Contents

1 First problem

1.1 PDF, initial and boundary conditions

In this example we are solving the following equation:

$$\frac{\partial \mathbf{X}}{\partial t} + [\nabla \times [\nabla \times \mathbf{X}]] = \mathbf{F} \tag{1}$$

on the rectangular manifold: $x \in [0, 1]$, $y \in [0, 0.2]$, $z \in [0, 1.5]$.
Let's consider the forcing vector

$$\mathbf{F} = \begin{pmatrix} -[2 + 3z(z - 1.5)] \exp(-3t) \\ 0 \\ -(\pi^2 + 3) \exp(\pi y - 3t) \end{pmatrix}$$

and boundary conditions

$$X_x\big|_{z=0} = 0;\ X_x\big|_{z=1.5} = 0$$
$$X_y\big|_{x=0} = 0;\ X_y\big|_{x=1} = 0$$
$$X_z\big|_{y=0} = \exp(-3t);\ \frac{\partial X_z}{\partial y}\bigg|_{y=0.2} = \pi \exp(\pi y - 3t).$$

The initial condition is

$$\mathbf{X}(x, y, z, t = 0) = \begin{pmatrix} z(z - 1.5) \\ 0 \\ \exp(\pi y) \end{pmatrix}$$

In this case, the solution of the equation (1) will be

$$\mathbf{X}(x, y, z, t) = \begin{pmatrix} z(z - 1.5) \exp(-3t) \\ 0 \\ \exp(\pi y - 3t) \end{pmatrix} \tag{2}$$

1.2 Finite element method and the weak form

Now we are going to write eq. (1) in the weak form. We will proceed as follows:

1 – multiply eq. (1) by a trial function $\mathbf{w}$ and integrate it over the whole domain

$$\int \mathbf{w} \cdot \frac{\partial \mathbf{X}}{\partial t} dV + \int \mathbf{w} \cdot [\nabla \times [\nabla \times \mathbf{X}]] dV = \int \mathbf{w} \cdot \mathbf{F} dV. \tag{3}$$

2 – make use of $[\nabla \times [\nabla \times \mathbf{X}]] = \nabla(\nabla \cdot \mathbf{X}) - \nabla^2 \mathbf{X}$ and rewrite eq. (3) in component notation

$$\int w_k \cdot \frac{\partial X_k}{\partial t} dV + \int w_k \partial_k (\partial_i X_i) dV - \int w_k \partial_i \partial_i X_k dV = \int w_k F_k dV. \tag{4}$$

3 – integrate by parts the middle two terms of eq. (4) and use the finite element method assumption that $\mathbf{w} = 0$ at the boundary regions, where Dirichlet boundary conditions are specified.

$$\int w_k \cdot \frac{\partial X_k}{\partial t} dV - \int (\partial_k w_k)(\partial_i X_i) dV + \int (\partial_i w_k)(\partial_i X_k) dV =$$
$$= - \int_{\text{Neumann b.c.}} w_k (\partial_i X_i) n_k dS + \int_{\text{Neumann b.c.}} w_k (\partial_i X_k) n_i dS + \int w_k F_k dV. \tag{5}$$

Again, the two surface integrals on the right-hand side are non-zero over the regions where Neumann boundary conditions are introduced and zero if we are dealing with Dirichlet boundary conditions.

4 – For the divergence-free case (i.e. $\partial_i X_i = 0$) the last equation has a simpler form:

$$\int w_k \cdot \frac{\partial X_k}{\partial t} dV + \int (\partial_i w_k)(\partial_i X_k) dV = \int_{\text{Neumann b.c.}} w_k (\partial_i X_k) n_i dS + \int w_k F_k dV, \tag{6}$$

so that in the finite element method's approach the divergence-free condition of the solution can be satisfied embarrassingly easily.

