AIAA 94-0236

Performance of a Finite Volume CEM Code on Multicomputers

J.S. Shang
Flight Dynamics Directorate
Wright Laboratory
Wright-Patterson AFB, OH
D.A. Calahan and **B. Vikstrom**
Electrical Engineering and Computing Sciences Dept.
University of Michigan
Ann Arbor, MI

32nd Aerospace Sciences Meeting & Exhibit
January 10-13, 1994 / Reno, NV

Performance of a Finite Volume CEM code on Multicomputers

J.S. Shang*
Flight Dynamics Directorate
Wright Laboratory
Wright-Patterson AFB, OH 45433
D.A. Calahan† and B. Vikstrom‡
Electrical Engineering and Computing Science Dept.
University of Michigan
Ann Arbor, MI

Abstract

A cell center, upwind biased, spatially third-order and temporally second-order finite volume procedure for solving the Maxwell equations in the time domain has been ported to a Intel Touchstone Delta and a Paragon multicomputer. Using a one-dimensional domain partitioning scheme, the computer code has attained a data processing rate of 7.56 Gigaops on the 512 node Delta and 3.56 Gigaops on the 179 node Paragon XP/S systems. The high parallel efficiency of the present computer program however is sustainable only up to a limited size of addressable memory ($nodes \times 48 \times 96$). The scalable performance range of the present code is significantly extended by operating on the Paragon system at present time.

Nomenclature

B	Magnetic flux density
D	Electric displacement
E	Electric field intensity
H	Magnetic field intensity
i, j, k	Index of discretized volume
J	Electric current density
n	Outward normal of a surface
r, θ, ϕ	Spherical coordinates
t	Time
U	Dependent variable
V	Elementary cell volume
ξ, η, ζ	General curvilinear coordinates
ϵ	Electric permittivity
μ	Magnetic permeability
λ	Wave length

Superscript

+,-	Denotes the flux vector associated with positive and negative eigenvalue

*Senior Scientist. Fellow AIAA

†Professor

‡Research Assistant

1 Introduction

The numerical efficiency and accuracy of computational electromagnetics (CEM) in analyzing refraction and diffraction phenomena are still the pacing items for aircraft signature technology. In this area of endeavor, the numerical accuracy requirement is rather stringent and this need is not separable from numerical efficiency. A desirable and commonly accepted predictive dynamic range is as high as 60 db over broad viewing ranges [1]. Although no specific values of absolute accuracy for target identification and signature processing have been defined for air vehicle applications, the compatible tolerance to the dynamic range requirement is anticipated to be very demanding. Moreover, in high frequency radar operation beyond X band the wavelength is reduced to the centimeter scale [2,3]. For wave scattering simulation, the numerical resolution of wave motions at a given frequency is dictated by the minimal wavelength of the media [1,4]. The concern in numerical analysis for description of standing waves, which requires a mesh spacing to be less than half of the wavelength, is easily satisfied by comparison with the need of scattering simulations. From previous investigations [1,4,5,6], each wavelength should be supported at least by ten elementary cells for finite volume or ten nodes for finite difference schemes to attain suitable numerical resolution. For dielectric media with large refraction indices, the required number of cells will be even greater.

The massive volume of data processing for air vehicle signature prediction can be illustrated by the following simple estimate. In the case of a RCS simulation for a ten meter span aircraft being illuminating by a gigahertz (1×10^9 Hz) radar wave, the number of cells required will exceed 10,000 in each ordinate. For three- dimensional simulation, the computational domain will contain around 1×10^{12} cells ($10^4 \times 10^4 \times 10^4$). The data base of this computation will be proportional to this 3-D array. Since the governing equations consist of six dependent variables and three spatial independent variables with nine direction cosines in a general curvilinear frame, a minimal 19 addressable memory units must be allocated for each individual cell. In practice, a two- step time integration algorithm is frequently used [4,7]. In order to achieve higher vector performance, additional vector arrays

will have to be included in the computer program. The typical CEM finite volume code will have addressable memory units as high as fifty per cell. Therefore for this RCS numerical simulation, the number of dependent variables is around 50 trillion. The number of arithmetic operations required to process data at each time level will be in the hundreds of trillions just to advance the temporal variation by one step. If multiple viewing angles are desired, the amount of computations is enormous.

At the present, high numerical efficiency of CEM simulation procedures can be attained either by algorithmic improvements to solve the Maxwell equations or by using scalable high performance computing systems. Several algorithms for solving the Maxwell equations in the time domain, a hyperbolic partial differential equations system, have been transitioned from the computational fluid dynamics (CFD) to the CEM community [4,5,6,7,8]. These numerical procedures are characteristic-based methods which are derived from eigenvalue and eigenvector analyses. Since the characteristic formulation addresses the fundamental issues of numerical stability and well- posedness of initial/boundary value problems, drastic improvements in numerical efficiency and accuracy have been reported in CEM simulations [1,4,6]. The theoretical approach to the eigenvalue and eigenvector analyses also has limitation, in that the coefficient matrix of the governing equation can only be diagonalized one dimension at a time. However the basically one-dimensional in time and space diagonalization approach does not impose a fundamental limit in their application to CEM [5,6,8]. The simple fact lies in a salient feature of the electromagnetic wave propagation, in that the direction of wave motion is always known to be the outer product of the electric and magnetic field intensity, $E \times H$ [9]. Therefore, a local mesh system can always be constructed to align with the principal axis of wave motion. In principle, the inherent dilemma of solving an initial value problem as a boundary value problem on a finite memory size computer can be eliminated [5,6,7].

In this regard the characteristic-based formulation satisfies the compatibility condition at the farfield boundary rather than using the approximated absorbing boundary conditions [10,11]. In fact, both finite difference [6] and finite volume [7] have successfully simulated an oscillating electric dipole. These numerical solutions generated by the characteristic-based method show no indication of reflecting waves from numerical boundaries [6,7]. Particularly, the finite volume computation has attained a data processing rate of 580 MFlops on a single Cray C90 processor at an averaged vector length of 84.4 [12]. In spite of that, the rate of data processing is still insufficient for engineering applications.

Another viable means of improving numerical efficiency for CEM is the use of scalable multicomputers. Recent progress in microchip and interconnect network technology has led to a host of high performance multiple address message-passing parallel computers. In theory, these scalable multi-computers or multi-processors are capable of providing essentially unlimited computing resources for data processing. However the performance limitation is intrinsically related to the numerical algorithms and the system architecture. The effective use of the scalable multi-computers still requires a balanced work load and minimal inter-processor communication.

