabla V - \mu_0 \sigma \mathbf{u} \times (\nabla \times \mathbf{A}) = \mathbf{0} \quad \text{in } \Omega_c$$

1.4 Weak formulation

Two function spaces,
$\mathcal{V}$, a vector space for $\mathbf{A}$
$\mathcal{Q}$, for scalar variable V.

$$\begin{aligned}&\int_{\Omega_n} \nabla \times \left(\mu_R^{-1} \nabla \times \mathbf{A}\right) \cdot \mathbf{v}\ \mathrm{d}x + \int_{\Omega_c} \nabla \times \left(\mu_R^{-1} \nabla \times \mathbf{A}\right) \cdot \mathbf{v} \\ &+ \mu_0 \sigma \frac{\partial \mathbf{A}}{\partial t} \cdot \mathbf{v} + \mu_0 \sigma \nabla V \cdot \mathbf{v}\ \mathrm{d}x - \mu_0 \sigma \mathbf{u} \times (\nabla \times \mathbf{A}) \cdot \mathbf{v}\ \mathrm{d}x = \int_{\Omega_n} \mu_0 \mathbf{J_0} \cdot \mathbf{v}\ \mathrm{d}x\end{aligned} \tag{7}$$

Coupled system

$$\begin{aligned}&\int_{\Omega_n \cup \Omega_c} \mu_R^{-1} (\nabla \times \mathbf{A}) \cdot (\nabla \times \mathbf{v})\ \mathrm{d}x + \int_{\partial\Omega} \mu_R^{-1} \mathbf{v} \cdot (\mathbf{n} \times \nabla \times \mathbf{A})\ \mathrm{d}s \\ &+ \int_{\Omega_c} \mu_0 \sigma \frac{\partial \mathbf{A}}{\partial t} \cdot \mathbf{v} + \mu_0 \sigma \nabla V \cdot \mathbf{v} + \mu_0 \sigma \nabla V \cdot \nabla q\ \mathrm{d}x \\ &- \int_{\Omega_c} \mu_0 \sigma \mathbf{u} \times (\nabla \times \mathbf{A}) \cdot \mathbf{v}\ \mathrm{d}x = \int_{\Omega_n} \mu_0 \mathbf{J_0} \cdot \mathbf{v}\ \mathrm{d}x\end{aligned} \tag{8}$$

1.5 Weak formulation for 2D

Input current $\mathbf{J_0} = (0, 0, J_0) \implies \mathbf{A} = (0, 0, A_z) \implies \nabla \times A = \mathbf{e}_x \frac{\partial A_z}{\partial y} - \mathbf{e}_y \frac{\partial A_z}{\partial x}$
We also assume that $\frac{\partial V}{\partial z} = 0$.
This simplifies the terms in the weak formulation to:

$$\begin{aligned}&\int_{\Omega_n \cup \Omega_n} \mu_R^{-1} \nabla A_z \cdot \nabla v_z\ \mathrm{d}x + \int_{\partial\Omega} \mu_R^{-1} v_z \left(n_x \frac{\partial A_z}{\partial x} - n_y \frac{\partial A_z}{\partial y}\right)\ \mathrm{d}s \\ &+ \int_{\Omega_c} \mu_0 \sigma \frac{\partial A_z}{\partial t} v_z\ \mathrm{d}x + \int_{\Omega_c} \mu_0 \sigma \frac{\partial V}{\partial z} v_x\ \mathrm{d}x + \int_{\Omega_c} \mu_0 \sigma \left(\frac{\partial V}{\partial x}\frac{\partial q}{\partial x} + \frac{\partial V}{\partial y}\frac{\partial q}{\partial y}\right)\ \mathrm{d}x \\ &+ \int_{\Omega_c} \mu_0 \sigma (\mathbf{u} \cdot \nabla A_z) v_z\ \mathrm{d}x = \int_{\Omega_n} \mu_0 J_{0,z} v_z\ \mathrm{d}x\end{aligned} \tag{9}$$

