The Finite Element Solution of Two-Dimensional Transverse Magnetic Scattering Problems on the Connection Machine

Scott Hutchinson Steven Castillo
Edward Hensel Kim Dalton

CRPC-TR90085
April, 1990

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

The Finite Element Solution of Two-Dimensional Transverse Magnetic Scattering Problems on the Connection Machine

Scott Hutchinson[1] Steven Castillo[1]
Edward Hensel[2]
Kim Dalton[1]

[1] *Department of Electrical and Computer Engineering*
[2] *Department of Mechanical Engineering*
scastill@nmsu.edu
Box 3-o
New Mexico State University, Las Cruces, New Mexico 88003

April 6, 1990

Abstract

A study is conducted of the finite element solution of the partial differential equations governing two-dimensional electromagnetic field scattering problems on a SIMD computer. A nodal assembly technique is introduced which maps a single node to a single processor. The physical domain is first discretized in parallel to yield the node locations of an O-grid mesh. Next, the system of equations is assembled and then solved in parallel using a conjugate gradient algorithm for complex-valued, non-symmetric, non-positive definite systems. Using this technique and Thinking Machines Corporation's Connection Machine-2 (CM-2), problems with more than 250k nodes are solved.

Results of electromagnetic scattering, governed by the 2-d scalar Helmholtz wave equations are presented for a variety of infinite cylinders and airfoil cross-sections. Solutions are demonstrated for a wide range of objects. A summary of performance data is given for the set of test problems.

1 Introduction

The finite element technique is a method which allows for the approximate solution of partial differential equations over some finite domain. Because partial differential equations govern various physical phenomena, the technique has applications in many disciplines. Here, a study is conducted of the finite element solution of the partial differential equations governing two-dimensional electromagnetic field scattering problems on a SIMD computer.

First, the weak form of the continuous governing equations are given. Second, the mapping of the finite element program onto Thinking Machines Corporation's Connection Machine using *nodal assembly* is described. Third, results are presented for a variety of scattering shapes. Lastly, conclusions are drawn and future research discussed.

2 Problem Formulation

The equations of interest are the 2-d scalar and vector Helmholtz wave equations [1]. The equations are applied over an open region artificially truncated with an absorbing boundary condition [2]. The scalar equation

$$\nabla \cdot \frac{1}{\mu_r} \nabla E_z + k_0^2 \epsilon_r E_z = 0 \tag{1}$$

governs the transverse magnetic (TM) normal incident case and the vector equation

$$\nabla \times \frac{1}{\epsilon_r} \nabla \times \mathbf{H} - k_0^2 \mu_r \mathbf{H} = 0 \tag{2}$$

governs the TM oblique incident case. $\mathbf{H}$ represents the unknown magnetic field and E_z represents the z-component of the unknown electric field. Each case can be written

$$E_z = E_z^i + E_z^s \tag{3}$$

$$\mathbf{H} = \mathbf{H}^i + \mathbf{H}^s \tag{4}$$

where E_z^i and $\mathbf{H}^i$ represent the known incident fields while E_z^s and $\mathbf{H}^s$ are the unknown scattered fields (Figure 1).

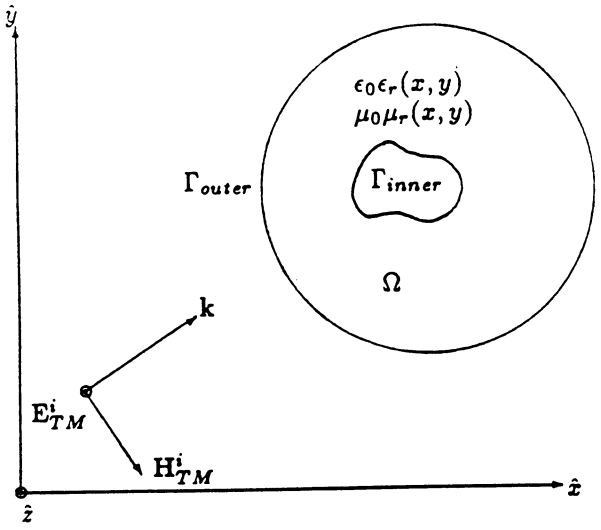

Figure 1: Open region scattering problem.

