A VECTORIZED MULTIPLE PLATE SCATTERING CODE

John W. Nehrbass and Inder J. Gupta

The Ohio State University ElectroScience Laboratory
Dept. of Electrical Engineering
1320 Kinnear Road, Columbus, Ohio 43212

Abstract

A computer code named Vectorized Multiple Plate Scattering (VMPS) code has been developed at the Ohio State University ElectroScience Laboratory to compute the scattered fields from structures that can be modelled using perfectly conducting flat plates. The VMPS code uses a moment method approach to solve an electric field integral equation for the scattered fields. The code utilizes the vectorization capability of CRAY supercomputers to compute the scattered fields very efficiently. In this paper, the operation of the VMPS code is described and its vectorization efficiency is demonstrated.

I. INTRODUCTION

A computer code named Vectorized Multiple Plate Scattering (VMPS) code has been developed at The Ohio State University ElectroScience Laboratory (OSU-ESL) to compute the scattered fields from structures that can be modelled using perfectly conducting multiple polygonal flat plates. In the VMPS code, an electric field integral equation is solved using a moment method (MM) solution to compute the scattered fields. The code utilizes the vectorization capability of CRAY supercomputers and, thus, computes the scattered fields very efficiently. The MM solution used in the code is basically the same as used in the Electromagnetic Surface Patch (ESP) code [1]. However, the code has been rewritten to facilitate vectorization on a CRAY supercomputer.

In the MM solution, the equivalent currents on the various plates are approximated by piecewise sinusoidal basis functions defined over quadrilateral patches. Therefore, this code divides the various plates into quadrilateral patches. A piecewise sinusoidal function (mode) is defined over every two quadrilateral patches that share a common side. Whenever two plates have a common edge, overlap modes, as shown in Figure 1, are placed near the edge to ensure the continuity of currents. The test sources in the moment method solution are filamentary dipoles placed along the axial center of the various modes.

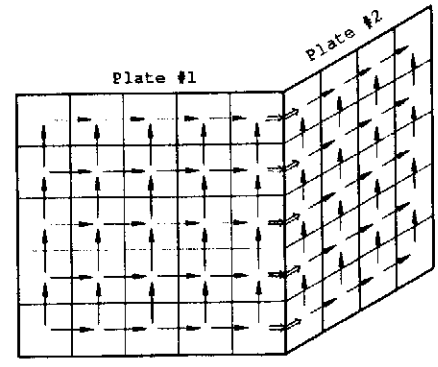

Figure 1: Quadrilateral patches on two flat plates and modal distribution. ($\longrightarrow$) plate modes, ($\Longrightarrow$) overlap modes.

In this paper, the operation of the code is described and the approach used to vectorize various section of the code is discussed. The CPU time used by the code on a VAX 8550 computer, an IBM 486 personal computer and a CRAY Y-MP computer are compared. The vectorization efficiency of the code is studied by executing the code on the CRAY Y-MP computer both in scalar and vector modes. It is shown that by developing a code so that vectorization is facilitated, one can decrease the CPU time by a factor as large as 8.

II. CODE DESCRIPTION

An EM moment method code, in general, consists of the following steps.

1. Read the geometry of the structure, frequency of operation and angular regions along which the scattered fields are to be computed.
2. Set up the modal distribution.
3. Compute the impedance matrix $[Z]$.
4. Compute the excitation vector $[V]$.
5. Solve for the unknown current coefficients $[I]$.
6. Using the equivalent currents, calculate the scattered fields.
7. Write the scattered fields to a data file.

For electrically large structures (defined by more than 50 modes), Steps 3 and 5 use most of the CPU time followed by steps 4 and 6. These four steps account for more than 98% of the total CPU time. The VMPS code is, therefore, written to facilitate the vectorization of these four steps. Since steps 1 and 7 involve input/output operations which can't be vectorized, no attempt has been made to vectorize these sections of the code. Further, existing software from the Electromagnetic Surface Patch Code [1] is used to set up the modal distribution. The approach used to vectorize steps 3 through 6 is discussed below.