Successful implementations of time dependent, three dimensional Navier Stokes equations [13,14] and three-dimensional Maxwell equations in the time domain [15] on multi-computers have been recorded. Specifically, the mapping of the fractional-step windward finite difference algorithm for solving the Maxwell equation in time domain onto the Intel Touchstone Delta system has attained a data processing rate of 6.6 Gigaops (1×10^{12} arithmetic operations per second) on a mesh system of ($512 \times 96 \times 96$). The parallel efficiency of this implementation is also persistently maintained at a value of 97.3 percent up to 512 numerical nodes. Unfortunately the scalable performance ceased to exist when the data array exceed the aforementioned dimension. Under this operational condition, the message length has reached a size of 36, 864 bytes and the frequency of data transmission is 4090 times per time step [15]. At that time, as the diagnostic tool was extremely limited, it was surmised that the scalable performance breakdown was incurred by the message traffic contention of the communication channels. In spite of this observed shortcoming, the application potential is still worthy of further investment.

In the present investigation, the cell center, upwind biased finite volume scheme design to solve the three-dimensional Maxwell equations in the time domain is mapped onto the Intel Delta and the Paragon distributed memory computers. The finite volume procedure has an added dimension for applications to scaterrers of arbitrary shape than the finite difference scheme that implemented earlier [15]. The present procedure was developed on a general curvilinear frame [7], and can be used to simulate a wide range of complex shape of scatters [12]. The strategy for concurrent computation is the same as developed by the earlier efforts [13,14,15].

The basic approach is based on a one- dimensional domain partition approach outlined in our earlier effort. The present effort is focused to seek the limitation of scalable performance of the finite volume procedure and to enlarge the performance envelope, if at all possible.

2 Numerical Algorithm

The time-dependent Maxwell equations for electromagnetic field can be given as [9]

$$\frac{\partial(\mu H)}{\partial t} + \nabla \times E = 0 \qquad (1)$$

$$\frac{\partial(\epsilon E)}{\partial t} + \nabla \times H = -J \qquad (2)$$

$$\nabla \cdot B = 0; \quad B = \mu H \qquad (3)$$

$$\nabla \cdot D = 0; \quad D = \epsilon E \tag{4}$$

where ϵ and μ are the electric permittivity and the magnetic permeablity which relate the electric displacement to the electric field intensity and magnetic flux density to the magnetic field intensity respectively.

The governing equations, (1) and (2), cast in flux vector form on a general curvilinear and body conformal coordinates system acquire the following form;

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} + \frac{\partial \hat{H}}{\partial \zeta} = -\hat{J} \tag{5}$$

where $\hat{U}$ is the transformed dependent variable. $\hat{F}, \hat{G}$ and $\hat{H}$ are the contravariant components of the flux vectors on the Cartesian coordinate system, the basic frame of reference of the present investigation.

$$\begin{aligned} &\hat{U} = UV; \hat{J} = JV; \hat{F} = \left(\xi_x F + \xi_y G + \xi_z H\right) V \\ &\hat{G} = \left(\eta_x F + \eta_y G + \eta_z H\right) V; \hat{H} = \left(\zeta_x F + \zeta_y G + \zeta_z H\right) V \end{aligned} \tag{6}$$

where

$$U = \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \qquad F = \begin{Bmatrix} 0 \\ -D_z/\epsilon \\ D_y/\epsilon \\ 0 \\ B_z/\mu \\ -B_y/\mu \end{Bmatrix} \tag{7}$$

$$G = \begin{Bmatrix} D_z/\epsilon \\ 0 \\ -D_x/\epsilon \\ -B_z/\mu \\ 0 \\ B_x/\mu \end{Bmatrix} \qquad H = \begin{Bmatrix} -D_y/\epsilon \\ D_x/\epsilon \\ 0 \\ +B_y/\mu \\ -B_x/\mu \\ 0 \end{Bmatrix} \qquad J = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ J_x \\ J_y \\ J_z \end{Bmatrix} \tag{8}$$

V denotes the discretized cell volume of the transformed coordinates and is identical to the inverse Jacobian of the coordinate transformation [7,12].

The above system of equations is solved by discretizing the physical space into contiguous elementary cells and by balancing all flux vectors on the control faces of the cell. Therefore in discretized form, the integration procedure degenerates into computations of the sum of all fluxes aligned with the surface area vectors. A cell centered, upwind biased finite volume scheme based on the MUSCL approach is adopted for the present analysis [16,17].

$$\frac{\Delta \hat{U}}{\Delta t} + \frac{\Delta \hat{F}}{\Delta \xi} + \frac{\Delta \hat{G}}{\Delta \eta} + \frac{\Delta \hat{H}}{\Delta \zeta} - \hat{J} = \frac{\Delta \hat{U}}{\Delta t} + R = 0 \tag{9}$$

The characteristic formulation in finite volume approximation is achieved by split the flux vector according to the signs of eigenvalues of the coefficient matrix in each spatial dimension [18,19]. In essence, the numerical procedure is designed to honor the zone of influence of the initial value problem. In other words, the numerical method is constructed to follow the direction of signal propagation by choosing the appropriate piecewise continuous data to form the directional difference for the simulated wave motion. The flux at any cell interface is represented by a superposition of two components; F^+, F^-, G^+, G^-, H^+, and H^- according to the direction of wave motion [16,17,18,19]. Follow earlier efforts [7,12], at the cell surface $i + 1/2$, is obtained as;

$$\begin{aligned} F_{i+\frac{1}{2}} &= F^+(U^L_{i+\frac{1}{2}}) + F^-(U^R_{i+\frac{1}{2}}) \\ G_{j+\frac{1}{2}} &= G^+(U^L_{j+\frac{1}{2}}) + G^-(U^R_{j+\frac{1}{2}}) \\ H_{k+\frac{1}{2}} &= H^+(U^L_{k+\frac{1}{2}}) + H^-(U^R_{k+\frac{1}{2}}) \end{aligned} \tag{10}$$

In practice, the F^+, F^- etc are reconstructed by the κ scheme [20] which is capable of generating approximations from first-order to third order accurate. The consistent numerical accuracy in time is obtained by a Runge-Kutta family of single-step multi-stage procedure permitting a wide range of order of temporal accuracy in one programming procedure [21]. In the present effort, only a two-stage procedure which is second order accurate in time is used.