Applying the Galerkin technique, the scalar equation is written

$$\int_\Omega \left(\frac{1}{\mu_r}\nabla T\cdot\nabla E_z^s - k^2\epsilon_r T E_z^s\right)\,d\Omega - \int_\Gamma T\left(\alpha T E_z^s - \beta\frac{\partial T}{\partial\phi}\frac{\partial E_z^s}{\partial\phi}\right)\,d\Gamma = \int_\Omega T\left[\nabla\cdot\left(\frac{1}{\mu_r}\nabla E_z^i\right) + k^2\epsilon_r E_z^i\right]\,d\Omega \quad (5)$$

where the unknown is the scattered electric field and the vector equation is written

$$\int_\Omega \left(\frac{1}{\epsilon_r}\nabla\times\mathbf{T}\cdot\nabla\times\mathbf{H} - k^2\mu_r\mathbf{T}\cdot\mathbf{H}\right)\,d\Omega + \int_\Gamma [\alpha\mathbf{T}\cdot\mathbf{H}_{tan} - \beta(\hat{r}\cdot\nabla\times\mathbf{T})(\hat{r}\cdot\nabla\times\mathbf{H})]\,d\Gamma = \int_\Gamma [\alpha\mathbf{T}\cdot\mathbf{H}^i_{tan} - \beta(\hat{r}\cdot\nabla\times\mathbf{T})(\hat{r}\cdot\nabla\times\mathbf{H}^i)]\,d\Gamma - \int_\Gamma \mathbf{T}\cdot\hat{r}\times\nabla\times\mathbf{H}^i\,d\Gamma \quad (6)$$

where the unknown is the total magnetic field. In each case the Bayliss-Turkel radiation condition has been applied to satisfy the Neumann boundary condition on the outer boundary.

In order to obtain the final finite-element form, these equations are discretized and presented as a linear system of equations

$$\mathbf{K}\mathbf{u} = \mathbf{b} \quad (7)$$

for u the unknown.

3 Nodal Mapping

The SIMD computer used is Thinking Machines Corporation's Connection Machine 2 (CM-2). Briefly, the CM-2 is described as a SIMD (Sequential Instruction Multiple Data) or data parallel type of parallel computer. This means that each computer instruction operates on data stored in a processor array. Each processor in the array holds a single data item. The CM-2 may be configured to have up to 64k (k=1024) physical processors each with its own local memory. Computationally, each physical processor may be subdivided into some number of virtual processors where the computational resources of the physical processor are shared among its virtual processors. The virtual processor ratio, then, is the ratio of the number of virtual processors assigned to each physical processors and must equal an integer power of 2. For a more complete description, see [3].

While SIMD computers have been in existence for a number of years, finite element algorithms for them have been few. One reason is that, as with all parallel architectures, techniques which may be mature on serial computers must be altered and sometimes discarded in favor of more applicable algorithms. This paper introduces a new nodal basis mapping of the finite element algorithm onto the CM-2.

One difficulty with implementing finite element algorithms on a SIMD computer is the choice of the data item. To achieve a relatively high level of efficiency as well as to limit communication, a data item which may be maintained throughout the algorithm is desirable. Typically, finite element algorithms operate on an elemental level during the calculation of the system of equations and then assemble these elemental equations to a global set of equations which exist on the nodal level. This global set of equations is then solved to yield results at the nodal level. This may be seen as having two different data items during different portions of the program and previous implementations of this mapping on the CM-2 have proved inefficient [4], [5]. To avoid this inefficiency, an algorithm which uses a nodal level data set throughout the program has been developed for use on the CM-2. While the solution on the nodal level remains basically the same as previous finite element algorithms on the CM-2 [6], [7], the calculation of the system of equations is done on the nodal level using what has been termed *nodal assembly*.

3.1 Mesh Generation

The nodal-basis mapping assigns a node to a processor. This mapping is maintained throughout the program, from discretization through solution. During discretization, each processor calculates its position in the prob-

lem domain based on information which describes the domain geometry. Each processor also determines its boundary status. To enhance the speed of the program, a parallel O-grid mesh generator is used to generate meshes. The O-grid meshes allow the use of nearest neighbor (NEWS) communication grid while the parallel mesh generation means that only geometry data need be specified on a front-end preprocessor.

A mesh is generated by the set of points formed by the intersection of the lines of a boundary conforming curvilinear coordinate system. The problem of interest is a two-dimensional, multiply-connected, arbitrary region with specified inner and outer boundaries. The boundary values are specified in cartesian coordinates (x, y) and are transformed to curvilinear coordinates (s, t). In the transformed region, algebraic interpolation is used to generate the physical cartesian coordinates (x, y). See [8] for a complete description.

