1 The initial problem

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

$$X_x|_{z=0} = 0; \ X_x|_{z=1.5} = 0$$
$$X_y|_{x=0} = 0; \ X_y|_{x=1} = 0$$
$$X_z|_{y=0} = \exp(-3t); \ \left.\frac{\partial X_z}{\partial y}\right|_{y=2} = \pi \exp(\pi y - 3t).$$

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 first equation will be

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

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 dirvergene-free case (i.e. $\partial_i X_i = 0$) the last equation has a simplier form:

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

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

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

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

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 freadom of a particular cell but that of the whole manifold, since we are performing the integration over the entire manifold, not just one cell.

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

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

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 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 immeideatly 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.

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.

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

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

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

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) $= \Delta t F_a^{n+1} + M_{ab} d_b^n$.

5 Analyzing the results of the initial problem

Here we are going to show 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.

For further notes please see the notebook.