The balancing of flux vectors on the transformed cell interface is greatly facilitated by introducing locally orthogonal coordinates. One of the ordinates is aligned with the outward normal of the cell surface. Since the rest coordinates are tangential components to the surface vector, they are easily constructed by executing consecutive outer products between the outward normal and one of the cell edges in consideration. Then the flux vectors on the cell interface are split into components parallel to the orthogonal coordinates and balanced locally. The necessary eigenvalue and eigenvector analyses required for flux splitting as well as the inverse mapping into the curvilinear coordinates are well known [5,6]. The detailed formulation can be found in reference [7], thus will not be repeated in here.

Numerical simulation of a three-dimensional, time dependint radiating electromagnetic wave is adopted for the present study. The initial and boundary conditions for an oscillating electric dipole simulation are straight forward. The pulsating source is described by a sinusoidal wave with an amplitude of unity and the wavelength equal to the half radius of the entire computational domain. The only physically meaningful initial condition required is the total inducted field at the dipole, for which the analytic solution is known to contain singular behavior [9]. In order to alleviate the extremely small time step required to resolve the finest cell volume near the coordinate origin, the initial values were imposed for four cells immediately adjacent to the dipole.

The farfield boundary condition on the truncated domain usually is the most difficult numerical boundary condition encountered in CEM [1,4,10,11], because the reflected wave from the artificial boundary will produce serious distortion of

wave patterns. For the present characteristic-based formulation, this boundary condition is described simply by the zero incoming flux component.

$$F^{-}(\xi, \eta_b, \zeta) = 0.0 \qquad (11)$$

where η_b denotes the radius of the farfield boundary. On the spherical coordinate, the above farfield boundary condition for a dipole is also the exact compatibility condition. Within the truncation error of numerical analysis, the spurious reflected wave commonly observed in CEM simulation should be completely eliminated [5,6]. The other numerical boundary conditions in circumferential and the azimuthal directions merely reflect a continuous field. Arrays of two overlapping cells are sufficient to enforce the functional continuity requirement.

3 Description of the Multicomputers

Both the Touchstone Delta and Paragon XP/S are mesh interconnected parallel processors [22,23,24]. The mesh topology is based on byte-wide communication channels rather than the bit-wide hypercube topology. These two multicomputers consist of a large number of heterogeneous nodes to perform computation, frame buffer, network link, and disk string functions. Individual computing nodes, i860 or i860XP, are connected to a Mesh Routing Chip (MRC) through a Interface Module.

On message passing or distributed memory multicomputers, the performance of concurrent computing is closely tied to node-to-node and node-to-memory communication bandwidth and latency. These system peculiarities always exist and have profound influence to parallel efficiency, particularly, when contention of communication path becomes significant in concurrent operation. The occurrence of contention is dictated by frequency of data movement, length of message, and the data transfer rate. The communication time of a message is proportional to the length of data string, but there is also some discontinuity in rate of data transmission at some specific message size [22]. This behavior is introduced by the communication protocol and operating system characteristics which can induce performance degradation. In applications, all message traffic may compete for passage in communication channels, and can lead to poor parallel computing performance of any distributed memory computers.

The Touchstone Delta system used consists of 576 heterogeneous nodes. Among these, 528 are i860 computational nodes, but only 512 nodes are allocated by a two-dimensional mesh up to a (16×32) configuration [23]. The i860 has a peak rate of 80 single- precision and 60 double-precision MFlops, a 16 Megabytes (Mb) nodal memory, and a 8 Kilobytes (Kb) data cache. The I/O function is carried out by a total of 32 80386 I/O nodes, and each has a bandwidth of 16 MHz. The complete system also includes 2 gateway nodes and 6 service nodes[23]. For a given numerical algorithm, the data processing rate and scalable performance are constrained by the communication latency and contention. The node-to-node communication latency of the Delta system is dominated by chores of argument checking, context switch on interrupt, and other protocol requirements [22]. Latency incurs extra time elapsed for multi-hop data movement between nodes and creates a situation for communication contention.

The message passing in the Delta system is row biased, such that the node near the boundaries of two-dimensional mesh will have to be routed through multi-hop to reach a node in the different row. A programming arrangement takes advantage of immediately adjacent node priority in message passing has shown a 7.3 percent gained in parallel computing efficiency of a fractional-step scheme [14,15]. However, the choke point of performance on the Delta system is still the node-to-node communication. Although the MRC has a rated bandwidth of 65 Mb/s, a measured peak is about 13.1 MB/s for a message size of 100,000 bytes [22].

The Paragon XP/S system, contains 240 i860XP computational nodes, 4 service nodes, and 16 multi-purpose I/O nodes. The i860XP has two CPUs and twice the size of nodal memory than the i860, (32Mb), and a rated peak performance of 75 double-precision and 100 single-precision MFlops. In contrast to the i860 used by the Delta system, the rated performance is 25 percent higher. The MRC of the paragon system has 10 unidirectional ports, and a bandwidth of 200 Mb/s per port. The data transfer rate is 3.077 times greater than that of the Delta system. For a numerical algorithm dominated by frequent message passing and high risk for communication contention, the improved data processing rate will have the potential to exceed 25 percent. The last major component of the Paragon system used is the 15 Redundant Array of Inexpensive Disk Drives (RAIDs), each having a capacity of 4.8 Gigabytes (Gb) [24].

There is no fundamental architectural differences between the two Intel systems, but hardware enhancements have been made to the computational node and MRC of the Paragon XP/S system. It may be interesting to point out that the enhancement to the data shipping rate is greater than the data processing rate of the Paragon system. As a consequence, a higher performance than 25 percent may be anticipated. On the other hand, the operating systems are significantly different from each other, but the developed software is still transportable from one system to other. Only minor changes of code that was developed on the Delta system was needed to operate on the Paragon system. The system information of the two- dimensional mesh partition to take advantage of the immediately adjacent nodes priority on the Delta system is not currently available for the Paragon.

