Application of Femap in Electromagnetic Field Simulation

Jing-Feng SUN, Qing-Xin YANG, Yong-Jian LI, You-Hua WANG, Chuang ZHANG
Province-Ministry Joint Key Laboratory of Electromagnetic Field and Electrical Apparatus Reliability
Hebei University of Technology
Tianjin, China
e-mail: sunjf@hebut.edu.cn

***Abstract*—A finite element analysis program was compiled by the author by means of Femap Application Programming Interface (API), and Femap became useful in electromagnetic filed simulation. In this paper, a model of current-carrying solenoid is built in Femap, the finite element analysis results are presented, and some key codes in the program are given.**

Keywords-fenite element method; Femap; electromagnetic filed simulation; API

I. INTRODUCTION

Finite element methods (FEM) have been used in science and industry since the 1960s [1,2], and they were widely applied in electromagnetic field simulation [3,4]. Some commercial software have been developed. And Femap is a good finite element modeling and post-processing software, which allows you to create your finite element models and to interpret results after your analysis easily and quickly. But it can not be applied directly to magnetic model, because it supports neither the magnetic properties of material, such as reluctivity, nor the load of current density.

In order to facilitate the application of Femap in electromagnetic field simulation, the author wrote a FEM program named Fert. Using Femap Application Programming Interface (API), Fert can read data from Femap, and write finite element analysis results into Femap database. This paper presents a step-by-step approach to the finite element simulation of a magnetic field using Femap.

II. PRE-PROCESSING

In pre-processing, you usually need to build a geometry model, define material properties, mesh the model, and finally apply constraints and loads.

A. Geometry

With robust Femap tools, you can build geometry from simple points to complex 3-D solids. In this paper, a 3-D model of a square current-carrying solenoid is constructed, as shown in figure 1. There are 5 solids in this model. Solid 2,3,4,5 make up the solenoid, and solid 1 is the air around it. The coil is divided into 4 parts for applying it a current density later.

B. Material and Property

Before meshing the model, all materials used in it should be defined. As mentioned above, we can not define reluctivity of a material in Femap, but it is really necessary in magnetic field simulation. So the job will be done by Fert later. But the materials should be created now, just to tell Fert how many kinds of materials have been used in this model.

Properties are used to define additional analysis information for one or more elements. It is meaningful for Fert that properties select the materials to be used in every element.

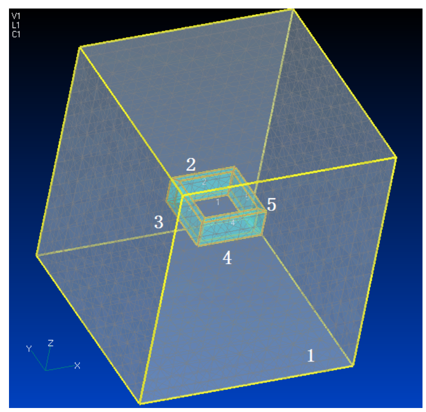

Figure 1. Model of a square current-carrying solenoid.

C. Load and Constraint

It is recommended that loads and constraints be applied after meshing, because any finite element analysis program will require the information of loads and constraints on nodes or elements. The meshing in Femap is automatic, and it needs not much attention. Femap allows you to create loads directly on elements.

Current density should be applied in this model, but it can not be defined in Femap. We can get another vector in place of current density, because what we need in calculation is just a set of values and their directions. And this vector is heat flux. In this model, four loads are applied corresponding to four solids of the solenoid. Two directions of theirs are parallel to X axis, and the other two are parallel to Y axis. Two values of theirs are positive, and the other two are negative. So these four loads compose a ring current that is shown in figure 2.

In Femap constraints could be applied to prevent nodes from moving in any of six degrees of freedom (DOF), X, Y and Z translation, and rotations about the X, Y and Z axes. In a magnetic field, there are only three DOF, the three axial components of magnetic vector potential $\boldsymbol{A}$, A_x, A_y *and* A_z.