1.6 Induction motor mesh

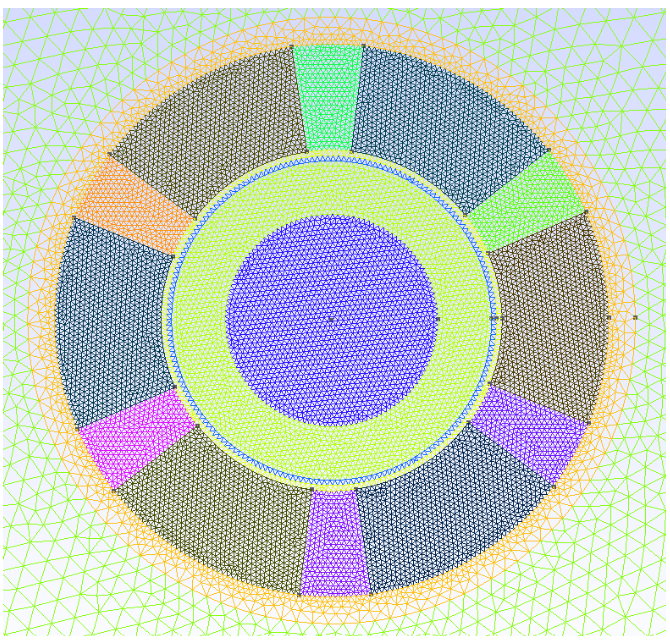
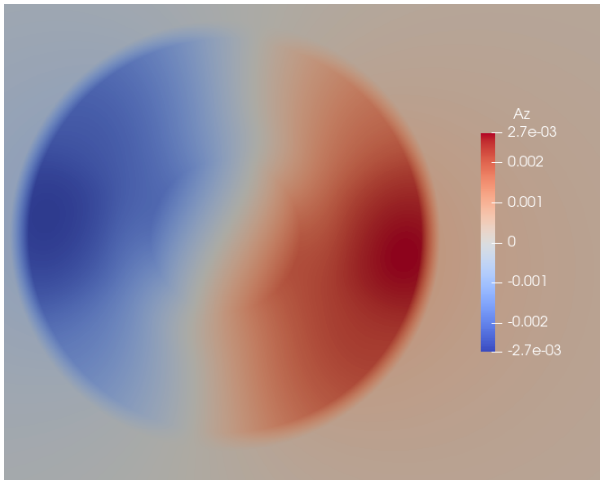
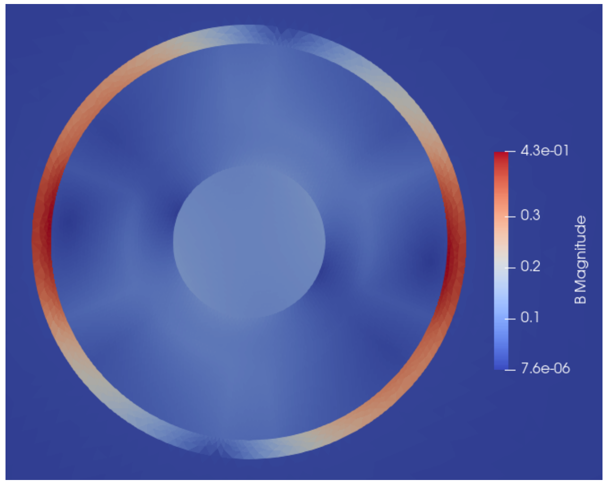


Figure 1: Three phase induction motor mesh built using GMSH

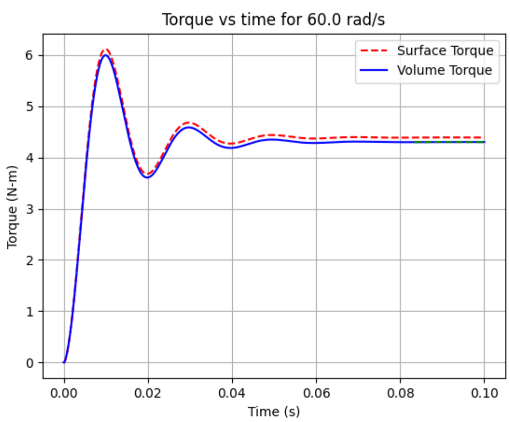
1.7 Results in Paraview



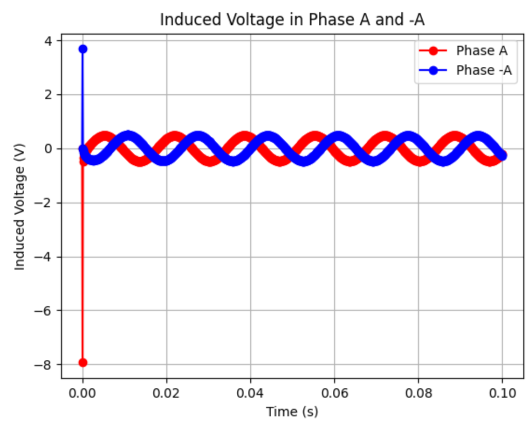
(a) Magnetic vector potential, A



(b) Magnetic flux density, B



(a) Electromagnetic torque, $\mathbf{T_e}$ (N·m)



