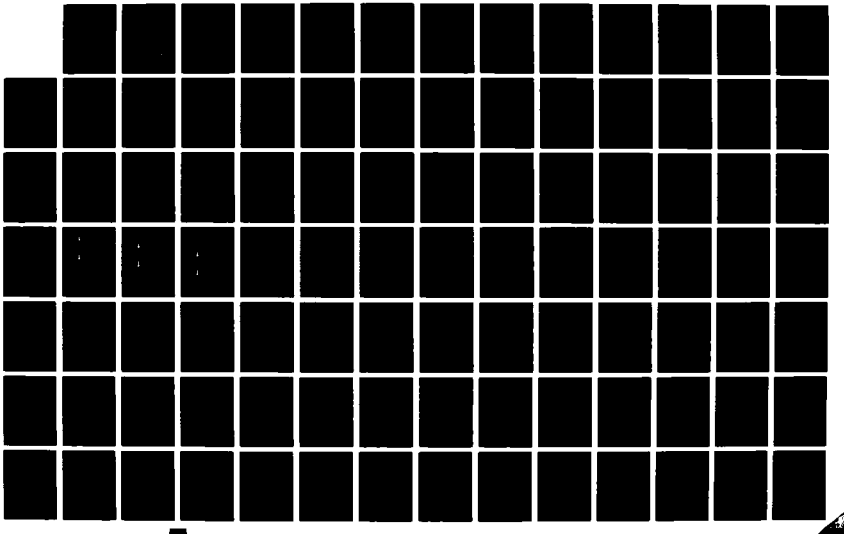
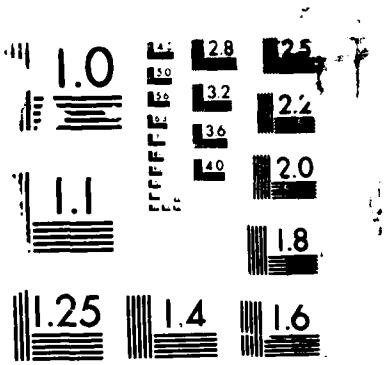
AD-A179 518 ARCHITECTURAL IMPLICATIONS OF A PARALLEL COMPUTATIONAL 1/2
APPROACH TO THE VE..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. J L STRAUSS
UNCLASSIFIED MAR 87 AFIT/GE/ENG/87M-5 F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

ARCHITECTURAL IMPLICATIONS OF A PARALLEL COMPUTATIONAL APPROACH TO THE VECTOR WAVE EQUATION

Thesis

Jack L. Strauss
Captain, USAF

AFIT/GE/ENG/87M-5

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

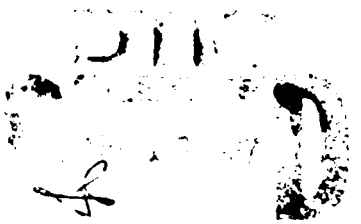
Wright-Patterson Air Force Base, Ohio

ARCHITECTURAL IMPLICATIONS OF A PARALLEL COMPUTATIONAL APPROACH TO THE VECTOR WAVE EQUATION

Thesis

Jack L. Strauss
Captain, USAF

AFIT/GE/ENG/87M-5



AFIT/GE/ENG/87M-5

ARCHITECTURAL IMPLICATIONS OF A PARALLEL COMPUTATIONAL APPROACH TO THE VECTOR WAVE EQUATION

Thesis

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Jack L. Strauss

Captain, USAF

March 1987

Approved for public release; distribution unlimited

Preface

This research is part of the continuing development effort from the Vector Wave Equation (VWE) Research Group. The work reported in this thesis is the algorithmic manipulation of the VWE to form a parallel computational approach which, when presented in a graphical form, provides a specification tool for Very Large Scale Integration (VLSI) implementation.

It is my firm belief that the effort of the VWE Research Group can make a significant contribution to the "State-of-the-Art" in the design and analysis of structures that are impacted by electromagnetic quantities or involve electromagnetic metrics as a design constraint. As such, it has been a pleasure and an honor to be involved in this work.

It is appropriate here, that I acknowledge and give thanks to the people and forces that made this research possible. The AFIT computer environment is the best that I have ever been involved with. My Thesis Advisor Maj Joe DeGroat, encouraged creative approaches to solutions rather than the security of classical methods. These factors combined to make for a research environment that was both exciting and powerful.

I would be terribly remiss if did not give special mention to my wife Debra, whose support, encouragement, and tolerance of the long hours required to do this work made this thesis possible.

Finally, I give thanks to the Holy Father and his son Jesus Christ in whom I believe.

Jack L. Strauss

Table of Contents

List of Figures

Abstract

An algorithm has been specified for the hardware implementation of the numerical solution of the electromagnetic fields of an arbitrary current source. The algorithm, defined as the Vector Wave Equation (VWE), solves for the magnetic (H) and electric (E) fields, as well as the vector potential (A) of a finite length arbitrary current source. The specified algorithm has been verified through FORTRAN simulation to produce results accurate to within 2 decimal places for a 500 sub-element dipole.

The VWE forms a model which is algorithmically symmetric with respect to the cartesian coordinate system. As such, the VWE lends itself to highly parallel and concurrent computational techniques. This property of the algorithm makes it an excellent candidate for implementation by a Very High Speed Integrated Circuit (VHSIC) class processor. Investigation of a parallel, highly concurrent architectural implementation has yielded preliminary results that computational savings of a factor of 3 and a throughput rate increase of 5 orders of magnitude is attainable.

This research has shown that an application specific VHSIC-class processor array has sufficient computing power to support interactive calculation of a set of equations which solve for the magnetic and electric fields, as well as the vector potential of an arbitrarily current source.

ARCHITECTURAL IMPLICATIONS OF A PARALLEL COMPUTATIONAL APPROACH TO THE VECTOR WAVE EQUATION

I. Introduction

This thesis is the study, development, and analysis of a parallel computational approach to the set of electromagnetic field equations defined below as the Vector Wave Equations (VWE). The charter of this effort was to specify a parallel processing engine for the VWE that could be implemented in either software or hardware. As such, the derivation of a parallel algorithm, specification of computational flow, and an analysis of the algorithm's behavior with respect to error/accuracy considerations became the focus of this work.

This study was motivated by the the end objective of developing an interactive CAD/CAM envi - ronment for aerodynamic systems design where a structure's electromagnetic observability, specifically its radar cross section, is known early in the design phase of a project. The capability of having a structure's observability known and interactively updated as the design progresses and is changed requires calculation of near- field electromagnetic wave equations. Numerical solutions to these equations is a computationally intensive problem that requires millions of floating point calculations.

At present, software approaches on classical computer architectures have extremely long run times. Because of this constraint, only simplified antenna configurations and geometries are examined. Conse - quently, there is a need for the development of a very fast customized VHSIC-class processor engine for calculating the radiated fields of arbitrary antennas. This processor engine would be part of a workstation which integrates the processor's electromagnetic computing capabilities along with interactive aerodynamic structural design techniques, such that electromagnetic metrics become an integral part of the design phase.

The design phase is the most effective time to influence the Radar Cross Section (RCS) of an object. Once the design has been implemented, RCS reduction must rely on the addition of heavy radar absorbing material, and active/passive cancellation. Radar absorbing material helps to dissipate the incident

energy upon its surface, thereby reducing the RCS of the target. However, this material can add con - siderable weight and thereby degrade the operational characteristics of the structure. Active/passive can - cellation techniques that reduce the RCS becomes intractable after design implementation due to the large size of the structures and current analysis capabilities. Consequently, it is more cost effective to make structural changes within a CAD environment during the design phase.

With the limitations to simplified geometries for numerical analysis, complex structures such as an aerospace vehicle require RCS measurements with scale models at compact ranges, or with full size models at outdoor ranges. This approach is time consuming, costly, and subject to scaling problems and poor weather conditions. However, the incorporation of a VHSIC-class processor engine, capable of high speed calculation of the electromagnetic equations into an interactive CAD workstation, makes possible real time observations of electromagnetic metrics. As such, the impact of modifications to the object's shape and composition material's on its RCS are known during its design.

This introductory text will include a discussion of the electromagnetic background and development of the VWE. The approach used in the derivation of the parallel algorithm will be stated. Motivation and basic assumptions in the derivation of the error expression will be discussed and referenced. Finally, some thoughts on the utilization and usefulness of the results of this study will be touched on.

Electromagnetic Background

The basis for modern antenna theory lies with the set of formulas known as Maxwell's equations. By utilizing Maxwell's equations, the total radiation pattern can be derived for a given antenna (acting as a scatterer). The total field of an antenna is the sum of the incident and the scattered fields of that antenna. To find the scattered field, the incident field is subtracted from the total field. Once the current distribution on the scatterer is found, the radiation integral given by Maxwell can be numerically computed.

A mid-point summation technique was employed to arrive at a numerical solution for the vector potential (**A**) [Jones, 1985:6]. Initially it was anticipated that the vector potential would be used as an intermediate result in obtaining the final electric (**E**) and magnetic (**H**) fields. The solution for the vector

potential involves an approximation where the dipole is divided into a finite number of sub-elements. Equation (1) defines the mathematical formula relating the vector potential solution to the general antenna problem as depicted in Figure 1. The evaluation of the complex integral in equation (1) is necessary to analyze even the simplest antenna. The desired magnetic and electric fields are derived from equations (2) and equation (3). The mathematical procedure is a vector curl operation, for which this study uses cartesian coordinates.

$$\mathbf{A} \approx \sum_{i=1}^{m} \frac{\Delta A_i \, J_i(\bar{r}_i) \, e^{-j\beta |\bar{r}-\bar{r}_i|}}{4\pi |\bar{r}-\bar{r}_i|} \qquad (1)$$

where
- ΔA_i = Surface Area of i^{th} element
- $J_i(\bar{r}_i)$ = Complex Current Density of i^{th} element
- $|\bar{r}-\bar{r}_i|$ = Distance from origin to the i^{th} element
- β = wave number $(2\pi/\lambda)$
- j = $\sqrt{-1}$.

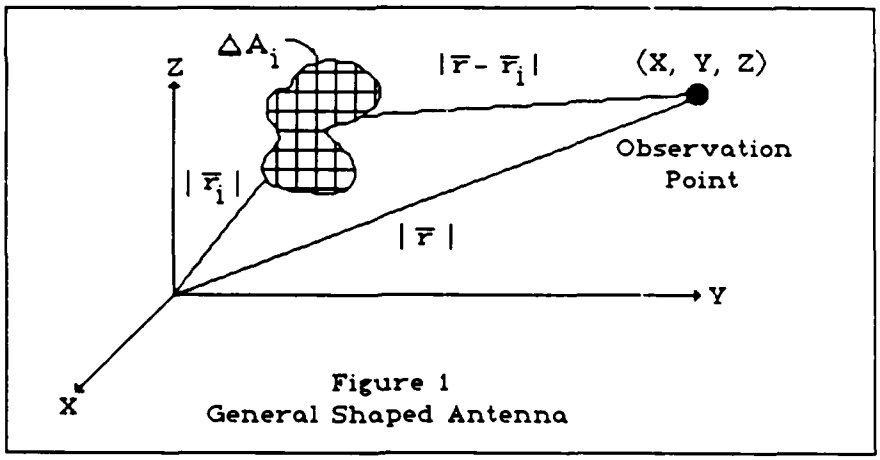


Figure 1
General Shaped Antenna

$$\mathbf{H} = \nabla \times \mathbf{A} \qquad (2)$$

$$\mathbf{E} = 1/j\omega\epsilon \, \nabla \times \mathbf{H} \qquad (3)$$

where
- ω = the radian frequency
- ϵ = the dielectric constant
- j = $\sqrt{-1}$.

However, it was found that the numerical result for the vector potential is not necessary in computing the magnetic and electric fields. The mid-point summation method employed to obtain the vector potential can be used for the magnetic and electric fields directly [Hoyt,1986]. Equations (4) and (5) below show the magnetic and electric fields in the "x" direction for cartesian coordinates. Equations for the y and z vector components are of equal complexity and similar in form to those in the x direction for the general antenna configuration. Evaluation of these equations is further complicated by current densities (J) being complex quantities in the general case.

$$H_x \approx h \sum_{i=1}^{m} \frac{\left(\frac{1}{r} + j\beta\right) e^{-j\beta r} \left(J_{yi}(z-z_i) - J_{zi}(y-y_i)\right)}{4\pi r^2} \quad (4)$$

$$E_x \approx h \sum_{i=1}^{m} \frac{e^{-j\beta r}}{4\pi r^3} \left[J_{xi}\left((z-z_i)^2 + (y-y_i)^2\right)\left(\beta^2 - \frac{1}{r^2} - \frac{3j\beta}{r}\right) + \right. \quad (5)$$

$$\left. \beta \left(-\beta^2 + \frac{3}{r^2} + \frac{3j\beta}{r}\right)\left[\left(J_{zi}(x-x_i)(z-z_i)\right) + \left(J_{yi}(x-x_i)(y-y_i)\right)\right] \right]$$

where h = length of the dipole per number of sub-elements

$r = |r - r_i| = \text{sqrt}\{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2\}$.

Hoyt derived and validated these equations, defined here to be the discrete Vector Wave Equations (VWE), with two current densities of different geometries on a simple z-directed dipole. The algorithm was evaluated with varying numbers of sub-elements to determine the number required to give an accuracy of two decimal places. The result of Hoyt's study show that the numerical summation algorithm produces results accurate to within 1% for 95 percent of the observation points for a 500 sub-element dipole. This study exploits the inherent symmetry of the algorithm with respect to the cartesian coordinate system and, as such, leads to a highly parallel and concurrent computational approach.

Approach

With Hoyt's results verifying that an acceptable level of accuracy can be obtained with 500 sub-elements, and the knowledge that any meaningful construct will be formed from thousands, possibly millions of finite length dipoles, clearly the critical path to reducing the computational intensity is to calculate the quantities within the summation formula as efficiently as possible. Therefore this thesis effort is concerned with the i^{th} calculation.

The approach used in the derivation of the computational expressions which yield the desired results is threefold. First, the technique of a top down functional decomposition was performed for each of the vector quan-tities of interest. This procedure, yielded a database of computational expressions in which intermediate results could be studied throughout the calculation process. Second, the calculation flow of each vector quantity was mapped graphically to form the construct of a data-flow graph. These data-flow graphs yield visual evidence of the inherent symmetry of these expressions and provide the parallel computational specification desired as a result of this work. Third, the data-flow graphs were overlayed to find the levels of concurrency that are attainable.

The processes of functional decomposition and the mapping into the directed graphs is detailed in chapter two of this thesis. Results in computational requirements reduction at various levels of computational concurrency is detailed in chapter four, the analysis section of the thesis. Finally, a complete set of calculation flow graphs are presented in appendix B for reference and concise access.

Error Expression

There are two types of error associated with any computer computation. The first type is error attributable to errors in data. This type of error is a function of the inherent error in finite word length and data representation. Also, the error transmitted as the value of some function to another process creates a cumulative input error to subsequent computations. The second type, generated error, is a grouping of factors such as truncation, algorithmic cancellation of significant digits, and others that are generated by the particular implementation of the function [Cody and Waite, 1980: 11]. Of the two types, the programmer

or designer has direct control of generated error but must observe the systems performance to gain insight into the "effects" and "behavior" of inherent and transmitted error.

The analysis of the cumulative effects of transmitted error requires extensive simulation for nontrivial systems. Analytic techniques rely on the Stochastic Model [Taylor, 1983: 391] where an arbitrary input is processed, and the result is of the form Y[X(n)] plus some delta representing the error indicative of the process. Manually tracking these types of data for a system of even moderate size is error prone, at best. The problem becomes intractable for complex systems.

Chapter three of this thesis implements the concept of the Stochastic Model in terms of inherent and generated error to gain visibility of the behavior of the Parallel VWE with respect to cumulative transmitted error. The process is performed in MACSYMA and is presented at a tutorial level.

Utilization of Results

In this final section of introductory text, a brief discussion of the intended utilization and resultant usefulness of the results of this thesis effort is appropriate. Three areas will be addressed: the target processing environment, intended use of the Error Expression, and the use of MACSYMA in handling large sets of equations.

Target Processing Environment. As stated by way of the end objective being a VHSIC-class processor engine for incorporation into a CAD work-station, the initial target processing environment is a hardware implementation. However, the results of this study have yielded, by means of the data-flow graphs, several levels of processing environment independent computation specifications that can and should be implemented and tested in software. Exploration into alternate architecture software implementation of any of the levels of computational concurrency is sure to yield time savings far greater than any classical computation techniques.

Error Expression. The intended use of the derived error expression discussed above is as an analytical model for hardware implementation design trade-off decisions. Because the level of abstraction is purely independent of any elementary function implementation, the result is a model

that a VLSI designer can use without constraint to any one particular elementary function implementation. This type of model, when complemented with empirical data from simulation or known levels of accuracy from previous designs, can aid in the assurance of overall system accuracy being known during the design phase.

Use of MACSYMA. One of the most daunting aspects of the analysis of nontrivial sets of equations is maintaining analytic clarity and accuracy of the expression as the set of equations is manipulated from one step to the next. MACSYMA, a symbolic equation processor, has proven through the course of this study to be a tool capable of handling the enormous book-keeping tasks as well as providing a full range of mathematical functions to handle any computational task required. The use of MACSYMA in this theses effort is extensive and fully documented. The intent of the presentation at this level of detail is to provide a guide for future applications of this type.

II. Development of Parallel VWE Algorithm

The process of functional decomposition of the VWE equations through the use of MACSYMA is presented in a step by step explanation of the Parallel Vector Potential derivation. The results of the more complex expressions for the electric and magnetic fields are given in a less tutorial form for brevity. The mapping of the parallel algorithm into the data-flow graph construct is also shown below.

The process is one of substituting variables into an equation. These variables represent "allowable" elementary functions. Allowable is connoted to mean either a hardware functional element or a software process implemented in the target processing environment.

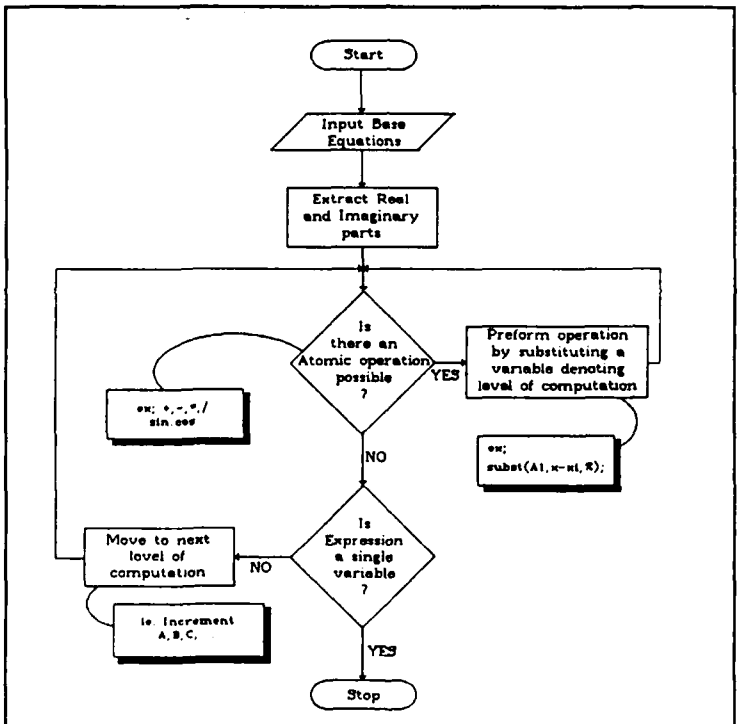


Figure 2 Decomposition Algorithm

The variables are coded such that computational and algorithmic clarity is maintained. That is, the coding scheme should clearly define computational levels in the overall calculation, while maintaining end-to-end algorithmic correctness. For this study the ALPHA characters A, B, C, ... denote distinct

computational levels while the concatenated numeric designators 1, 2, 3, ... uniquely define a specific elementary function. For example, "E5" might mean a specific multiplication at the "E" computational level. This process is achieved by the algorithm given above in Figure 2.

Vector Potential Derivation

The cartesian expression for the Discrete Vector Potential is :

$$\mathbf{A} = \sum_{i} (\mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z)$$

where $\mathbf{A}_x = G * (jxri + jxii)$
$\mathbf{A}_y = G * (jyri + jyii)$
$\mathbf{A}_z = G * (jzri + jzii)$

noting that jxri is the real part of the complex current density in the x direction, jxii is the imaginary part with similar notation being used for the other coordinate system elements,

$$G = \frac{e^{-j\beta r}}{4\pi r}$$ which is the free space form of Greens Function,

and r is the vector distance between the i^{th} element and the observation point; β is the wave number ($2\pi/\lambda$).

These expressions are entered into MACSYMA as follows:

```
g:(%E^(-%i*b*r))/(4*%pi*r);

ax:g*(jxri+%i*jxii);

ay:g*(jyri+%i*jyii);

az:g*(jzri+%i*jzii);
```

The resulting MACSYMA presentation for the three vector coordinate system elements are shown below:

ax;

$$\frac{(jxri + \%i\ jxii)\ \%e^{-\%i\ b\ r}}{4\ \%pi\ r}$$

```
                                                    - %i b r
                                  (jyri + %i jyii) %e
ay;                               ---------------------
                                      4 %pi r

                                                    - %i b r
                                  (jzri + %i jzii) %e
az;                               ---------------------    .
                                      4 %pi r
```

Once entered, the top-down functional decomposition of the x directed vector potential may begin. The process is identical for the y and z coordinate system elements.