The Paragon operating system OSF/1 is Unix like, and written by the Open Systems Foundation (OSF) and Intel [24]. The OSF/1 operating system provides the virtual memory capability that in turn also consumes considerable memory, and at present the performance is not necessarily optimal,

is not a production system. In this regard, the timing information obtained from the Paragon system is still subject to continuous improvements.

4 Domain Decomposition Strategy

The data structure partition is the first critical step toward high parallel efficiency on distributed memory systems. On the mesh-connected parallel processors, the data flow and data management lead to four different partitions of data [13,15]. The most elementary approach of domain partition is the one dimensional parallelization in which planar data coresponding to the first index of the 3-D array, IL, is assigned to individual processors. As a consequence, each numerical node is assigned to process data only containing a two-dimensional cross plane ($JL \times KL$). Although the initial and boundary values within the two- dimensional array are self-contained, communication is required between contiguous planes identified by the computational node. Along the ordinate associated with each node, data communication between nodes is needed to construct the balanced flux vectors on cell interfaces, and to satisfy the boundary conditions at the edges of the computational domain. Since the data overlapping regions are significant in comparison with the total data base, the nodal communication is frequent during a numerical sweep for time advancements. The other data partition schemes including the page structure, the pencil structure, and the block structure [15] also have some degrees of control for data management and to meet the basic criterion for parallel computing, balance the work load among all nodes, and keep the data transfer between nodes at the absolute minimum. A graphic depiction of these data partition schemes is given in Figure 1.

Only the one dimensional (1D) domain partition accomplished for now, because this approach is the most straightforward implementation of numerical algorithms on structured grids to parallel computers [6,12,15]. In the present approach, the data are organized into equal-in-length and contiguous cross planes which can also be viewed as a form of task partition. The major advantages of the 1D partition is that the approach can automatically achieve perfect load balancing among nodes. The advantage is however acquired at the expense of a greater amount of inter-nodal communications than that of the page, pencil, and block structures [13,14]. On the Delta system, even full use of message-passing priority of the immediately adjacent nodes still can not overcome the basic shortcoming. The communication contention eventually will set the performance ceiling of the scalable operations for both a finite difference [15] and the present finite volume algorithms.

The timing results of a related study of a fractional-step scheme [6,15] revealed that, the pencil partition has demonstrated a potential 37 percent higher data processing rate than that of the 1D partition, but the pencil partition also encountered a similar ceiling of scalable performance up to a size of problem of ($512 \times 96 \times 96$). Nevertheless, the 1D scheme has achieved a scalable data processing rate of 6.55 Gigaops and a sustained parallel efficiency of 92.7 percent up to 512 nodes on the Delta system [15]. The scalable performance of the finite difference code developed for Cartesian coordinates breaks down when the mesh system exceeds a dimension of ($512 \times 96 \times 96$). The limiting behavior of performance is also observed for the pencil partition scheme. This performance limitation of a distributed memory multicomputer is closely tied to memory bandwidth and memory latency, and is also the issue to be examined here.

In mapping the upwind biased finite volume code onto the Intel systems using the 1D partition, each node processes a grid plane containing $JL \times KL$ cells. For this partition scheme, numerous message passings are performed per grid point for each temporal step advancement, and the length of message is the product of JL, KL and numbers of dependent variables in the formulation. For the baseline case of ($24 \times 24 \times 48$), each individual message has a length of 221,184 bytes. If the cell density of the finite volume computation is doubled in both directions of the cross sectional plane, the length of message will be quadrupled (884,736 bytes). The frequency of message passing is still linearly proportional to the number of nodes in use.

Specifically for each time step advancement, a minimum of 3 global synchronization operations were implemented for the directional numerical sweeps in each of the three ordinates. Within each global synchronization, a total of ($2 \times IL$) synchronous message receptions, an equal number of asynchronous message transmissions, as well as the blocking of an asynchronous call are required. For the present analysis, the number of nodes is identical to the cell index number IL.

5 Numerical Results

The basic computer program was developed on an Iris workstation, 4D/440VGX. For the baseline mesh system, ($24 \times 24 \times 48$), 200 time steps required 1,507.95 cpu seconds for a wave to travel about a quarter length of the computational domain. Therefore, for the electromagnetic wave to traverse the entire computational domain will require 800 time step, the unit of time defined as one characteristic time, t_{ch}. In spite of a short average vector length for all operations of only 15.78, this program still can delivered a 173.9 MFlops on a single processor of a Cray Y-MP8/8128 and 229.49 MFlops on a Cray C916/16256 respectively. On a single node of the Delta system without the message passing requirement, this code processed the same data in 19.85 MFlops. The single-node data processing rate on the Paragon system shows a greater sensitivity to vector optimization than of the Delta system, the data rate ranges from as low as 23.82 and up to 26.77 MFlops. The performance variation is generated by using different compiler options. The comparative

data however lacks precision, because there are modifications made in codes for parallel computing, particularly the implementation of system calls. A better approach is given by first analysing the performance difference on both computer systems for the fractional-step algorithm of which the number of floating point operations per time advancement is known [15].

The parallel computing efficiency of the finite difference code is thoroughly verified, therefore is adopted for a base of comparison [15]. Calculations on the ($node \times 96 \times 96$ grid system, the largest scalable dimension, were carried on both the Delta and the Paragon systems. In Figure 2, a set of timing results from the Delta System and two sets timing results from the Paragon system are given. The two sets of the Paragon data reflect the greater sensitivity of data execution rate to options of compiler. The slower rate (the longer period of computation) produced by the Paragon system was generated with neither vector operation nor compiling optional enhancements, yielding a data rate only 46.13 percent to that of the Delta system. The results from Paragon with vector operation enhancement indicate a significant improvement and show a data rate of 14.2 MFlops per node. The recorded data processing rate is nearly identical to that of the Delta sytem, operating under the identical condition. From this comparison, a similarly scalable parallel performance of the fractional step algorithm is established for the Paragon system. The degradation of concurrent computations is 3.1 and 3.8 percent over the full complement of nodes available (512 nodes on Delta, 179 nodes on Paragon) for both systems respectively.

The issue of non-scalability of I/O on distributed memory computers is well known [13,14,15] and is revealed by the presentation of Figure 3. The period of time required to output computed results on tapes and disks are given in seconds, only the fastest and the slowest timing data are included. For both systems, the I/O performance degraded rapidly as the number of nodes in use was increased, and the relationship is almost linearly proportional to the number of nodes used. The operating system of the Paragon multicomputer is presently under constant upgrading. Thus the level of performance is not consistent. The present I/O data scattering band is much greater than that of the Delta system. In all, the timing results demonstrated that the I/O performances are not scalable for both systems. This issue should remain as a pacing item for research in concurrent computing.