(b) Induced voltage (V)

2 Permanent Magnet Synchronous Motor (PMSM)

Reference: Modelling a permanent magnet synchronous motor in FEniCSx for parallel high-performance simulations

2.1 Numerical formulation

Reference: PMSM Numerical formulation

2.1.1 Maxwell's equations

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{10a}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J} \tag{10b}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{10c}$$

$$\nabla \cdot \mathbf{D} = \rho \tag{10d}$$

2.1.2 Constitutive relations

$$\mathbf{B} = \mu \mathbf{H} \tag{11a}$$

$$\mathbf{J} = \sigma \mathbf{E} \tag{11b}$$

$$\mathbf{D} = \epsilon \mathbf{E} \tag{11c}$$

Within low-frequency machines, the displacement current will have a negligible effect on the magnetic field close to the current sources. Hence, the problem is simplified by omitting Eq. (10d), and asserting in Eq. (10b) to reduce it to $\nabla \times \mathbf{H} = \mathbf{J}$

2.1.3 A-V formulation

The A-V formulation begins by introducing the magnetic vector potential through the following definition

$$\mathbf{B} = \nabla \times \mathbf{A}$$

which is then substituted into Faraday's law (10a) to define the electric scalar potential

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla V$$

2.1.4 Modelling permanent magnets

Magnetic flux density is specified by the following unidirectional function:

$$\mathbf{B} = \mu \mathbf{H} + \mathbf{B_r}$$

$\mathbf{B_r}$ is the remanent magnetic flux density of the ferromagnet. The magnetisation $\mathbf{M}$ is calculated by:

$$\mathbf{M} = \nu_0 \mathbf{B_r}$$

Hence the behaviour of the permanent magnets is described by the equation:

$$\mathbf{B} = \mu \mathbf{H} + \mu_0 \mathbf{M} \tag{12}$$

2.1.5 Modelling rotation

The rotation of the motor is modelled by using the motion voltage term $\mathbf{u} \times \mathbf{B}$

$$\mathbf{E}' = \mathbf{E} + \mathbf{u} \times \mathbf{B}$$
$$\mathbf{J} = \sigma \left(-\frac{\partial \mathbf{A}}{\partial t} - \nabla V + \mathbf{u} \times \mathbf{B} \right)$$

$$\mathbf{J} = \sigma \left(-\frac{\partial \mathbf{A}}{\partial t} - \nabla V + \omega_r r \times \mathbf{B} \right) \quad \text{in } \Omega_r \cup \Omega_{pm} \tag{13}$$

2.1.6 Governing equations

The Coulomb gauge is defined as $\nabla \cdot A = 0$
The solution of the magnetic vector potential will be calculated using

$$\nu \nabla^2 \mathbf{A} = \sigma \frac{\partial \mathbf{A}}{\partial t} + \nabla V - \sigma \omega_v r \times (\nabla \times \mathbf{A}) \quad \text{in } \Omega_r \tag{14a}$$

$$\nu \nabla^2 \mathbf{A} = \sigma \frac{\partial \mathbf{A}}{\partial t} + \nabla V \quad \text{in } \Omega_s \tag{14b}$$

$$\nu \nabla^2 \mathbf{A} = -\mathbf{J}_s \quad \text{in } \Omega_c \tag{14c}$$

$$\nu \nabla^2 \mathbf{A} = -\nabla \times (\nu \mu_0 \mathbf{M}) - \sigma \omega_v r \times (\nabla \times \mathbf{A}) \quad \text{in } \Omega_{pm} \tag{14d}$$

the electric scalar potential will be calculated using

$$\sigma \nabla^2 V = \nabla \cdot \sigma \left(-\frac{\partial \mathbf{A}}{\partial t} + \omega_v r \times (\nabla \times \mathbf{A}) \right) \quad \text{in } \Omega_r \tag{15a}$$

$$\sigma \nabla^2 V = \nabla \cdot \left(-\sigma \frac{\partial \mathbf{A}}{\partial t} \right) \quad \text{in } \Omega_s \tag{15b}$$

$$\sigma \nabla^2 V = \nabla \cdot (\sigma \omega_v r \times (\nabla \times \mathbf{A})) \quad \text{in } \Omega_{pm} \tag{15c}$$