From the above expression for A_x (ax) the real and imaginary parts are extracted. The real and imaginary parts of the Vector Potential in the x direction are defined as axr and axi below:

```
realpart(ax)
 axr;                             jxii sin(b r) + jxri cos(b r)
                                  -----------------------------
                                          4 %pi r
```

```
imagpart(ax)

 axi;                             jxii cos(b r) - jxri sin(b r)
                                  -----------------------------    .
                                          4 %pi r
```

These real and imaginary parts may be operated on according to the above substitution algorithm. First, however, recall that "r" is the vector distance between the i^{th} element and the observation point. As the discrete VWE given here in cartesian coordinates, the cartesian expression for r is required. This substi - tution gives an initial, fully decomposed form of the real and imaginary parts (denoted by axr0 and axi0) as:

```
axr0;
                                                  2          2          2
        (jxii sin(b sqrt((z - zi)  + (y - yi)  + (x - xi)  ))

                                                    2          2          2
        + jxri cos(b sqrt((z - zi)  + (y - yi)  + (x - xi)  )))

                                                  2          2          2
        /(4 %pi sqrt((z - zi)  + (y - yi)  + (x - xi)  ))
```

axi0; $$(jxii\ \cos(b\ \sqrt{(z - zi)^2 + (y - yi)^2 + (x - xi)^2}) - jxri\ \sin(b\ \sqrt{(z - zi)^2 + (y - yi)^2 + (x - xi)^2}))$$ $$/(4\ \%pi\ \sqrt{(z - zi)^2 + (y - yi)^2 + (x - xi)^2}).$$

Following the development process, the first substitution is A1, A2, and A3 for (x-xi), (y-yi), and (z-zi). At this point in the substitution process, these are the only allowable elementary operations capable of being performed. All other calculations within the equation are contingent upon the completion of these. Therefore, there is a delineation of a computational level. The status of the equation after this first level of substitution is:

axr1; $$\frac{jxii\ \sin(b\ \sqrt{A3^2 + A2^2 + A1^2}) + jxri\ \cos(b\ \sqrt{A3^2 + A2^2 + A1^2})}{4\ \%pi\ \sqrt{A3^2 + A2^2 + A1^2}}$$

and axi1; $$\frac{jxii\ \cos(b\ \sqrt{A3^2 + A2^2 + A1^2}) - jxri\ \sin(b\ \sqrt{A3^2 + A2^2 + A1^2})}{4\ \%pi\ \sqrt{A3^2 + A2^2 + A1^2}}.$$

This process follows the procedures defined above and the status at each computational level is generated. The status after the second substitution where $A1^2$, $A2^2$, and $A3^2$ is replaced by B1, B2, and B3, respectively is:

axr2; $$\frac{jxii\ \sin(b\ \sqrt{B3 + B2 + B1}) + jxri\ \cos(b\ \sqrt{B3 + B2 + B1})}{4\ \%pi\ \sqrt{B3 + B2 + B1}}$$

and axi2; $$\frac{jxii\ \cos(b\ \sqrt{B3 + B2 + B1}) - jxri\ \sin(b\ \sqrt{B3 + B2 + B1})}{4\ \%pi\ \sqrt{B3 + B2 + B1}}$$

We note two things here. First, the quantity 4π (4 %pi) is considered as an input constant such that a separate multiplication is not required. Secondly, in the third level of computation shown below, the function C1 is a triple summation. The development of the VLSI Triple Summer will be a future thesis project and was accepted as an allowable elementary function for this study. With the triple summer, the status after the third level of substitution is:

$$\text{axr3;} \quad \frac{\text{jxii sin(b sqrt(C1))} + \text{jxri cos(b sqrt(C1))}}{\text{4 \%pi sqrt(C1)}}$$

$$\text{and axi3;} \quad \frac{\text{jxii cos(b sqrt(C1))} - \text{jxri sin(b sqrt(C1))}}{\text{4 \%pi sqrt(C1)}} .$$

The square root operation is performed next via substitution of D1. The status after the forth level is:

$$\text{axr4;} \quad \frac{\text{jxii sin(b D1)} + \text{jxri cos(b D1)}}{\text{4 \%pi D1}}$$

$$\text{and axi4;} \quad \frac{\text{jxii cos(b D1)} - \text{jxri sin(b D1)}}{\text{4 \%pi D1}} .$$

Next, the multiplication of β and D1 is performed, for which we substitute E1. Here we also have the multiplication of the input constant 4π (4 %pi) and D1 replaced with E2. The status after the fifth sub - stitution is:

$$\text{axr5;} \quad \frac{\text{jxii sin(E1)} + \text{jxri cos(E1)}}{\text{E2}}$$

$$\text{and axi5;} \quad \frac{\text{jxii cos(E1)} - \text{jxri sin(E1)}}{\text{E2}} .$$

The process continues with the substitutions F1, F2, and F3 for cos(E1), sin(E1), and the inversion process of E2, respectively. Note that an inversion process is used rather than a division to provide a more general specification for the hardware implementation. The status after level six is:

axr6; (jxii F2 + jxri F1) F3

and axi6; (jxii F1 - jxri F2) F3 .

The next substitutions are for the multiplications of F1 and F2 with the complex components of the current density. The status after the seventh level of computation is:

axr7; F3 (G4 + G1)

and axi7; F3 (G3 - G2) .

The next operations to be performed are the remaining addition and subtraction from above. The status after level eight is:

axr8; F3 H1

and axi8; F3 H2 .

Finally, we multiply the quantities above. Recall that F3 is the resultant inversion quantity from level six. The final results are:

axr9;	J1	The real part of the vector potential in the "X" direction
axi9;	J2	The imaginary part of the vector potential in the "X" direction.

Mechanically the process is one of scanning the the equation to check if an allowable elementary function can be performed. If one can, it is coded by the conventions stated above. That is, the function is given the ALPHA designator for the current computational level followed by a numeric designator uniquely defining that specific function. The resultant coded variable is substituted into the equation, and the same process is continued until no more elementary functions can be performed. The ALPHA designator is incremented, and the process is continued until a single variable remains signifying that the calculation process is complete.

Note that for this study, the above process was accomplished manually. However, clearly this process is one that lends itself to and would benefit by automated techniques. Also the letter "I" was not used in defining computational levels to avoid confusion between alpha/numeric characters.

The results of this process provide us with a database of computational expressions at each level where the substitution designators, A1, B3, etc., indicate floating point operations required in the evaluation of the equation. Once this process is accomplished for each of the coordinate system elements, a listing of the required substitutions is generated. The listing for the entire cartesian set is generated by eliminating the redundant calculations. The combined calculations form a parallel computational approach to the vector potential. Below is a listing for the discrete calculation of the real part of A_x and the combined cartesian set with the redundant calculations removed:

	Discrete A_x real	Combined Cartesian Set (**A**)
Level one:	subst("A1",(x-xi),%)$ subst("A2",(y-yi),%)$ subst("A3",(z-zi),%)$	subst("A1",(x-xi),%)$ subst("A2",(y-yi),%)$ subst("A3",(z-zi),%)$
Level two:	subst("B1","A1"^2,%)$ subst("B2","A2"^2,%)$ subst("B3","A3"^2,%)$	subst("B1","A1"^2,%)$ subst("B2","A2"^2,%)$ subst("B3","A3"^2,%)$
Level three:	subst("C1","B1"+"B2"+"B3",%);	subst("C1","B1"+"B2"+"B3",%);
Level four:	subst("D1",sqrt("C1"),%);	subst("D1",sqrt("C1"),%);
Level five:	subst("E1",b*"D1",%)$ subst("E2",(4*%pi*"D1"),%);	subst("E1",b*"D1",%)$ subst("E2",(4*%pi*"D1"),%);
Level six:	subst("F1",cos("E1"),%)$ subst("F2",sin("E1"),%)$ subst("F3",1/"E2",%);	subst("F1",cos("E1"),%)$ subst("F2",sin("E1"),%)$ subst("F3",1/"E2",%);

Level seven:	subst("G1",jxri*"F1",%)$
	
	subst("G4",jxii*"F2",%);
		subst("G1",jxri*"F1",%)$
		subst("G2",jxri*"F2",%)$
		subst("G3",jxii*"F1",%)$
		subst("G4",jxii*"F2",%);
		subst("G5",jyri*"F1",%)$
		subst("G6",jyri*"F2",%)$
		subst("G7",jyii*"F1",%)$
		subst("G8",jyii*"F2",%);
		subst("G9",jzri*"F1",%)$
		subst("G10",jzri*"F2",%)$
		subst("G11",jzii*"F1",%)$
		subst("G12",jzii*"F2",%);

Level eight:	subst("H1","G1"+"G4",%);	subst("H1","G1"+"G4",%);

		subst("H2","G3"-"G2",%);
		subst("H3","G5"+"G8",%);
		subst("H4","G7"-"G6",%);
		subst("H5","G9"+"G12",%);
		subst("H6","G11"-"G10",%);

Level nine:	subst("J1","F3""H1",%);	subst("J1","F3""H1",%);
		subst("J2","F3""H2",%);
		subst("J3","F3""H3",%);
		subst("J4","F3""H4",%);
		subst("J5","F3""H5",%);
		subst("J6","F3""H6",%);

It is noted that there is a one to one correspondence of the substitution designators to the nodes of a data-flow graph. A graphical representation is generated by mapping the above listings into data-flow graphs. The data-flow graphs for the real part of the "x" directed vector potential and the combined cartesian set are shown in Figures 3 and 4.

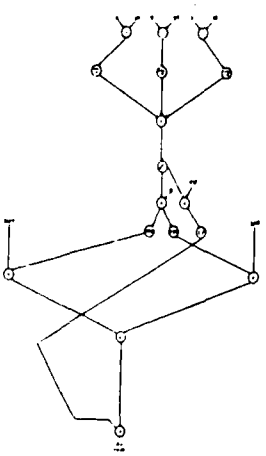


Figure 3 Real Part of "X" Directed Vector Potential

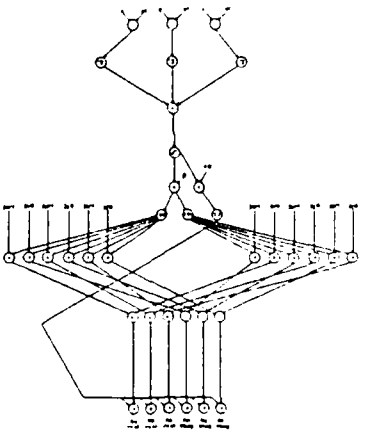


Figure 4 Combined Cartesian Set for Vector Potential

The substitution process for the vector potential is presented in MACSYMA syntax in Appendix A. Data-flow graphs for all vector quantities are given in Appendix B.

Magnetic Field Derivation

The equation for the Discrete Magnetic Field in cartesian coordinates is:

$$\mathbf{H} = \sum_{i} (\mathbf{H}_x + \mathbf{H}_y + \mathbf{H}_z).$$

The complexity of the equations precludes a tutorial presentation. The processing of these field equations is of such magnitude that manipulation becomes intractable without an equation processor such as MACSYMA. For example, at the point when the cartesian expression for the vector distance "r" is substituted into the expression for the real part of the magnetic field in the "x" direction forming "Hxr0", the equation has the form:

$$((\cos(b\ \mathrm{sqrt}((z - zi)^2 + (y - yi)^2 + (x - xi)^2)) (jyri\ (z - zi) - jzri\ (y - yi)) + \sin(b\ \mathrm{sqrt}((z - zi)^2 + (y - yi)^2 + (x - xi)^2))\ (jyii\ (z - zi) - jzii\ (y - yi)))\ /\mathrm{sqrt}((z - zi)^2 + (y - yi)^2 + (x - xi)^2)$$

$$- b\ (\cos(b\ \mathrm{sqrt}((z - zi)^2 + (y - yi)^2 + (x - xi)^2))\ (jyii\ (z - zi) - jzii\ (y - yi)) - \sin(b\ \mathrm{sqrt}((z - zi)^2 + (y - yi)^2 + (x - xi)^2))\ (jyri\ (z - zi) - jzri\ (y - yi))))$$

$$/(4\ \%pi\ ((z - zi)^2 + (y - yi)^2 + (x - xi)^2)).$$

The substitution process for the magnetic field is carried out similarly to that of the vector poten - tial described above. The listings are generated and the data-flow graphs are formed. The data-flow graphs for the real part of the "x" directed magnetic field and the combined cartesian set is given below in Figures 5 and 6. The complete substitution process for the magnetic field is presented in MACSYMA syntax in Appendix A. Data-flow graphs for all vector quantities are given in Appendix B.

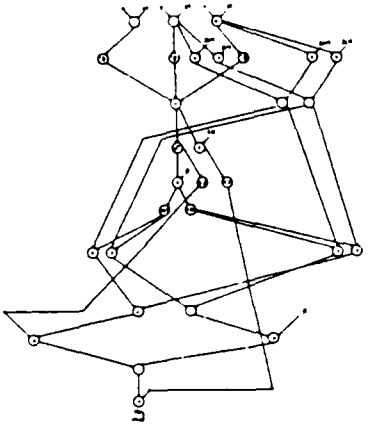


Figure 5 Real Part of "X" Directed Magnetic Field

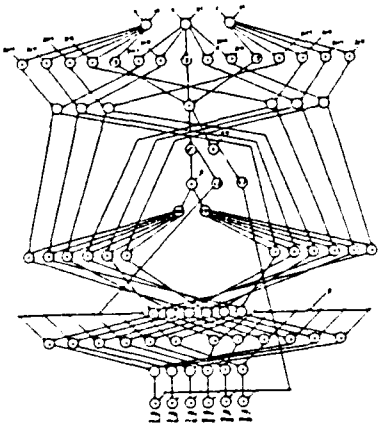


Figure 6 Combined Cartesian Set for Magnetic Field

Electric Field Derivation

The equation for the Discrete Electric Field in cartesian coordinates is:

$$E = \sum_{i} (E_x + E_y + E_z).$$

The complexity of the electric field equations is even greater than the equations for the magnetic field. For example, at the same point illustrated above, when the cartesian expression for the vector distance "r" is substituted into the expression for the real part of the electric field in the "x" direction forming "Exr0", the equation has the form:

$$(\sin(b\sqrt{(z - zi)^2 + (y - yi)^2 + (x - xi)^2})) \left(\frac{3b\,(jzri\,(x - xi)\,(z - zi) + jyri\,(x - xi)\,(y - yi))}{\sqrt{(z - zi)^2 + (y - yi)^2 + (x - xi)^2}}\right.$$

$$\left. + (jxii\,(b^2 - \frac{1}{(z - zi)^2 + (y - yi)^2 + (x - xi)^2}) - \frac{3b\,jxri}{\sqrt{(z - zi)^2 + (y - yi)^2 + (x - xi)^2}}) ((z - zi)^2 + (y - yi)^2)\right.$$

$$+\left(\frac{3}{(z-zi)^2+(y-yi)^2+(x-xi)^2}-b^2\right)(jzii\,(x-xi)\,(z-zi)+jyii\,(x-xi)\,(y-yi)))$$

$$+\cos\left(b\,\mathrm{sqrt}((z-zi)^2+(y-yi)^2+(x-xi)^2)\right)\left(-\frac{3\,b\,(jzii\,(x-xi)\,(z-zi)+jyii\,(x-xi)\,(y-yi))}{\mathrm{sqrt}((z-zi)^2+(y-yi)^2+(x-xi)^2)}\right.$$

$$+\left(\frac{3\,b\,jxii}{\mathrm{sqrt}((z-zi)^2+(y-yi)^2+(x-xi)^2)}\right.$$

$$\left.+jxri\left(b^2-\frac{1}{(z-zi)^2+(y-yi)^2+(x-xi)^2}\right)\right)((z-zi)^2+(y-yi)^2)$$

$$+\left(\frac{3}{(z-zi)^2+(y-yi)^2+(x-xi)^2}-b^2\right)(jzri\,(x-xi)\,(z-zi)+jyri\,(x-xi)\,(y-yi))))$$

$$/(4\,\%pi\,((z-zi)^2+(y-yi)^2+(x-xi)^2)^{3/2})$$

The substitution process for the electric field is carried out similar to that for the vector potential and magnetic field described above. The listings are generated, and the data-flow graphs are formed. The data-flow graphs for the real part of the "x" directed electric field and the combined cartesian set is given below in Figures 7 and 8. The complete substitution process for the electric field is presented in MACSYMA syntax in Appendix A. Data-flow graphs for all vector quantities are given in Appendix B.

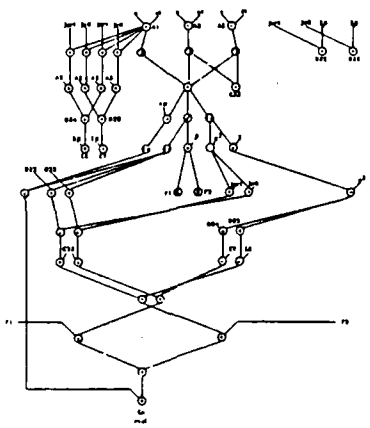


Figure 7 Real Part of "X" Directed Electric Field

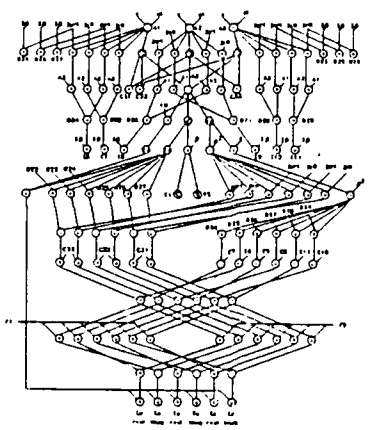


Figure 8 Combined Cartesian Set for Electric Field

It is important to note that the data-flow graphs developed here provide several types of specification data. A programmer can use these graphs as the algorithmic specification to reduce processing by means of eliminating redundant computations. The computer architect will note the high degree of concurrent processing that can be achieved in an "Alternate Architecture" approach. Further, the VLSI designer now has a system architecture specification at the functional element level. The impact and implications of this parallel computational approach is developed in chapter four of this thesis.

III. Development of VWE Error Expression

In this section we seek to develop an analytic expression which will provide a model for implementation trade-off analysis for the parallel VWE with respect to accuracy/error considerations. In doing so we define an algorithmic entity as the VWE Core Equation. This expression is common to all three vector quantities of interest to this thesis effort. The development of the error expression is a bottom-up process in which error terms are assigned to each of the allowable elementary functions and to the input data. The resultant terms are substituted into the VWE core and the computation, as specified in chapter two, is carried out. The final expression includes the arithmetic quantity desired and an upper bound on the cumulative results of the functional and generated error. This resultant expression forms a model for analysis of the VWE Core Equation with error terms as the parametric variables.

The text below is organized into the definition of the VWE core and assignment of the error terms for the allowable elementary functions, detailed presentation of the bottom-up derivation of the error expression, and the resulting error model which can be used for implementation trade-off analysis.

The VWE Core

We see from the data-flow graphs that the VWE Core Equation forms two quantities:

$$\text{Cos}(\beta \, \text{sqrt}((x-xi)^2 + (y-yi)^2 + (z-zi)^2)$$

and

$$\text{Sin}(\beta \, \text{sqrt}((x-xi)^2 + (y-yi)^2 + (z-zi)^2)$$

For this discussion we will consider the cosine term only. Graphical representation of the VWE Core Equation is shown below in figure 9.

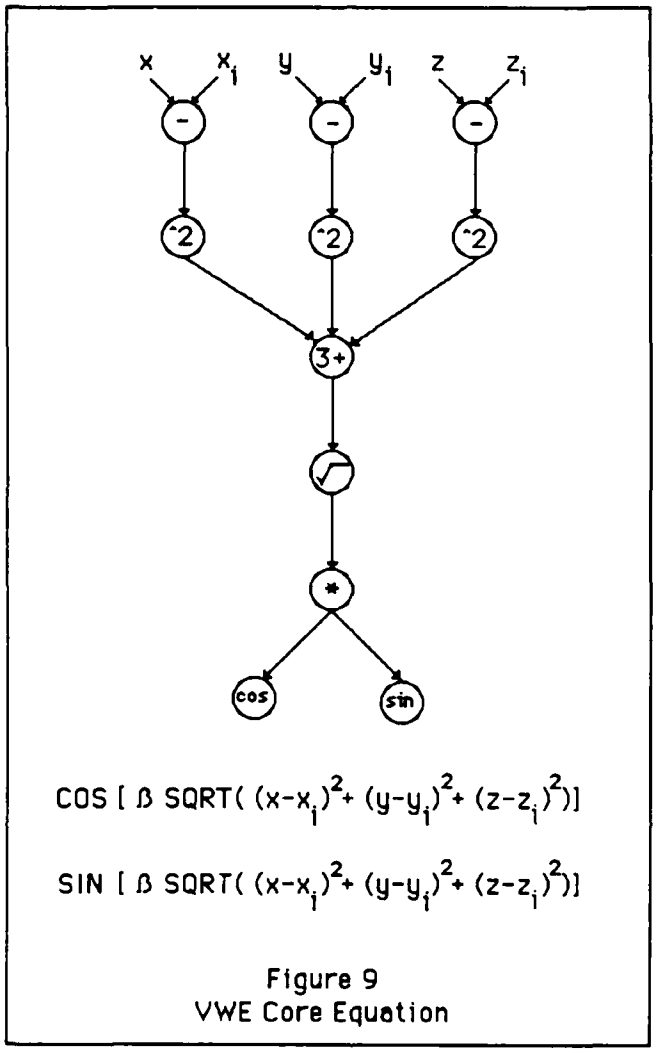


Figure 9
VWE Core Equation

Assignment of Error Terms

The approach for deriving the error model will be to assign an error term by arithmetic operation. That is, all additions will be assigned the same error term, all multiplications ... and so forth. A summary of the error terms assigned to the arithmetic operations are below:

Operation	Error Term
Data Representation	e0
Addition/Subtraction	e1
Multiplication	e2
Square Root	e3
Sin/Cos	e4
Inversion	e5
Triple Summation	e6
Cumulative Error (Level: L)	egL

Table 1 Error Terms

These error terms are defined as the total error inherent in the input data (bit/word length), and the error generated as the function value (resolution, truncation, etc.). A notational convenience will be used when the cumulative expression detracts from analytical clarity. It is given a designator defining its comp - utational level and is treated as a separate arithmetic quantity at higher levels. This approach facilitates im - plementation of the stochastic error model for complex systems and yields a final error expression that is independent of any specific elementary function implemen-tation. Thus, the error expression provides only specification data and does not restrict the VLSI designer to any one (analyzed) implementation.

Bottom-up Derivation of Error Model

As in chapter two, this section will be tutorial with respect to the level of MACSYMA syntax used in its presentation. The intent is to provide an example to future users for similar applications.