Impedance Matrix

Using the MM solution of an electric field integral equation (EFIE), the elements of the impedance matrix are given by

$$Z_{mn} = -\iint_{S_n} \vec{J}_n \cdot \vec{E}_m dS_n \qquad (1)$$
$$m, n = 1, 2 \cdots N$$

where $\vec{E}_m$ is the field of the m^{th} test source, $\vec{J}_n$ is the current distribution on the n^{th} mode, N is the total number of modes, and the integration is carried over the surface defining the n^{th} mode. In the MM solution used in the code, the test sources are filamentary dipoles placed along the axial center of the various modes. Thus, a test source consists of two sinusoidal monopole filaments placed end-to-end. The radiated fields of a monopole filament with sinusoidal current distribution are known in closed form. For example, if the monopole is positioned along the z axis, as shown in Figure 2, the field at a point $P(\rho, z)$ is given by

$$\vec{E} = \frac{\eta_0}{4\pi \sinh(\gamma d)} \left[\left\{ \frac{\exp^{-\gamma R_2}}{R_2} - \cosh(\gamma d) \frac{\exp^{-\gamma R_1}}{R_1} \right\} \hat{z} + \frac{1}{\rho} \left\{ \sinh(\gamma d) \exp^{-\gamma R_1} + \cosh(\gamma d) \cos\theta_1 \exp^{-\gamma R_1} - \cos\theta_2 \exp^{-\gamma R_2} \right\} \hat{\rho} \right] \qquad (2)$$

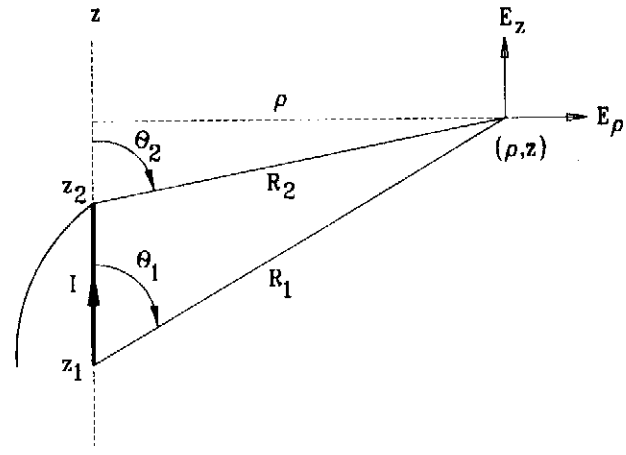

Figure 2: A filament with a sinusoidal current distribution positioned along the z axis.

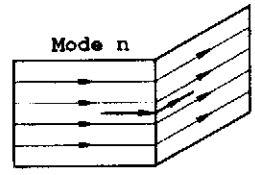

Figure 3: Mode n is defined by filamentary V-dipoles.

where η_0 is the free space intrinsic impedance, d is the length of the monopole, and γ is the complex propagation constant. For this work, γ is purely imaginary. The distances (ρ, R_1, R_2), angles (θ_1, θ_2) and unit vectors $(\hat{\rho}, \hat{z})$ are defined in Figure 2. Note that (2) does not include the field contribution from the point charge at the end of the monopole. Since two of these monopoles are placed end-to-end to form a test source, the contributions from the two point charges are cancelled. The field, $\vec{E}_m$, of the m^{th} test source is found by adding the fields of the individual monopoles. Next, numerical integration can be carried out to solve (1).

Note that when the observation point P moves closer to the filamentary dipole, the distance R_1 and R_2 will become small and the field as given in (2) will approach infinity. Thus, (2) can not be used to calculate the self impedance terms or the impedance terms for overlapping or touching modes. Let these terms be defined as near-zone terms. To compute the near zone terms, the n^{th} mode in (1) is represented by an array of filamentary V-dipoles (see Figure 3). The impedance term is then found as the weighted sum of the reaction between the test

source and these filamentary V-dipoles. To compute these reactions, the distance between the test source and each filamentary V-dipole is examined. Numerical integration is performed whenever the distance is large enough such that (2) is not singular. Otherwise Richmond's [2] closed form solution is used. The closed form solution involves complicated exponential integrals and requires more CPU time.