978-0-7695-3634-7/09 $25.00 © 2009 IEEE
DOI 10.1109/ICIC.2009.360

You can select three DOF of the six arbitrarily in femap as long as you know what they represent and they would be treated correctly in your finite element analysis program. In this model, all nodes on six outside surfaces are treated as infinite boundary condition, and all their three DOF are constrained. Figure 3 shows these nodal constraints.

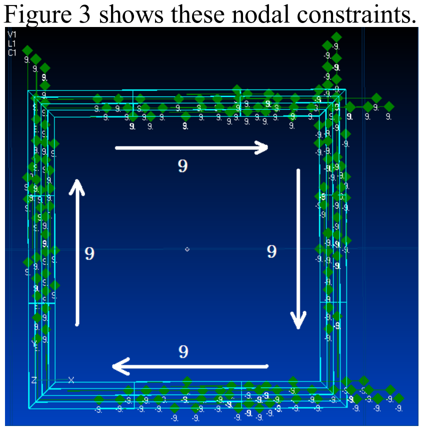

Figure 2. Loads on elements.

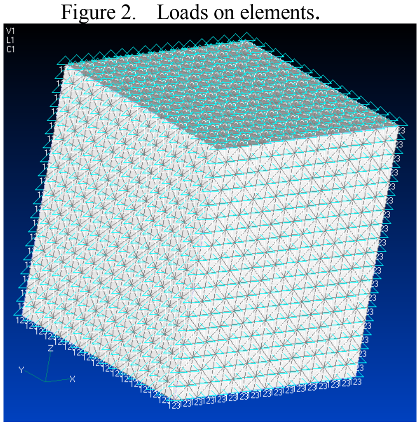

Figure 3. Nodal constraints.

III. TRANSLATING DATA

Although Femap supports some finite element analysis programs, such as NX Nastran, we can not use them in electromagnetic field simulation for the reason mentioned above. To use Femap in electromagnetic field simulation, a new program that can translate data with Femap is required. However, the finite element method is not the main focus in this paper. What we discuss here is how the program translates data using Femap API. Visual Basic 6.0 is taken as an example to introduce the corresponding program codes. So this paper is more meaningful for those who are studying on finite element method but not going to mesh the model themselves.

The Femap API is an object-oriented system, and many objects, methods and properties are available, all of which are described in detail in [5]. Only those used in this model are introduced in this paper.

Firstly the program should be connected with Femap. And the following codes will attach to the model in a Femap session that is already running.

```
Dim femap As Object
Set femap = GetObject(,"femap.model")
```

Then the program can read or write data Femap through the object named femap.

A. Checking and Renumbering

Before reading nodal or elemental data, there are two necessary operations to aim at in Femap.

The first one is to check coincident nodes. Because some curves were meshed with two or more meshing operations, there will be nodes in the model which are at the same location. This will make the elements separating logically, and the program will get a wrong coefficient matrix. In order to solve the problem, use the pull-down menu Tools > Check > Coincident Nodes to check and merge these nodes, and these elements will be connected together effectively.

The other one is to renumber all nodes and elements in the model, because usually the numbers of them are not consecutive when they are created, and this will raise program error when the program is reading data.

B. Reading Nodal Data

The precondition of reading nodal data is that there is a nodal object, and it could be created as follows.

```
Dim nd As Object
Set nd = femap.feNode
```

In order to read nodal data, we can use method Getallarray. This method retrieves arrays of values for a set of nodes. The first parameter is set to 0 to read the data of all nodes, and the ellipsis represents the data we do not concern. The first calling GetAllArray is to get the total number of nodes. Assign it to a long variable LngNon, and use it to define the array DblNode to save the coordinates of all the nodes. The second calling is to save the coordinates in DblNode. The example is as follows.

```
Dim rc As Long
rc = nd.getallarray(0, LngNon, …)
ReDim DblNode(1 To LngNon, 1 To 3)
rc = nd.getallarray(0, …, DblNode, …)
```