The validation of numerical simulations was performed on a ($24 \times 24 \times 48$) and a ($48 \times 48 \times 96$) mesh systems. All calculations were first carried out on the Cray C916/16256 computer. On the higher cell density grid, the average vector length for all operations was 34.69, and achieved a data processing rate of 396.36 MFlops. In order to alleviate the aggregated I/O time required from both the distributed memory systems and the significant slow down of the non-scalable performance, only selective planes of concurrent results were compared with the validation data generated by the Cray computer. However, the numerical results generated by the Delta system were verified only for the coarse mesh. The finer mesh numerical simulations were not obtainable on the Delta system, because the CPU time required by non-scalable computations became prohibitive. For the cases considered, all numerical solutions generated by both sytems are identical within the order of the truncation error.

In Figure 4, time evolution of the electric field intensity of an oscillating dipole is sampled at a fixed point in space. This point is defined by the spherical coordinates (r = 0.4, θ = 79.5, ϕ = 184) and the sampling duration is $5.5t_{ch}$. Three Cartesian components of the electric field are presented along with the analytic solution. At this physical location and instance, the Y component of the electric field dominated over the two others and the X component assumed the smallest wave amplitude. It is observed that, except in the initial transition period, the numerical simulation lags the pulsing radiating wave. The agreement with the classic theoretical result [9,12] is excellent in both wave amplitude and phase angle.

The comparison with theoretical results for the magnetic field intensity at the same physical location and point in time is depicted in Figure 5. Again, the Y component of the magnetic field dominated over the X component, and the Z component vanished identically. The overall agreement with the theoretical result is excellent. After the initial transition period, numerical results did not reveal any detectable wave modulations due to dispersive or dissipative numerical errors. Numerical results produced by the Paragon system and the Cray computers are essentially identical at the selected data plane.

Figure 6 presents the instantaneous distributions of the electric field intensity of an oscillating dipole along a radius. The robustness of the upwind biased scheme in resolving the steep gradient region near the dipole center is clearly illustrated. All three computed electric field components have a singular behavior proportional to an inverse cubic power of radius at the coordinate origin, r = 0 [9]. The present numerical simulation is able to adequately resolve the locally rapid changes. The most important feature is however the good agreement between theory and the numerical results at the farfield, r = 1, where only the characteristic-based numerical algorithm can satisfy the perfect no-reflection numerical boundary through the compatibility condition [5,6,7,19].

Figure 7 gives the comparison of instantaneous magnetic field distributions with the classic results. As it has been mentioned previously, only two Cartesian components of the magnetic field intensities exist for the oscillating electric dipole. The singular behavior of the magnetic field is less severe than that of the electric field intensity, but is proportional to the inverse square power of the radius at the coordinate origin [7,8,9]

The direct comparison with theory indicates an excellent agreement.

Now, the verification of parallel version of the computer program for solving the Maxwell equations in time domain is considered complete for the present purpose.

6 Timing Results

For the present investigation, the period of times needed for program initiation and I/O function consists of a relatively a small fraction of the total computing resource, and the I/O problem is well known. The timing information is thus focused on only the time required for computations and recorded for the bare minimum sampling period. The operating system of the Paragon multicomputers is still in the shakedown stage. Although attention has been paid to ensure repeatable performances for the data collection procedure, the timing results must be regarded as preliminary information. A substantial improvement of performance is anticipated, after operating system enhancements are installed.

In Figure 8, the timing data of the baseline case calculations ($nodes \times 24 \times 48$) by the Delta and the Paragon systems is presented. At each 2-D mesh configuration, the maximum and the minimum nodal CPU times are included to reflect the range of total performance (computation and node-to-node communication) among the most and the least effcient node. The results generated by the Delta system cover the range of 512 nodes, and at present 180 nodes of the Paragon system are available to users. For the upwind biased finite volume algorithm, the Paragon revealed a higher data processing rate than the Delta system. Over the overlapping number of nodes, the Paragon outperforms the Delta system by 34.84 percent on average. The improved performance is greater than just the ratio of the peak CPU rates between i860XP and the i860, 1.25. The additional efficiency gain must be derived from the wider bandwidth of the MRC, the layers of software implemented, and the operating system.

Based on profile information gethered from the Cray computers, the nodal data processing rate is estimated around 19.92 MFlops. Another estimate is also reached a result of 26.77 MFlops, based on the statistic data of arithmetic operations from the Cray computers; In the sampling period, a total of $1,121,277,163$ float point operations has been executed. The estimate from this statistic information is considered higher than actual, because the latancy and the execution rate of the synchronous and the asynchronous system calls was not able to be taken into consideration.

Timing results from both systems indicated a jump in CPU time when more than 16 nodes were employed. This pecularity is introduced by the imposition of initial and boundary values at the boundaries of the computational domain and is formulation specific. The first four cells near the coordinate origin and the last two cells at the farfield are excluded from the computation. Therefore the realistic scalable performace should adopted the timing data at 24 nodes as the base for normalization. Again, the scattering of the data band of different nodal performance is greater on the Paragon than the Delta system. The existence of communication hot spot may also occur [22], but no confirmation to this speculation can be offered at the present time.

Finally the scalability of the Delta performance is beginning to show signs of failing for the number of nodes exceeding 256. Admittedly, a greater number of nodes were employed for the timing information on Delta, but even in the overlapping range of nodes in use, the degradation in performance becomes noticable. As it was found later, the scalable performance of the Delta system on the refined cell density grid ($nodes \times 48 \times 96$) was not achievable beyond 8 nodes.

The performance of the Paragon system for two grid systems ($nodes \times 24 \times 48$) and ($nodes \times 48 \times 96$) of the upwind finite volume scheme is given by Figure 9. The greater range of scalability of the upwind biased finite volume code operating on the Paragon system definitely contributed to the wider bandwidth of the MRC of the Paragon than that of the Delta sytems. Although the number of floating point operations per node required to advance a time step is quadrupled, the execution time is increased by an average ratio of 2.15. Two factors may have contributed to the results. First the vector length is doubled in all do loop structures of the codes which enhances greatly the vector operations. Second, the number of messages passed remains the same for both mesh systems and the change is restricted to the message length. The communication latency in start-up time is also unaltered. The only additional communication contentions can only be incurred by the increased message length. Unfortunately, substantiation of aforementioned observations has to be acquired from the detailed profile information of the Paragon, and it is not available at the present time.