We start by entering the core into MACSYMA:

```
core:cos(b*sqrt( (x-xi)^2 + (y-yi)^2 + (z-zi)^2 ));

(c24) core;

                                  2         2         2
(d24)          cos(b sqrt((z - zi)  + (y - yi)  + (x - xi)  ))
```

We proceed similarly to the approach taken in the calculation flow analysis. The text below il - lustrates the steps taken in this analysis. The following series of substitutions replace the input values x, xi, y, ...zi with a construct of the form x + e0, xi + e0, ... zi +e0. The "e0" term accounting for the in - herent error of the finite length binary representation used in any computer architecture.

```
subst(x-xi+2*e0,x-xi,%);

                                  2         2                     2
      cos(b sqrt((z - zi)  + (y - yi)  + (- xi + x + 2 e0)  ))

subst(y-yi+2*e0,y-yi,%);

                                  2                     2                     2
      cos(b sqrt((z - zi)  + (- yi + y + 2 e0)  + (- xi + x + 2 e0)  ))

subst(z-zi+2*e0,z-zi,%);

                                  2                     2                     2
          cos(b sqrt((- zi + z + 2 e0)  + (- yi + y + 2 e0)  + (- xi + x + 2 e0)  ))
```

Next we perform the arithmetic operations of level "A" and add the associated error terms for the subtraction operation "e1". We note here the procedure that will be used in the derivation of the error ex - pression. To form an upper bound expression, we assume the form of the general addition of error terms. That is, we will not attempt to arithmetically cancel error terms at the atomic level. To do so would be to assume more about a target architecture than is intended by this study. As such, the following MACSYMA substitution processes are of the form $(x\text{-}x_i) \Rightarrow (x\text{-}x_i) + e1$.

```
subst("A1" + e1+2*e0,x-xi+2*e0,%)$
subst("A2" + e1+2*e0,y-yi+2*e0,%)$
subst("A3" + e1+2*e0,z-zi+2*e0,%);
```

The resulting form of the VWE Core after the first level of comp-utation is significantly more complex than that of chapter two, again exemplifying the need to use an equation processor. The status is shown as core1 below:

```
core1;
                                    2                  2                  2
cos(b sqrt((A3 + e1 + 2 e0)  + (A2 + e1 + 2 e0)  + (A1 + e1 + 2 e0)  ))
```

Next we multiply the inner terms to enable use to proceed. We can also see the effect of cum - ulative error, noting the "3e1" and "12e0" terms.

```
expand(%);
                  2                          2                          2
cos(b sqrt(A3  + 2 e1 A3 + 4 e0 A3 + A2  + 2 e1 A2 + 4 e0 A2 + A1

                                  2                  2
          + 2 e1 A1 + 4 e0 A1 + 3 e1  + 12 e0 e1 + 12 e0  ))
```

The calculations of level two are performed by first, substituting for the A^2 operation. Then, we substitute in the error terms associated with the squaring operation (same as multiplication). the process is shown below:

```
subst("B1","A1"^2,%)$
subst("B2","A2"^2,%)$
subst("B3","A3"^2,%);

cos(b sqrt(B3 + B2 + B1 + 2 e1 A3 + 4 e0 A3 + 2 e1 A2 + 4 e0 A2 + 2 e1 A1

                              2                  2
          + 4 e0 A1 + 3 e1  + 12 e0 e1 + 12 e0  ))

subst("B1" + e2,"B1",%)$
subst("B2" + e2,"B2",%)$
subst("B3" + e2,"B3",%);
```

The status of the computation at level two is:

```
core2;
cos(b sqrt(B3 + B2 + B1 + 2 e1 A3 + 4 e0 A3 + 2 e1 A2 + 4 e0 A2 + 2 e1 A1

                                  2                  2
          + 4 e0 A1 + 3 e2 + 3 e1  + 12 e0 e1 + 12 e0  ))
```

To perform the computation of level three (the triple summation), we need to substitute the entire inner expression as MACSYMA will not allow sub-atomic substitutions. The process and the status at level three is given below:

```
subst("C1"+e6+2*e1+"A3"+4*e0*"A3"+2*e1*"A2"+4*e0*"A2"+2*e1*"A1"+4*e0*"A1"+3*e2+3*
e1^2+12*e0*e1+12*e0^2,"B3" + "B2" +
"B1"+2*e1*"A3"+4*e0*"A3"+2*e1*"A2"+4*e0*"A2"+2*e1*"A1"+4*e0*"A1"+3*e2+3*e1^2+12*e
0*e1+12*e0^2,%);
```

Note, we have added the error term e6 as part of the substitution.

```
core3;
```

$$\cos(b\ \mathrm{sqrt}(C1 + e6 + 4\ e0\ A3 + A3 + 2\ e1\ A2 + 4\ e0\ A2 + 2\ e1\ A1 + 4\ e0\ A1 + 3\ e2 + 3\ e1^2 + 12\ e0\ e1 + 2\ e1 + 12\ e0^2))$$

At this point we wish to simplify the expression. We will define "eg3" as the generated error at level three. The development is as follows:

$$\text{let}\quad \lambda = + e6 + 4\ e0\ A3 + A3 + 2\ e1\ A2 + 4\ e0\ A2 + 2\ e1\ A1 + 4\ e0\ A1 + 3\ e2 + 3\ e1^2 + 12\ e0\ e1 + 2\ e1 + 12\ e0^2$$

this expression represents all of the terms inside the square root operator except C1. Recall the identity :

$$\mathrm{sqrt}(A+B) = \mathrm{sqrt}[(A+B)*(A/A)] = \mathrm{sqrt}\,[(A+B)/A] * \mathrm{sqrt}\,(A)$$

we then define eg3 = sqrt[(C1+λ)/C1] and perform the substitution with resulting expression shown below:

```
subst(b*sqrt("C1")*eg3,  b*sqrt("C1" +e6+2*e1+"A3"+4*e0*"A3"+2*e1*"A2"
+4*e0*"A2"+2*e1*"A1"+4*e0*"A1"+3*e2+3*e1^2+12*e0*e1+12*e0^2),%),
```

```
core3.1;
```

$$\cos(b\ eg3\ \mathrm{sqrt}(C1))$$

We note here that eg3 has a multiplicative relationship to the desired atomic expression "sqrt(C1)". As we proceed to level four, we have not violated any arithmetic axiom as eg3 has the form of the square root operation inherent in it as shown in the development above.

The computation and status at level four is given below:

```
subst(b*("D1" + e3 )* eg3,b*eg3*sqrt("C1") ,%);
core4;
```

cos(b eg3 (D1 + e3))

We can expand the inner expression as follows:

```
core4.1;
```

cos(b eg3 D1 + b e3 eg3)

At this step we see that the expression yields a multiplicative error term which is not readily separable from within the argument of the cosine term. As such, we proceed to level five with the cumulative error as shown above. The operation is performed and the error term is substituted below:

```
subst("E1",b*"D1",%);
subst("E1"+e2,"E1",%);
```

This last substitution yields:

$$\cos[\, E1\ eg3\ +(\beta\ e3\ eg3\ )+e2\,]$$

which by trigonometric substitution has the form:

$$\cos(E1\ eg3)\cos((\beta\ e3\ eg3\ )+e2)-\sin(E1\ eg3)\sin((\beta\ e3\ eg3\ )+e2)$$

We see that at this stage that we can only get the desired term:

$$\cos(E1)$$

if and only if:

$$eg3 = 1$$

and

$$(\beta\ e3\ eg3\ )+e2 = n2\pi\ ,\ n= 0,1,2,3,...$$

We now have a closed form expression that in a limited sense models the behavior of the VWE calculation. With the exception of the inversion or division function we now have an expression which includes all of the arithmetic operations (and the associated error terms) required in the calculation of the VWE.

The VWE Error Model

The VWE Error Model follows from the derivation above. We saw that the desired value "cos(E1)" is attained if and only if:

$$eg3 = 1$$

and

$$(\beta\ e3\ eg3\ ) + e2 = n2\pi\ ,\ n= 0,1,2,3,...$$

where $eg3 = sqrt[(C1+\lambda)/C1] = sqrt[1 + \lambda/C1]$

$$C1 = (x-xi)^2 + (y-yi)^2 + (z-zi)^2$$

$$= (x^2 - 2\ x\ xi + xi^2 + y^2 - 2\ y\ yi + yi^2 + z^2 - 2\ z\ zi + zi^2)$$

$$\lambda = 4\ e0\ A3 + A3 + 2\ e1\ A2 + 4\ e0\ A2 + 2\ e1\ A1 + 4\ e0\ A1 + e6 + 3\ e2 + 3\ e1^2 + 12\ e0\ e1 + 2\ e1 + 12\ e0^2$$

$$A1 = x-xi$$

$$A2 = y-yi$$

$$A3 = z-zi$$

These relationships constitute a model. The designer with known error metrics for the functional elements which implement the required elementary operations can substitute those metrics into the expres - sions and gain visibility of the impact that any particular implementation will have on the system's accu - racy well before bread boarding a design. This process supports design trade-off decisions with respect to "system accuracy" specifications at a much earlier and cost effective phase of the project.

It is imperative to note that this model, alone, may not be suitable for such design decisions. Exhaustive simulation as well as derivation of a family of VWE error expressions and models may be required. The complexity of the VWE and the many possible architectural implementations will drive this issue.

The model developed in this study is a baseline. Further development and analysis is beyond the scope of this thesis.

IV. Analysis

In this section, the total computational requirements of the parallel VWE is analyzed and the calculated savings at various levels of computational concurrency is presented. The data-flow graphs developed in chapter two are used to extract significant data for this analysis. The significant data derived from the graphical representation includes visibility of operational redundancy within the coordinate components of a specific vector quantity and across the three vector quantities of interest, **A**, **E**, and **H**. Visibility of data dependencies and maximal concurrency possible in the calculation of all components of **A**, **E**, and **H** is readily available. Finally, the graphs provide a means for comparison of implementation of the equation at various levels of computational concurrency.

Removal of Operational Redundancy

Classically, the real and imaginary parts of each of the vector components would be discretely calculated without regard to the fact that several of the intermediate results are reusable in the calculation of other coordinate system elements. For example, when calculating the real part of $\mathbf{H}_x$, there are 30 floating point operations involved. There are, noting symmetry, 30 floating point operations involved in the calc - ulation of the imaginary part of $\mathbf{H}_x$. However, as clearly visible in the graphical representation, 26 of these operation are identical. Therefore both quantities may be obtained in 34 flops. Further analysis of the graphical representation shows that the complete vector component set for both the real and imaginary parts of the magnetic field, **H**, can be calculated in just 74 flops as apposed to 180 when computed classically. The analysis is summarized in table 2 for all vector quantities of interest.

Also shown in table 2 is an entry for "**E** & **H** fields". This entry represents the removal of all computational redundancy within the complete **E** and **H** graphical representations and is the overlaying of the 2 computational processes to form one computational process. The savings of 26 flops achieved by this overlaying is due to a common core of operations present in these vector quantities similar to that

	Discrete	Paired	Combined
Vector Potential	17/102	21/63	37
H Field	30/180	34/102	74
E Field	52/312	56/168	130
E &H Fields	—	—	170
E&H & Vector Potential	—	—	202

Table 2 Computational Requirements

defined in chapter three as the VWE Core Equation. Because of this common core of operations, the vector potential, A, can be integrated into the fully parallel calculation of the three vector quantities of interest, such that all results are obtainable in 202 flops as apposed to 594 flops when computed classically, representing nearly 66% computational savings. The impact of these results are shown in graphical form in figure 10 where the computational requirements are plotted for a given number of dipole elements.

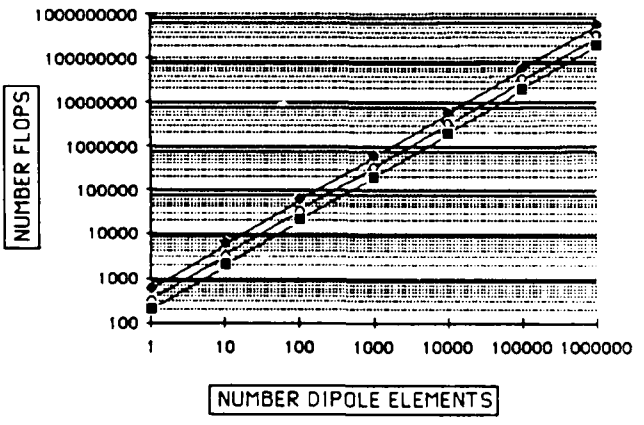


Figure 10 Number FLOPS vs Dipole Elements for a given Computational Approach

Maximal Concurrency

Concurrency in this thesis is understood in an architectural sense. That is, concurrency means the simultaneous execution of a number of discrete operations. Specifically, in this thesis, concurrency means the simultaneous execution of floating point arithmetic operations.

The visibility into the data dependencies that the data-flow graphs provides allows us to observe 12 distinct computational levels when the entire set of vector quantities are overlayed as depicted on the following page in figure 11.

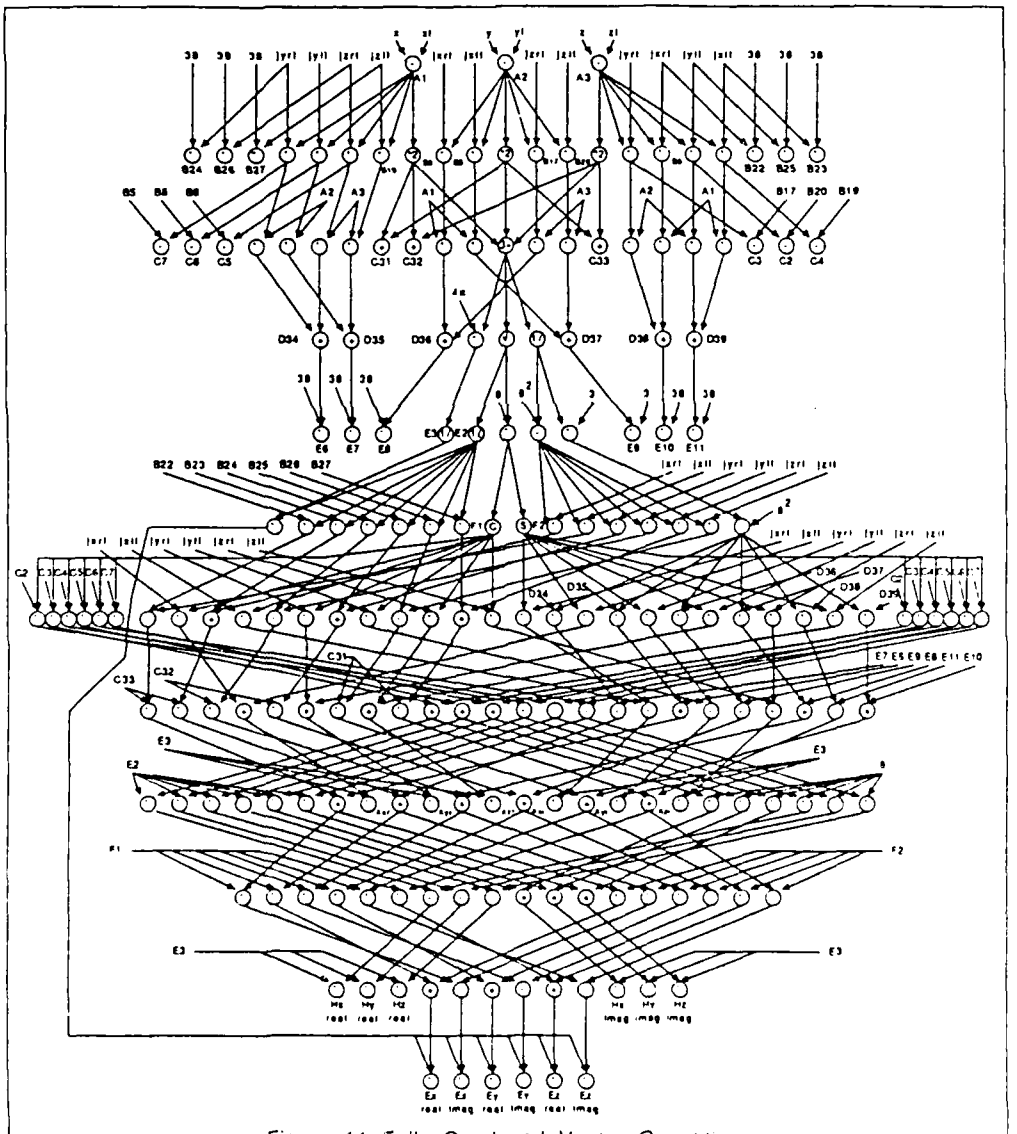


Figure 11 Fully Overlayed Vector Quantities

This represents a 98% reduction in linear time when compared to the classical approach. However this dramatic time reduction is attainable only if the processing environment supports concurrent floating point arithmetic operations at the maximum width of the VWE calculation. The maximum operational width of the VWE is 36, occurring at the seventh computational level, G. These 36 operations include 30 floating point multiplies and six adder/subtractors.

Architectural Considerations

With the end objective of developing an interactive CAD/CAM environment for aerodynamic systems design, the target processing environment is assumed to be a VHSIC class processor, that is, a hardware implementation. However, the computational specification data available from the parallel approach is target processing environment independent in that any current or future architecture can realize processing gain by implementing the parallel VWE at the maximum level of computational concurrency supportable by that architecture.

As shown in the table 2 above, discrete calculation of the i^{th} vector element requires 594 FLOPS. Assuming that the accuracy verified by Hoyt is sufficient, we will have 500 of these i^{th} calculations per dipole element. If we define the execution time of the average floating point arithmetic operation to be T_a, then we derive an expression to approximate the time required for the solution of the discrete VWE which we define as $T_{discrete}$:

$$T_{discrete} \approx [500 * 594T_a] * N,$$

where N = the number of dipole elements in the structure.

Using the results of the parallel approach where all redundant computations have been eliminated, as shown in table 2, there are 202 FLOPS required. Defining $T_{combined}$ to be time required for this solution we have:

$$T_{combined} \approx [500 * 202T_a] * N.$$

This approach when compared to the discrete approach provides a speed up of nearly 3. It is significant to note that this speedup can be achieved on the same architecture as that of the discrete by simply recoding the algorithm to use all partial results. The speedup actually achieved on a given architecture may be somewhat less than 3 due to the width of the computation and the resulting need to store partial results in memory versus registers.

If however the the given architecture supports concurrent floating point arithmetic then the parallelism shown in the graphical representation may be exploited. In the fully concurrent algorithm there are 12 distinct computational levels. We define the average execution time of the concurrent floating point arithmetic process to be $T_{a\text{-}con}$. The resultant expression for the time required to calculate the parallel VWE, defined to be $T_{parallel}$ is:

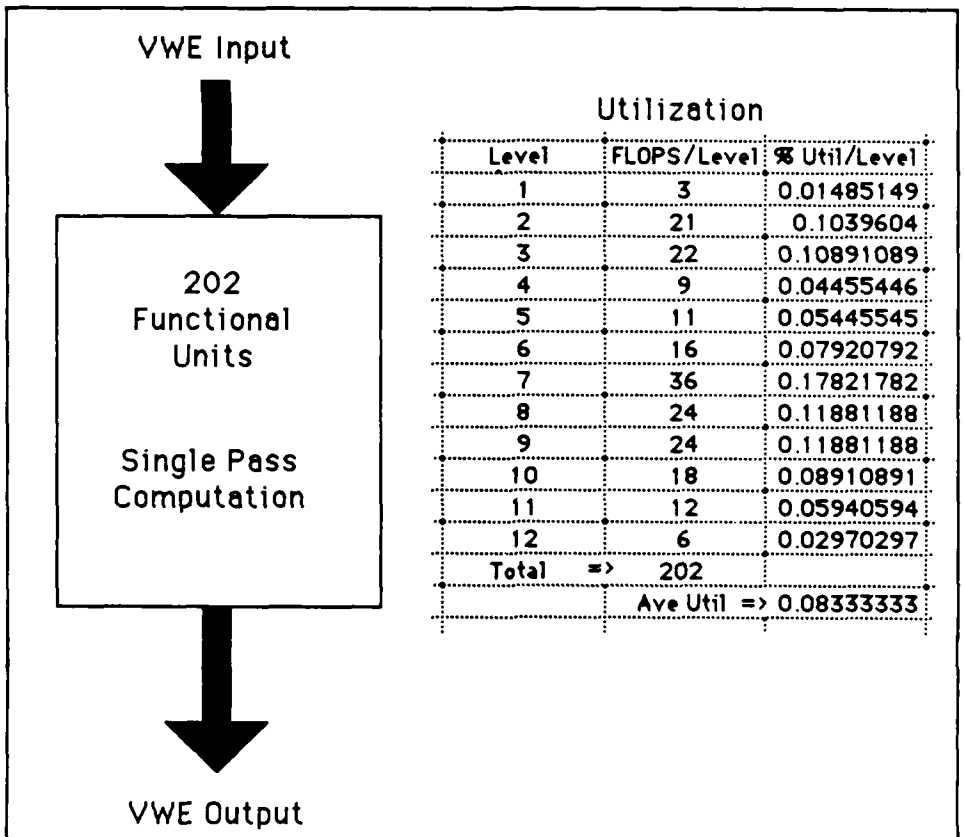
$$T_{parallel} \approx [500 * 12T_{a\text{-}con}] * N.$$

Comparing with $T_{discrete}$ gives:

$$\frac{T_{discrete}}{T_{parallel}} = 49.5 * \frac{T_a}{T_{a\text{-}con}},$$

this speed up of 49.5 can be achieved if the architecture supports concurrent floating point operations at the maximum width of the parallel VWE and the ratio T_a to T_{acon} is ≈ 1. The maximum width is 36. If less than 36 functional units are available, the speedup would be proportionally reduced.

A block diagram for a possible architecture that is defined as the hardware implementation of the fully overlayed data-flow graph of figure 11 is shown below in figure 12. The architecture depicted in Figure 12 is one in which there is a functional unit for each node in the data-flow graph. The functional units are hardwired so that as results become available they are passed to the next unit. Due to the computation's data dependencies, processor functional units at all but the currently active level sit idle. This results in very low component utilization.



Utilization

Level	FLOPS/Level	% Util/Level
1	3	0.01485149
2	21	0.1039604
3	22	0.10891089
4	9	0.04455446
5	11	0.05445545
6	16	0.07920792
7	36	0.17821782
8	24	0.11881188
9	24	0.11881188
10	18	0.08910891
11	12	0.05940594
12	6	0.02970297
Total =>	202	
	Ave Util =>	0.08333333

Figure 12 Block Diagram and Utilization of 202 Funtional Unit Architecture

Further improvements in speedup can be achieved by applying pipelining and other architectural techniques. The limit of these approaches would be a hardwired pipelined architecture capable of producing the complete vector component set every major cycle. A block diagram of this architecture is given in figure 13 below.

The depicted pipelined architecture requires the same 202 functional units as the previous architecture with the addition of the interstage registers and pipeline control hardware. The improvement in system performance is due to a result being available every major cycle (once the pipe is full) as mentioned earlier and a significant increase in component utilization when a large number of dipole elements are to be

calculated. Utilization figures for this architecture are not static as before but are dependent on the number of dipole elements to be input into the pipe. The limit of component utilization approaches 100% for large numbers of dipoles to be calculated and the curve is shown in figure 13.

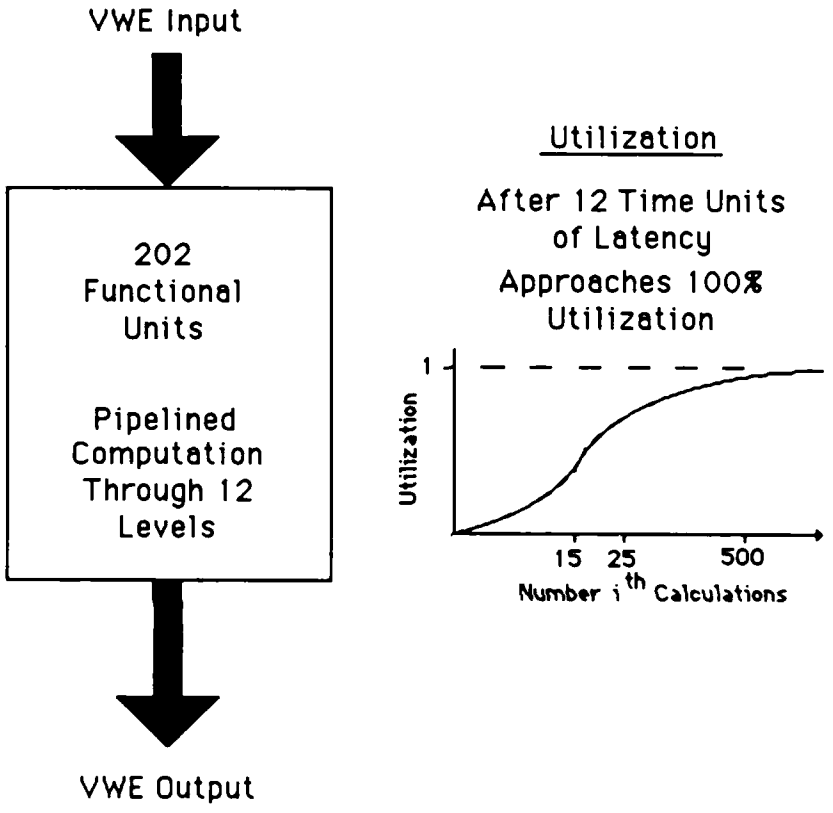


Figure 13 Block Diagram and Utilization of 202 Funtional Unit Pipelined Architecture

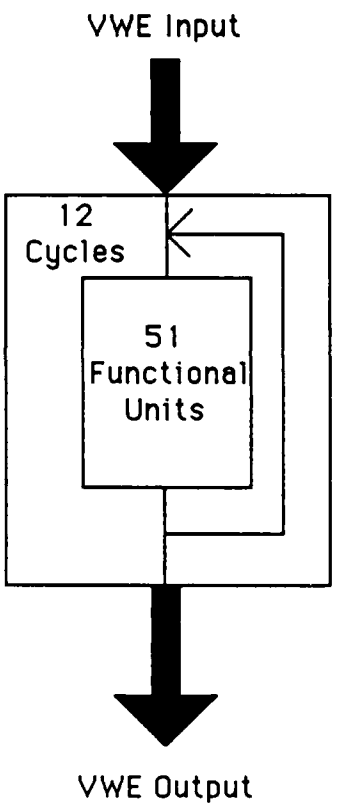
The above architecture's performance can be expressed as follows:

$$T_{piplined} \approx (500 * T_{mc}) * N,$$

where T_{mc} is the time of a major cycle. This expression shows that the speed-up is limited to approx - imately two orders of magnitude for a single processor system.