C. Reading Elemental Data

The processing of reading elemental data is almost the same with reading nodal data. The difference is that the object is changed from node to element. For each element, five variables should be saved, four of which are node numbers (Femap supports twenty) and the other is material number, if the elements have been selected as tetrahedron without midside nodes. The example is as follows.

```
Dim elem As Object
Set elem = femap.feElem
rc = elem.getallarray(0, LngNoe, …)
ReDim LngElem(1 To LngNoe, 1 To 5)
rc = elem.getallarray(0, …, propID, …, Nodes, …)
For ID = 1 To LngNoe
   LngElem(ID, 1) = Nodes(20 * (ID − 1))
   LngElem(ID, 2) = Nodes(20 * (ID − 1) + 1)
   LngElem(ID, 3) = Nodes(20 * (ID − 1) + 2)
   LngElem(ID, 4) = Nodes(20 * (ID − 1) + 4)
   LngElem(ID, 5) = propID(ID − 1)
Next
```

D. Reading Number of Materials

The electromagnetic properties of materials must be defined in your finite element analysis program. Before this, the number of materials should be clear. Method Last loads the last (with the highest ID) available entity. The highest ID is the total number of materials because the materials have been numbered consecutively (if not, renumber them), and it is assigned to LngNom. The example is as follows.

```
Dim mt As Object
Set mt = femap.feMatl
rc = mt.Last
LngNom = mt.ID
```

Then the material properties could be defined respectively in the program.

E. Reading Loads and Constraints

LoadMesh object corresponds to the nodal and elemental loads in the model. Reading loads data from the object needs one method and several object properties. Method next retrieves the next available entity with a larger ID. Property meshID finds the ID of the element that is loaded. Heatflux is the value for a heat flux load. Fluxdir(0), fluxdir(1) and fluxdir(2) are the components of the direction for the load. The example is as follows.

```
Dim ld As Object
Set ld = femap.feLoadMesh
While ld.Next
    DblLoad(ld.meshID, 1) = ld.heatflux
    DblLoad(ld.meshID, 2) = ld.fluxdir(0)
    DblLoad(ld.meshID, 3) = ld.fluxdir(1)
    DblLoad(ld.meshID, 4) = ld.fluxdir(2)
Wend
```

BCNode object corresponds to the nodal constraints in the model, and dof, one of its properties, is an array of flags telling whether or not the corresponding degrees of freedom are constrained (True) or free (False) in this constraint set. There are six elements in the array and three of them are required in magnetic fields.

Symmetrical boundary condition and infinite boundary condition are differently treated in the coefficient matrix. So there should be a difference between their records. To infinite boundary, only ID is recorded, and to symmetrical boundary, the dof is recorded too.

```
Dim bc As Object
Set bc = femap.feBCNode
While bc.Next
   If bc.dof(0) And bc.dof(1) And bc.dof(2) Then
      LngNb1 = LngNb1 + 1
      LngArrNb1(LngNb1) = bc.ID
   ElseIf bc.dof(0) Or bc.dof(1) Or bc.dof(2) Then
      LngNb2 = LngNb2 + 1
      VNb2(LngNb2, 1) = bc.ID
      VNb2(LngNb2, 2) = bc.dof(0)
      VNb2(LngNb2, 3) = bc.dof(1)
      VNb2(LngNb2, 4) = bc.dof(2)
   End If
Wend
```

F. Writing Results

After these reading, a calculation of finite element method could be executed, and the ultimate objective is the vectors of magnetic flux density $\boldsymbol{B}$. Inside one element, $\boldsymbol{B}$ is a constant, and each element has one vector. In Femap, you can initialize the current object to prepare it for creating output containing scalar data, but not vector, at each element. So the program writes the X, Y and Z component of the vector as scalars, and they will form a vecter again in post-processing.

The following codes shows how to write X components of these vectors into Femap, and the codes for the other two components are identical to these. Method InitScalarAtElem initializes the object and PutScalarAtElem stores the data. IDeV and ValV are two Variants in Visual Basic 6.0.

```
Dim o As Object
Set o = femap.feOutput
rc = o.InitScalarAtElem(1, 1001, "BX of Elements", 1, True)
For i = 1 To LngNoe
   IDe(i) = i
   Val(i) = DblB(3 * i— 2)
Next
IDeV = IDe
ValV = Val
rc = o. PutScalarAtElem (LngNoe, IDeV, ValV)
rc = o.put(—1)
```

IV. POST-PROCESSING

After importing the finite element analysis results into Femap, we can see them in the graphics windows. First, select the view style using the pull-down menu, View > Select > Contour Style > Vector. It displays the model with vectors in contour colors at elemental centroid or nodal position. To choose $\boldsymbol{B}$ displayed as vector in the view, use Deformed and Contour Data command button, select Vector Type > 3D Components, and select the X, Y and Z components of $\boldsymbol{B}$ in Vector 1.