Figure 10 depicts the scalability of the upwind biased finite volume scheme mapped onto the Paragon system by the 1D domain decomposition strategy. For the mesh system of ($nodes \times 24 \times 48$), the parallel efficiency, normalized by the timing data obtained from 24-node computation, suffered a 15.6 percent degradation at 179-node configuration. Similarly, the results from the enriched mesh attained 84.3 percent parallel efficiency over the full range of available nodes. Unfortunately, the absolute limit of scalable performance of the present implementation is still undetermined at the present time.

7 Conclusions

The upwind biased finite volume scheme has been successfully implemented on both the Intel Touchstone Delta and Paragon XP/S multi- computers by a one-dimensional domain decomposition strategy.

The preliminary timing results indicate that the Paragon system can generate a 34.84 percent higher data processing rate

for the upwind biased finite volume algorithm than that of the Delta system. The data processing rate is estimated to be about 19.92 Megaflops per node.

Most importantly, the wider communication bandwidth of the Mesh Routing Chip (MRC) and the operating system of the Paragon system permit a greater range of scalable performance of the present numerical scheme. The extended capability allows four times greater number of cells to be processed efficiently.

Continuous research for a better data structure partitioning for parallel computing is still urgently needed.

8 Acknowledgement

Invaluable support and assistance from Drs D. Gaitonde, D. Rizzetta, and Mr S. Scherr of the CFD Section, as well as Mr D. Wagoner of the Computer Science Corp. are gratefully acknowledged.

References

1. Taflove, A., Re-inventing Electromagnetics: Supercomputing Solution of Maxwell's Equations Via Direct Time Integration on Space Grids, Computing Systems in Engineering, Vol 3, no 1-4, 1992, pp. 153-168.

2. Crispin, J.W. and Seigel, K.M., Edit., Methods of Radar Cross- Section Analysis, Academic Press, New York, 1968

3. Knott, E.F., Shaeffer, J.F., and Tuley, M.T., Radar Cross Section, 2nd Edition, Artech House, Boston, 1993.

4. Shankar, V., Research to Application - Supercomputing Trends for the 90's and Opportunities for Interdisciplinary Computations, AIAA paper 91-0002, Jan. 1991.

5. Shang, J.S., Characteristic Based Methods for the Time Domain Maxwell Equations, AIAA Paper 91-0606, Jan. 1991.

6. Shang, J.S., A Fractional-Step Method for Solving 3-D, Time- Domain Maxwell Equations, AIAA Paper 93-0461, Jan. 1993.

7. Shang, J.S. and Gaitonde, D., Characteristic-Based Time-Dependent Maxwell Equations Solvers on a General Curvilinear Frame, AIAA Paper 93-3178, July 1993.

8. Shang, J.S., A Characteristic-Based Algorithm for Solving 3-D, Time Domain Maxwell Equations, AIAA Paper 92-0452, Jan. 1992.

9. Harrington, R.R., Time-Harmonic Electromagnetic Fields, McGraw- Hill Book Co., New York, 1961.

10. Enquist, B. and Majda, A., Absorbing Boundary Conditions for the Numerical Simulation of Waves, Math of Comp., Vol 31, July 1977, pp. 629-651.

11. Higdon, R., Absorbing Boundary Conditions for Difference Approximations to the Multidimensional Wave Equation, Math. of Comp., Vol 47, No 175, 1986, pp. 437-459.

12. Shang, J.S. and Gaitonde, D., Scattered Electromagnetic Field of a Reentry Vehicle, AIAA Paper 94-0231, Jan. 1994.

13. Wesley, R., Wu, E., and Calahan, D.A., A Massively-parallel Navier-Stokes Implementation, AIAA Paper 89-1940CP, Jan. 1989.

14. Scherr, S.J., Implementation of an Explicit Navier-Stokes Algorithm on a Distributed Memory Parallel Computer, AIAA Paper 93- 0063, Jan. 1993.

15. Shang, J.S., Hill, K.C., and Calahan, D.A., Performance of a Characteristic-Based , 3-D, Time-Domain Maxwell Equations Solver on a Massively Parallel Computer, AIAA paper 93-3179, July 1993. 16. Van Leer, B., Flux Vector Splitting for the Euler Equations, Tech Rept 82-30, ICASE, NASA Langley Research Center, Hampton, VA. Sept. 1982.

17. Anderson, W.K., Thomas, J.L., and Van Leer, B., A Comparison of Finite Volume Flux Splittings for the Euler Equations, AIAA Paper 85-0122, Jan. 1985.

18. Steger, J.L. and Warming R.F., Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite Difference Methods, J. Comp. Physics, Vol 20, No 2, April 1981, pp. 263-293.

19. Roe, P.L., Characteristic-Based Schemes for the Euler Equations, Ann Rev. Fluid Mech., Vol 18, 1986, pp. 337-365.

20. Thomas, J.L. and Walters R.W., Upwind Relaxation Algorithms for the Navier-Stokes Equations. AIAA J., Vol 25, No 4, April 1987.

21. Carnahan, B., Luther, H.A., and Wilkes, J.O., Applied Numerical Methods, John Wiley & Sons, Inc., 1969.

22. Dunigan T.H., Communication Performance of the Intel Touchstone Delta Mesh, ORNL/TM-11983, Oak Ridge National Lab., TN, Jan. 1992.

