Up-to-Date Finite Element Based Simulation of Permanent Magnet

G. Kovács

Széchenyi István University, Department of Automation
Egyetem tér 1, H-9026 Győr, Hungary
e-mail: kovacsg@maxwell.sze.hu

Abstract: The paper presents a two dimensional finite element based solver for static magnetic field problems. The application has been written in C programming language. Magnetic vector potential of some reference models have been calculated by the help of two environments. The first one is the MATLAB environment and the second one is the C programming language based finite element code. The simulation results of the two environments were compared to each other focusing on the magnetic vector potential and the simulation time. Differences of the simulation results are showed in this paper, as well.

Keywords: Finite Element Method, C programming language, MATLAB, permanent magnet

1. Introduction

The computer-aided design is one of the important parts of the electric engine development. Electric engines have been improved at the Széchenyi István University, as well and one of the parts of this development is to design and optimization a brushless DC (direct current) motor family, which are will be applied with bicycles and smaller motors. There are also two projects where these motors will be applied. One of them is the development of hybrid E-VAN, which is an adapted Ford-truck. The second one is an electric car which is an individual development by the students and teachers of the university. The main aspect of this PMS motor (Permanent Magnet Synchronous motor) development is to reduce the weight and the size of the engine but the torque and losses of the motor should not decrease. There are different ways to design the PMS motors. This paper shows a C programming language based finite element simulation of the permanent magnets which are used in permanent magnet synchronous motors.

2. Structure of the development environment

The new finite element package consists of two main parts which can be seen in Fig. 1. The first part is the GMSH software [1] which is a two and three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. The second part of the finite element development environment is the C language based package under
Linux operating system, where partial differential equations and matrix operations can be solved. The finite element based solver has been written by the help of this package.

### Figure 1. Structure of the development environment

Many data have to be used in the FEM (Finite Element Method) structure. For example number of nodes, number of elements, material parameters, etc., which are given from the geometry and the mesh in the pre-processing step? These parameters and data are used in the Analysis step to solve the equations of partial differential equations. And finally the data of the simulation results are shown in the post-processing step.

Fig. 2. shows the graphical user interface of the finite element based package which consists of two parts.

### Figure 2. Graphical User Interface
On the left side the mesh, material parameters and boundary conditions can be set. And on the right side the model and the simulation results can be seen which a permanent magnet is.

3. Governing Equations

The basic model consists of a permanent magnet, a ferrite core and an excited single coil which can be seen in Fig. 3.

\[ \nabla \times \mathbf{H} = \mathbf{J}_0, \text{in } \Omega_0 \cup \Omega_m, \quad (1) \]
\[ \nabla \cdot \mathbf{B} = 0, \text{in } \Omega_0 \cup \Omega_m. \quad (2) \]
Here $H$ is the magnetic field intensity, $J_0$ is the source current density, $B$ is the magnetic flux density. The $H$ magnetic field intensity can be expressed as

$$H = \begin{cases} \nu_0 B, & \text{in } \Omega_0 \\ \nu_0 \nu_r B, & \text{in } \Omega_m \end{cases}.$$  \hspace{1cm} (3)

Here $\nu_0$ is the reluctivity of vacuum and $\nu_r$ is the relative reluctivity of magnet. The air region is denoted by $\Omega_0$ and the magnetic region is denoted by $\Omega_m$. The $A$ magnetic flux density can be expressed as

$$B = \nabla \times A,$$ \hspace{1cm} (4)

where $A$ is the magnetic vector potential. This expression satisfies (2), because of the identity $\nabla \cdot \nabla \times \mathbf{v}$ for any vector function $\mathbf{v} = \mathbf{v}(r)$.

When the domain contains permanent magnets, their magnetic characteristic are given by [3]

$$B = \mu H + B_0,$$ \hspace{1cm} (5)

where $B_0$ is the remanent flux density. Substituting (1) and (4) to (5) and using the constitutive relations IS (3) the following partial differential equation can be obtained:

$$\nabla \times \frac{1}{\mu} \nabla \times A - \nabla \times \frac{1}{\mu} B_0 = J_0.$$ \hspace{1cm} (6)

The divergence of the magnetic vector potential can be selected according to Coulomb’s gauge [5,7],

$$\nabla \cdot A = 0,$$

which is satisfied automatically in two dimensional problems [4, 5]. Using some mathematical identity and using some formulations [4, 5], the following weak equation can be obtained

$$\int_{\Omega} \frac{1}{\mu} \nabla \times W \cdot \nabla \times A \, d\Omega - \int_{\Omega} \frac{1}{\mu} \nabla \times W \cdot B_0 \, d\Omega = \int_{\Omega} W \cdot J_0 \, d\Omega,$$ \hspace{1cm} (7)

which solution results in the approximation of the magnetic vector potential.

4. Simulation results of the models

Five simple different arrangements had been created from the basic model. These models are calculated in the C programming language based on the finite element environment and in MATLAB environment. The simulation results were compared each other focusing the time and the accurate of the magnetic vector potential. In both environments the unknowns are the same, or closely the same which is 12790.