By assuming that the outer boundary is located far from the motor, where the magnetic fields are negligible, the following homogeneous Dirichlet condition can be imposed $\mathbf{A} = 0 \quad \text{in } \partial\Omega$
The unknown variables to be solved are represented by a trial function, and the trial and test functions must be defined in suitable function spaces, such as the Sobolev space.

2.1.7 Temporal discretisation

Backward Euler method was employed and the time derivative in the strong formulation is discretised as below:

$$\frac{\partial \mathbf{A}}{\partial t} = \frac{A_{n+1} - A_n}{\Delta t_n} \tag{16}$$

where Δt_n is the time-step. $t_{n+1} = t_n + \Delta t_n$

2.2 Weak formulation

The weak formulation of the problem reads: given A_n at time t_n, and $J_{s,n+1}$ and M_{n+1} at time t_{n+1}, find $A_{n+1} \in [Q_h]^3$ and $V_{n+1} \in Q_h$ such that

$$\begin{aligned} F_A(A_{n+1};\mathbf{v}) := &\int_\Omega \nu \nabla A_{n+1} \cdot \nabla \mathbf{v}\, dx + \int_{\Omega_{r,s}} \sigma \frac{A_{n+1} - A_n}{\Delta t} \cdot \mathbf{v}\, dx \\ &+ \int_{\Omega_{r,s}} \sigma V_{n+1} \cdot \mathbf{v}\, dx + \int_{\Omega_{r,pm}} \sigma \omega_r \times (\nabla \times A_{n+1}) \cdot \mathbf{v}\, dx \\ &- \int_\Omega J_{s,n+1} \cdot \mathbf{v}\, dx - \int_{\Omega_{pm}} \nu \mu_0 M_{n+1} \cdot (\nabla \times \mathbf{v})\, dx = 0, \quad \forall \mathbf{v} \in [Q_h]^3 \end{aligned} \tag{17}$$

$$\begin{aligned} F_V(V_{n+1};q) := &\int_\Omega \sigma \nabla V_{n+1} \cdot \nabla q\, dx - \int_{\Omega_{r,s}} \sigma \frac{A_{n+1} - A_n}{\Delta t} \cdot \nabla q\, dx \\ &+ \int_{\Omega_{r,pm}} \sigma \omega_r \times (\nabla \times A_{n+1}) \cdot \nabla q\, dx = 0, \quad \forall q \in Q_h \end{aligned} \tag{18}$$

where Q_h is a finite element space and $Q_h \subset H^1(\Omega)$.

2.3 Weak formulation in 2D

Input current $\mathbf{J_0} = (0, 0, J_0) \implies \mathbf{A} = (0, 0, A_z) \implies \nabla \times A = \mathbf{e}_x \frac{\partial A_z}{\partial y} - \mathbf{e}_y \frac{\partial A_z}{\partial x}$
We also assume that $\frac{\partial V}{\partial z} = 0$.

Coupled system after simplifying to 2D:

$$\begin{aligned} &\int_{\Omega_c \cup \Omega_n} \mu^{-1} \nabla A_z \cdot \nabla v_z \;\mathrm{d}x + \int_{\Omega_c} \sigma \frac{\partial A_z}{\partial t} v_z \;\mathrm{d}x \\ &+ \int_{\Omega_c} \sigma \frac{\partial V}{\partial z} v_x \;\mathrm{d}x + \int_{\Omega_c} \sigma \left(\frac{\partial V}{\partial x}\frac{\partial q}{\partial x} + \frac{\partial V}{\partial y}\frac{\partial q}{\partial y} \right) \;\mathrm{d}x \\ &+ \int_{\Omega_c} \sigma (\mathbf{u} \cdot \nabla A_z) v_z \;\mathrm{d}x = \int_{\Omega_n} J_{0,z} v_z \;\mathrm{d}x + \int_{\Omega_{pm}} \mu^{-1} \mu_0 \left(M_x \frac{\partial v_z}{\partial y} - M_y \frac{\partial v_z}{\partial x} \right) \;\mathrm{d}x \end{aligned} \tag{19}$$