For the Neumann boundary conditions we have

$$n_i \partial_i X_k|_{\delta\Omega} = f_k(\mathbf{x}, t),$$

thus the final expression for the weak form of eq. (1) is

$$\int w_k \cdot \frac{\partial X_k}{\partial t} dV + \int (\partial_i w_k)(\partial_i X_k) dV = \int_{\text{Neumann b.c.}} w_k f_k dS + \int w_k F_k dV, \tag{7}$$

1.3 The matrix-vector form

Now we need to divide our manifold into cells and in each cell perform a decomposition of $\mathbf{w}$ and $\mathbf{X}$ by means of dealII basic functions which are uniquely associated with nodes of the cells (these basis functions live not in the manifold of our problem, but in a different manifold. Therefore, in our code when integrating something we introduce Jacobian to go from one coordinate system to another)

In the simplest and most convenient form, the decomposition would look like

$$w_k(\mathbf{x}, t) = \sum_l N_{kl}(\xi(\mathbf{x})) c_{kl}(t), \quad X_k(\mathbf{x}, t) = \sum_l N_{kl}(\xi(\mathbf{x})) d_{kl}(t)$$

where $N_{kl}(\xi(\mathbf{x}))$ are the dealII basis functions, $c_{kl}(t)$ and $d_{kl}(t)$ are time-dependent weights of the decomposition process. In the sum l goes over all nodes of a cell within which the position vector $\mathbf{x}$ is located.

Actually, we can even further simplify this expression, if we recall that each node has 3 degrees of freedom (because of 3 dimensions). Therefore, N_{kl} can be written as N_{3l+k}, where $3l + k$ is a unique number. The maximum value of $3l + k$ for a cell represents the number of degrees of freedom in the cell, which is always three times larger than the number of nodes for the cell.

We can then plug this decomposition expression into eq. (7), but now in the sum the index $a = 3l + k$ will cover not only degrees of freedom of a particular cell but that of the whole manifold, since we are performing the integration over the entire manifold, not just one cell.

$$\sum_{l,m} \int N_{3l+k} N_{3m+k} c_{3l+k} \dot{d}_{3m+k} dV - \sum_{l,m} \int N_{3l+k,i} N_{3m+k,i} c_{3l+k} d_{3m+k} dV =$$

$$= \sum_l \int_{\text{Neumann b.c.}} N_{3l+k} c_{3l+k} f_k dV + \sum_l \int N_{3l+k} c_{3l+k} F_k dV \tag{8}$$

Several things to note: (1) the sum over i and k is also implied, (2) a dot over d_{3m+k} denotes a time derivative, (3) a comma in $N_{3l+k,i}$ represents a derivative of N_{3l+k} with respect to the ith spatial coordinate, (4) we can take the coefficients d and c out of the integrals.

The next step is to introduce the global matrices

$$M_{ab} = \int N_a N_b dV, \quad K_{ab} = \int N_{a,i} N_{b,i} dV$$

and the global forcing vector

$$F_a \equiv F_{3l+k} = \int N_{3l+k} F_k dV + \int_{\text{Neumann b.c.}} N_{3l+k} f_k dS.$$

This step allows us to rewrite eq. (8) in the matrix-vector form:

$$c_a M_{ab} \dot{d}_b + c_a K_{ab} d_b = c_a F_a. \tag{9}$$

Since the last equation has to be true for any trial function $\mathbf{w}$ (i.e. for any coefficients c_a), it immediately yields

$$\boxed{M_{ab} \dot{d}_b + K_{ab} d_b = F_a.} \tag{10}$$

All the matrices and vectors in this equation are constructed in the "Assembly part of our code" and then the solution for this equation (i.e. the d-coefficients) is calculated in the "Solve" part of the code.

Note, that Dirichlet boundary conditions sit directly in d-coefficients, while Neumann's are in the forcing vector F_a.

1.4 The last step. Dealing with time-dependence of the matrix-vector form

To solve eq. (10) we need to discretize it in time. We will use the backward Euler method. The time now takes only discrete values:

$$d_b(t) \approx d_b(t_n) \equiv d_b^n,$$

i.e. the upper index denotes a point in time at which d_b is evaluated.

Thus, at some point $t = t_n$ we obtain

$$M_{ab}^{n+1} \dot{d}_b^{n+1} + K_{ab}^{n+1} d_b^{n+1} = F_a^{n+1}.$$

We assume that both the matrices and the vectors can be time-dependent.

$$\dot{d}_b^{n+1} \approx \frac{d_b^{n+1} - d_b^n}{\Delta t}$$

$$M_{ab} \frac{d_b^{n+1} - d_b^n}{\Delta t} + K_{ab} d_b^{n+1} = F_a$$

$$\boxed{(M_{ab} + \Delta t K_{ab}) d_b^{n+1} = \Delta t F_a + M_{ab} d_b^n} \tag{11}$$

Therefore, by numerically solving eq. (11) n times we can obtain the solution at t^{n+1} if the initial conditions are specified.