A means of achieving further speed-up is through the application of multiple processors each capable of computing the VWE component set and taking advantage of the architectural considerations presented thus far.

An architecture capable of supporting 30 floating point multiplies, 18 floating point additions/subtractions, a square root, sin/cos and an inverse or division operation may compute the complete i^{th} VWE calculation in 12 computational levels. This is an architecture with 51 functional units and a block diagram is given in figure 14. These 51 functional units are reconfigurable to implement the various



Utilization

Cycle	FLOPS/Cycle	% Util/cycle
1	3	0.05882353
2	21	0.41176471
3	22	0.43137255
4	9	0.17647059
5	11	0.21568627
6	16	0.31372549
7	36	0.70588235
8	24	0.47058824
9	24	0.47058824
10	18	0.35294118
11	12	0.23529412
12	6	0.11764706
Total =>	202	
	Ave Util =>	0.33006536

Figure 14 Block Diagram and Utilization of 51 Funtional Unit Architecture

computational levels specified in figure 11 and hardwired in the previous architectures. This architecture will require extensive use of registers and control circuitry that will increase complexity and area con -

siderations. However, because of the significantly reduced number of functional units, component util - ization factors are improved as shown in figure 14. Implementation of this architecture utilizing VHSIC technology and possible wafer scale integration, yields a processing engine that may be incorporated as the processor of a multiprocessing array system.

The capabilities of the above system is shown in the definition of $T_{concurrent}$ below:

$$T_{concurrent} \approx \frac{[500 * 12T_{vhsic}] * N}{\text{Number of Processors}},$$

where T_{vhsic} is the execution time of the average arithmetic operation on the VHSIC processor engine.

Comparing an array of the above processors to the classical single processor architecture mentioned previously results in a speedup (S) of:

$$S = 49.5 * \frac{T_a}{T_{vhsic}} * \text{Number of Processors}.$$

We now seek an expression that bounds S in terms of the number of processors used in the array. A lower bound given by assuming one processor and that the ratio $T_a/T_{vhsic} \approx 1$. That is, a VHSIC implementation of the functional units are of the same order of speed as the average execution time of the classic architecture. The resultant expression is:

$$49.5 \leq S \leq 49.5 * \frac{T_a}{T_{vhsic}} * \text{Number of Processors}.$$

The upper bound makes no assumption as to T_a/T_{vhsic} or number of processors. Considering a conservative lower limit of 2 for T_a/T_{vhsic} and rounding the scaler on the right hand side of the inequality to 50 we can express the equation as:

$$49.5 \leq S \leq 100 * \text{Number of Processors}.$$

To achieve a desired 5 orders of magnitude improvement, the above expression indicates that an array of 1000 VHSIC processor engines are required.

A realistic evaluation of an aerodynamic structure may easily require 1 million dipole elements. Using the discrete approach on a Cray 1, this computation would require 8.25 hours assuming that the 100 MFLOP rate could be uniformly sustained. Even if the parallel algorithm's results are utilized to remove all redundant operations, the computation would still require 2.8 hours. This result clearly does not support an interactive environment. However, the computation for these same 1 million dipole elements would require only 0.6 seconds on the 1000 processor array as described above. This is a processing rate capable of supporting an interactive environment.

The performance of the various architectures discussed above are compared with respect to time in figure 15 below. For this comparison, the specific architecture time variables are all assumed to be 10-ns. The effect of this generalization is nominal with respect to the magnitudes calculated below.

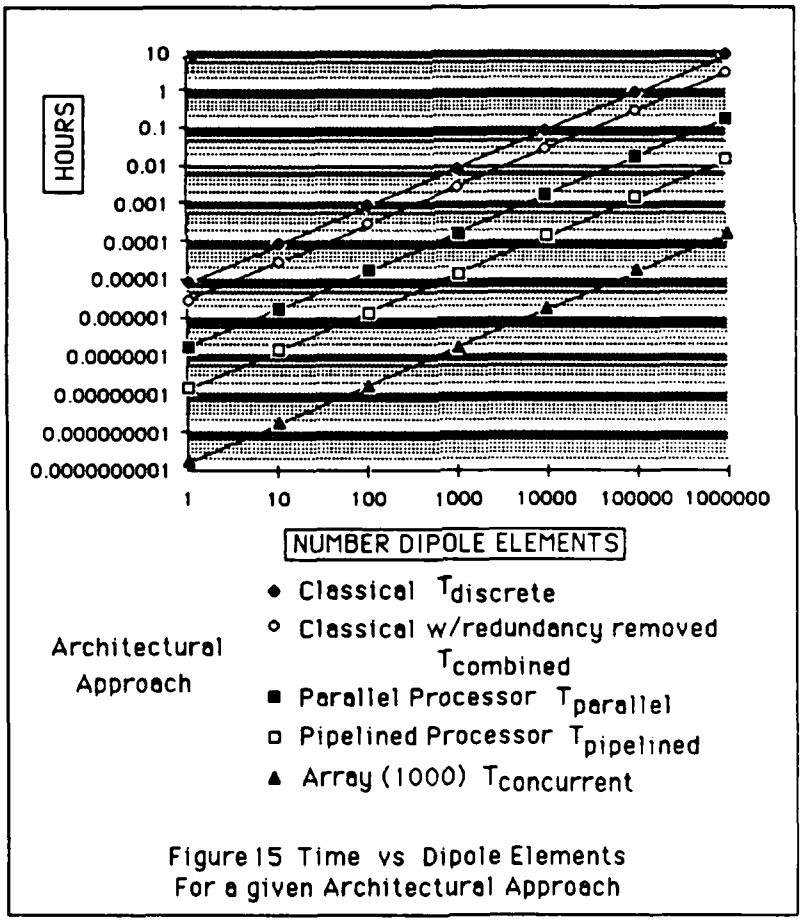


Figure 15 Time vs Dipole Elements
For a given Architectural Approach

V. Conclusions and Recommendations

This study was motivated by the the end objective of developing a CAD/CAM environment for aerodynamic systems design. In particular, the capability of having a structure's electromagnetic observ - ability known and interactivly updated as the design progresses. Current analytical approaches require calculation of near-field electromagnetic wave equations. Numerical solutions to these equations is a computationally intensive problem that requires millions of floating point calculations.

This thesis represents the study, development, and analysis of a parallel computational ap - proach to the set of electromagnetic field equations defined as the Vector Wave Equations (VWE). This effort has produced an initial specification for a parallel processing engine for the VWE that can be imple - mented in either software or hardware. The analysis points excitingly to a conclusion that an interactive environment with respect to electromagnetic metrics is attainable. The cost of this capability is a complex array of VHSIC class processors. Further cost must be considered to address the following issues.

The perspective of this development effort was that a model of the structure existed in a form com - patible with the cartesian representation of the VWE. As such, the issue of creating that model must be ad - dressed. That task will likely be of a similar magnitude as this forward processing task has been. The issue of the model with respect to it's shear size must be grappled with. Finally, the grim task of I/O handling and post processing must be completed in order to attain a truly interactive design environment.

Other, possibly more attractive, issues are those of alternate applications for the VWE. An example might be the study of it's utility in long lead, high frequency layout analysis where the wave length of the frequency is a significant fraction of the conduction path. Further, the applicability of these results to a general class of related partial differential equations may make finite element calculations practical for heat flow, fluid dynamics, and other "vector intensive" areas of study.

Final recommendations for follow-up research with respect to this thesis are in the areas of empirical verification of the computational time savings and hardware implementation. It was stressed throughout this text that, although the objective was a hardware implementation, the algorithmic

specification of the data-flow graphs should be coded and tested in software. This area would include development of a VWE simulator at various levels of computational concurrency and testing the algorithms performance on systems such as the Cray. Further software projects should include a simulator of the VWE with the allowable elementary functions parameterized by function resolution and word size. This project will give empirical visibility into the algorithms performance with respect to accuracy considerations.

Hardware implementation issues include time-area analysis, optimal scheduling and utilization research, and VHSIC technology concerns. The form of the Parallel VWE, as developed here, may lead to implementation by wafer scale integration. Further investigation into alternate architectures combined with the resolution of some of the above issues may likely be the path to attaining a very "state-of-the-art" goal of an interactive design environment.

Appendix A

This appendix provides the supplementary material for the development of the Parallel VWE. Presented here is the full development process of the three vector quantities (**A**, **H**, and **E**) in MACSYMA syntax. The text is organized into three sections. The sections are the development of the Parallel Vector Potential, the Parallel Magnetic Field, and the Parallel Electric Field. The individual sections are further divided into the presentation of the computational flow which is the development of the parallel electromagnetic quantity and the command listing which implements that process. These command listings are what is mapped into the data-flow graphs. The data-flow graphs are presented in Appendix B.

Vector Potential Development

This section illustrates the calculation analysis of the Vector Potential. The following pages will show the step by step procedure of the computation of the Parallel Vector Potential.

The vector potential in the "x" direction will be the example through level four with the complete complex six-tuple shown for the remainder of the process. The text is in MACSYMA syntax and presented as the user might see it on the screen.

Ax

```
(c22) ax,
```

$$(d22) \qquad \frac{(jxri + \%i\ jxii)\ \%e^{-\%i\ b\ r}}{4\ \%pi\ r}$$

realpart(ax) The real part is extracted with MACSYMA.

```
(c23) axr;

                                jxii sin(b r) + jxri cos(b r)
(d23)                           -----------------------------
                                        4 %pi r
```

imagpart(ax) The imaginary part is extracted with MACSYMA.

```
(c24) axi;

                                jxii cos(b r) - jxri sin(b r)
(d24)                           -----------------------------
                                        4 %pi r
```

Here we begin the computation process:
The example below recalls the real part of Axr, then substitutes the expanded version of "r" into the expression.

Axr0; Substitute the expanded version of "r" into the expression.

```
(c25) axr0;

                                                  2          2          2
(d25)           (jxii sin(b sqrt((z - zi)  + (y - yi)  + (x - xi)  ))

                                                2          2          2
                + jxri cos(b sqrt((z - zi)  + (y - yi)  + (x - xi)  )))

                                                2          2          2
                /(4 %pi sqrt((z - zi)  + (y - yi)  + (x - xi)  ))
```

Axi0; Substitute the expanded version of "r" into the expression.

```
(c26) axi0;

                                                  2          2          2
(d26)           (jxii cos(b sqrt((z - zi)  + (y - yi)  + (x - xi)  ))

                                                2          2          2
                - jxri sin(b sqrt((z - zi)  + (y - yi)  + (x - xi)  )))

                                                2          2          2
                /(4 %pi sqrt((z - zi)  + (y - yi)  + (x - xi)  ))
```

After the first level of computation:

Axr1;

```
(c27) axr1;
```

$$(d27)\quad \frac{jxii\ \sin\left(b\ \mathrm{sqrt}\left(A3^2 + A2^2 + A1^2\right)\right) + jxri\ \cos\left(b\ \mathrm{sqrt}\left(A3^2 + A2^2 + A1^2\right)\right)}{4\ \%pi\ \mathrm{sqrt}\left(A3^2 + A2^2 + A1^2\right)}$$

Axi1;

```
(c28) axi1;
```

$$(d28)\quad \frac{jxii\ \cos\left(b\ \mathrm{sqrt}\left(A3^2 + A2^2 + A1^2\right)\right) - jxri\ \sin\left(b\ \mathrm{sqrt}\left(A3^2 + A2^2 + A1^2\right)\right)}{4\ \%pi\ \mathrm{sqrt}\left(A3^2 + A2^2 + A1^2\right)}$$

After the second level of computation:

Axr2;

```
(c35) axr2;
```

$$(d35)\quad \frac{jxii\ \sin\left(b\ \mathrm{sqrt}(B3 + B2 + B1)\right) + jxri\ \cos\left(b\ \mathrm{sqrt}(B3 + B2 + B1)\right)}{4\ \%pi\ \mathrm{sqrt}(B3 + B2 + B1)}$$

Axi2;

```
(c47) axi2;
```

$$(d47)\quad \frac{jxii\ \cos\left(b\ \mathrm{sqrt}(B3 + B2 + B1)\right) - jxri\ \sin\left(b\ \mathrm{sqrt}(B3 + B2 + B1)\right)}{4\ \%pi\ \mathrm{sqrt}(B3 + B2 + B1)}$$

After the third level of computation:

Axr3;

(c38) axr3;

$$(d38)\qquad \frac{jxii\ \sin(b\ \mathrm{sqrt}(C1)) + jxri\ \cos(b\ \mathrm{sqrt}(C1))}{4\ \%pi\ \mathrm{sqrt}(C1)}$$

Axi3;

(c39) axi3;

$$(d39)\qquad \frac{jxii\ \cos(b\ \mathrm{sqrt}(C1)) - jxri\ \sin(b\ \mathrm{sqrt}(C1))}{4\ \%pi\ \mathrm{sqrt}(C1)}$$

After the forth level of computation:

Axr4;

(c66) axr4;

$$(d66)\qquad \frac{jxii\ \sin(b\ D1) + jxri\ \cos(b\ D1)}{4\ \%pi\ D1}$$

Axi4;

(c67) axi4;

$$(d67)\qquad \frac{jxii\ \cos(b\ D1) - jxri\ \sin(b\ D1)}{4\ \%pi\ D1}$$

After the fifth level of computation:

Axr5;

(c98) axr5;

$$(d98) \qquad \frac{jxii\ \sin(E1) + jxri\ \cos(E1)}{E2}$$

Axi5;

(c99) axi5;

$$(d99) \qquad \frac{jxii\ \cos(E1) - jxri\ \sin(E1)}{E2}$$

Ayr5;

(c100) ayr5;

$$(d100) \qquad \frac{jyii\ \sin(E1) + jyri\ \cos(E1)}{E2}$$

Ayi5;

(c101) ayi5;

$$(d101) \qquad \frac{jyii\ \cos(E1) - jyri\ \sin(E1)}{E2}$$

Azr5;

(c102) azr5;

$$(d102) \qquad \frac{jzii\ \sin(E1) + jzri\ \cos(E1)}{E2}$$

Azi5;

(c103) azi5;

$$(d103) \qquad \frac{jzii\ \cos(E1) - jzri\ \sin(E1)}{E2}$$

After the sixth level of computation:

Axr6;

(c140) axr6;

(d140) (jxii F2 + jxri F1) F3

Axi6;

(c141) axi6;

(d141) (jxii F1 - jxri F2) F3

Ayr6;

(c142) ayr6;

(d142) (jyii F2 + jyri F1) F3

Ayi6;

(c143) ayi6;

(d143) (jyii F1 - jyri F2) F3

Azr6;

(c144) azr6;

(d144) (jzii F2 + jzri F1) F3

Azi6;

(c145) azi6;

(d145) (jzii F1 - jzri F2) F3

After the seventh level of computation:

Axr7;

(c203) axr7;

(d203) F3 (G4 + G1)

Axi7;

(c204) axi7;

(d204) F3 (G3 - G2)

Ayr7;

(c205) ayr7;

(d205) F3 (G8 + G5)

Ayi7;

(c206) ayi7;

(d206) F3 (G7 - G6)

Azr7;

(c207) azr7;

(d207) F3 (G9 + G12)

Azi7;

(c208) azi7;

(d208) F3 (G11 - G10)

After the eighth level of computation:

Axr8;

(c234) axr8;

(d234) F3 H1

Axi8;

(c235) axi8;

(d235) F3 H2

Ayr8;

(c236) ayr8;

(d236) F3 H3

Ayi8;

(c237) ayi8;

(d237) F3 H4

Azr8;

(c238) azr8;

(d238) F3 H5

Azi8;

(c239) azi8;

(d239) F3 H6

After the ninth and final level of computation:

Axr9;

(c265) axr9;

(d265) J1

Axi9;

(c266) axi9;

(d266) J2

Ayr9;

(c267) ayr9;

(d267) J3

Ayi9;

(c268) ayi9;

(d268) J4

Azr9;

(c269) azr9;

(d269) J5

Azi9;

(c271) azi9;

(d271) J6

MACSYMA Command Listing for Parallel Vector Potential

The text below is the actual commands used to perform the calculation analysis for the Vector Potential. The commands are separated into the computational levels developed in the previous text.

These first commands are in the preparation stage of the process.

```
realpart(ax);
imagpart(ax);
subst(((((x-xi)^2)+((y-yi)^2)+((z-zi)^2))^(1/2)),r,%);
```

Level one:

```
subst("A1",(x-xi),%)$
subst("A2",(y-yi),%)$
subst("A3",(z-zi),%)$
```

Level two:

```
subst("B1","A1"^2,%)$
subst("B2","A2"^2,%)$
subst("B3","A3"^2,%)$
```

Level three:

```
subst("C1","B1"+"B2"+"B3",%);
```

Level four:

```
subst("D1",sqrt("C1"),%);
```

Level five:

```
subst("E1",b*"D1",%)$
subst("E2",(4*%pi*"D1"),%);
```

Level six:

```
subst("F1",cos("E1"),%)$
subst("F2",sin("E1"),%)$
subst("F3",1/"E2",%);
```

Level seven:

```
subst("G1",jxri*"F1",%)$
subst("G2",jxri*"F2",%)$
subst("G3",jxii*"F1",%)$
subst("G4",jxii*"F2",%);

subst("G5",jyri*"F1",%)$
subst("G6",jyri*"F2",%)$
subst("G7",jyii*"F1",%)$
subst("G8",jyii*"F2",%);

subst("G9",jzri*"F1",%)$
subst("G10",jzri*"F2",%)$
subst("G11",jzii*"F1",%)$
subst("G12",jzii*"F2",%);
```

Level eight:

```
subst("H1","G1"+"G4",%);
subst("H2","G3"-"G2",%);
subst("H3","G5"+"G8",%);
subst("H4","G7"-"G6",%);
subst("H5","G9"+"G12",%);
subst("H6","G11"-"G10",%);
```

Level nine:

```
subst("J1","F3"*"H1",%);
subst("J2","F3"*"H2",%);
subst("J3","F3"*"H3",%);
subst("J4","F3"*"H4",%);
subst("J5","F3"*"H5",%);
subst("J6","F3"*"H6",%);
```

Magnetic Field Development

This section illustrates the calculation analysis of the Magnetic Field. The following pages will show the step by step procedure of the computation of the Parallel Magnetic Field.

The magnetic field in the "x" direction will be the example through level one with the complete complex six-tuple shown for the remainder of the process. The text is in MACSYMA syntax and presented as the user might see it on the screen.

Hx:

$$\frac{\left(-\frac{1}{r}+\%i\,b\right)\%e^{-\%i\,b\,r}\left((jyri+\%i\,jyii)(z-zi)-(jzri+\%i\,jzii)(y-yi)\right)}{4\,\%pi\,r^2}$$

realpart(hx) The real part is extracted with MACSYMA.

```
((cos(b r) (jyri (z - zi) - jzri (y - yi))
 + sin(b r) (jyii (z - zi) - jzii (y - yi)))/r
 - b (cos(b r) (jyii (z - zi) - jzii (y - yi))
                                                    2
 - sin(b r) (jyri (z - zi) - jzri (y - yi))))/(4 %pi r )
```

imagpart(hx) The imaginary part is extracted with MACSYMA.

```
((cos(b r) (jyii (z - zi) - jzii (y - yi))
 - sin(b r) (jyri (z - zi) - jzri (y - yi)))/r
 + b (cos(b r) (jyri (z - zi) - jzri (y - yi))
                                                    2
 + sin(b r) (jyii (z - zi) - jzii (y - yi))))/(4 %pi r )
```

Here we begin the computation process

The example below recalls the real part of Hx, then substitutes the expanded version of "r" into the expression.

realpart(hx);

hxr;

```
((cos(b r) (jyri (z - zi) - jzri (y - yi))

+ sin(b r) (jyii (z - zi) - jzii (y - yi)))/r

- b (cos(b r) (jyii (z - zi) - jzii (y - yi))

                                                    2
- sin(b r) (jyri (z - zi) - jzri (y - yi))))/(4 %pi r )
```

subst(((((x-xi)^2)+((y-yi)^2)+((z-zi)^2))^(1/2)),r,%);

HxrO;

```
                                2          2          2
((cos(b sqrt((z - zi)  + (y - yi)  + (x - xi) ))

(jyri (z - zi) - jzri (y - yi)) + sin(b

           2          2          2
sqrt((z - zi)  + (y - yi)  + (x - xi) )) (jyii (z - zi) - jzii (y - yi)))

            2          2          2
/sqrt((z - zi)  + (y - yi)  + (x - xi) )

                          2          2          2
- b (cos(b sqrt((z - zi)  + (y - yi)  + (x - xi) ))

(jyii (z - zi) - jzii (y - yi)) - sin(b

           2          2          2
sqrt((z - zi)  + (y - yi)  + (x - xi) )) (jyri (z - zi) - jzri (y - yi))))

               2          2          2
/(4 %pi ((z - zi)  + (y - yi)  + (x - xi) ))
```

After the first level of computation the real part of the X directed Magnetic Field has the form

hxr1;

$$\begin{aligned}
&(((jyii\ A3 - jzii\ A2)\sin(b\ sqrt(A3^2 + A2^2 + A1^2)) \\
&+ (jyri\ A3 - jzri\ A2)\cos(b\ sqrt(A3^2 + A2^2 + A1^2)))/sqrt(A3^2 + A2^2 + A1^2) \\
&- b((jyii\ A3 - jzii\ A2)\cos(b\ sqrt(A3^2 + A2^2 + A1^2)) \\
&- (jyri\ A3 - jzri\ A2)\sin(b\ sqrt(A3^2 + A2^2 + A1^2))))/(4\ \%pi\ (A3^2 + A2^2 + A1^2))
\end{aligned}$$

After the second level of computation:

hxr2;

$$\begin{aligned}
&(B15 - B20)\sin(b\ sqrt(B3 + B2 + B1)) + (B12 - B17)\cos(b\ sqrt(B3 + B2 + B1)) \\
&(\text{---} \\
&\qquad sqrt(B3 + B2 + B1) \\
&- b((B15 - B20)\cos(b\ sqrt(B3 + B2 + B1)) \\
&- (B12 - B17)\sin(b\ sqrt(B3 + B2 + B1))))/(4\ \%pi\ (B3 + B2 + B1))
\end{aligned}$$

hxi2;

$$\begin{aligned}
&(b((B15 - B20)\sin(b\ sqrt(B3 + B2 + B1)) \\
&+ (B12 - B17)\cos(b\ sqrt(B3 + B2 + B1))) \\
&+ ((B15 - B20)\cos(b\ sqrt(B3 + B2 + B1)) \\
&- (B12 - B17)\sin(b\ sqrt(B3 + B2 + B1)))/sqrt(B3 + B2 + B1)) \\
&/(4\ \%pi\ (B3 + B2 + B1))
\end{aligned}$$

hyr2;

```
 sin(b sqrt(B3 + B2 + B1)) (B19 - B9) + cos(b sqrt(B3 + B2 + B1)) (B16 - B6)
(-------------------------------------------------------------------------
                        sqrt(B3 + B2 + B1)

 - b (cos(b sqrt(B3 + B2 + B1)) (B19 - B9)

 - sin(b sqrt(B3 + B2 + B1)) (B16 - B6)))/(4 %pi (B3 + B2 + B1))
```

hyi2;

```
(b (sin(b sqrt(B3 + B2 + B1)) (B19 - B9)

 + cos(b sqrt(B3 + B2 + B1)) (B16 - B6))

 + (cos(b sqrt(B3 + B2 + B1)) (B19 - B9)

 - sin(b sqrt(B3 + B2 + B1)) (B16 - B6))/sqrt(B3 + B2 + B1))

/(4 %pi (B3 + B2 + B1))
```

hzr2;

```
 sin(b sqrt(B3 + B2 + B1)) (B8 - B13) + cos(b sqrt(B3 + B2 + B1)) (B5 - B10)
(-------------------------------------------------------------------------
                        sqrt(B3 + B2 + B1)

 - b (cos(b sqrt(B3 + B2 + B1)) (B8 - B13)

 - sin(b sqrt(B3 + B2 + B1)) (B5 - B10)))/(4 %pi (B3 + B2 + B1))
```

hzi2

```
(b (sin(b sqrt(B3 + B2 + B1)) (B8 - B13)

 + cos(b sqrt(B3 + B2 + B1)) (B5 - B10))

 + (cos(b sqrt(B3 + B2 + B1)) (B8 - B13)

 - sin(b sqrt(B3 + B2 + B1)) (B5 - B10))/sqrt(B3 + B2 + B1))

/(4 %pi (B3 + B2 + B1))
```