3.2 Nodal Assembly

The nodal assembly technique makes use of the concept of a *nodal region* which contains a given node and its neighboring nodes and elements as in Figure 2. Each processor simply calculates the local interaction coefficients associated with its row in the global system of equations as well as the forcing value. Since the interactions are local, nearest-neighbor communication is used. This portion of the algorithm is somewhat inefficient in applying boundary conditions since only processors which represent boundary nodes are active during this phase of the program. However, this may only be slightly detrimental to the overall efficiency of the program if the boundary–condition calculations are not too laborious.

3.3 System Solution

Once calculated, the system of equations is solved using a conjugate-gradient based algorithm [9]. Conjugate gradient algorithms have been used previously on the CM-2 for the solution of linear systems [6], [7]. This is because they are a collection of various matrix and vector operations which can be performed with a high level of concurrency. Further, in the case of a regular grid, all the system coefficients represent local interactions and so any interprocessor communication will be nearest neighbor. Thus, communication is also optimized using this solution technique. However, in contrast with previous finite element algorithms on the CM-2, the conjugate gradient algorithm used here is one which must handle a complex–valued, non-positive definite system of equations. It is given as

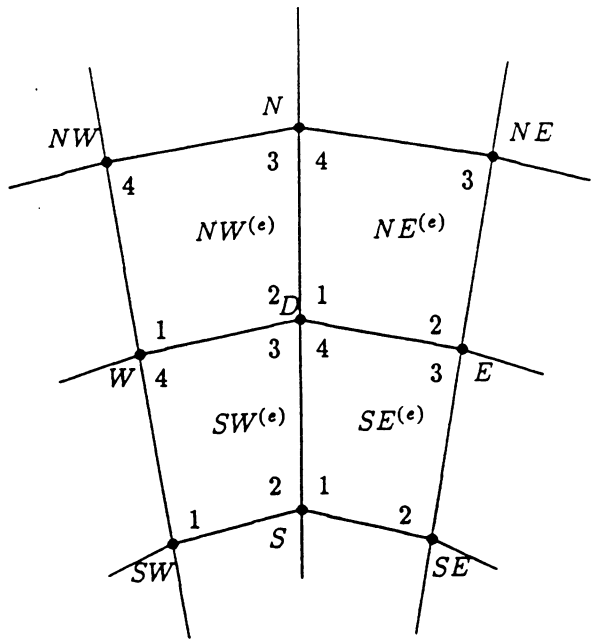

Figure 2: Nodal Region of node "D" indicating nearest neighbor nodes and adjacent elements.

Initialize:

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{K}\mathbf{u}_0 \quad (8)$$

$$\mathbf{p}_0 = \mathbf{K}^T\mathbf{r}_0 \quad (9)$$

Iterate:

$$a_i = \frac{|\,\mathbf{K}^T\mathbf{r}_i\,|^2}{|\,\mathbf{K}\mathbf{p}_i\,|^2} \quad (10)$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i + a_i\mathbf{p}_i \quad (11)$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i - a_i\mathbf{K}\mathbf{p}_i \quad (12)$$

$$b_i = \frac{|\,\mathbf{K}^T\mathbf{r}_{i+1}\,|^2}{|\,\mathbf{K}^T\mathbf{r}_i\,|^2} \quad (13)$$

$$\mathbf{p}_{i+1} = \mathbf{K}^T\mathbf{r}_{i+1} + b_i\mathbf{p}_i \quad (14)$$

where the choice of $\mathbf{u}_0$ is arbitrary. Note here that the matrix-transpose implies the conjugate-transpose.

Figure 3 is a flow chart of the finite element program as implemented on the CM-2.

4 Results

The method described above has been implemented on the CM-2 using the C-Paris (PARallel Instruction Set) programming protocol for the Connection Machine [10]. This program was used to obtain results for the solution of electromagnetic wave scattering from a variety

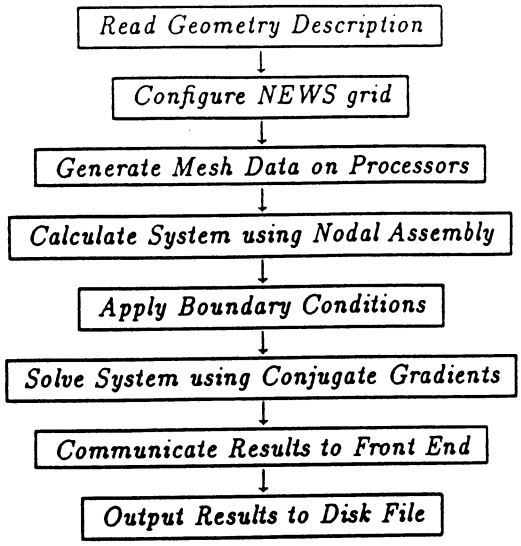

Figure 3: Flow chart for CM-2 nodal-basis finite element program.

