1994

N95- 18974

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

S7-33
30897
P. 6

A FINITE ELEMENT CODE FOR ELECTRIC MOTOR DESIGN

Prepared By:	C. Warren Campbell, Ph.D., P.E.
Academic Rank:	Associate Professor
Institution and Department:	The University of Alabama in Huntsville Department of Civil & Environmental Engineering
NASA/MSFC:	
Laboratory:	Propulsion
Division:	Component Development
Branch:	Control Mechanisms
MSFC Colleague:	Rae Ann Weir

INTRODUCTION

NASA is studying the replacement of rocket engine hydraulic systems with electromechanical actuators (EMAs). Repairing and refurbishing hydraulic systems is one of the major costs of preparing the Space Shuttle for flight. Previously, EMAs could not match the weight, volume, and performance characteristics of hydraulic systems, but new rare earth magnet technology makes EMAs competitive.

Motor design technology has improved recently with the development of finite element methods for magnetic analysis[1-3]. Codes are available to do the analysis, but these codes are either costly or run on relatively unfriendly main frames. Furthermore, the codes are usually general purpose codes that solve a variety of problems (structural, thermal, dynamic, etc.). They are relatively costly, do not allow access to source codes, nor to all the variables that are required for the analysis. Furthermore, priorities for upgrades to the codes are determined by the marketplace. For these reasons, a NASA code that runs on user friendly PCs is desirable.

FEMOT PHILOSOPHY

The specifications for FEMOT were as follows:

1. Provide nonlinear solutions to the magnetostatic problem
2. Able to solve problems up to 15,000 nodes
3. Run on PCs
4. Requires a minimum of commercial software
5. Allow run time access to every variable in the code

FEMOT as implemented will run on a PC. The only commercial software required is Borland C/C++. The version used was 3.1. Version 4 is now available for less than $150. A spline interpolation function from *Numerical Recipes in C*[4] was the only other piece of software used. The spline routine can be obtained free from Internet. These requirements satisfy specification 4. Borland C/C++ has a symbolic debugger that permits stepping line by line through the code and allows access to every variable in the code during run time. This feature is useful for debugging the code as well as debugging motor runs on FEMOT. Hence, specification 5 is satisfied.

Memory is the greatest restriction to running on a PC. The code was developed on a 386 PC with 16 M RAM. To minimize the memory required, the code should store and solve linear systems using only the nonzero elements of the coefficient matrix. Using only the nonzero elements also reduces the number of calculations required. If a 10,000 node problem is to be solved, the full coefficient matrix will be 10,000 x 10,000 or 10^8 words of storage. Hence, sparse matrix technology must be applied.

FEMOT DATA STRUCTURES

The data structure adopted for this effort is a modification of the method used by Kundert[5] for nonsymmetric matrices. FEMOT generates positive definite, symmetric coefficient matrices so that Kundert's method could be simplified. The basic structure for the first three rows of a 7x7 matrix are shown in Figure VII-1.