In the VMPS code, to facilitate vectorization, a whole row of the impedance matrix is computed at a time rather than as individual elements. This approach increases the size of the DO loop structures, which increases the vectorization efficiency. Note that the m^{th} row of the impedance matrix represents the reaction from the test source m to all modes. This section of the code involves the following steps.

1. Calculate the distance between the test source and the various modes.
2. Separate the near zone terms and select the number of integration points for the other terms (far zone terms).
3. Select the location of the integration points on the various modes and calculate the weights for these points.
4. Find the reaction between the field of the test source and the currents at the selected points.
5. Sum the weighted point reactions to calculate the far zone elements of the impedance matrix.
6. Compute near zone terms.

In the first step, the distances from the m^{th} test source to the various modes are calculated. These distances are used in the second step to identify the near zone terms and to determine the number of integration points for the other terms. If the distance between the test source and a mode is less than a predetermined distance, that element of the impedance matrix is computed using the near zone approach in step 6 and the number of integration points for the mode is set equal to zero. Otherwise, numerical integration is used to compute the element. The number of points used in the integration is selected based on the distance between the source and the test mode. For distances less than or equal to $\lambda/4$, 50 integration points are used. For distances between $\lambda/4$ and $\lambda/2$, 18 integration points are used. For distances between $\lambda/2$ and 2λ, 8 integration points are used, and for distances greater than 2λ, only two integration points (a single point on each quadrilateral defining the mode) are used.

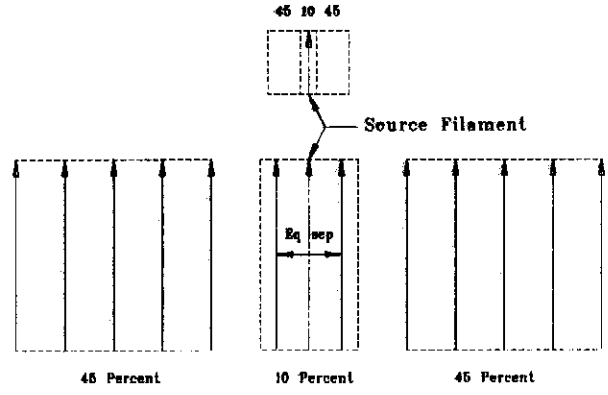

Figure 4: Filamentary monopole distribution when the source monopole lies along the center line of a quadrilateral.

Note that this step provides a map of the number of integration points for various modes.

In step 3, the locations of the integration points on the various modes are found and the integration weights for these points are defined. In step 4, the reaction between the fields of the m^{th} test source and the currents at all the integration points selected in step 3 is calculated. Note that for large structures the total number of integration points will be very large. Thus, the DO loop structure in this step has a large index. In the VMPS code, this DO loop is written to maximize vectorization. In step 5, the point reactions associated with a mode are combined to obtain the elements of the impedance matrix.

The above five steps compute all the elements of the m^{th} row of the impedance matrix except the near zone terms. The near zone terms are computed in step 6. Again, to facilitate vectorization, all the near zone terms are calculated as a group, rather than as individual elements. The procedure used to compute the near zone terms is briefly described below.

To compute a near zone term of the impedance matrix, each quadrilateral of a mode is represented by an array of filamentary monopoles. The distribution of the filamentary monopoles depends on the location of the source monopole (remember that the test source consists of two monopoles) with respect to the quadrilateral. If the source monopole lies along the centerline of the quadrilateral, then the filamentary monopole distribution on that quadrilateral is given in Figure 4. Otherwise, the filamentary monopoles are equally spaced on the quadrilateral. Using the above approach, the filamentary monopole distribution on all the near zone modes for a given source monopole is found and the integration

weights for the various monopoles are calculated. Next, these filamentary monopoles are grouped in two sets. The first set contains all of the filaments for which the reactance can be computed using numerical integration, and the second set contains the filaments for which Richmond's closed form solution is used. The reactances for the two sets are computed in two separate DO loops. Again, the DO loops are written to facilitate vectorization. Next, these reactances are combined to obtain the elements of the impedance matrix. It should be pointed out that Richmond's solution involves extensive computations and is very hard to vectorize.

This completes the calculation of one row of the impedance matrix. The procedure is repeated over all of the test sources to compute the whole impedance matrix.