In the code we introduce *the system matrix* $= M_{ab} + \Delta t K_{ab}$ and *the right hand side vector* (*RHS*) $= \Delta t F_a + M_{ab} d_b^n$.

1.5 Analyzing the solution of the first problem

Here we are going to present the analytical and numerical solutions of the eq. (1) with the boundary and initial conditions specified above. We have divided our rectangular mesh into 5, 10 and 10 cells along x, y and z dimensions respectively. Time t goes from 0 to 1.

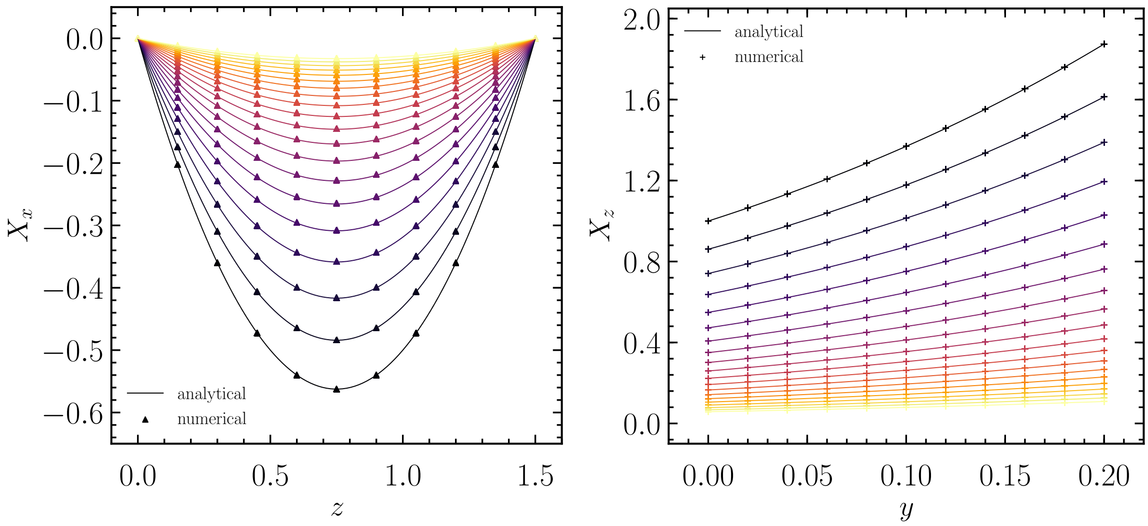

Figure 1: Solution of the first problem. The xth component as a function of coordinate z (left) and the zth component as a function of y (right).