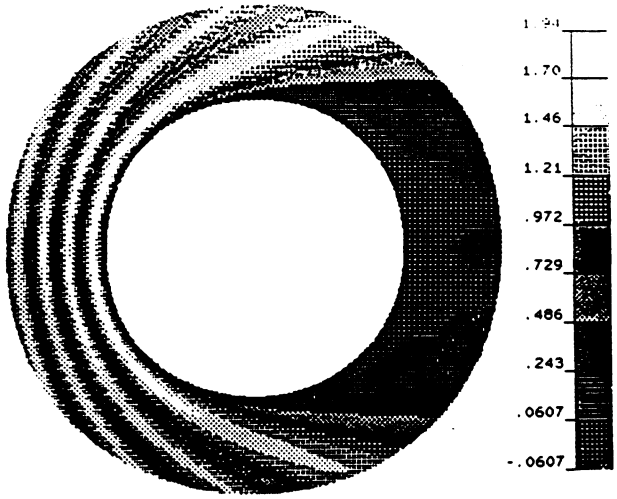

Figure 4: Case 1 magnitude: Total field magnitude for scattering from a perfect electric conducting cylinder with $a = 3\lambda$ and $b = 5\lambda$

of 2-dimensional objects. Table 1 gives some test problems for scattering from perfect electric conducting objects. The first 4 cases are all cylindrical shapes for which a semi-analytical solution is available for accuracy verification. The last case is an airfoil with NACA number 0010. All floating point calculations are done using 32 bit arithmetic and the floating–point acceleration hardware available on the CM-2. The conjugate gradient algorithm was halted when the following was satisfied

$$\frac{|\,\mathbf{r}_i\,|}{|\,\mathbf{b}\,|} < 10^{-4} \qquad (15)$$

Figures 4 – 13 represent magnitude and phase plots of the fields for the cases listed in Table 1. In each case, the incident plane wave is taken as traveling in the x-direction. The total field magnitude becomes zero on the boundary and displays a shadow region behind the conducting body. The phase plots illustrate that lines of constant phase approach the perfect conducting inner boundary at normal incidence. This follows from the boundary condition that

$$E_{tan} = 0$$

on the boundary.

Table 2 gives timing and Megaflop ratings achieved on the same problems. All timing results were obtained using the CM timing facility. As Table 2 illustrates, projected floating point computations from 200–400 MFlops have been achieved during both phases of the algorithm. Further, the MFlop ratings extrapolated to the same virtual processor ratio run on a full 64k CM-2

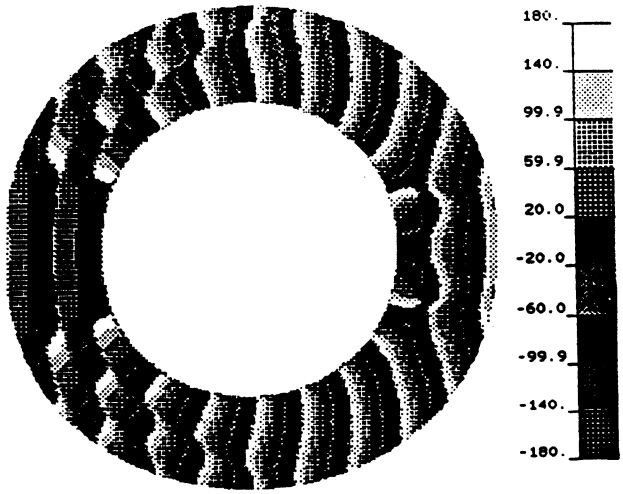

Figure 5: Case 1 phase: Total field phase for scattering from a perfect electric conducting cylinder with $a = 3\lambda$ and $b = 5\lambda$