Excitation Vector

Using the MM solution of EFIE, the elements of the excitation vector are given by

$$V_m = \iiint_{V_m} \vec{E}_i \cdot \vec{J}_m dv_m \qquad (3)$$

where $\vec{E}_i$ is the incident field, $\vec{J}_m$ is the current along the mth test source and the integration is performed over the volume of the test source. Since filamentary test sources are used in the VMPS code, (3) can be written as

$$V_m = \int_{l_m} \vec{E}_i \cdot \vec{J}_m dl_m \qquad (4)$$

where integration is carried out over both monopoles that comprise the test source m.

Let the impressed current source be located at (r_0, θ_o, ϕ_0) and separated by a large (electrically) distance from the scatterer. Then the incident field $\vec{E}_i$ can be considered as locally planar and can be written as

$$\vec{E}_i = \vec{E}_0 e^{jk\vec{r}\cdot\hat{r}_0} \qquad (5)$$

where $\vec{E}_0$ is a constant containing the polarization of the incident field, $\hat{r}_0$ is a unit vector from the coordinate origin to the source and $\vec{r}$ is a radial vector from the origin to the observation point. Substituting (5) into (4), one obtains

$$V_m = \vec{E}_0 \cdot \int_{l_m} \vec{J}_m e^{jk\vec{r}\cdot\hat{r}_0} dl_m. \qquad (6)$$

Or,

$$V_m = -\frac{4\pi j}{\omega\mu} \vec{E}_0 \cdot \vec{E}_m \qquad (7)$$

where

$$\vec{E}_m = \frac{j\omega\mu}{4\pi} \int_{l_m} \vec{J}_m e^{jk\vec{r}\cdot\hat{r}} dl_m \qquad (8)$$

is the far-zone field of the m^{th} source. The closed form expression for the far-zone fields of an electric line source with sinusoidal current distribution is known [3]. Since the test source consists of two monopoles with sinusoidal current distribution, $\vec{E}_m$ and, thus, the elements of the excitation vector are known in a closed form. In the VMPS code, all the elements of the excitation vector are calculated in a single DO loop, and this DO loop is written to facilitate vectorization.

Current Coefficients

Once the impedance matrix and the excitation vector have been determined, it is a straight-forward task to compute the current coefficients. The current coefficients are given by the following set of linear equations.

$$[Z][I] = [V] \qquad (9)$$

where $[Z]$ is the impedance matrix, $[V]$ is the excitation vector, and $[I]$ is a vector containing the current coefficients. Most of the vector computers have very efficient routines to solve a set of simultaneous linear equations. For CRAY supercomputers, NAGLIB library routines are recommended. In the VMPS code, routines F03AHE and F03AKE are used. Routine F03AHE is used for LU decomposition, and routine F03AKE is used in the second step (back substitution).

Scattering Field Calculations

To calculate the scattered fields, the current flowing on the surface of various modes is represented by 5 filamentary V-dipoles with piecewise sinusoidal current distribution. The scattered field for each mode is then given by the weighted sum of the 5 V-dipoles representing the mode. The fields radiated by a given filamentary V-dipole can be calculated by summing the fields radiated by the two monopoles forming the dipole. As pointed out before, the closed form expression for the far-zone fields of an electric line source with sinusoidal current distribution is known. Thus, to calculate the scattered fields, one needs to compute the fields radiated by $10N$ monopoles, where N is the total number of modes. Next, all these fields can be summed to compute the total scattered fields. In the VMPS code, a single DO loop is used to calculate the fields radiated from

Table 1: CPU time (in seconds) used by the VMPS code to analyze 1 meter square plate on various computers.

Freq. (MHz)	No. of modes	VAX 8550	IBM 486 PC	CRAY Y-MP scalar mode	CRAY Y-MP vector mode
300	40	113.75	109.14	10.36	1.48
400	84	301.31	282.10	26.55	3.81
500	144	596.88	569.75	52.72	7.32
600	180	800.02	757.41	69.26	9.03
700	264	1400.18	1344.58	117.89	15.62
800	364	2275.81	2171.21	184.78	23.64
900	420	2809.23	2664.00	223.83	28.33
1000	544	4308.07	4050.88	327.86	42.10

Table 2: CPU time used to calculate the impedance matrix on a CRAY Y-MP computer.