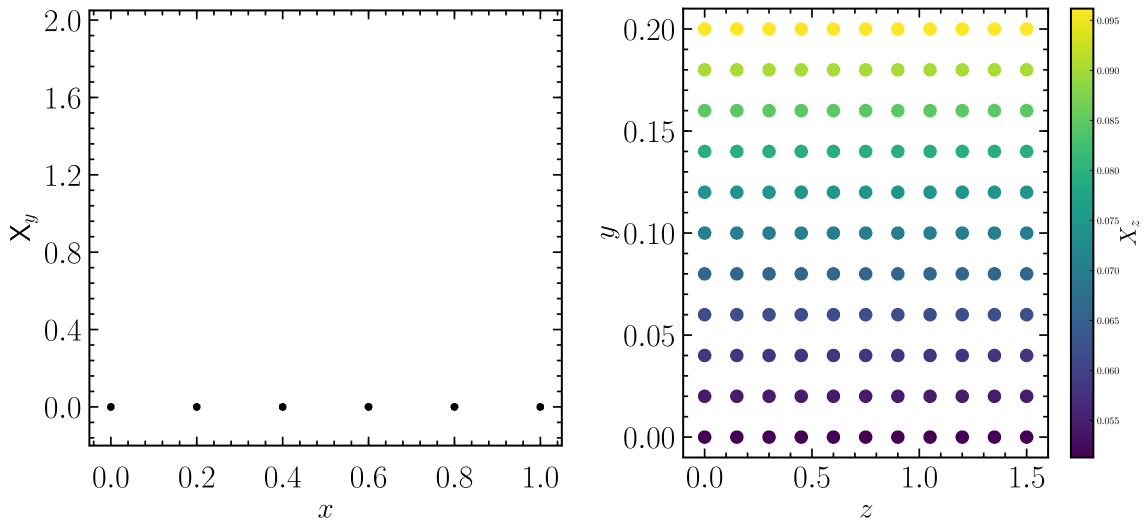

Figure 2: Solution of the first problem. The yth component as a function of coordinate x (left) and the zth component as a function of z and y coordinates (right).

2 Second problem

2.1 PDF, initial and boundary conditions

In this section we solve the problem outlined in (reference to Reisenegger 2017), namely, the growth of perturbation in the weak poloidal field in the presence of strong background toroidal component. The equation we have to deal with in the first approximation is given as follows

$$\frac{\partial \mathbf{B}_p}{\partial t} = -\nabla \times \left[\frac{c}{4\pi n} (\nabla \times \mathbf{B}_t) \times \mathbf{B}_p \right], \tag{12}$$

where the toroidal field $\mathbf{B}_t$ can be decomposed using the basis vectors of the cylindrical coordinates:

$$\mathbf{B}_t = 0 \cdot \mathbf{e}_R + B \cdot \mathbf{e}_\varphi + 0 \cdot \mathbf{e}_z, \tag{13}$$

and a scalar function B is defined as

$$B = (\chi_0/\chi)^2 = R^4 n^2, \tag{14}$$

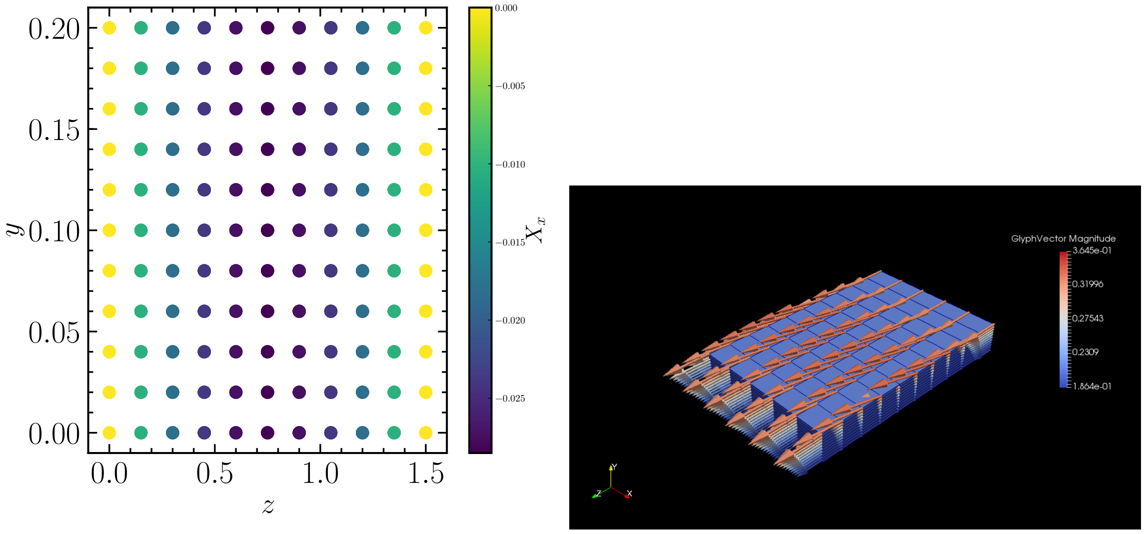

Figure 3: Solution of the first problem. The xth component as a function of coordinated y and z (left) and the 3D view at the rectangular manifold on which the equation (1) is solved. (right).

$$n = 1 - r^2,$$
$$R = \sqrt{x^2 + y^2},$$
$$r = \sqrt{x^2 + y^2 + z^2}.$$

In the Cartesian coordinates' basis the toroidal field is given in the form of

$$\mathbf{B}_t = -\sin\varphi B \cdot \mathbf{e}_x + \cos\varphi B \cdot \mathbf{e}_y + 0 \cdot \mathbf{e}_z =$$
$$\mathbf{B}_t = -\frac{y}{R} B \cdot \mathbf{e}_x + \frac{x}{R} B \cdot \mathbf{e}_y + 0 \cdot \mathbf{e}_z.$$

The toroidal field does not evolve with time so that the curl of $\mathbf{B}_t$ has to be computed just once in the beginning of the simulation.