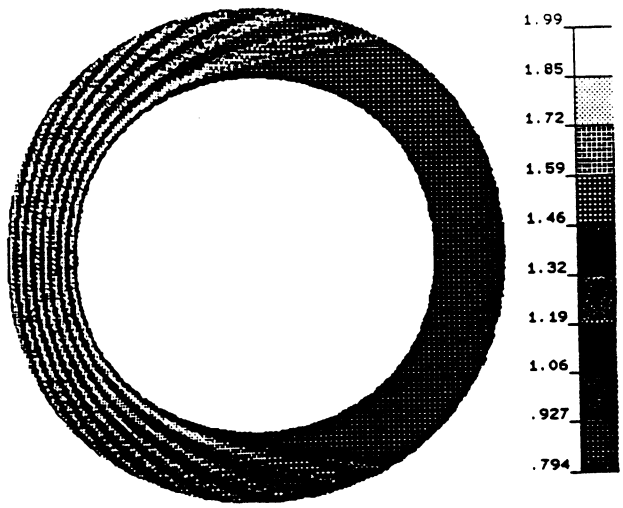

Figure 6: Case 2 magnitude: Total field magnitude for scattering from a perfect electric conducting cylinder with $a = 10\lambda$ and $b = 14\lambda$

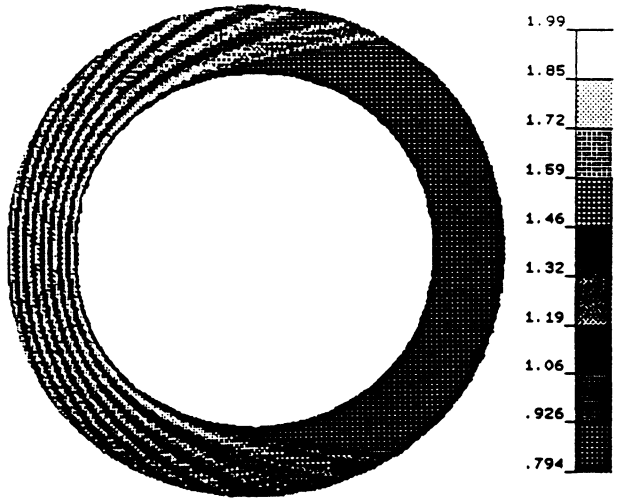

Figure 8: Case 3 magnitude: Total field magnitude for scattering from a perfect electric conducting cylinder with $a = 10\lambda$ and $b = 14\lambda$. Twice the nodal density was used in both the radial and circumferential directions as for Case 2

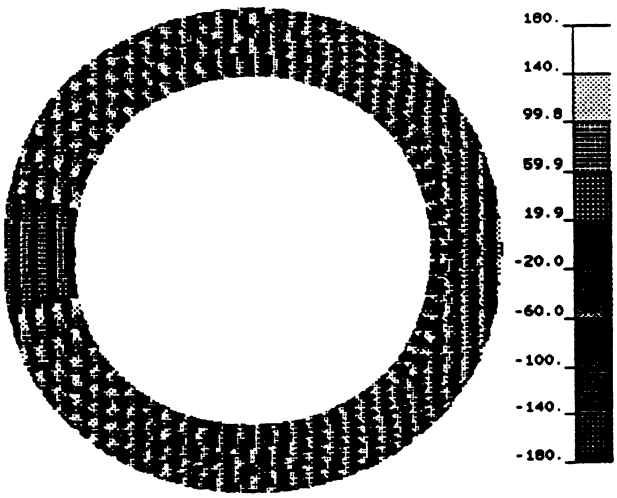

Figure 7: Case 2 phase: Total field phase for scattering from a perfect electric conducting cylinder with $a = 10\lambda$ and $b = 14\lambda$

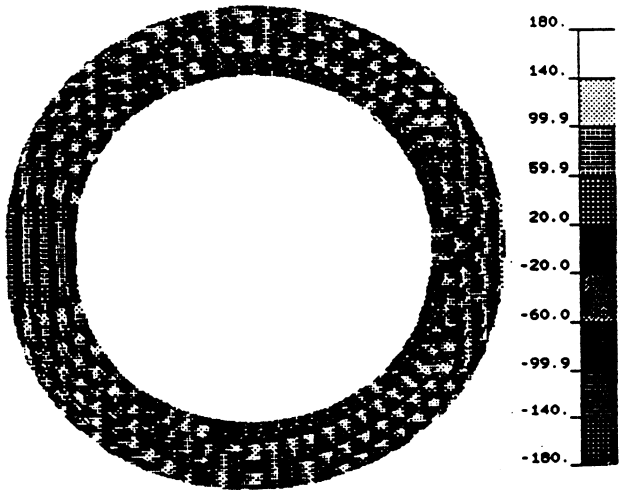

Figure 9: Case 3 phase: Total field phase for scattering from a perfect electric conducting cylinder with $a = 10\lambda$ and $b = 14\lambda$. Twice the nodal density was used in both the radial and circumferential directions as for Case 2