Freq. (MHz)	CPU (seconds) Scalar Mode	CPU (seconds) Vector Mode	Improvement
300	7.30	0.96	7.60
400	19.37	2.73	7.10
500	38.52	5.29	7.28
600	49.97	6.53	7.65
700	84.18	11.48	7.33
800	128.61	17.07	7.53
900	152.46	20.05	7.60
1000	215.53	29.65	7.27

all monopoles. Again, the DO loop is written to facilitate vectorization.

Thus, the four major steps (3–6) in the moment method code have been vectorized. The improvement in the computation speed due to this vectorization is demonstrated in the next section.

III. CODE EFFICIENCY

The VMPS code was executed on various computers to calculate the backscattered fields from a square plate and the CPU time used by the code on these computers is listed in Table 1. The plate dimensions are 1 meter × 1 meter. The backscattered fields are calculated at different frequencies along a 45° conical cut in 1° steps (361 aspect angles). The CPU time listed in the table is the time spent in steps 3 through 6 of the computer program. The number of modes used to define the equivalent currents on the plate at various frequencies is also listed in the table. Note that, as expected, the number of modes increases with the frequency of operation, and so does the CPU time used to calculate the plate back scattered fields. For fair comparisons, only a single processor was used on the CRAY computer. Note that the code uses the maximum CPU time on VAX 8550 computer and the minimum CPU time on CRAY Y-MP computer. This is true even if the vector option of the CRAY computer is not utilized; i.e. the code is compiled with no vectorization option (scalar mode operation). When the vectorization option is utilized on the CRAY, the CPU time shows further improvement. At higher frequencies, the CPU times used in the vector mode of operation is only one eighth of the CPU time used in the scalar mode of operation, which is a significant improvement. In the vector mode of operation, the code uses only 42 CPU seconds to analyze the plate at 1,000 MHz, which is very efficient.

When the code was run on the VAX 8550, IBM 486 and in the CRAY scalar mode of operation, a subroutine based on Crout's method [4] was used to solve for the unknown current coefficients in step 5. In the CRAY vector mode of operation, this subroutine was replaced by NAGLIB subroutines. The reason for using the Crout based subroutine in the scalar mode of operation is that the NAGLIB subroutines are compiled using the vectorization option and we did not have access to the source subroutines.

Next, to study the extent to which various sections of the code have been vectorized, the CPU time used in the various sections of the code in scalar mode and vector mode of operation are compared. Tables 2, 3, 4 and 5, respectively, compare the CPU time used to calculate the impedance matrix, excitation vector, current coefficients and the backscattered fields. Note that, as expected, all sections of the code use less CPU time in the vector mode of operation. The improvement in the CPU time used to calculate the impedance matrix is approximately a factor of 7.5; whereas, the improvement in the CPU time used to calculate the excitation vector is approximately a factor of 8.8. Similarly, the improvement in the CPU time used to calculate the scattered fields is approximately 7.8. In general, for an optimally vectorized computer code the improvement in the CPU time in the vector mode of operation is a factor of 9–10. Thus, these sections of the code have been vectorized effectively. The improvement factor for the CPU time used to calculate the current coefficients

Table 3: CPU time used to calculate the voltage vector on a CRAY Y-MP computer.

Freq. (MHz)	CPU (seconds) Scalar Mode	CPU (seconds) Vector Mode	Improvement
300	0.41	0.053	7.74
400	0.86	0.11	7.82
500	1.47	0.19	7.74
600	1.84	0.21	8.76
700	2.70	0.31	8.71
800	3.72	0.42	8.86
900	4.30	0.49	8.78
1000	5.56	0.63	8.82

Table 4: CPU time used to calculate the current coefficients on a CRAY Y-MP computer.

Freq. (MHz)	CPU (seconds) Scalar Mode	CPU (seconds) Vector Mode	Improvement
300	0.39	0.097	4.02
400	1.63	0.27	6.04
500	4.75	0.68	6.98
600	7.47	0.94	7.95
700	16.43	1.88	8.74
800	32.31	3.52	9.18
900	43.88	4.73	9.28
1000	76.80	7.93	9.68

Table 5: CPU time used to calculate the scattered fields on a CRAY Y-MP computer.