After the third level of computation:

hxr3;

$$\frac{\cos(b\sqrt{C1})\,C3 + \sin(b\sqrt{C1})\,C2}{\sqrt{C1}} - b\,(\cos(b\sqrt{C1})\,C2 - \sin(b\sqrt{C1})\,C3)/(4\,\%pi\,C1)$$

hxi3;

$$\frac{\cos(b\sqrt{C1})\,C2 - \sin(b\sqrt{C1})\,C3}{\sqrt{C1}} + b\,(\cos(b\sqrt{C1})\,C3 + \sin(b\sqrt{C1})\,C2)/(4\,\%pi\,C1)$$

hyr3;

$$\frac{\cos(b\sqrt{C1})\,C5 + \sin(b\sqrt{C1})\,C4}{\sqrt{C1}} - b\,(\cos(b\sqrt{C1})\,C4 - \sin(b\sqrt{C1})\,C5)/(4\,\%pi\,C1)$$

hyi3;

$$\frac{\cos(b\sqrt{C1})\,C4 - \sin(b\sqrt{C1})\,C5}{\sqrt{C1}} + b\,(\cos(b\sqrt{C1})\,C5 + \sin(b\sqrt{C1})\,C4)/(4\,\%pi\,C1)$$

hzr3;

$$\frac{\cos(b\sqrt{C1})\,C7 + \sin(b\sqrt{C1})\,C6}{\sqrt{C1}} - b\,(\cos(b\sqrt{C1})\,C6 - \sin(b\sqrt{C1})\,C7)/(4\,\%pi\,C1)$$

hzi3;

$$\frac{\cos(b\sqrt{C1})\,C6 - \sin(b\sqrt{C1})\,C7}{\sqrt{C1}} + b\,(\cos(b\sqrt{C1})\,C7 + \sin(b\sqrt{C1})\,C6)/(4\,\%pi\,C1)$$

After the forth level of computation:

hxr4;

$$\frac{\frac{C2\sin(b\,D1) + C3\cos(b\,D1)}{D1} - b\,(C2\cos(b\,D1) - C3\sin(b\,D1))}{D2}$$

hxi4;

$$\frac{\frac{C2\cos(b\,D1) - C3\sin(b\,D1)}{D1} + b\,(C2\sin(b\,D1) + C3\cos(b\,D1))}{D2}$$

hyr4;

$$\frac{\frac{C4\sin(b\,D1) + C5\cos(b\,D1)}{D1} - b\,(C4\cos(b\,D1) - C5\sin(b\,D1))}{D2}$$

hyi4;

$$\frac{\frac{C4\cos(b\,D1) - C5\sin(b\,D1)}{D1} + b\,(C4\sin(b\,D1) + C5\cos(b\,D1))}{D2}$$

hzr4;

$$\frac{\frac{C6\sin(b\,D1) + C7\cos(b\,D1)}{D1} - b\,(C6\cos(b\,D1) - C7\sin(b\,D1))}{D2}$$

hzi4;

$$\frac{\frac{C6\cos(b\,D1) - C7\sin(b\,D1)}{D1} + b\,(C6\sin(b\,D1) + C7\cos(b\,D1))}{D2}$$

After the fifth level of computation:

hxr5;

((C2 sin(E1) + C3 cos(E1)) E2 - b (C2 cos(E1) - C3 sin(E1))) E3

hxi5;

((C2 cos(E1) - C3 sin(E1)) E2 + b (C2 sin(E1) + C3 cos(E1))) E3

hyr5;

((C4 sin(E1) + C5 cos(E1)) E2 - b (C4 cos(E1) - C5 sin(E1))) E3

hyi5;

((C4 cos(E1) - C5 sin(E1)) E2 + b (C4 sin(E1) + C5 cos(E1))) E3

hzr5;

((C6 sin(E1) + C7 cos(E1)) E2 - b (C6 cos(E1) - C7 sin(E1))) E3

hzi5;

((C6 cos(E1) - C7 sin(E1)) E2 + b (C6 sin(E1) + C7 cos(E1))) E3

After the sixth level of computation:

hxr6;

E3 (E2 (C2 F2 + C3 F1) - b (C2 F1 - C3 F2))

hxi6;

E3 (E2 (C2 F1 - C3 F2) + b (C2 F2 + C3 F1))

hyr6;

E3 (E2 (C4 F2 + C5 F1) - b (C4 F1 - C5 F2))

hyi6;

E3 (E2 (C4 F1 - C5 F2) + b (C4 F2 + C5 F1))

hzr6;

E3 (E2 (C6 F2 + C7 F1) - b (C6 F1 - C7 F2))

hzi6;

E3 (E2 (C6 F1 - C7 F2) + b (C6 F2 + C7 F1))

After the seventh level of computation:

hxr7;

E3 (E2 (G7 + G2) - b (G1 - G8))

hxi7;

E3 (E2 (G1 - G8) + b (G7 + G2))

hyr7;

E3 (E2 (G9 + G4) - b (G3 - G10))

hyi7;

E3 (b (G9 + G4) + E2 (G3 - G10))

hzr7;

E3 (E2 (G6 + G11) - b (G5 - G12))

hzi7;

E3 (b (G6 + G11) + E2 (G5 - G12))

After the eighth level of computation:

hxr8;

E3 (E2 H1 - b H4)

hxi8;

E3 (E2 H4 + b H1)

hyr8;

E3 (E2 H2 - b H5)

hyi8;

E3 (E2 H5 + b H2)

hzr8;

E3 (E2 H3 - b H6)

hzi8;

E3 (E2 H6 + b H3)

After the ninth level of computation:

hxr9;

E3 (J1 - J10)

hxi9;

E3 (J7 + J4)

hyr9;

E3 (J2 - J11)

hyi9;

E3 (J8 + J5)

hzr9;

E3(J3 - J12)

hzi9;

E3 (J9 + J6)

After the tenth level of computation:

hxr10;

E3 K1

hxi10;

E3 K4

hyr10;

E3 K2

hyi10;

E3 K5

hzr10;

E3 K3

hzi10;

E3 K6

After the eleventh and final level of computation:

hxr11;

L1 Hx real

hxi11;

L4 Hx imag

hyr11;

L2 Hy real

hyi11;

L5 Hy imag

hzr11;

L3 Hz real

hzi11;

L6 Hy imag

MACSYMA Command Listing for Parallel Magnetic Field

The text below is the actual commands used to perform the calculation analysis for the Magnetic Field. The commands are separated into the computational levels developed in the previous text.

These first commands are in the preparation stage of the process.

```
realpart(hx);
imagpart(hx);
subst(((((x-xi)^2)+((y-yi)^2)+((z-zi)^2))^(1/2)),r,%);
```

Level one:

```
subst("A1",(x-xi),%);
subst("A2",(y-yi),%);
subst("A3",(z-zi),%);
```

Level two

```
subst("B1","A1"^2,%);
subst("B2","A2"^2,%);
subst("B3","A3"^2,%);

subst("B4",jxri*"A1",%);          not used
subst("B5",jxri*"A2",%);
subst("B6",jxri*"A3",%);
subst("B7",jxii*"A1",%);          not used
subst("B8",jxii*"A2",%);
subst("B9",jxii*"A3",%);
subst("B10",jyri*"A1",%);
subst("B11",jyri*"A2",%);         not used
subst("B12",jyri*"A3",%);
subst("B13",jyii*"A1",%);
subst("B14",jyii*"A2",%);         not used
subst("B15",jyii*"A3",%);
subst("B16",jzri*"A1",%);
subst("B17",jzri*"A2",%);
subst("B18",jzri*"A3",%);         not used
subst("B19",jzii*"A1",%);
subst("B20",jzii*"A2",%);
subst("B21",jzii*"A3",%);         not used
```

Level three

```
subst("C1","B1"+"B2"+"B3",%);
subst("C2","B15"-"B20",%);
subst("C3","B12"-"B17",%);
subst("C4","B19"-"B9",%);
subst("C5","B16"-"B6",%);
subst("C6","B8"-"B13",%);
subst("C7","B5"-"B10",%);
```

Level four

```
subst("D1",sqrt("C1"),%);
subst("D2",4*%pi*"C1",%);
```

Level five

```
subst("E1",b*"D1",%);
subst("E2",(1/"D1"),%);
subst("E3",(1/"D2"),%);
```

Level six

```
subst("F1",cos("E1"),%),
subst("F2",sin("E1"),%);
```

Level seven

```
subst("G1","C2"*"F1",%);
subst("G2","C3"*"F1",%);
subst("G3","C4"*"F1",%);
subst("G4","C5"*"F1",%);
subst("G5","C6"*"F1",%);
subst("G6","C7"*"F1",%);
subst("G7","C2"*"F2",%);
subst("G8","C3"*"F2",%);
subst("G9","C4"*"F2",%);
subst("G10","C5"*"F2",%);
subst("G11","C6"*"F2",%);
subst("G12","C7"*"F2",%);
```

Level eight

```
subst("H1","G7"+"G2",%);
subst("H2","G9"+"G4",%);
subst("H3","G6"+"G11",%),
subst("H4","G1"-"G8",%);
subst("H5","G3"-"G10",%),
subst("H6","G5"-"G12",%);
```

Level nine

```
subst("J1","E2"*"H1",%);
subst("J2","E2"*"H2",%);
subst("J3","E2"*"H3",%);
subst("J4","E2"*"H4",%);
subst("J5","E2"*"H5",%);
subst("J6","E2"*"H6",%);
subst("J7",b*"H1",%);
subst("J8",b*"H2",%);
subst("J9",b*"H3",%);
subst("J10",b*"H4",%);
subst("J11",b*"H5",%);
subst("J12",b*"H6",%);
```

Level ten

```
subst("K1","J1"-"J10",%);
subst("K2","J2"-"J11",%);
subst("K3","J3"-"J12",%);
subst("K4","J7"+"J4",%);
subst("K5","J8"+"J5",%);
subst("K6","J9"+"J6",%);
```

Level eleven

```
subst("L1","E3"*"K1",%);
subst("L2","E3"*"K2",%);
subst("L3","E3"*"K3",%);

subst("L4","E3"*"K4",%);
subst("L5","E3"*"K5",%);
subst("L6","E3"*"K6",%);
```

Electric Field Development

This section illustrates the calculation analysis of the Electric Field. The following pages will show the step by step procedure of the computation of the Parallel Electric Field.

The Electric Field in the "x" direction will be the example through level four with the complete complex six-tuple shown for the remainder of the process. The text is in MACSYMA syntax and presented as the user might see it on the screen.

Ex;

```
(c9) ex;

       - %i b r                          3 %i b   1    2          2          2
(d9) %e         ((jxri + %i jxii) (- ------------ - -- + b ) ((z - zi)  + (y - yi) )
                                        r         2
                                                  r

    3 %i b    3   2
 + (------ + -- - b ) ((jzri + %i jzii) (x - xi) (z - zi)
       r     2
             r

                                                      3
 + (jyri + %i jyii) (x - xi) (y - yi)))/(4 %pi r )
```

realpart(ex) The real part is extracted with MACSYMA.

```
(c10) exr;
                                2  1    3 b jxri              2         2
(d10) (sin(b r) ((jxii (b  - --) - --------) ((z - zi)  + (y - yi) )
                                  2     r
                                 r

      3 b (jzri (x - xi) (z - zi) + jyri (x - xi) (y - yi))
 + ----------------------------------------------------
                              r
   3   2
 + (-- - b ) (jzii (x - xi) (z - zi) + jyii (x - xi) (y - yi)))
     2
    r
             3 b jxii          2  1            2          2
 + cos(b r) ((-------- + jxri (b  - --)) ((z - zi)  + (y - yi) )
               r                  2
                                 r
   3   2
 + (-- - b ) (jzri (x - xi) (z - zi) + jyri (x - xi) (y - yi))
     2
    r
       3 b (jzii (x - xi) (z - zi) + jyii (x - xi) (y - yi))                  3
 - ----------------------------------------------------))/(4 %pi r )
                              r
```

imagpart(ex) The imaginary part is extracted with MACSYMA.

```
(c11) exi;
                                2  1    3 b jxri              2         2
(d11) (cos(b r) ((jxii (b  - --) - --------) ((z - zi)  + (y - yi) )
                                  2     r
                                 r

      3 b (jzri (x - xi) (z - zi) + jyri (x - xi) (y - yi))
 + ----------------------------------------------------
                              r
   3   2
 + (-- - b ) (jzii (x - xi) (z - zi) + jyii (x - xi) (y - yi)))
     2
    r
             3 b jxii          2  1            2          2
 - sin(b r) ((-------- + jxri (b  - --)) ((z - zi)  + (y - yi) )
               r                  2
                                 r
   3   2
 + (-- - b ) (jzri (x - xi) (z - zi) + jyri (x - xi) (y - yi))
     2
    r
       3 b (jzii (x - xi) (z - zi) + jyii (x - xi) (y - yi))                  3
 - ----------------------------------------------------))/(4 %pi r  )
                              r
```

Here we begin the computation process:
The example below recalls the real part of Ex, then substitutes the expanded version of "r" into the expression.

Exr0: Substitute the expanded version of "r" into the expression.

```
(c12) exr0;
                                  2          2          2
(d12) (sin(b sqrt((z - zi)  + (y - yi)  + (x - xi)  ))

        3 b (jzri (x - xi) (z - zi) + jyri (x - xi) (y - yi))
 (-------------------------------------------------------
                          2          2          2
              sqrt((z - zi)  + (y - yi)  + (x - xi)  )
          2                   1
 + (jxii (b  - ----------------------------------)
                          2          2          2
                 (z - zi)  + (y - yi)  + (x - xi)
                  3 b jxri                                2          2
 - ----------------------------------------) ((z - zi)  + (y - yi)  )
                    2          2          2
          sqrt((z - zi)  + (y - yi)  + (x - xi) )
                    3                     2
 + (---------------------------------- - b  )
                 2          2          2
          (z - zi)  + (y - yi)  + (x - xi)

 (jzii (x - xi) (z - zi) + jyii (x - xi) (y - yi)))

                          2          2          2
 + cos(b sqrt((z - zi)  + (y - yi)  + (x - xi)  ))

        3 b (jzii (x - xi) (z - zi) + jyii (x - xi) (y - yi))
 (- -------------------------------------------------------
                          2          2          2
              sqrt((z - zi)  + (y - yi)  + (x - xi)  )
                 3 b jxii
 + (---------------------------------------
                    2          2          2
          sqrt((z - zi)  + (y - yi)  + (x - xi)  )
           2                   1                          2          2
 + jxri (b  - ----------------------------------)) ((z - zi)  + (y - yi)  )
                          2          2          2
                 (z - zi)  + (y - yi)  + (x - xi)
                   3                     2
 + (---------------------------------- - b  )
                 2          2          2
          (z - zi)  + (y - yi)  + (x - xi)

                                                                2          2          2  3/2
 (jzri (x - xi) (z - zi) + jyri (x - xi) (y - yi)))) /(4 %pi ((z - zi)  + (y - yi)  + (x - xi)  )      )
```

Ex10; Substitute the expanded version of "r" into the expression.

```
(c13) ex10;

                                  2         2         2
(d13) (cos(b sqrt((z - zi)  + (y - yi)  + (x - xi) ))

        3 b (jzri (x - xi) (z - zi) + jyri (x - xi) (y - yi))
 (-------------------------------------------------------
                          2         2         2
              sqrt((z - zi)  + (y - yi)  + (x - xi) )

         2                    1
 + (jxii (b  - ---------------------------------)
                       2         2         2
                 (z - zi)  + (y - yi)  + (x - xi)
                    3 b jxri                                  2         2
 - ----------------------------------------) ((z - zi)  + (y - yi)  )
                   2         2         2
        sqrt((z - zi)  + (y - yi)  + (x - xi)  )
                   3                        2
 + (--------------------------------- - b  )
              2         2         2
        (z - zi)  + (y - yi)  + (x - xi)

 (jzii (x - xi) (z - zi) + jyii (x - xi) (y - yi)))

                        2         2         2
 - sin(b sqrt((z - zi)  + (y - yi)  + (x - xi)  ))

          3 b (jzii (x - xi) (z - zi) + jyii (x - xi) (y - yi))
 (- -------------------------------------------------------
                          2         2         2
              sqrt((z - zi)  + (y - yi)  + (x - xi)  )
                    3 b jxii
 + (---------------------------------------
                   2         2         2
        sqrt((z - zi)  + (y - yi)  + (x - xi) )

         2                  1                                  2         2
 + jxri (b  - ---------------------------------)) ((z - zi)  + (y - yi)  )
                       2         2         2
                 (z - zi)  + (y - yi)  + (x - xi)

                   3                        2
 + (--------------------------------- - b  )
              2         2         2
        (z - zi)  + (y - yi)  + (x - xi)

 (jzri (x - xi) (z - zi) + jyri (x - xi) (y - yi))))

                    2         2         2  3/2
 /(4 %pi ((z - zi)  + (y - yi)  + (x - xi)  )     )
```

After the first level of computation:

Exr1;

```
(c14) exr1;

                  2     2        2          1                    3 b jxri
(d14) (((A3  + A2 ) (jxii (b  - ----------------) - ----------------------)
                                    2    2    2                2    2    2
                                 A3 + A2 + A1          sqrt(A3 + A2 + A1 )

                                          3          2
 + (jzii A1 A3 + jyii A1 A2) (---------------- - b )
                                        2    2    2
                                     A3 + A2 + A1

   3 b (jzri A1 A3 + jyri A1 A2)                    2    2    2
 + ----------------------------) sin(b sqrt(A3  + A2  + A1  ))
                 2    2    2
          sqrt(A3 + A2 + A1  )

       2     2        2          1                    3 b jxii
 + ((A3  + A2  ) (jxri (b  - ----------------  ) + ----------------------)
                                    2    2    2                2    2    2
                                 A3 + A2 + A1          sqrt(A3 + A2 + A1  )

                                          3          2
 + (jzri A1 A3 + jyri A1 A2) (---------------- - b )
                                        2    2    2
                                     A3 + A2 + A1

   3 b (jzii A1 A3 + jyii A1 A2)                    2    2    2
 - ----------------------------) cos(b sqrt(A3  + A2  + A1 )))
                 2    2    2
          sqrt(A3 + A2 + A1  )

             2    2    2 3/2
 /(4 %pi (A3  + A2  + A1 )    )
```

Exi1;

```
(c15) exi1;

                2    2          2          1                    3 b jxri
(d15) (((A3 + A2  ) (jxii (b - ---------------  ) - --------------------)
                                    2    2    2              2    2    2
                                 A3 + A2 + A1     sqrt(A3 + A2 + A1  )

                                      3           2
 + (jzii A1 A3 + jyii A1 A2) (---------------- - b  )
                                    2    2    2
                                 A3 + A2 + A1

    3 b (jzri A1 A3 + jyri A1 A2)                  2    2    2
 + ----------------------------) cos(b sqrt(A3 + A2 + A1  ))
                    2    2    2
          sqrt(A3 + A2 + A1  )

       2    2          2          1                    3 b jxii
 - ((A3 + A2  ) (jxri (b - ---------------) + --------------------)
                              2    2    2              2    2    2
                           A3 + A2 + A1     sqrt(A3 + A2 + A1  )

                                      3           2
 + (jzri A1 A3 + jyri A1 A2) (---------------- - b  )
                                    2    2    2
                                 A3 + A2 + A1

    3 b (jzii A1 A3 + jyii A1 A2)                  2    2    2
 - ----------------------------) sin(b sqrt(A3 + A2 + A1  )))
                    2    2    2
          sqrt(A3 + A2 + A1  )

             2    2    2  3/2
/(4 %pi (A3 + A2 + A1  )    )
```

After the second level of computation:

Exr2;

(c18) exr2;

$$(d18)\ (((B3+B2)\,(jxii\,(b^2-\frac{1}{B3+B2+B1})-\frac{B22}{\mathrm{sqrt}(B3+B2+B1)})$$
$$+(A3\,B19+A2\,B13)\,(\frac{3}{B3+B2+B1}-b^2)+\frac{3\,b\,(A3\,B16+A2\,B10)}{\mathrm{sqrt}(B3+B2+B1)})$$
$$\sin(b\,\mathrm{sqrt}(B3+B2+B1))+((B3+B2)$$
$$(jxri\,(b^2-\frac{1}{B3+B2+B1})+\frac{B23}{\mathrm{sqrt}(B3+B2+B1)})$$
$$+(A3\,B16+A2\,B10)\,(\frac{3}{B3+B2+B1}-b^2)-\frac{3\,b\,(A3\,B19+A2\,B13)}{\mathrm{sqrt}(B3+B2+B1)})$$
$$\cos(b\,\mathrm{sqrt}(B3+B2+B1)))/(4\,\%pi\,(B3+B2+B1)^{3/2})$$

Exi2;

(c19) exi2;

$$(d19)\ (((B3+B2)\,(jxii\,(b^2-\frac{1}{B3+B2+B1})-\frac{B22}{\mathrm{sqrt}(B3+B2+B1)})$$
$$+(A3\,B19+A2\,B13)\,(\frac{3}{B3+B2+B1}-b^2)+\frac{3\,b\,(A3\,B16+A2\,B10)}{\mathrm{sqrt}(B3+B2+B1)})$$
$$\cos(b\,\mathrm{sqrt}(B3+B2+B1))-((B3+B2)$$
$$(jxri\,(b^2-\frac{1}{B3+B2+B1})+\frac{B23}{\mathrm{sqrt}(B3+B2+B1)})$$
$$+(A3\,B16+A2\,B10)\,(\frac{3}{B3+B2+B1}-b^2)-\frac{3\,b\,(A3\,B19+A2\,B13)}{\mathrm{sqrt}(B3+B2+B1)})$$
$$\sin(b\,\mathrm{sqrt}(B3+B2+B1)))/(4\,\%pi\,(B3+B2+B1)^{3/2})$$

After the third level of computation:

Exr3;

```
(c22) exr3;

                                3 b (C38 + C36)    3    2
(d22) (sin(b sqrt(C1)) (------------------- + (--- - b  ) (C37 + C35)
                                  sqrt(C1)         C1

          2  1     B22
 + (jxii (b  - -- ) - -------- ) C33) + cos(b sqrt(C1))
             C1    sqrt(C1)

   3    2                  3 b (C37 + C35)      B23            2  1
((-- - b  ) (C38 + C36) - ------------------- + (-------- + jxri (b  - -- )) C33))
   C1                        sqrt(C1)          sqrt(C1)           C1

          3/2
/(4 %pi C1    )
```

Exi3;

```
(c23) exi3;

                                3 b (C38 + C36)    3    2
(d23) (cos(b sqrt(C1)) (------------------- + ( --- - b  ) (C37 + C35)
                                  sqrt(C1)         C1

          2  1      B22
 + (jxii (b  - -- ) - --------) C33) - sin(b sqrt(C1))
             C1    sqrt(C1)

   3    2                  3 b (C37 + C35)      B23            2  1
((-- - b  ) (C38 + C36) - ------------------- + (-------- + jxri (b  - -- )) C33))
  C1                         sqrt(C1)          sqrt(C1)           C1

          3/2
/(4 %pi C1    )
```

After the forth level of computation:

Exr4;

```
(c51) exr4;
                                  2          3 b D34                2          B22
(d51) (sin(b D1) ((3 D33 - b  ) D35 + ------- + C33 (jxii (b  - D33) - --- ))
                                        D1                              D1

                 3 b D35              2                          2          B23
 + cos(b D1) (- ------- + (3 D33 - b  ) D34 + C33 (jxri (b  - D33) + --- )))
                   D1                                                 D1

/(D1 D2)
```

Exi4;

```
(c52) exi4;

                                  2          3 b D34                2          B22
(d52) (cos(b D1) ((3 D33 - b  ) D35 + ------- + C33 (jxii (b  - D33) - --- ))
                                        D1                              D1

                 3 b D35              2                          2          B23
 - sin(b D1) (- ------- + (3 D33 - b  ) D34 + C33 (jxri (b  - D33) + --- )))
                   D1                                                 D1

/(D1 D2)
```

After the fifth level of computation:

Exr5;

```
(c53) exr5;

                                                    2
(d53) E2 E3 (cos(E1) (- E2 E7 + D34 (E5 - b  ) + C33 (jxri E4 + B23 E2))

                                                    2
                + sin(E1) (E2 E6 + D35 (E5 - b  ) + C33 (jxii E4 - B22 E2)))
```

Exi5;

```
(c54) exi5,

                                                    2
(d54) E2 E3 (cos(E1) (E2 E6 + D35 (E5 - b  ) + C33 (jxii E4 - B22 E2))

                                                    2
              - sin(E1) (- E2 E7 + D34 (E5 - b  ) + C33 (jxri E4 + B23 E2)))
```

Eyr5;

(c55) eyr5;

$$(d55)\ E2\ E3\ (\cos(E1)\ (-\ E2\ E9 + D36\ (E5 - b^2) + C32\ (jyrI\ E4 + B25\ E2))$$

$$+ \sin(E1)\ (E2\ E8 + D37\ (E5 - b^2) + C32\ (jyii\ E4 - B24\ E2)))$$

Eyi5;

(c57) eyi5;

$$(d57)\ E2\ E3\ (\cos(E1)\ (E2\ E8 + D37\ (E5 - b^2) + C32\ (jyii\ E4 - B24\ E2))$$

$$- \sin(E1)\ (-\ E2\ E9 + D36\ (E5 - b^2) + C32\ (jyri\ E4 + B25\ E2)))$$

Ezr5;

(c58) ezr5;

$$(d58)\ E2\ E3\ (\sin(E1)\ (D39\ (E5 - b^2) + C31\ (jzii\ E4 - B26\ E2) + E10\ E2)$$

$$+ \cos(E1)\ (D38\ (E5 - b^2) + C31\ (jzri\ E4 + B27\ E2) - E11\ E2))$$

Ezi5;

(c59) ezi5;

$$(d59)\ E2\ E3\ (\cos(E1)\ (D39\ (E5 - b^2) + C31\ (jzii\ E4 - B26\ E2) + E10\ E2)$$

$$- \sin(E1)\ (D38\ (E5 - b^2) + C31\ (jzri\ E4 + B27\ E2) - E11\ E2))$$

After the sixth level of computation:

Exr6;
(c60) exr6;

(d60) F3 (F2 (C33 (F8 - F5) + D35 F4 + E6) + F1 (C33 (F7 + F6) + D34 F4 - E7))

Exi6;
(c61) exi6;

(d61) F3 (F1 (C33 (F8 - F5) + D35 F4 + E6) - F2 (C33 (F7 + F6) + D34 F4 - E7))

Eyr6;
(c62) eyr6;

(d62) F3 (F2 (C32 (F12 - F9) + D37 F4 + E8) + F1 (D36 F4 + C32 (F11 + F10) - E9))

Eyi6;
(c63) eyi6;

(d63) F3 (F1 (C32 (F12 - F9) + D37 F4 + E8) - F2 (D36 F4 + C32 (F11 + F10) - E9))

Ezr6;
(c64) ezr6;

(d64) F3 (F2 (D39 F4 + C31 (F16 - F13) + E10) + F1 (D38 F4 + C31 (F15 + F14) - E11))

Ezi6;
(c65) ezi6;

(d65) F3 (F1 (D39 F4 + C31 (F16 - F13) + E10) - F2 (D38 F4 + C31 (F15 + F14) - E11))

After the seventh level of computation:

Exr7;
(c66) exr7;

(d66) F3 (F1 (G22 + C33 G21 - E7) + F2 (G23 + C33 G20 + E6))

Exi7;
(c67) exi7;

(d67) F3 (F1 (G23 + C33 G20 + E6) - F2 (G22 + C33 G21 - E7))

Eyr7;
(c68) eyr7;

(d68) F3 (F1 (G32 + C32 G31 - E9) + F2 (G33 + C32 G30 + E8))

Eyi7;
(c69) eyi7;

(d69) F3 (F1 (G33 + C32 G30 + E8) - F2 (G32 + C32 G31 - E9))

Ezr7;
(c70) ezr7;

(d70) F3 (F1 (G42 + C31 G41 - E11) + F2 (G43 + C31 G40 + E10))

Ezi7;
(c71) ezi7;

(d71) F3 (F1 (G43 + C31 G40 + E10) - F2 (G42 + C31 G41 - E11))

After the eighth level of computation:

Exr8;
(c72) exr8;

(d72) F3 (F2 (H23 + H21) + F1 (H22 + H20))

Exi8;
(c73) exi8;

(d73) F3 (F1 (H23 + H21) - F2 (H22 + H20))

Eyr8;
(c74) eyr8;

(d74) F3 (F2 (H33 + H31) + F1 (H32 + H30))

Eyi8;
(c75) eyi8;

(d75) F3 (F1 (H33 + H31) - F2 (H32 + H30))

Ezr8;
(c76) ezr8,

(d76) F3 (F2 (H43 + H41) + F1 (H42 + H40))

Ezi8;
(c77) ezi8,

(d77) F3 (F1 (H43 + H41) - F2 (H42 + H40))

After the ninth level of computation:

Exr9;
(c78) exr9;

(d78) F3 (F1 J21 + F2 J20)

Exi9;
(c79) exi9,

(d79) F3 (F1 J20 - F2 J21)

Eyr9;
(c80) eyr9,

(d80) F3 (F1 J31 + F2 J30)

Eyi9;
(c81) eyi9;

(d81) F3 (F1 J30 - F2 J31)

Ezr9;
(c82) ezr9,

(d82) F3 (F1 J41 + F2 J40)

Ezi9;
(c83) ezi9,

(d83) F3 (F1 J40 - F2 J41)

After the tenth level of computation:

Exr10;
(c84) exr10;

(d84) F3 (K21 + K20)

Exi10;
(c85) exi10;

(d85) F3 (K22 - K23)

Eyr10;
(c86) eyr10;

(d86) F3 (K31 + K30)

Eyi10;
(c87) eyi10;

(d87) F3 (K32 - K33)

Ezr10;
(c88) ezr10;

(d88) F3 (K41 + K40)

Ezi10;
(c89) ezi10;

(d89) F3 (K42 - K43)

After the eleventh level of computation:

Exr11;
(c90) exr11;

(d90) F3 L20

Exi11;
(c91) exi11;

(d91) F3 L21

Eyr11;
(c92) eyr11;

(d92) F3 L30

Eyi11;
(c93) eyi11;

(d93) F3 L31

Ezr11;
(c94) ezr11;

(d94) F3 L40

Ezi11;
(c95) ezi11;

(d95) F3 L41

After the twelfth and final level of computation:

Exr12;
(c96) exr12;

(d96) M1 Ex real

Exi12;
(c97) exi12;

(d97) M2 Ex imag

Eyr12;
(c98) eyr12;

(d98) M3 Ey real

Eyi12;
(c99) eyi12;

(d99) M4 Ey imag

Ezr12;
(c100) ezr12;

(d100) M5 Ez real

Ezi12;
(c101) ezi12;

(d101) M6 Ez imag

MACSYMA Command Listing for Parallel Electric Field

The text below is the actual commands used to perform the calculation analysis for the Electric Field The commands are separated into the computational levels developed in the previous text.

Note here that certain MACSYMA conventions require what would seem a very round about way to substitute into equat'ons. For example, in level two, notice the required steps to perform the operations that were quite straight forward in the Magnetic field computations. These extra steps were necessary because of the way these equations were loaded into MACSYMA originally. The only impact on the behavior of the calculation flow is that certain choices of variable combinations are made by the order of these input commands and as such, other combinations are certainly possible and indeed may have been more natural. However, the overall behavior of the equations proved to be symmetric and therefore caused no real concern.

The substitution commands that were identical to those of the Magnetic field commands are in standard print. The commands required that were strictly for the Electric fields are in **bold face** and commands that are strictly to accommodate MACSYMA are underlined.

These first commands are in the preparation stage of the process.

```
realpart(ex);
imagpart(ex);
subst(((((x-xi)^2)+((y-yi)^2)+((z-zi)^2))^(1/2)),r,%);
```

Level one:

```
  subst("A1",(x-xi),%)$
  subst("A2",(y-yi),%)$
  subst("A3",(z-zi),%)$
```

Level two:

```
subst("B1","A1"**2,%)$
subst("B2","A2"**2,%)$
subst("B3","A3"**2,%)$

subst("B4"*"A2", jxri*"A1"*"A2",%)$
subst("B4"*"A3",jxri*"A1"*"A3",%)$
subst("B5"*"A1", jxri*"A2"*"A1",%)$
subst("B5"*"A3",jxri*"A2"*"A3",%)$
subst("B6"*"A1", jxri*"A3"*"A1",%)$
subst("B6"*"A2",jxri*"A3"*"A2",%)$
subst("B7"*"A2", jxii*"A1"*"A2",%)$
subst("B7"*"A3",jxii*"A1"*"A3",%)$
subst("B8"*"A1", jxii*"A2"*"A1",%)$
subst("B8"*"A3",jxii*"A2"*"A3",%)$
subst("B9"*"A1", jxii*"A3"*"A1",%)$
subst("B9"*"A2",jxii*"A3"*"A2",%)$
subst("B10"*"A2", jyri*"A1"*"A2",%)$
subst("B10"*"A3",jyri*"A1"*"A3",%)$
subst("B11"*"A1", jyri*"A2"*"A1",%)$
subst("B11"*"A3",jyri*"A2"*"A3",%)$
subst("B12"*"A1", jyri*"A3"*"A1",%)$
subst("B12"*"A2",jyri*"A3"*"A2",%)$
subst("B13"*"A2", jyii*"A1"*"A2",%)$
subst("B13"*"A3",jyii*"A1"*"A3",%)$
subst("B14"*"A1", jyii*"A2"*"A1",%)$
subst("B14"*"A3",jyii*"A2"*"A3",%)$
subst("B15"*"A1", jyii*"A3"*"A1",%)$
subst("B15"*"A2",jyii*"A3"*"A2",%)$
subst("B16"*"A2", jzri*"A1"*"A2",%)$
subst("B16"*"A3",jzri*"A1"*"A3",%)$
subst("B17"*"A1", jzri*"A2"*"A1",%)$
subst("B17"*"A3",jzri*"A2"*"A3",%)$
subst("B18"*"A1", jzri*"A3"*"A1",%)$
subst("B18"*"A2",jzri*"A3"*"A2",%)$
subst("B19"*"A2", jzii*"A1"*"A2",%)$
subst("B19"*"A3",jzii*"A1"*"A3",%)$
subst("B20"*"A1", jzii*"A2"*"A1",%)$
subst("B20"*"A3",jzii*"A2"*"A3",%)$
subst("B21"*"A1", jzii*"A3"*"A1",%)$
subst("B21"*"A2",jzii*"A3"*"A2",%)$

subst("B22",3*b*jxri,%)$
subst("B23",3*b*jxii,%)$
subst("B24",3*b*jyri,%)$
subst("B25",3*b*jyii,%)$
subst("B26",3*b*jzri,%)$
subst("B27",3*b*jzii,%)$
```

Level three:

subst("C1","B1"+"B2"+"B3",%)$

subst("C31","B1"+"B2",%)$
subst("C32","B1"+"B3",%)$
subst("C33","B2"+"B3",%);

subst("C35","A3"*"B19",%)$
subst("C36","A3"*"B16",%)$
subst("C37","A2"*"B13",%)$
subst("C38","A2"*"B10",%);

subst("C39","A3"*"B20",%)$
subst("C40","A3"*"B17",%)$
subst("C41","A2"*"B7",%)$
subst("C42","A2"*"B4",%);

subst("C43","A3"*"B7",%)$
subst("C44","A3"*"B14",%)$
subst("C45","A3"*"B4",%)$
subst("C46","A3"*"B11",%);

Level four:

subst("D1",sqrt("C1"),%)$
subst("D2"*"D1",4*%pi*"C1"^(3/2),%)$

subst("D33",1/"C1",%);

subst("D34","C38"+"C36",%)$
subst("D35","C37"+"C35",%);

subst("D36","C40"+"C42",%)$
subst("D37","C39"+"C41",%);

subst("D38","C45"+"C46",%)$
subst("D39","C43"+"C44",%);

Level five:

```
subst("E1",b*"D1",%)$
subst("E2",(1/"D1"),%)$
subst("E3",(1/"D2"),%)$
subst("E4",b^2-"D33",%)$
subst("E5",3*"D33",%)$


subst("E6"*"E2",3*b*"D34"*"E2",%)$
subst("E7"*"E2",3*b*"D35"*"E2",%);
subst("E8"*"E2",3*b*"D36"*"E2",%)$
subst("E9"*"E2",3*b*"D37"*"E2",%);
subst("E10"*"E2",3*b*"D38"*"E2",%)$
subst("E11"*"E2",3*b*"D39"*"E2",%);
```

Level six:

```
subst("F1",cos("E1"),%)$
subst("F2",sin("E1"),%)$
subst("F4","E5"-b^2,%);


subst("F5","E2"*"B22",%)$
subst("F6","E2"*"B23",%)$
subst("F7","E4"*jxri,%)$
subst("F8","E4"*jxii,%)$
subst(1,"E2",%)$
subst(1,"E3",%)$
subst(%*"F3",%,%);

subst("F9","E2"*"B24",%)$
subst("F10","E2"*"B25",%)$
subst("F11","E4"*jyri,%)$
subst("F12","E4"*jyii,%)$
subst(1,"E2",%)$
subst(1,"E3",%)$
subst(%*"F3",%,%);

subst("F13","E2"*"B26",%)$
subst("F14","E2"*"B27",%)$
subst("F15","E4"*jzri,%)$
subst("F16","E4"*jzii,%)$
subst(1,"E2",%)$
subst(1,"E3",%)$
subst(%*"F3",%,%);
```

```
Level seven:

subst("G20","F8"-"F5",%)$
subst("G21","F7"+"F6",%)$
subst("G22","F4"*"D34",%)$
subst("G23","F4"*"D35",%);

subst("G30","F12"-"F9",%)$
subst("G31","F11"+"F10",%)$
subst("G32","F4"*"D36",%)$
subst("G33","F4"*"D37",%);


subst("G40","F16"-"F13",%)$
subst("G41","F15"+"F14",%)$
subst("G42","F4"*"D38",%)$
subst("G43","F4"*"D39",%);


Level eight:

subst("H22","C33"*"G21",%)$
subst("H23","C33"*"G20",%)$
subst("H20"+"H22","H22"+"G22"-"E7",%)$
subst("H21"+"H23","H23"+"G23"+"E6",%);


subst("H32","C32"*"G31",%)$
subst("H33","C32"*"G30",%)$
subst("H30"+"H32","H32"+"G32"-"E9",%)$
subst("H31"+"H33","H33"+"G33"+"E8",%);


subst("H42","C31"*"G41",%)$
subst("H43","C31"*"G40",%)$
subst("H40"+"H42","H42"+"G42"-"E11",%)$
subst("H41"+"H43","H43"+"G43"+"E10",%);


Level nine:

subst("J20","H23"+"H21",%)$
subst("J21","H22"+"H20",%);
subst("J30","H33"+"H31",%)$
subst("J31","H32"+"H30",%);
subst("J40","H43"+"H41",%)$
subst("J41","H42"+"H40",%);
```

```
Level ten:

subst("K20","F1"*"J21",%)$
subst("K21","F2"*"J20",%);
subst("K22","F1"*"J20",%)$
subst("K23","F2"*"J21",%);

subst("K30","F1"*"J31",%)$
subst("K31","F2"*"J30",%);
subst("K32","F1"*"J30",%)$
subst("K33","F2"*"J31",%);

subst("K40","F1"*"J41",%)$
subst("K41","F2"*"J40",%);
subst("K42","F1"*"J40",%)$
subst("K43","F2"*"J41",%);

Level eleven:

subst("L20","K21"+"K20",%);
subst("L21","K22"-"K23",%);

subst("L30","K31"+"K30",%);
subst("L31","K32"-"K33",%);

subst("L40","K41"+"K40",%);
subst("L41","K42"-"K43",%);

Level twelve:

subst("M1","F3"*"L20",%);

subst("M2","F3"*"L21",%);

subst("M3","F3"*"L30",%);

subst("M4","F3"*"L31",%);

subst("M5","F3"*"L40",%);

subst("M6","F3"*"L41",%);
```

Appendix B

This appendix provides the supplementary material for the development of the Parallel VWE. Presented here are the full size data-flow graphs which provide a graphical representation of the specification data developed as a result of this thesis effort. The text is organized into three sections. The sections are the presentation of the Parallel Vector Potential, the Parallel Magnetic Field, and the Parallel Electric Field. The order of presentation for all sections are:

1. The real part of the "X" directed vector quantity.
2. The imaginary part of the vector quantity.
3. The real and imaginary part (paired) of the vector quantity.
4. The combined cartesian set for the vector quantity

The final data-flow graph, Figure 28, is the presentation of the fully overlayed vector quantities.