2.4 Comparison with Induction motor 2D

1. Defined the Permanent Magnet (PM) Region Ω_{pm}.
2. Added PM region to the conducting region. $\Omega_c = \Omega_c \cup \Omega_{pm}$
3. **Magnetisation source term:** $\int_{\Omega_{pm}} \mu^{-1} \mu_0 \left(M_x \frac{\partial v_z}{\partial y} - M_y \frac{\partial v_z}{\partial x} \right) \;\mathrm{d}x$
4. Modelled rotation for PM.
5. Remaining regions are kept the same and the same boundary conditions are applied.

2.5 FEniCSx code for PMSM 2D Model

PMSM 2D variational form code snippet

GitHub Link - github.com/abhinavtk7/team30-demo/blob/main/pmsm.py

```python
cell = mesh.ufl_cell()
FE = ufl.FiniteElement("Lagrange", cell, degree)    # cell = 'triangle', degree = 1
ME = ufl.MixedElement([FE, FE])
VQ = fem.FunctionSpace(mesh, ME)

# Define test, trial and functions for previous timestep
Az, V = ufl.TrialFunctions(VQ)
vz, q = ufl.TestFunctions(VQ)
AnVn = fem.Function(VQ)
An, _ = ufl.split(AnVn)  # Solution at previous time step
J0z = fem.Function(DG0)  # Current density


# Create integration sets
domains = {"Air": (1,), "AirGap": (2, 3), "Al": (4,), "Rotor": (5, ),
           "Stator": (6, ), "Cu": (7, 8, 9, 10, 11, 12),
           "PM": (13, 14, 15, 16, 17, 18, 19, 20, 21, 22)}

Omega_n = domains["Cu"] + domains["Stator"] + domains["Air"] + domains["AirGap"]
Omega_c = domains["Rotor"] + domains["Al"] + domains["PM"]
Omega_pm = domains["PM"]

# Magnetization part
coercivity = 8.38e5  # [A/m]
DG0v = fem.FunctionSpace(mesh, ("DG", 0, (2,)))
Mvec = fem.Function(DG0v)

pm_spacing = (np.pi / 6) + (np.pi / 30)
pm_angles = np.asarray([i * pm_spacing for i in range(10)], dtype=np.float64)

# link pm orientation angle to each marker
pm_orientation = {}
for i, pm_marker in enumerate(Omega_pm):
    pm_orientation[pm_marker] = pm_angles[i]

# Create integration measures
dx = ufl.Measure("dx", domain=mesh, subdomain_data=ct)

# Define temporal and spatial parameters
dt = fem.Constant(mesh, dt_)
x = ufl.SpatialCoordinate(mesh)

omega = fem.Constant(mesh, default_scalar_type(omega_u))

# Motion voltage term
u = omega * ufl.as_vector((-x[1], x[0]))

# Magnetization term
curl_vz = ufl.as_vector((vz.dx(1), -vz.dx(0)))
mag_term =  (mu_0/mu) * ufl.inner( Mvec , curl_vz) * dx(Omega_pm)
```

```python

# Define variational form
f_a =   + dt / mu * ufl.inner(ufl.grad(Az), ufl.grad(vz)) * dx(Omega_n + Omega_c) \
        + dt / mu * vz * (n[0] * Az.dx(0) - n[1] * Az.dx(1)) * ds \
        + sigma * (Az - An) * vz * dx(Omega_c) \
        + dt * sigma * ufl.dot(u, ufl.grad(Az)) * vz * dx(Omega_c) \
        - dt * J0z * vz * dx(Omega_n) \
        - dt * mag_term

f_v =   + dt * sigma * (V.dx(0) * q.dx(0) + V.dx(1) * q.dx(1)) * dx(Omega_c)

form_av = f_a + f_v
a, L = ufl.system(form_av)
```