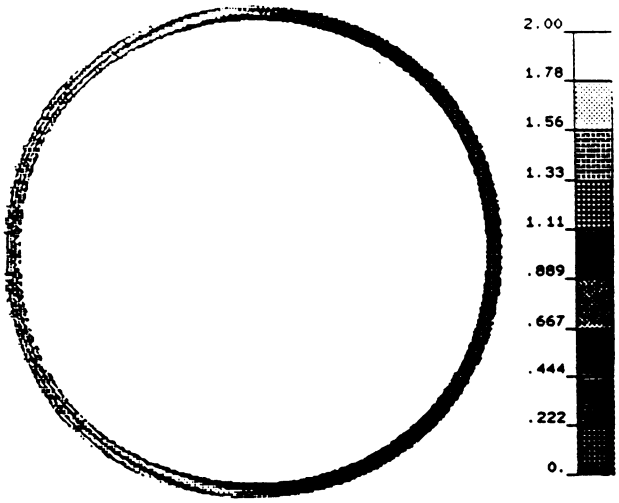

Figure 10: Case 4 magnitude: Total field magnitude for scattering from a perfect electric conducting cylinder with $a = 30\lambda$ and $b = 32\lambda$

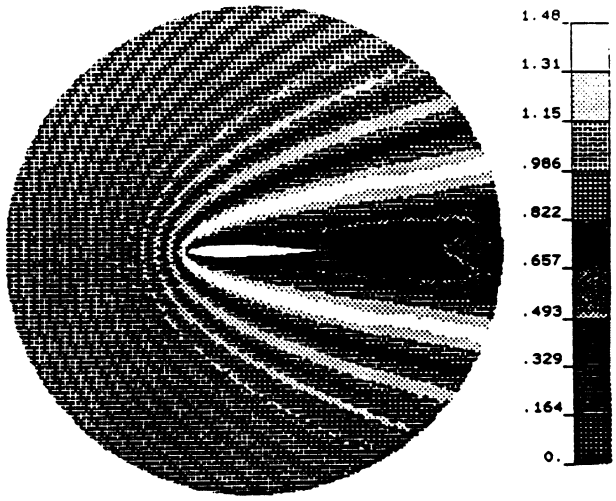

Figure 12: Airfoil magnitude: Total field magnitude for scattering from a perfect electric conducting airfoil with chord length = 5λ and $b = 9.5\lambda$. NACA number is 0010

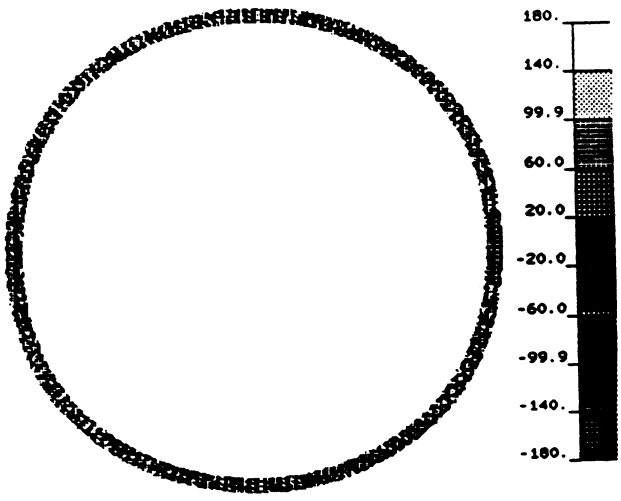

Figure 11: Case 4 phase: Total field phase for scattering from a perfect electric conducting cylinder with $a = 30\lambda$ and $b = 32\lambda$