Freq. (MHz)	CPU (seconds) Scalar Mode	CPU (seconds) Vector Mode	Improvement
300	2.19	0.30	7.30
400	4.62	0.63	7.33
500	7.91	1.11	7.13
600	9.91	1.28	7.74
700	14.51	1.87	7.76
800	20.06	2.56	7.84
900	23.11	2.99	7.73
1000	29.87	3.82	7.82

Table 6: CPU time (in seconds) used by the VMPS code and the ESP4 code on a CRAY Y-MP computer.

Freq. (MHz)	Scalar mode ESP4	Scalar mode VMPS	Vector mode ESP4	Vector mode VMPS
300	6.48	10.36	6.24	1.48
400	16.33	26.55	15.36	3.81
500	33.58	52.72	30.39	7.32
600	49.95	69.26	40.46	9.03
700	81.09	117.89	69.19	15.62
800	134.41	184.78	110.41	23.64
900	169.71	223.83	136.87	28.33
1000	263.18	327.86	204.21	42.10

increases with an increase in the frequency of operation and reaches as high as 9.68. This improvement may be misleading in the sense that different routines have been used in the scalar mode and vector mode of operation. Remember that in the vector mode of operation NAGLIB subroutines are used; whereas, a Crout based subroutine is used in the scalar mode of operation. In any event, the CPU time in the vector mode of operation is quite small and it is clear that NAGLIB subroutines are very efficient.

Next, to demonstrate the computation efficiency of the VMPS code, its CPU times are compared with the CPU times of the ESP4 code. Both codes were executed on the CRAY computer in scalar as well as vector mode of operation using the same options. The same modal distribution was used in the two codes. Table 6 shows the CPU time used by the two codes at different frequencies. Note that in the scalar mode of operation, the ESP4 code is a little more efficient than the VMPS code. This is because the VMPS code performs more calculations and is written to calculate the impedance matrix more accurately. The ratio of the CPU times used by the two codes, however, is approaching unity at higher frequencies. In the vector mode of operation, the VMPS code is completely outperforming the ESP4 code. The VMPS code is 4–5 times faster than the ESP4 code. The ratio of the CPU time used by the ESP4 code to the CPU time used by the VMPS code increases with an increase in the frequency of operation. Thus, the VMPS code utilizes the vectorization capability of CRAY supercomputers better than the original ESP4 code.

IV. SUMMARY AND CONCLUSIONS

The operation of the VMPS code was described and the approach used to vectorize the various parts of the code was discussed. It was demonstrated that the code runs very efficiently on CRAY supercomputers. For example, to compute the back scattered fields of a square plate with 544 modes, the code used only 42.1 CPU seconds on a CRAY Y-MP machine.

For the VMPS code, the CPU time may not be the limiting factor in analysing electrically large structures. The available memory space may be the limitation. This problem can be addressed using an out-of-core matrix solver. In the future, the VMPS code will be modified to incorporate out-of-core matrix solver.

Acknowledgement

This work was supported by NASA, Langley Research Center, Hampton, VA. The authors also want to express their thanks to the Ohio Supercomputer Center for the Grant #PAS355 which was a major factor in starting this research.

REFERENCES

[1] E.H. Newman, "A User's Manual for Electromagnetic Surface Patch Code: ESP Version IV," Technical Report 716199-11, The Ohio State University ElectroScience Laboratory, Columbus, OH, August 1988.

[2] J.H. Richmond and N.H. Geary, "Mutual Impedance of Non-Planar Skew Sinusoidal Dipoles," *IEEE Trans. Antennas and Propagat.*, Vol. AP-23, pp. 412–414, May 1975.

[3] J.H. Richmond, "Radiation and Scattering by Thin-Wire Structures in the Complex Frequency Domain," Technical Report 2902-10, The Ohio State University ElectroScience Laboratory, Columbus, OH, July 1973.

[4] G. Forsythe and C.B. Moler, Computer Solution of Linear Algebraic Systems, Prentice Hall, Englewood Cliffs, NJ, 1967.