Magnetization code
GitHub Link - https://github.com/abhinavtk7/team30-demo/blob/main/utils.py

```python
# Magnetization code

def update_magnetization(Mvec, coercivity, omega_u, t, ct, domains, pm_orientation):
    block_size = 2 # Mvec.function_space.dofmap.index_map_bs = 2 for 2D
    sign = 1

    for (material, domain) in domains.items():
        if material == 'PM':
            for marker in domain:
                inout = 1 if marker in [13, 15, 17, 19, 21] else -1
                angle = pm_orientation[marker] + omega_u * t
                Mx = coercivity * np.cos(angle) * sign * inout
                My = coercivity * np.sin(angle) * sign * inout

                cells = ct.find(marker)
                for cell in cells:
                    idx = block_size * cell
                    Mvec.x.array[idx + 0] = Mx
                    Mvec.x.array[idx + 1] = My

    Mvec.x.scatter_forward()
    # Mvec.vector.ghostUpdate(addv=PETSc.InsertMode.INSERT,
                              # mode=PETSc.ScatterMode.FORWARD)
```

2.6 PMSM mesh

GitHub Link - team30-demo/generate_pmsm_2D.py
Three-phase AC motor with 10 permanent magnets.
Rectangular magnets are approximated as segments of a circular ring.
Sub domains: Rotor steel, Rotor Aluminium, Permanent magnets, Air Gap, Copper coils, Stator, Air.

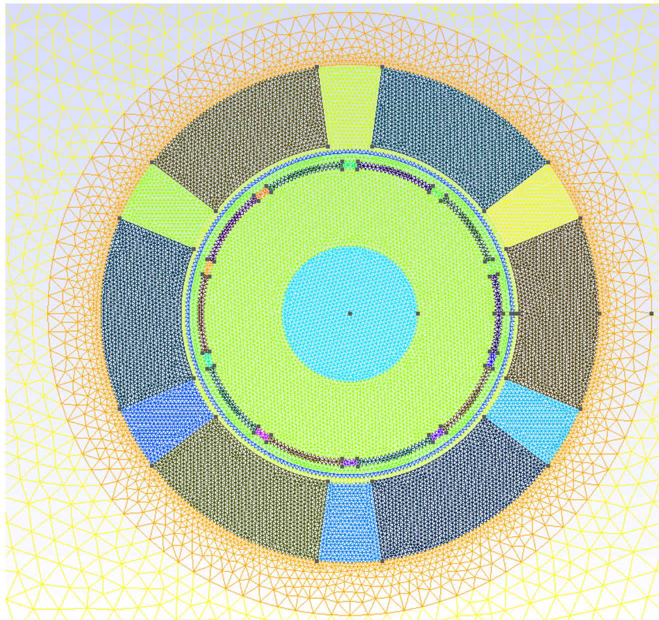
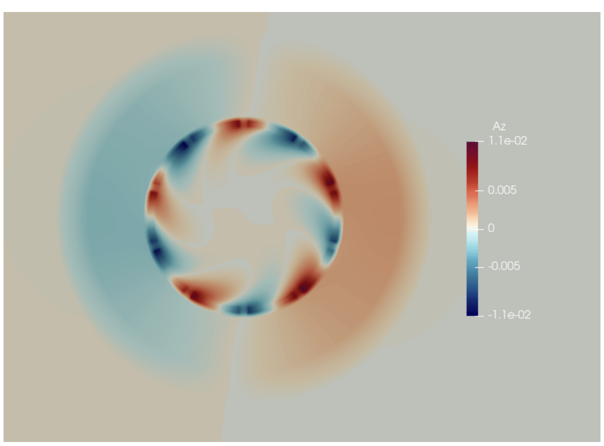


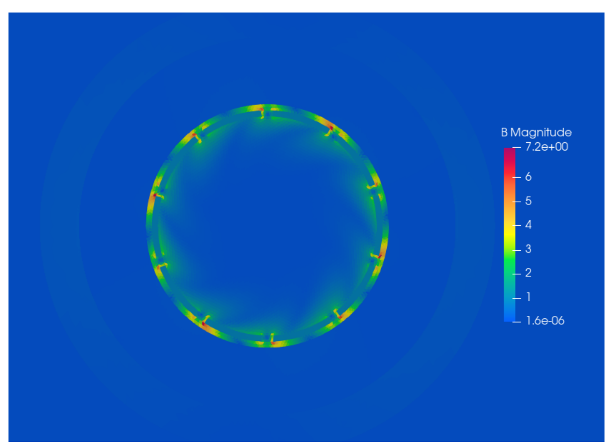
Figure 4: PMSM Mesh built using GMSH

Remarks:
Try different meshes by changing parameters.