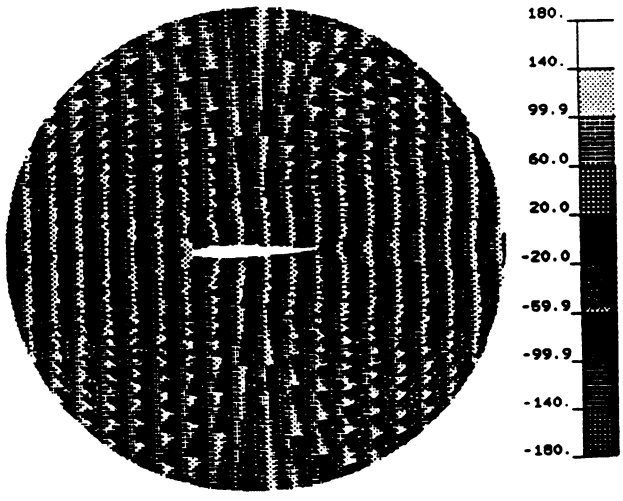

Figure 13: Airfoil phase: Total field phase for scattering from a perfect electric conducting airfoil with chord length = 5λ and $b = 9.5\lambda$. NACA number is 0010

show capabilities on the order of 1.5 GFlops for both portions. The finite element mapping described above will allow the solution of problems in excess of 4 million nodes on a fully configured CM-2 with the larger 256-kbit memory chips. Because of this capability, objects of electrical sizes (dimension in terms of wavelengths) exceeding 100 wavelengths may studied using the finite element method. This has not previously been possible.

Figure 14 shows a plot of the "relative speedup" demonstrated by the Fill and Solve portions of the program. This "relative speedup" is defined as

$$rsp \equiv \frac{t_{minp}}{t_{np}} \tag{16}$$

where t_{minp} is the execution time of a given problem on the minimum number of processors possible (highest virtual processor ratio) and t_{np} is the execution time on some number of processors. Note that the graph illustrates the speedup over the largest possible range of these ratios for this program using a CM-2 with processor memories of 64k-bits.

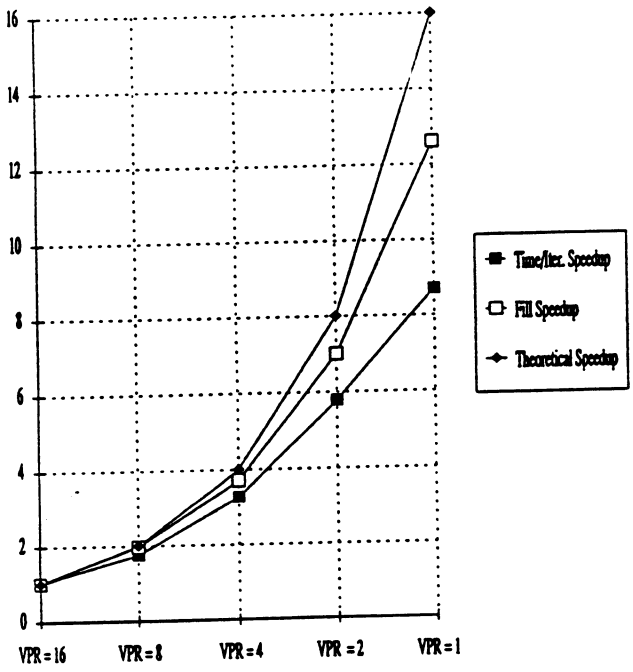

Figure 14: Relative Speedup.

Table 1: Test cases.

Case	a	b	Num. Nodes			V.P. Ratio
			Circ.	Radial	Total	
1	3 λ	5 λ	512	32	16384	1
2	10 λ	14 λ	1024	64	65536	4
3	10 λ	14 λ	2048	128	262144	16
4	30 λ	32 λ	2048	32	65536	4
5	5 λ^*	9.5 λ	1024	128	131072	8

* – the chord length of the airfoil was taken as 5λ.

Table 2: Timings and Mflop ratings for the above cases.

Case	Time (s)			MFlops	
	Fill	Solve	Total	Fill	Solve
1	0.05	33.52	113.25	365	211
2	0.17	137.88	314.65	430	322
3	0.78	1811.82	1967.79	375	399
4	0.17	125.24	224.72	430	318
5	0.32	-	6798	457	-

Table 3: Mflop ratings extrapolated to the virtual processor ratio implemented on a full 64k processor CM-2.

VP Ratio	Projected MFlops	
	Fill	Solve
1	1461	846
4	1719	1289
16	1500	1596

5 Conclusions and Future Research

Using the nodal–assembly technique, a finite element program is implemented on a data parallel computer in a manner which allows the use of the same data structure throughout the program, from discretization through solution. Nodal–assembly mapping provides for a relatively efficient program.