SYMMETRIC SPARSE MATRIX STORAGE SCHEME

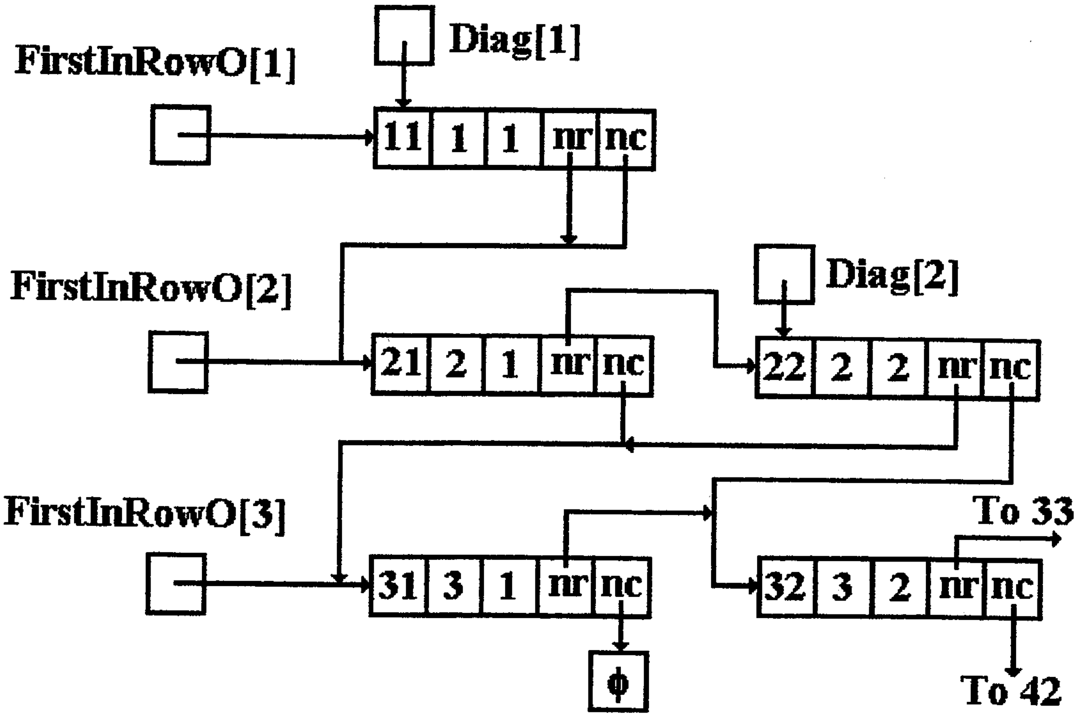

Figure VII-1. FEMOT Coefficient matrix data structure.

In this storage scheme, each matrix element is stored in a structure containing the element value, the row index i, the column index j, a pointer to the next element in the same row, and a pointer to the next element in the same column. If the element is on the main diagonal (i = j), the pointer to the next element in the row points to the first element in the next row. If the matrix contains no nonzero elements in the remainder of the column, the pointer points to the NULL pointer.

To overcome the problem of sequential access, two additional pointer arrays are used. FirstInRowO[i] points to the first element in row i, and Diag[i] points to the diagonal element. This is a slight modification of Kundert's method which used an additional pointer array to point to the first element in each column. For a symmetric matrix, the first element in each column is the diagonal element so that this pointer would be redundant with Diag.

This linked list permits dynamic memory allocation while the two pointer arrays increase the speed of linear equation solutions.

DATA INPUT AND MATRIX STRUCTURE

FEMOT uses the triangular, first order, Newton nonlinear element. Newton's method is used for fast solution of the nonlinear problem. The first version of FEMOT will update the coefficient matrix at each iteration, but this may be changed in a later version to speed the solution.

Figure VII-2 shows a simple example that illustrates the data input.

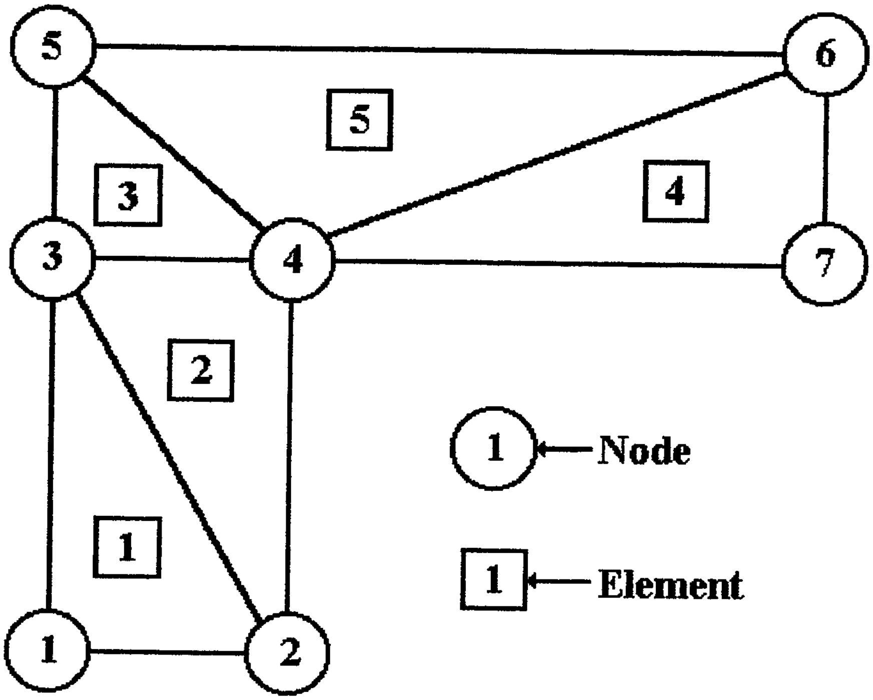

Figure VII-2 A simple finite element example.