23. Touchstone Delta System User's Guide, Intel Corp., Publication No 312125-001, Oct. 1991.

24. Paragon User's Guide, Intel Corp., Publication No 312489-002, Oct 1993.

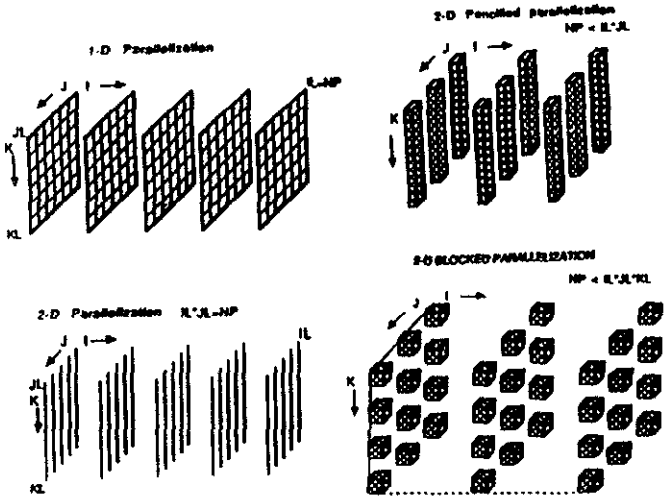

Figure 1: Selected Data Partition Options

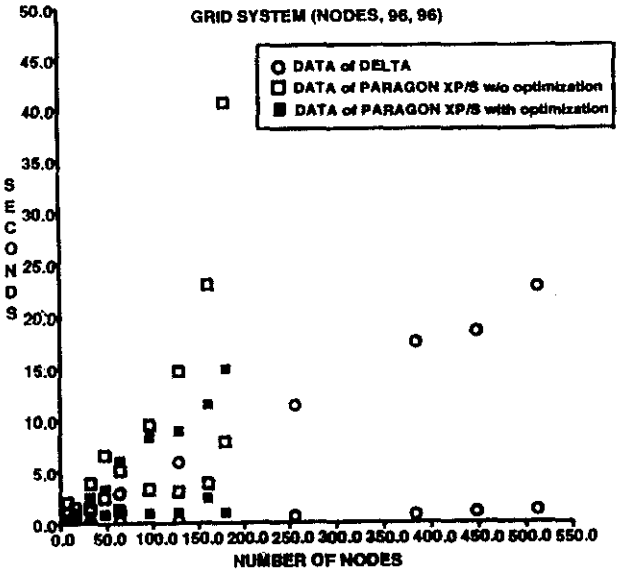

Figure 3: I/O Performance of Fractional-Step Algorithm on Delta and Paragon Systems ($nodes \times 96 \times 96$)

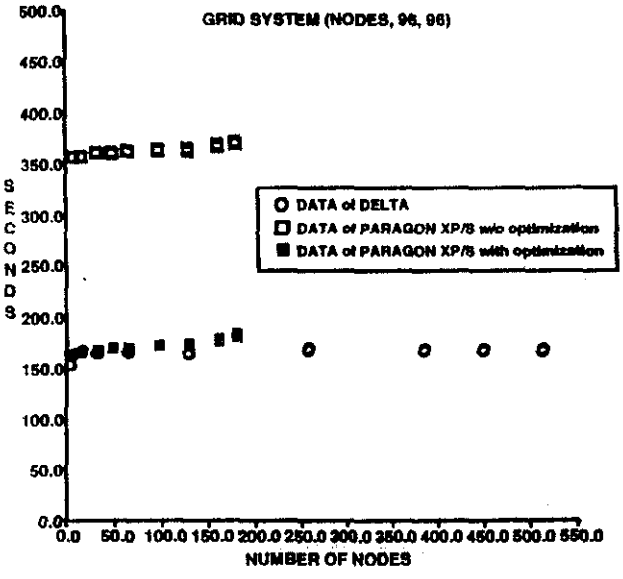

Figure 2: Data Processing Rates of Fractional-Step Algorithm on Delta and Paragon Systems ($nodes \times 96 \times 96$)