From the work presented here, conclusions may be drawn.

- Nodal assembly allows the mapping of one finite element node onto one virtual processor. This mapping is maintained throughout the program.
- Using first order quadrilaterals and a regular mesh, the mapping may be configured in a NEWS grid, allowing nearest-neighbor communication.
- The mapping described above will permit a maximum virtual processor ratio of 16 under the current CM-2 memory limitations (64k bits per processor). On a 64k processor machine, this allows a maximum of 1048576 nodes.
- Nodal assembly is inefficient when handling boundary conditions. This is because only processors on a given boundary are active during this portion of a program.
- Nodal basis mapping is well suited for use with a conjugate-gradient iterative solution. All the matrix and vector operations can be computed with a high level of concurrency. Nearest neighbor communication is again used in performing the matrix-vector products.

With respect to the CM-2, several things need be said. First, although all these examples were run on a machine with only a 32 bit floating point accelerator, 64 bit accelerators are now available to allow double precision floating point calculations in hardware. Second, the individual processor memory has been increased from 64k bits to 256k bits on some machines. This would effectively allow the solution of problems with 4 times the number of nodes. The 64k bits of processor memory allowed for a maximum virtual processor ratio of 16 for 10485760 nodes. The new memory size would allow a virtual processor ratio of 64 for 41943040 nodes.

Further research into data parallel techniques and their use in the solution of scattering problems is ongoing. These include an extension to 3-dimensional finite elements, effectiveness of the absorbing boundary condition for both 2- and 3-dimensional problems and convergence properties of the conjugate gradient algorithm when applied to complex geometries.

Also, with respect to mesh generation, several areas require further investigation. These include parallel mesh generation, mesh refinement techniques as well as other interpolation schemes. Mesh refinement techniques allow the program to actively alter the node distribution in the physical domain. This permits more nodes to be allocated in regions where the solution is expected to vary rapidly and fewer nodes in regions where the solution is expected to be relatively constant. Thus, a better approximation to the exact solution is obtained for a given number of nodes.

6 Acknowledgements

This work was supported by NSF grant # EET-8812958.

Computational resources were provided by Los Alamos National Laboratories, Los Alamos, New Mexico and by the Northeast Parallel Architectures Center (NPAC) at Syracuse University, which is funded by and operates under contract to, DARPA, and the Air Force Systems command, Rome Air Development Center (RADC), Griffiss AFB, NY, under contract number F30602-88-C-0031.

References

[1] R.F. Harrington, *Time Harmonic Electromagnetic Fields*, McGraw-Hill: San Francisco, 1961.

[2] A. Bayliss and E. Turkel, "Radiation boundary conditions for wave-like equations," *Communications on Pure and Applied Mathematics*, 33,707-725.

[3] Thinking Machines Corp. *Connection Machine Model CM-2 Technical Summary,* Version 5.1, Thinking Machines Corp., 1989.

[4] S.A. Hutchinson, S.P. Castillo and E. Hensel, "A basic finite element code on the Connection Machine," *Proceedings of the Fourth Annual Conference on Hypercubes, Concurrent Computers and Applications,* to be published, Monterey CA, 1989.

[5] S.A. Hutchinson, S.P. Castillo and E. Hensel, "Solving 2-d electrostatic problems on the Connection Machine using the finite element method," *Proceedings of the 5th Annual Review of Progress in Applied Computational Electromagnetics,* Monterey CA, 1989.

[6] R.E. Cline *et al.*, "Towards the development of engineering production codes for the Connection Machine," *Proceedings of the Fourth Annual Conference on Hypercubes, Concurrent Computers and Applications,* to be published, Monterey CA, 1989.

[7] S.L. Johnsson and K. Mathur, *Data Structures and Algorithms for the Finite Element Method on a Data Parallel Supercomputer,* Technical Report CS89-1, Thinking Machines Corp., 1988.

[8] J.F. Thompson, Z.U.A. Warsi and C.W. Mastin, *Numerical Grid Generation - Foundations and Applications,* North Holland: New York, 1985.

[9] M.R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solution of linear systems," *J. Res. Nat. But. Standards,* vol. 49, pp.409-436, 1952.

[10] Thinking Machines Corp. *Introduction to Programming in C/Paris,* Version 5, Thinking Machines Corp., 1989.