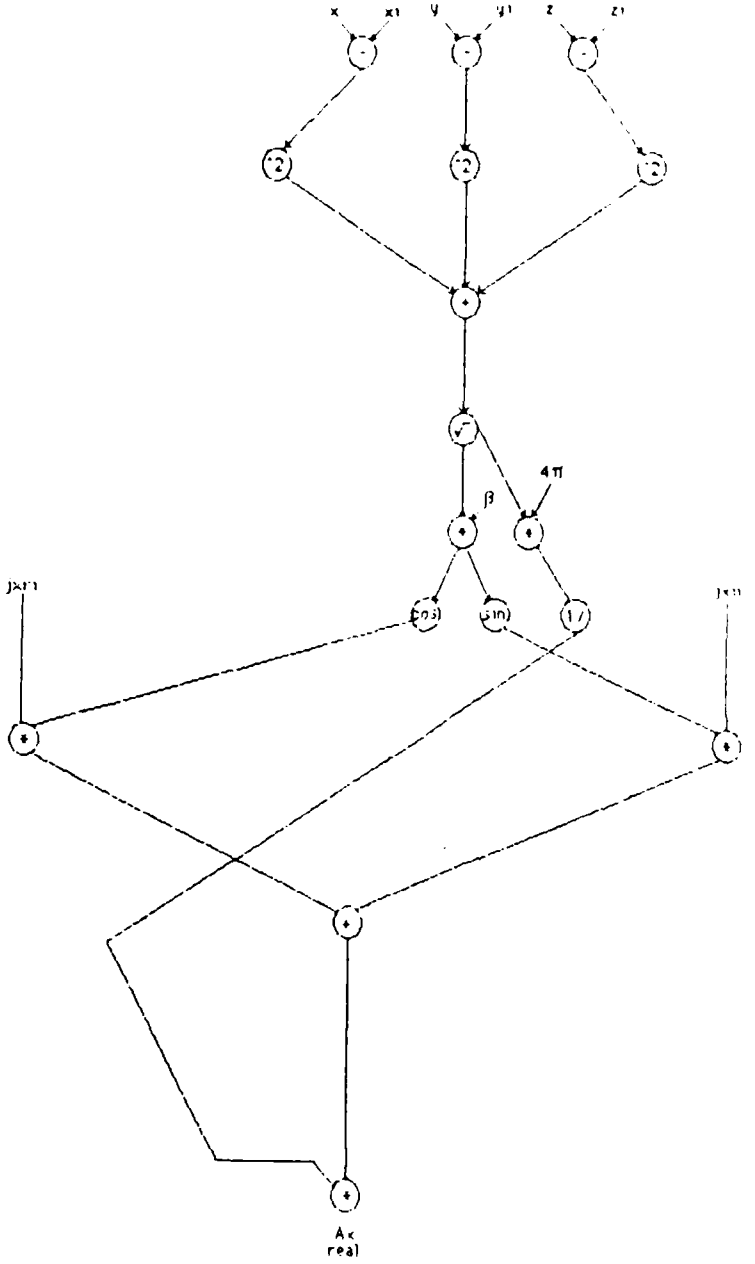


Figure 16. Real Part of "X" Directed Vector Potential

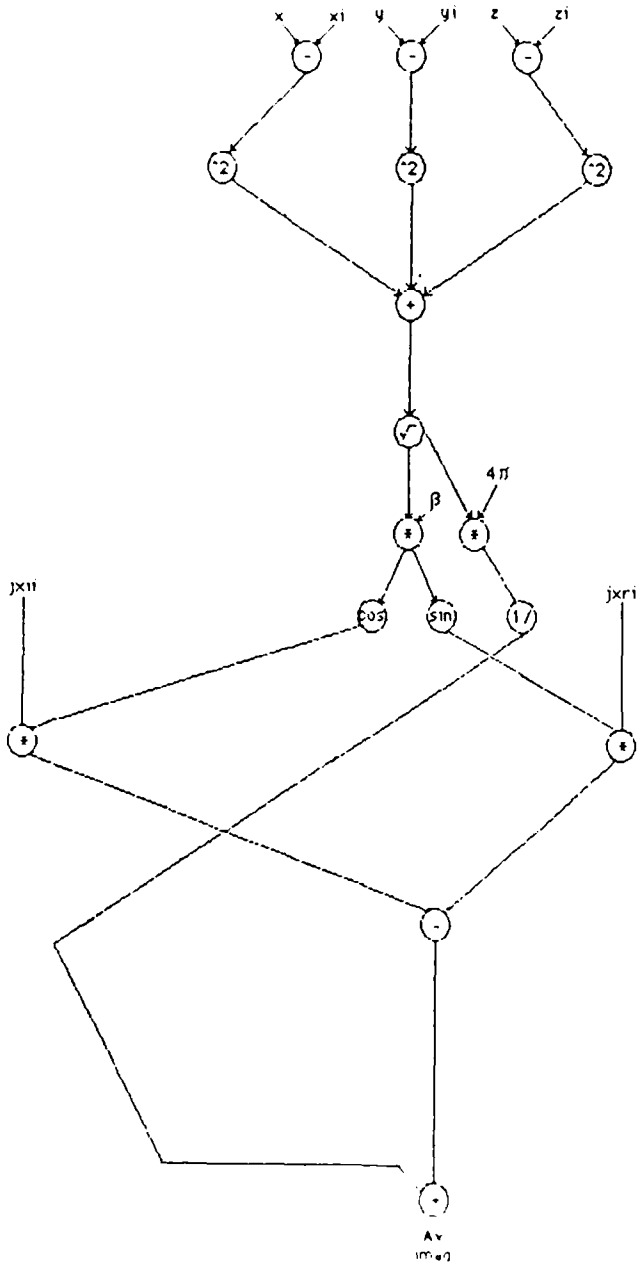
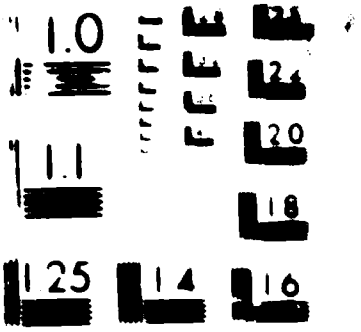
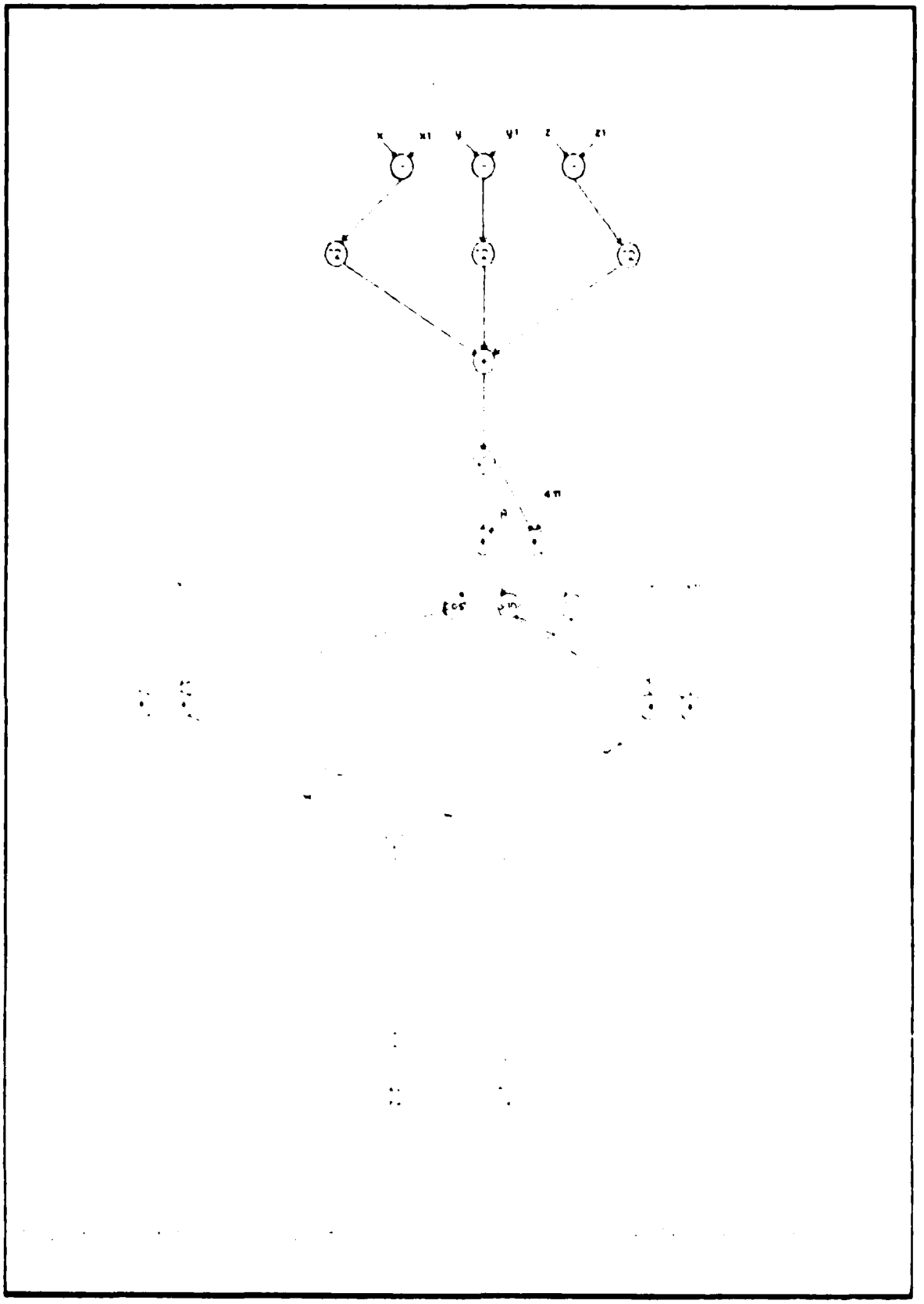
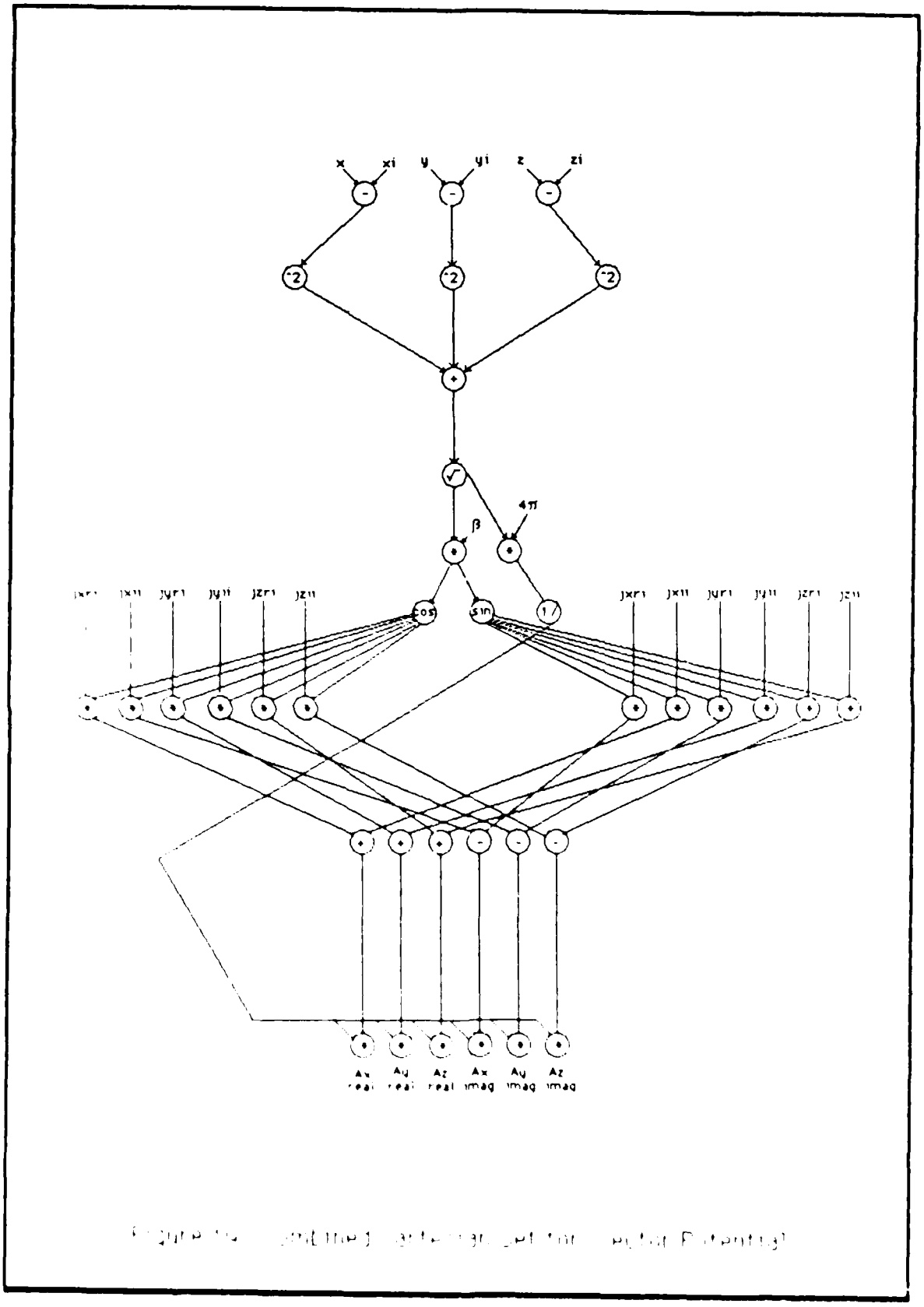


Figure 17 Imaginary Part

AD-A179 518 ARCHITECTURAL IMPLICATIONS OF A PARALLEL COMPUTATIONAL 1/2
APPROACH TO THE VE..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. J L STRAUSS
UNCLASSIFIED MAR 87 AFIT/GE/ENG/87M-5 F/G 9/2 NL


1.0
1.1
1.25
1.4
1.6
2.2
2.0
1.8




x
xi
y
yi
z
zi
-
-
-
^2
^2
^2
+
√
4π
β
+
+
cos
sin
1/
jxr1
jxi1
jyr1
jyi1
jzr1
jzi1
jxr1
jxi1
jyr1
jyi1
jzr1
jzi1
+
+
+
-
-
-
Ax real
Ay real
Az real
Ax imag
Ay imag
Az imag