2.7 PMSM 2D Model outputs in Paraview



(a) Magnetic vector potential, A



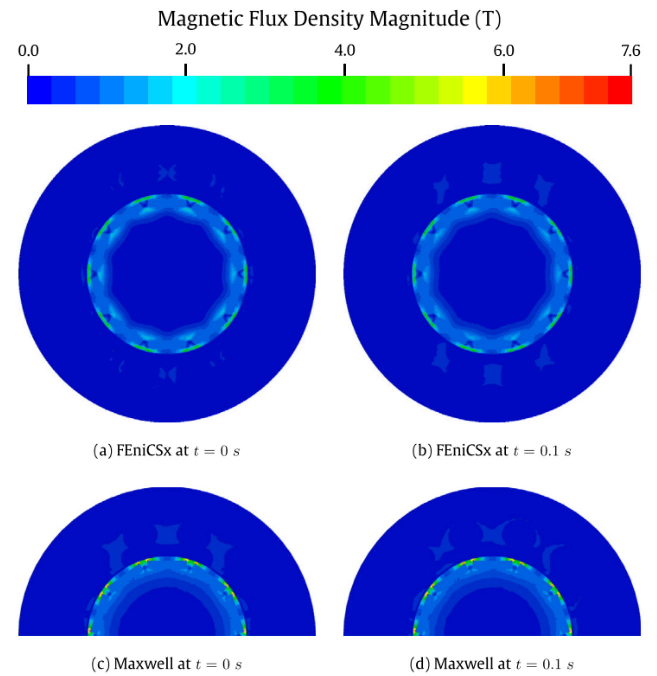
(b) Magnetic flux density, B

Remarks:
Need to analyze outputs by varying parameters.

2.8 Verification of the 2D model

2.8.1 Comparison of B with PMSM reference paper outputs

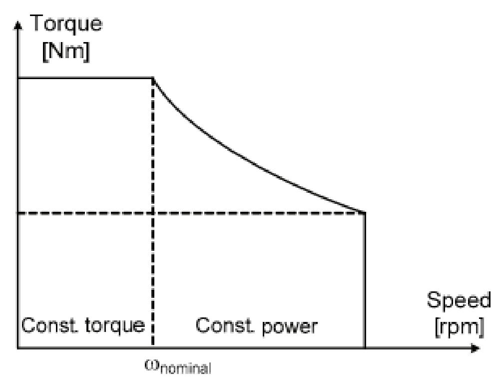
Reference: Modelling a permanent magnet synchronous motor in FEniCSx - Results & Discussion section



Contour plots of the magnetic flux density solution.

2.8.2 Comparing the torque-speed curve

Reference: Performance analysis of a PMSM drive with torque and speed control, Tomasz Rudnicki; Robert Czerwinski



The ideal torque characteristic curves for the PMSM motor

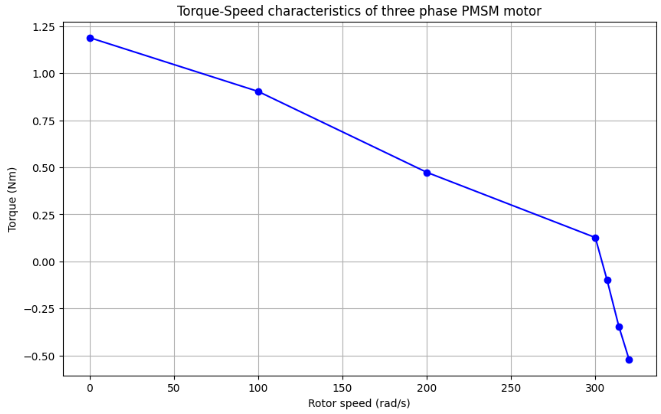


Figure 6: Average torque values for different rotor angular speed

The torque-speed characteristic of a motor illustrates the relationship between its torque and varying angular speed. This plot is unique to each motor. Here, we compare the ideal torque-speed characteristic of a PMSM motor with our simulated model. The curves exhibit similar variations, which validates our results.

Remarks:
Can we use torque-speed characteristic for verification?