4.1. Comparison of the computation time of the models

Computation times of two different development environment are compared with each other. The sum up of the computation times of the simulations can be seen in Table 1.
Table 1. Computation times of models in two different environments

<table>
<thead>
<tr>
<th>model</th>
<th>MATLAB</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>excited coil</td>
<td>60sec</td>
<td>13sec</td>
</tr>
<tr>
<td>ferrite core and excited coil</td>
<td>60sec</td>
<td>13sec</td>
</tr>
<tr>
<td>magnet</td>
<td>60sec</td>
<td>12sec</td>
</tr>
<tr>
<td>magnet and ferrite core</td>
<td>59sec</td>
<td>10sec</td>
</tr>
<tr>
<td>magnet, ferrite core and excited coil</td>
<td>63sec</td>
<td>11sec</td>
</tr>
</tbody>
</table>

On the left side the computation times in MATLAB environment are shown, and on the right side the computation times in C environment are shown in Table 1. In case of every arrangement the problem was calculated almost five times faster by the help of the C environment than with MATLAB.

4.2. Comparison of the magnetic vector potential of the models

Simulation results were also compared with each other focusing on the accuracy of the magnetic vector potential. Fig. 4. shows the simulation result of the magnetic potential in the case of excited one turned coil model. Fig 4a shows the simulation results in GMSH environment. Fig 4b shows the simulation results in MATLAB environment.

Comparing Fig. 4a and 4b, it can be seen that the magnitude of the magnetic vector potential in the model are similar.

Fig. 5. shows the simulation result of the magnetic potential, where the model consists an excited one turned coil with ferrite core. Fig. 5a shows the simulation results in GMSH environment. Fig. 5b shows the simulation results in MATLAB environment.

176
Comparing the Fig. 5a and 5b figures with each other, it can be seen that the magnitude of the magnetic vector potential in the model are similar.

Fig. 6. shows the simulation result of the magnetic potential, where the model consists a permanent magnet. Fig. 6a shows the simulation results in GMSH environment. Fig. 6b shows the simulation results in MATLAB environment.

Comparing the Fig. 6a and 6b figures with each other, it can be seen that the magnitude of the magnetic vector potential in the model are similar.
Fig. 7. shows the simulation result of the magnetic potential, where the model consists a permanent magnet with ferrite core. Fig. 7a shows the simulation results in GMSH environment. Fig. 7b shows the simulation results in MATLAB environment.

Comparing the Fig. 7a and 7b, it can be seen that the magnitude of the magnetic vector potential in the model are similar.

Fig. 8. shows the simulation result of the magnetic potential, where the model consists of a permanent magnet with ferrite core and one turned excited coil. Fig. 8a shows the simulation results in GMSH environment. Fig. 8b shows the simulation results in MATLAB environment.

Figure 7. Simulation results of magnet with ferrite core

Figure 8. Simulation results of magnet, ferrite core and excited coil
Comparing the Fig. 8a and 8b, it can be seen that the magnitude of the magnetic vector potential in the model are similar.

4.3. Differences of the simulation results

Simulation results were also compared with each other focusing on the magnitude of the differences of models.

Fig. 9. shows the differences of the simulation results of the two different development environment in the case of the model of one turned excited coil.

![Figure 9. Margin between the simulation results, model: excited coil](image)

The magnitude of the difference of the simulation results is about $10^{-5}$.

Fig. 10. shows the differences of the simulation results of the two different development environment in the case of the model of one turned excited coil with ferrite core.

![Figure 10. Margin between the simulation results, model: excited coil with ferrite core](image)
The magnitude of the difference of the simulation results is about $10^{-4}$.

Fig. 11. shows the differences of the simulation results of the two different development environment in the case of the model of permanent magnet.

![Figure 11](image1.png)

*Figure 11. Margin between the simulation results, model: magnet*

The magnitude of the difference of the simulation results is about $10^{-4}$.

Fig. 12. shows the differences of the simulation results of the two different development environment in the case of the model of permanent magnet with ferrite core.

![Figure 12](image2.png)

*Figure 12. Margin between the simulation results, model: magnet with ferrite core*

The magnitude of the difference of the simulation results is about $10^{-4}$.

Fig. 13. shows the differences of the simulation results of the two different development environment in the case of the model of permanent magnet with ferrite core and one turned excited coil.

![Figure 13](image3.png)
The magnitude of the difference of the simulation results is about $10^{-4}$. It can be seen in every case that the differences of the simulation results are very small.

5. Conclusion

The new, finite element package based on C programming language is faster than the MATLAB solver, but the accuracy of the simulation results is adequate.

In the future work this C programming language based finite element package will be used for simulation and optimization of permanent magnet synchronous motors in a more fast and accurate way. This solution will have used in a research work where arrangements of the magnets and air gaps of the rotor of the BLDC motor to develop more energy efficient BLDC motors will have investigated.

Acknowledgement

This research was realized in the frames of TÁMOP 4.2.4. A/2-11-1-2012-0001 „National Excellence Program – Elaborating and operating an inland student and researcher personal support system convergence program” The project was subsidized by the European Union and co-financed by the European Social Fund.

References