The data structure for this example are given in Figure VII-3. For the simple geometry above, The top box in the figure gives the node number, x and y coordinates, whether or not the node is a fixed potential node (1 = yes) or not, and if the node potential is fixed, its value. Node zero at the bottom is a flag indicating that the node list is finished. The second box is the element list. From Figure VII-2, element 1 contains nodes 1, 2, 3 and element 2 contains nodes 2, 4, and 3. Each line of the element input gives the element number, the vertices of the element (nodes i, j, and k), the nodal sources (current densities), and the material number of the element. Because nodal sources are input, the current density has a linear variation in x and y across the element.

INPUT DATA

Node Data

Node	x	y	Fixed?	Potential
1	0.0	0.0	0	0.0
2	1.0	0.0	0	0.0
3	0.0	4.0	0	0.0
4	1.0	4.0	0	0.0
5	0.0	4.5	0	0.0
6	3.0	4.5	1	0.0
7	3.0	4.0	1	0.0
0	0.0	0.0	0	0.0

Element Data

Element	Node	Node	Node	Source	Source	Source	Material
No.	i	j	k	i	j	k	No.
1	1	2	3	1.0	1.0	1.0	1
2	2	4	3	1.0	1.0	1.0	1
3	3	4	5	0.0	0.0	0.0	2
4	4	7	6	0.0	0.0	0.0	2
5	4	6	5	0.0	0.0	0.0	3

Figure VII-3 Sample data input.

Figure VII-4 shows the coefficient matrix structure corresponding to Figures VII-2 and VII-3. The stars represent nonzero elements. The circled x is an element that is initially zero, but becomes non-zero during factorization. Since the matrix is symmetric, only the lower triangular part is shown.

$$\begin{pmatrix} * & & & & & & \\ * & * & & & & & \\ * & * & * & & & & \\ & * & * & * & & & \\ & & * & * & * & & \\ & & & * & * & * & \\ & & & * & \otimes & * & * \end{pmatrix}$$

Figure VII-4. Coefficient matrix structure

SUMMARY

FEMOT is a finite element program for solving the nonlinear magnetostatic problem. This version uses nonlinear, Newton first order elements. The code can be used for electric motor design and analysis. FEMOT can be embedded within an optimization code that will vary nodal coordinates to optimize the motor design. The output from FEMOT can be used to determine motor back EMF, torque, cogging, and magnet saturation. It will run on a PC and will be available to anyone who wants to use it.

REFERENCES

1. Silvester, P. P., Cabayan, H. S., and Browne, B. T., "Efficient Techniques for Finite Element Analysis of Electric Machines," *IEEE Transactions on Power App. and Syst.*, Vol. PAS-92, 1973, pp. 1274-1281.

2. Chari, M. V. K., and D'Angelo, J., "Finite Element Analysis of Magneto-Mechanical Devices," Fifth International Workshop in Rare Earth-Cobalt Permanent Magnets and their Applications, Roanoke, VA, 7-10 June, 1981, Paper No. VI-1.

3. Anderson, O. W., "Transform Leakage Flux Program Based on the Finite Element Method, *IEEE Transactions on Power App. and Syst.*, Vol. PAS-92, No. 2, 1973.

4. Press, William H., et al., *Numerical Recipes in C*, Cambridge University Press, New York, 1990.

5. Kundert, Kenneth S., "Sparse Matrix Techniques," in *Circuit Analysis, Simulation, and Design*, ed. by A. E. Ruehli, Elsevier Science Publishing Company, Inc., New York, 1986.