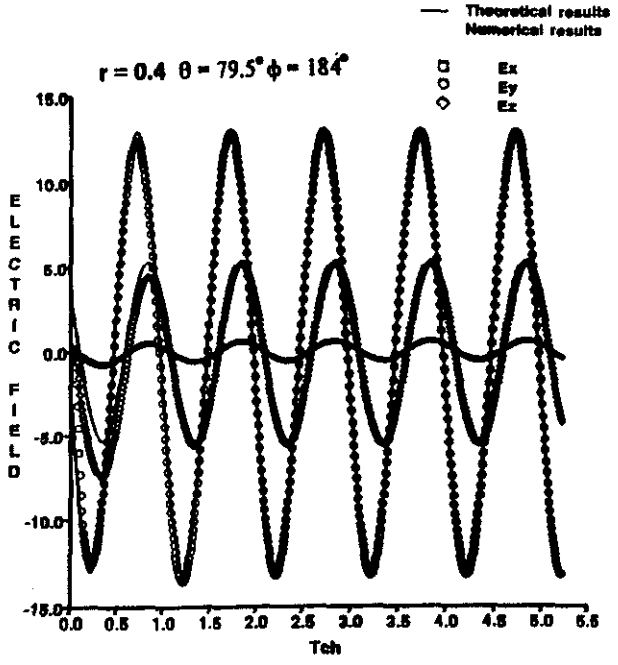

Figure 4: Temporal Variation of Electric Field of an Oscillating Dipole

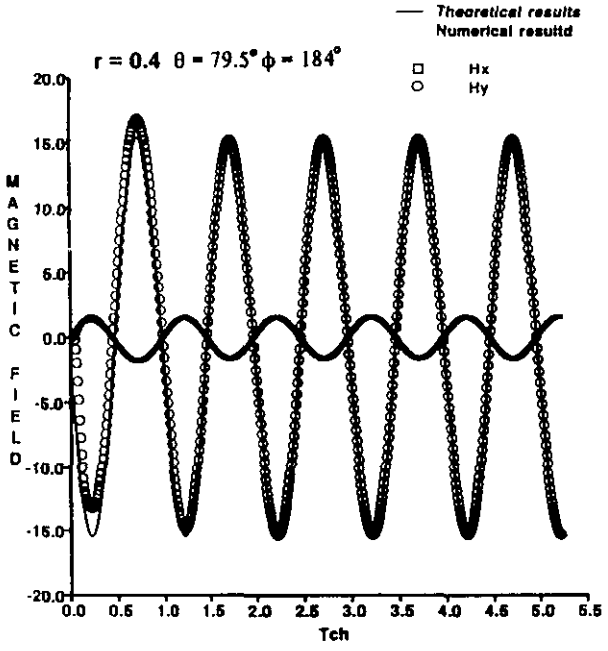

Figure 5: Temporal Variation of Magnetic Field of an Oscillating Dipole

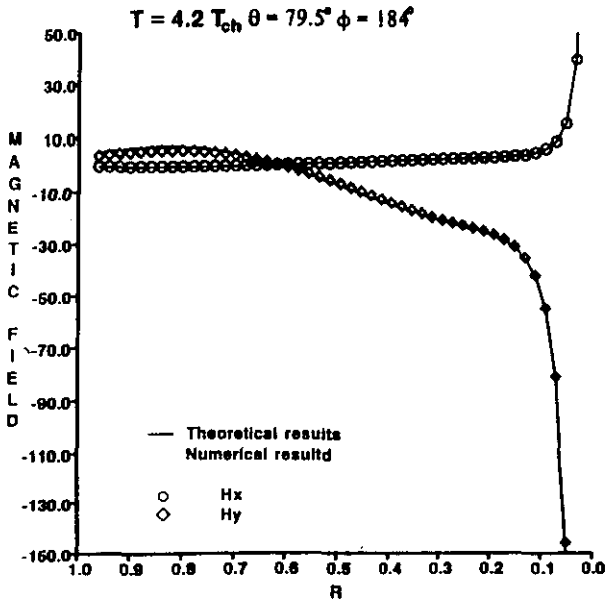

Figure 7: Instantaneous Magnetic Field of an Oscillating Dipole

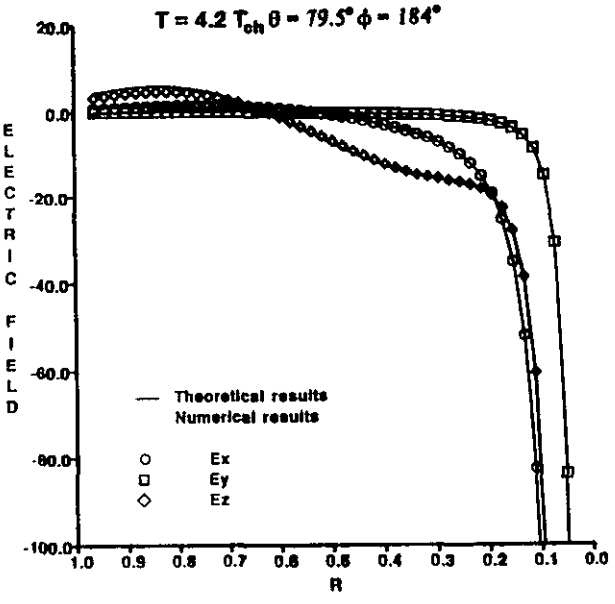

Figure 6: Instantaneous Electric Field of an Oscillating Dipole

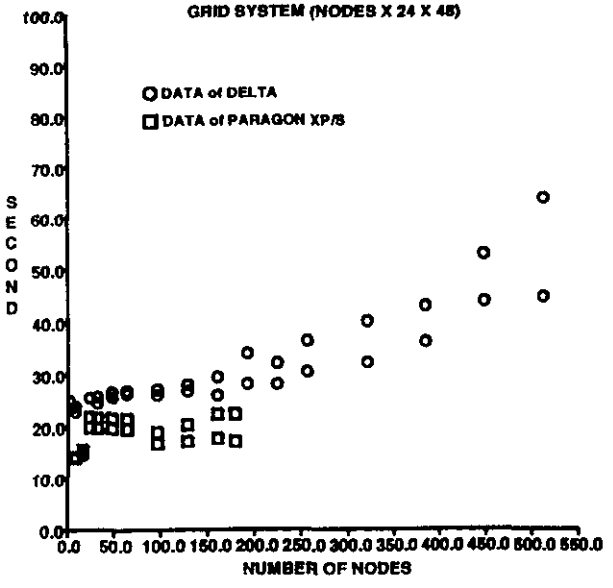

Figure 8: Data Processing Rate of Upwind-Biased Algorithm on Delta and Paragon Systems ($nodes \times 24 \times 48$)

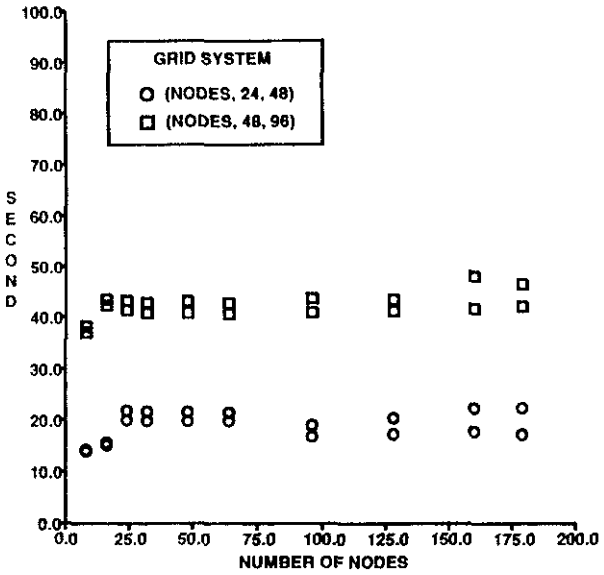

Figure 9: Data Processing Rate of Upwind-Biased Algorithm on Paragon System ($nodes \times 24 \times 48$) & ($node \times 48 \times 96$)

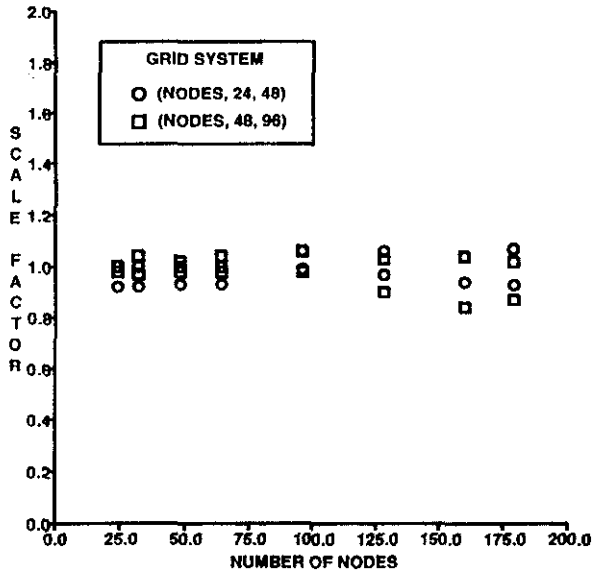

Figure 10: Scalability of Computations