The initial configuration for the poloidal perturbation is the following:

$$\mathbf{B}_p = 0 \cdot \mathbf{e}_R + 0 \cdot \mathbf{e}_\varphi + 10^{-3} \cdot \mathbf{e}_z, \tag{15}$$

The problem is solved on a spherical shell-like manifold, where the variable r takes values from some $r_{\min} = 0.1 - 0.3$ to $r_{\max} = 1.0$.

2.2 First step. Solving for velocities

As the first step, we calculate the velocity field $\mathbf{u}$ owing to the background toroidal magnetic field by merely taking a curl of $\mathbf{B}_t$ and then multiplying the result by a scalar function. We do it in a bit sophisticated manner though. We also introduce an auxiliary vector field $\mathbf{A} = 4\pi n \cdot \mathbf{u}$, so that we solve

$$4\pi n \cdot \mathbf{u} \equiv \mathbf{A} = [\nabla \times \mathbf{B}_t]\,. \tag{16}$$

We make use of a trial function $\mathbf{w}$ and integrate the equation (16) over the whole domain. This is a classic procedure carried out in any calculations that are somehow related to the finite element method. Henceforth we will omit the subscript t in the toroidal magnetic field $\mathbf{B}_t$ to not confuse it with the vector components (i.e. B_k, where k takes values 0, 1 and 2). Thus, we obtain

$$\int w_k A_k dV = \int w_k e_{kij} \partial_i B_j dV, \tag{17}$$

where summation under the repeating indices is assumed and e_{kij} denotes the Levi-Civita tensor. Also, below we will again use the notation $\partial_k B_i \equiv B_{i,k}$. After integrating by parts the right term we get

$$\int w_k A_k dV = \int B_k e_{kij} \partial_i w_j dV - \int n_k e_{kij} w_i B_j dS. \tag{18}$$

Remember, that the toroidal magnetic field $\mathbf{B}$ is known. The next steps are

$$w_k(\mathbf{x}) = \sum_l N_{kl}(\xi(\mathbf{x}))c_{kl}, \quad A_k(\mathbf{x}) = \sum_l N_{kl}(\xi(\mathbf{x}))d_{kl}, \tag{19}$$

$$\sum_{l,m} \int N_{3l+k} N_{3m+k} c_{3l+k} d_{3m+k} dV = \sum_l \int B_k e_{kij} N_{3l+j,i} c_{3l+j} dV - \sum_l \int n_k e_{kij} N_{3l+i} c_{3l+i} B_j dS,$$

$$K_{ab} = \int N_a N_b dV, \quad F_a \equiv F_{3l+j} = \int B_k e_{kij} N_{3l+j,i} dV - \int n_k e_{kji} N_{3l+j} B_i dS,$$

$$c_a K_{ab} d_b = c_a F_a,$$

$$\boxed{K_{ab} d_b = F_a} \tag{20}$$

The last equation is solved iteratively and the coefficients d_i are computed. The proper description of the last five lines of equations is given in the subsection 1.3. After the coefficients are specified, the vector field $\mathbf{A}$ is constructed and the velocity field $\mathbf{u} = \mathbf{A}/(4\pi n)$ is found.

2.3 Second step. Solving for poloidal perturbation

Now we seek for the solution of the partial differential equation

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times [\mathbf{u} \times \mathbf{B}], \tag{21}$$

where from now on $\mathbf{B}$ denotes the poloidal perturbation, not the toroidal magnetic field!