Figure 4 shows the vectors of magnetic flux density from Fert in every element, and Figure 5 and 6 are their front view and top view. And this simulation result accords with the experiment.

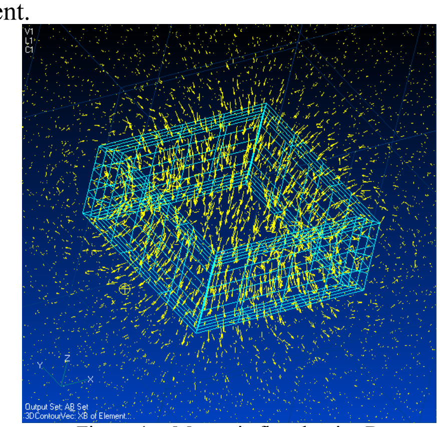

Figure 4. Magnetic flux density $\boldsymbol{B}$.

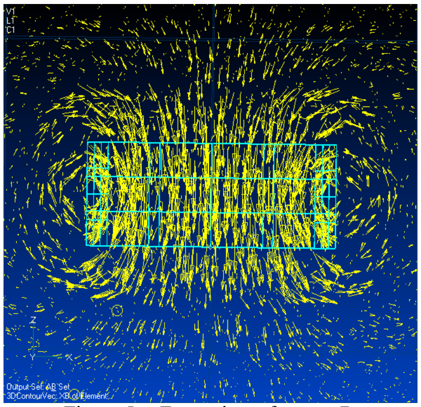

Figure 5. Front view of vector ***B***.

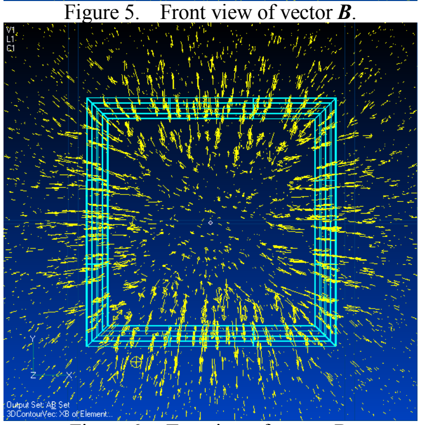

Figure 6. Top view of vector ***B***.

V. CONCLUSION

By the post-processing of the result from the program we can see that the whole simulation process is reliable. Data could be translated accurately between Femap and external program based on API. The method in this paper could be used to verify new algorithm of FEM. Femap is a good pre-processing and post-processing software for FEM, and there is scope for it in electromagnetic field simulation.

REFERENCES

[1] A. Nacar, A. Needleman, and PM. Ortiz, "A finite element method for analyzing localization in rate dependent solids at finite strains," Computer Methods in Applied Mechanics and Engineering, vol. 73, Aug. 1989, pp. 235-258, doi:10.1016/0045-7825(89)90067-4.

[2] A. I. Gasanov and I. E. Kaporin, "Use of the exclusion method in solving strictly elliptic systems using the finite element method," USSR Computational Mathematics and Mathematical Physics, vol. 26, Sep. 1987, pp. 120-130, doi:10.1016/0041-5553(86)90125-4.

[3] Bodo Heise, "Analysis of a fully discrete finite element method for a nonlinear magnetic field problem," SIAM Journal on Numerical Analysis, vol. 31, Jun. 1994, pp. 745-759, doi:10.1137/0731040

[4] Changfeng Ma, "Finite-element method for time-dependent Maxwell's equations based on an explicit-magnetic-field scheme," Journal of Computational and Applied Mathematics, vol. 194, Oct. 2006, pp. 409-424, doi:10.1016/j.cam.2005.08.008.

[5] Electronic Publication: UGS Corporation, Femap 9.2 Online Help, 2006.