Taking the same steps as in the subsection 1.3 we obtain

$$\int w_k \frac{\partial B_k}{\partial t} dV = -\int w_i e_{ijk} \partial_j (e_{klm} u_l B_m) dV,$$

The term on the right hand side is integrated by parts

$$\int w_k \frac{\partial B_k}{\partial t} dV = -\int (e_{ilm} u_l B_m) e_{ijk} \partial_j w_k dV + \int n_i e_{ijk} w_j (e_{klm} u_l B_m) dS, \tag{22}$$

and DealII basis function decomposition of $\mathbf{B}$ and $\mathbf{w}$ is fulfilled

$$w_k(\mathbf{x}, t) = \sum_l N_{kl}(\xi(\mathbf{x}))c_{kl}(t), \quad B_k(\mathbf{x}, t) = \sum_l N_{kl}(\xi(\mathbf{x}))d_{kl}(t).$$

Now we plug the series representation of $\mathbf{B}$ and $\mathbf{w}$ back into (22)

$$\sum_{p,s} \int N_{3p+k} N_{3s+k} c_{3p+k} \dot{d}_{3s+k} dV = -\sum_{p,s} \int (e_{ilm} u_l N_{3s+m} d_{3s+m}) e_{ijk} N_{3p+k,j} c_{3p+k} dV +$$

$$+ \sum_{p,s} \int n_i e_{ijk} N_{3p+j} c_{3p+j} (e_{klm} u_l N_{3s+m} d_{3s+m}) dS. \tag{23}$$

Here we rewrite (23) in the matrix-vector notation

$$M_{ab} = \int N_a N_b dV, \quad K_{ab} \equiv K_{3s+m\,3p+k} = -\int e_{ilm} u_l N_{3s+m} e_{ijk} N_{3p+k,j} dV + \int n_i e_{ikj} N_{3p+k} e_{jlm} u_l N_{3s+m} dS.$$

The equation below has to hold true independently of the coefficients c_a

$$c_a M_{ab} \dot{d}_b = c_a K_{ab} d_b.$$

The final form is

$$\boxed{M_{ab} \dot{d}_b = K_{ab} d_b} \tag{24}$$

2.4 Third step. Evolution in time

Analogously to the cases described in the subsection 1.4, we proceed as follows

$$d_b(t) \approx d_b(t_n) \equiv d_b^n,$$

so that the upper index denotes a point in time at which d_b is evaluated.

At some point $t = t_n$ we thus get

$$M_{ab}^{n+1} \dot{d}_b^{n+1} = K_{ab}^{n+1} d_b^{n+1}$$

$$\dot{d}_b^{n+1} \approx \frac{d_b^{n+1} - d_b^n}{\Delta t}$$

$$M_{ab} \frac{d_b^{n+1} - d_b^n}{\Delta t} = K_{ab} d_b^{n+1}$$

$$\boxed{(M_{ab} - \Delta t K_{ab}) d_b^{n+1} = M_{ab} d_b^n} \tag{25}$$

The equation (25) is solved iteratively in the code, the set of coefficients d_b is found and the poloidal perturbation is evolved in time by Δt.

2.5 Illustration and analysis of the solution

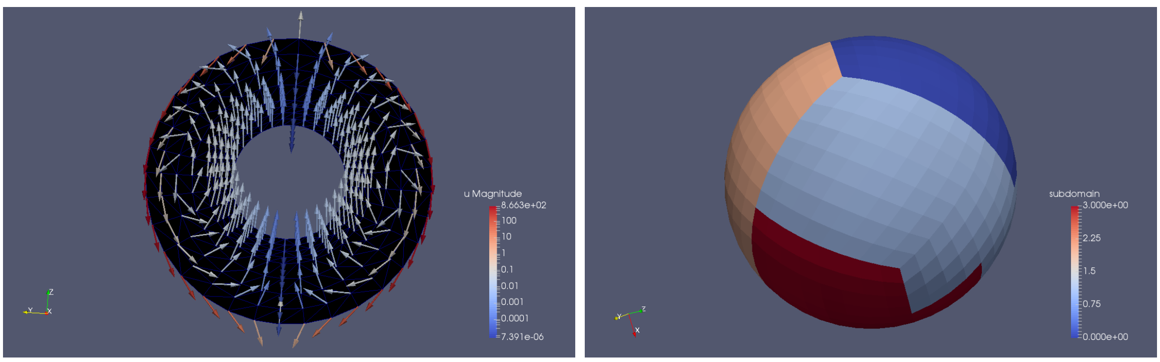

Figure 4: The velocity field $\mathbf{u}$ in the plane $x = 0$ (left) and the 4-processors' subdomains of the manifold (right)