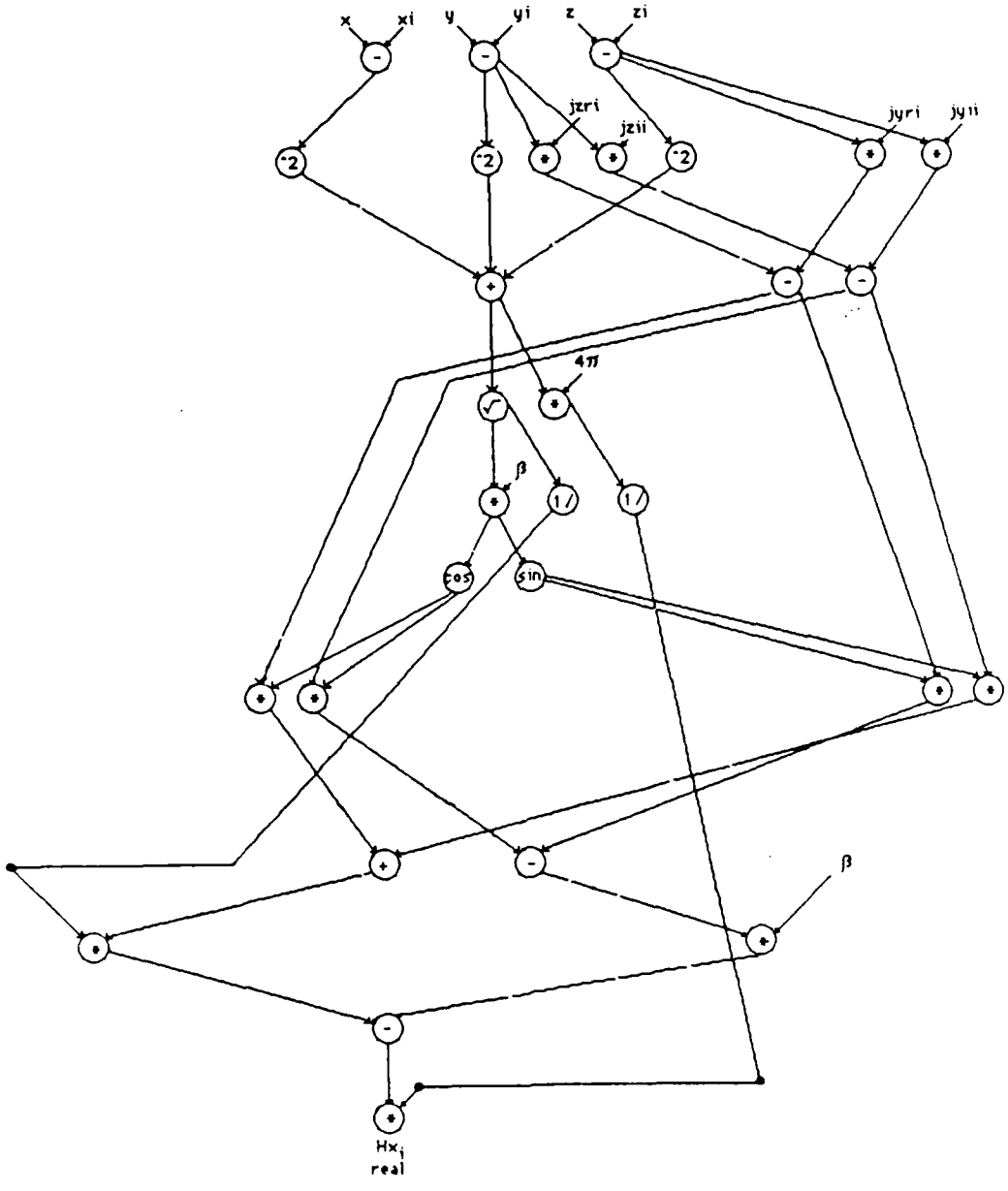


Figure 20 Real Part of "X" Directed Magnetic Field

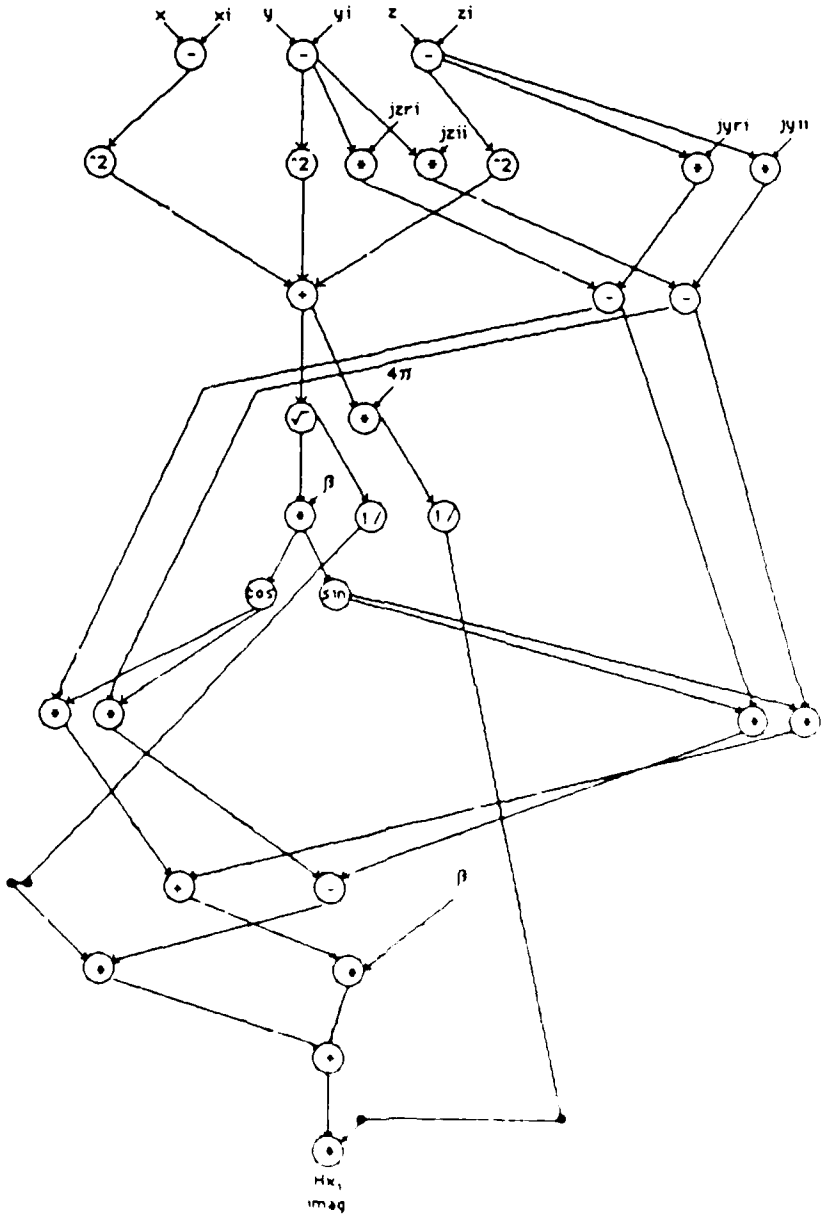


Figure 21 Imaginary Part of [illegible] Magnetic Field

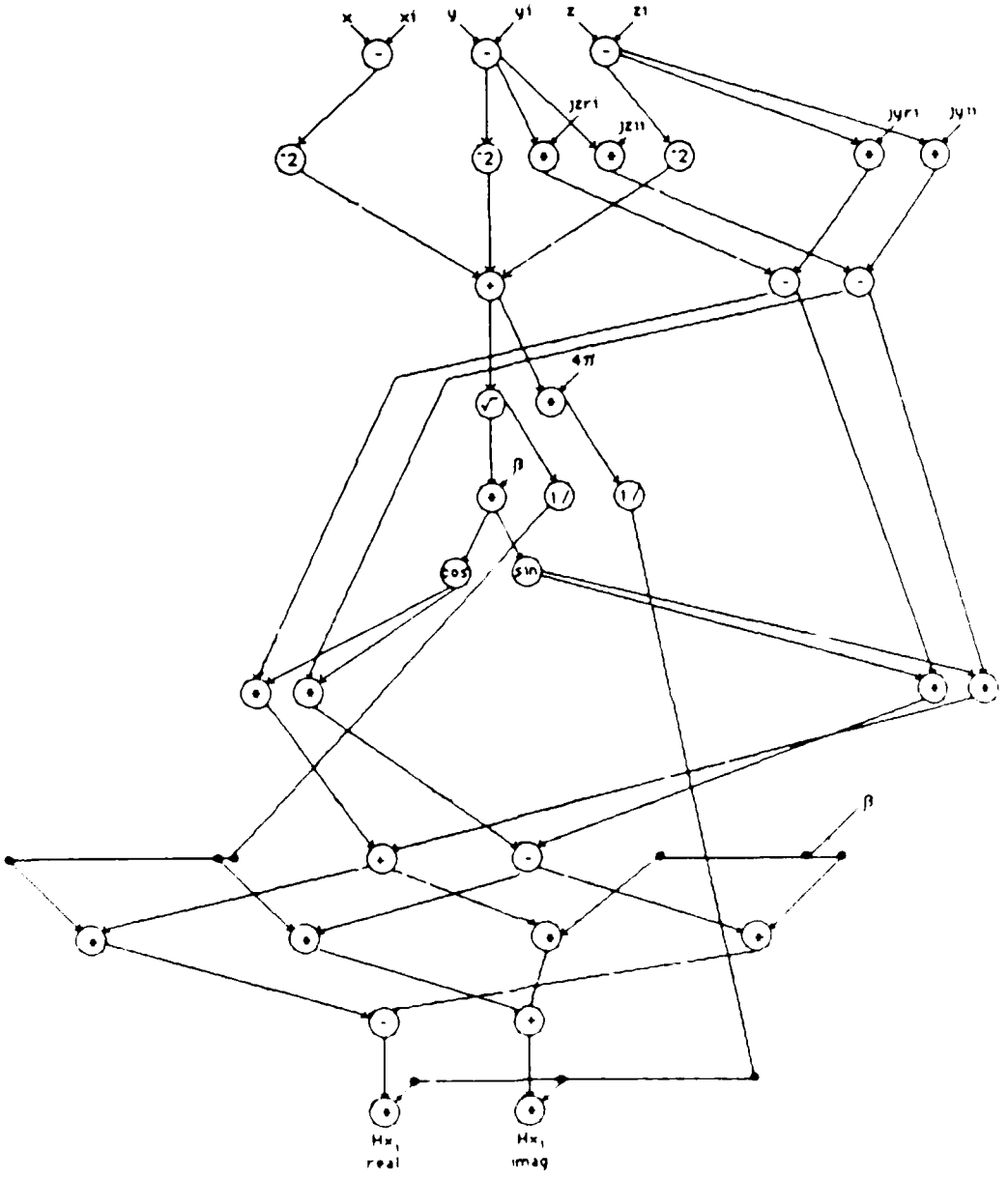


Figure 22 Real and Imaginary Part (Paired) of "X" Directed Magnetic Field

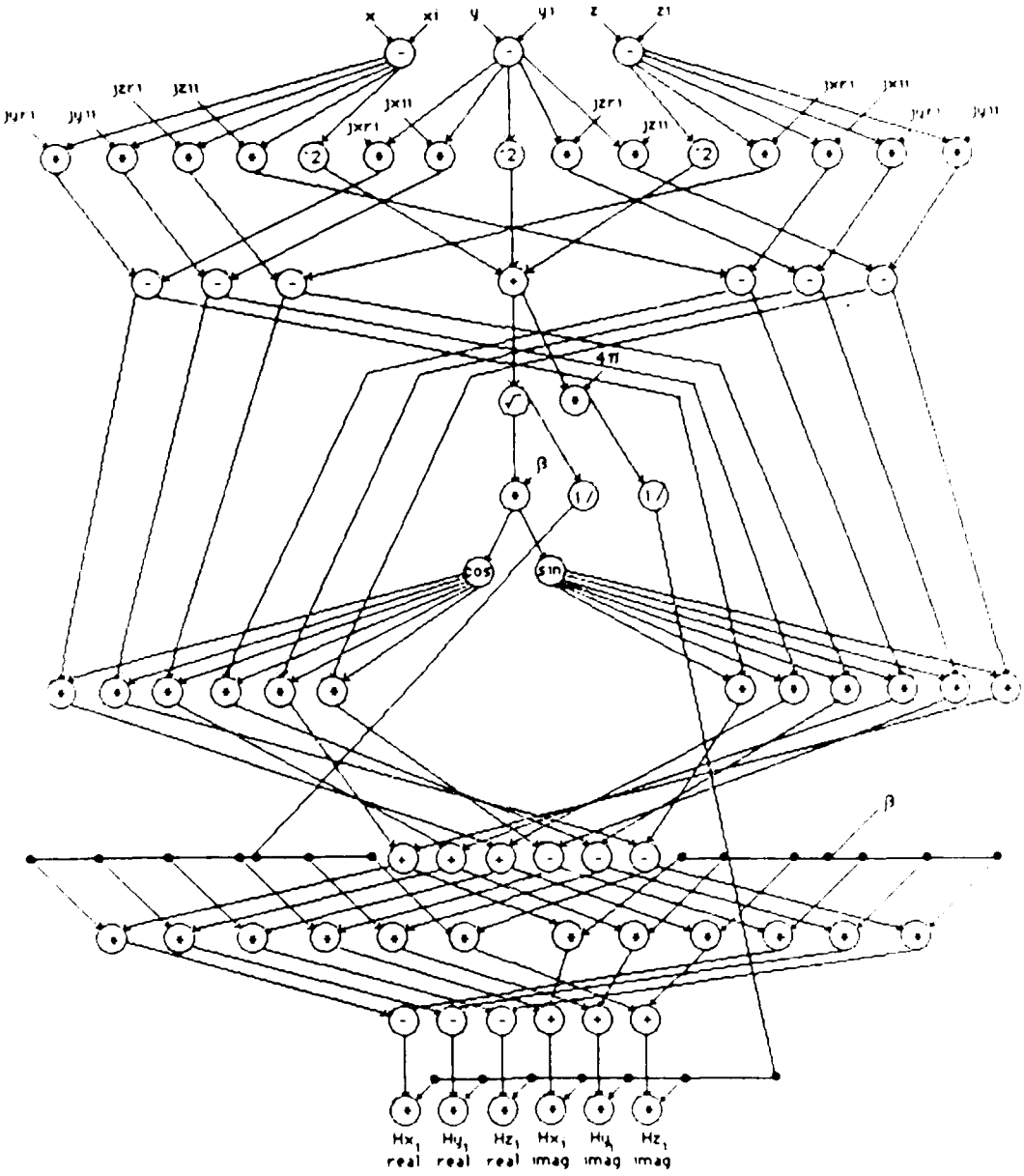


Figure 23. Combined Cartesian Set for Magnetic Field

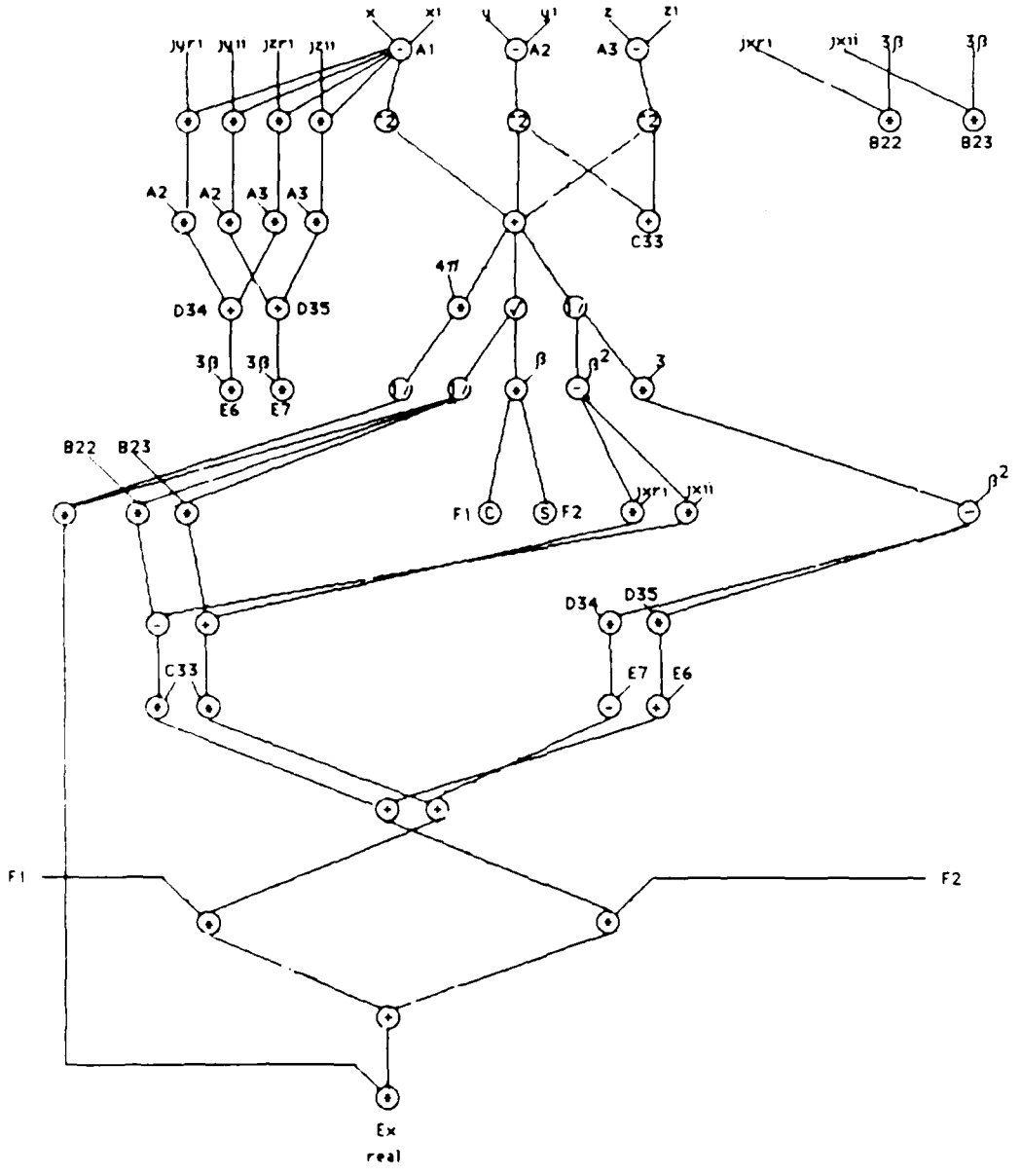


Figure 24 Real Part of "X" Directed Electric Field

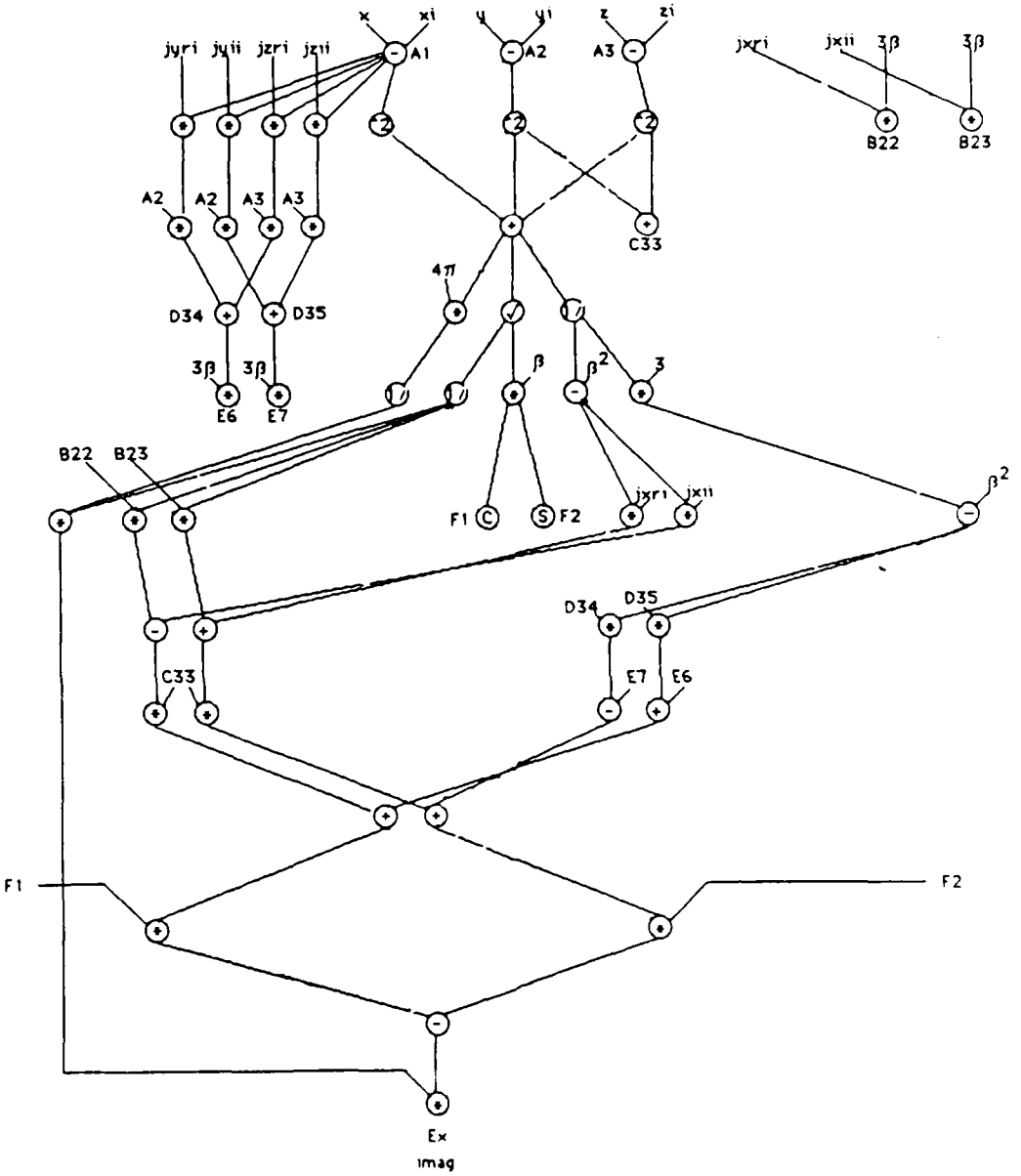


Figure 25 Imaginary Part of "X" Directed Electric Field

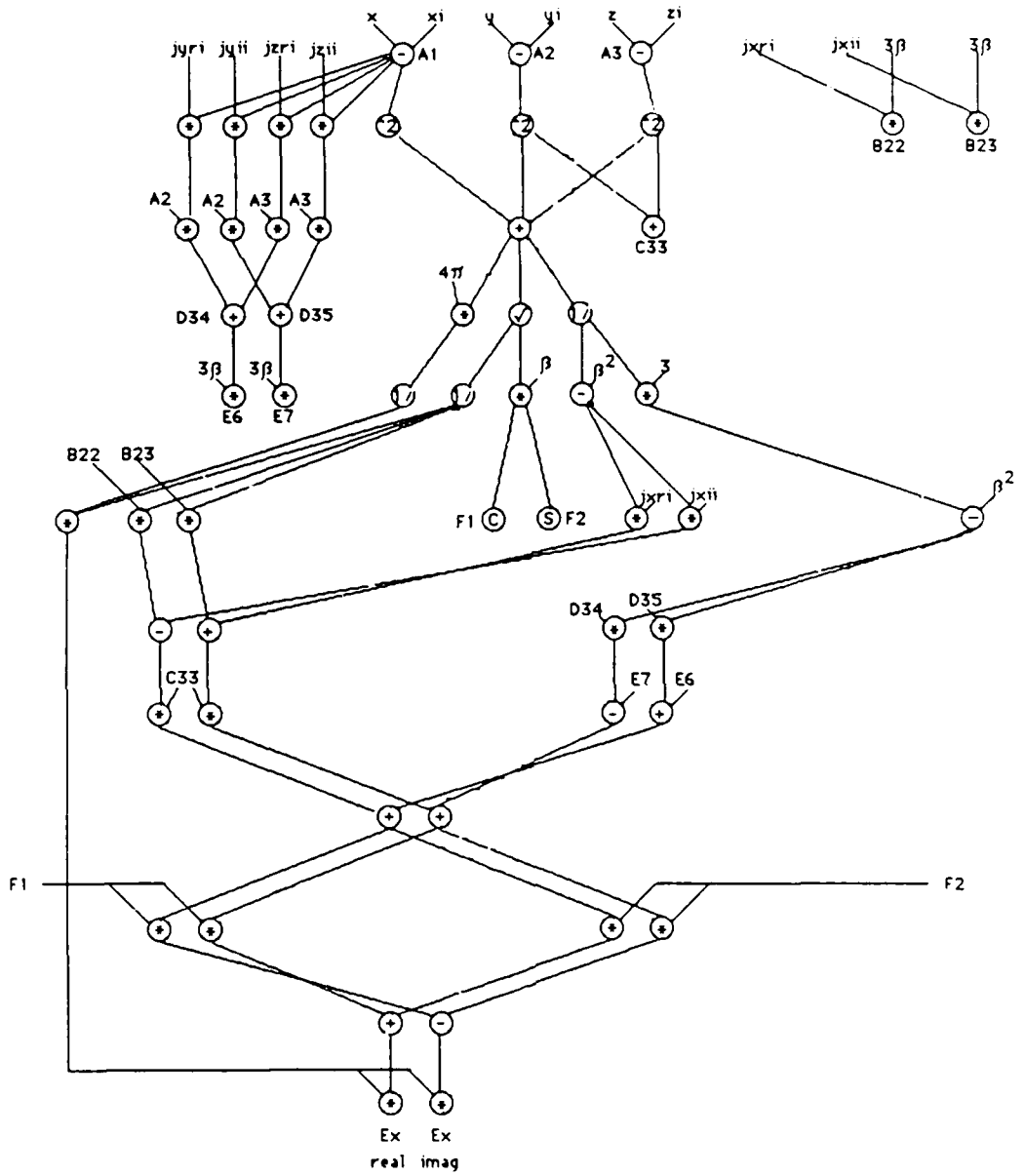


Figure 26 Real and Imaginary Part (Paired) of "X" Directed Electric Field

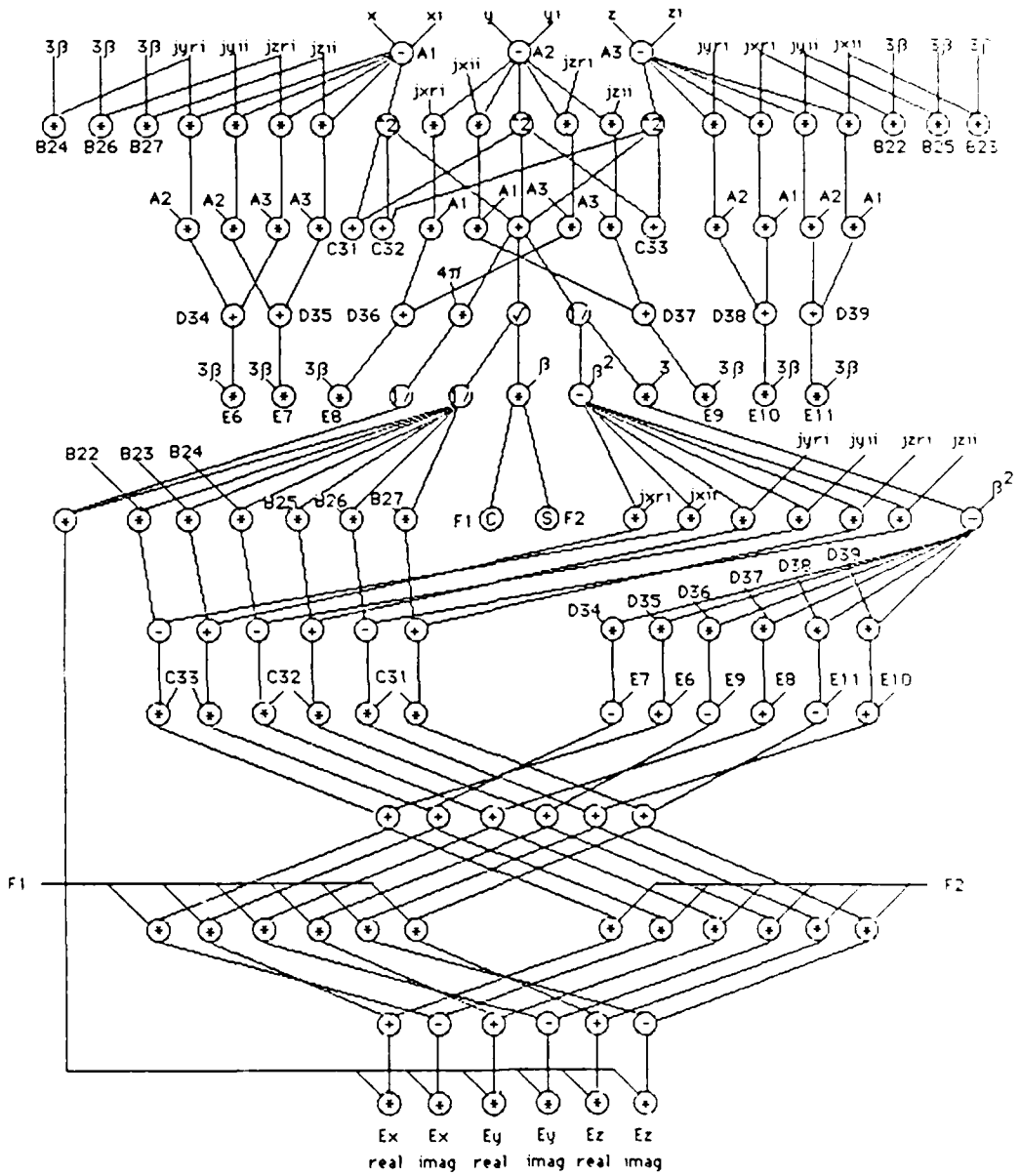


Figure 27 Combined Cartesian Set for Electric Field

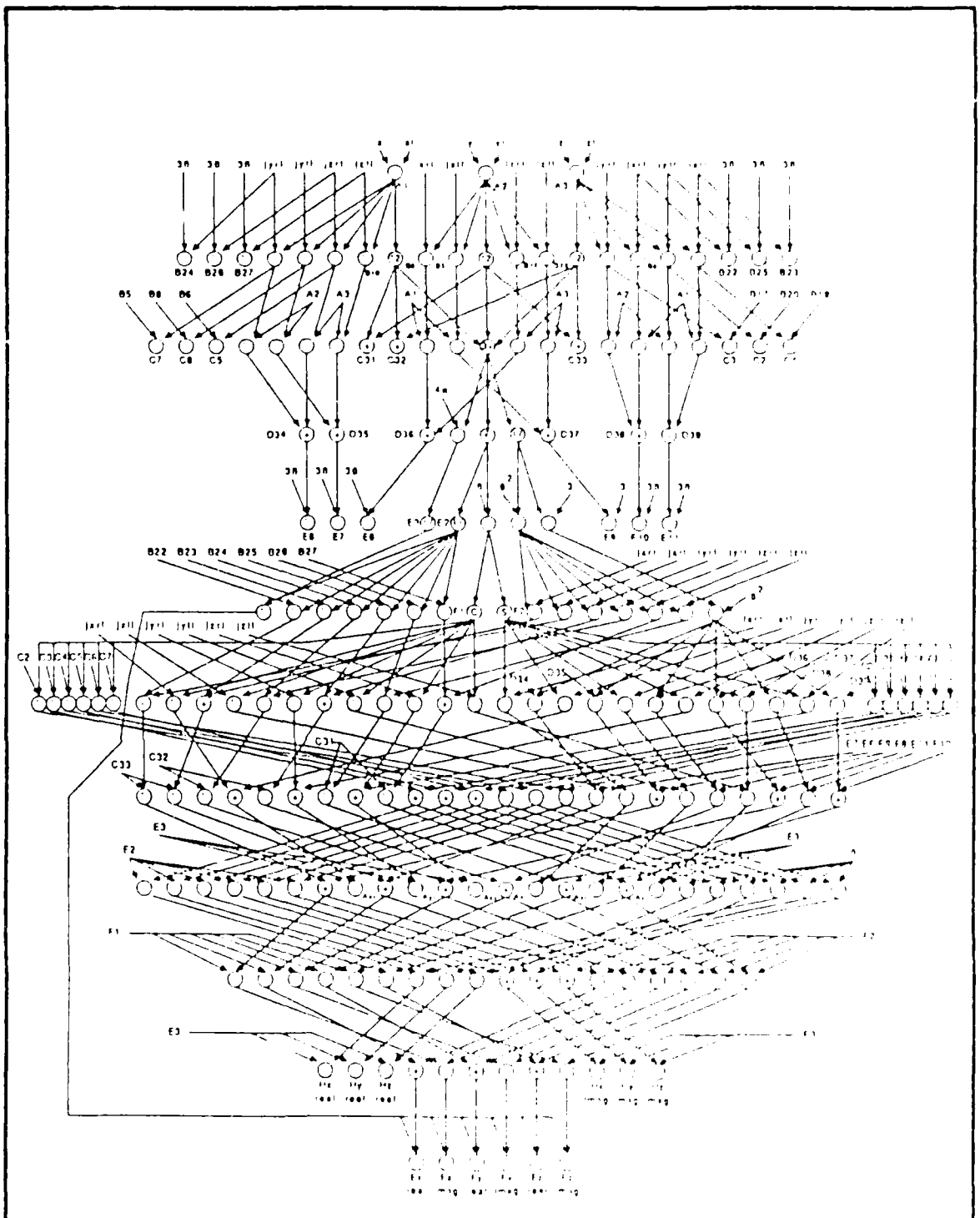


Figure 28 Fully Overlayed Vector Quantizer

Bibliography

Cody, William J. and William Waite. Software Manual for the Elementary Functions. New Jersey: Prentice-Hall, 1980.

Hoyt, B. A. Digital Algorithm Specification for the VLSI Implementation of the Electromagnetic Field of an Arbitrary Current Source. MS Thesis AFIT/GE/ENG/86D18, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1986.

Jones, Lawrence E. Algorithm Definition for the VLSI Design Implementation of the Electromagnetic Radiation Integral. MS Thesis AFIT/GE/ENG/85D23, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985.

Taylor, Fred J. Digital Filter Design Handbook. New York: Marcel Decker, 1983.

Vita

Captain Jack L. Strauss was born 6 April 1957 in Inglewood California. He graduated from High School in 1975 and attended Orange Coast College for a year prior to enlisting in the Air Force. As an enlisted member he performed duties in the 3055X career field (Electronic Computer Systems Specialist). He was accepted into the Airman Scholarship and Commissioning Program in 1979 and ultimately graduated from Loyola Marymount University with a BSEE. He was assigned to HQ SISD, Offutt AFB, Nebraska as a Communications Systems Development Engineer and became the SAC test director for the SACDIN program. He was awarded the Air Force Commendation Medal upon leaving Offutt and entered the School of Engineering at the Air Force Institute of Technology WPAFB, Ohio in May 1985.

Permanent Address: P.O. Box 1774
Garden Grove, CA 92642

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/87M-5	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION School of Engineering	6b OFFICE SYMBOL (If applicable) AFIT/ENG
6c ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433	7a NAME OF MONITORING ORGANIZATION 7b ADDRESS (City, State, and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)
8c ADDRESS (City, State, and ZIP Code)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER 10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
see box 19

12 PERSONAL AUTHOR(S)
Jack L. Strauss, Capt, USAF

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
MS Thesis	FROM _____ TO _____	1987 March	1

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
32	01	3	Electromagnetic Scattering, Radar Absorbing Materials, Finite Element Method

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: ARCHITECTURAL IMPLEMENTATION OF A PARALLEL MOMENTUM APPROACH TO THE VECTOR WAVE EQUATION

Thesis Chairman: J. W. DeGroat, Capt, USAF
Assistant Professor, Electrical and Computer Engineering

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	
22a NAME OF RESPONSIBLE INDIVIDUAL J. W. DeGroat, Capt, USAF	22b TELEPHONE (Include Area Code) 22c OFFICE SYMBOL

DD Form 1473, JUN 86 Previous editions are obsolete

From Block 19

Abstract

An algorithm has been specified for the hardware implementation of the numerical solution of the electromagnetic fields of an arbitrary current source. The algorithm, Defined as the Vector Wave Equation (VWE), solves for the magnetic (H) and electric (E) fields, as well as the vector potential (A) of a finite length arbitrary current source. The specified algorithm has been verified through FORTRAN simulation to produce results accurate to within 3 decimal places for a 500 sub element dipole.

The VWE forms a model which is algorithmically symmetric with respect to the Cartesian coordinate system. As such, the VWE lends itself to highly parallel and concurrent computational techniques. This property of the algorithm makes it an excellent candidate for implementation by a Very High Speed Integrated Circuit (VHSIC) class processor. Investigation of a parallel, highly concurrent architectural implementation has yielded preliminary results that computational savings of a factor of 5 and a throughput rate increase of 5 orders of magnitude is attainable.

This research has shown that an application specific VHSIC class processor array has sufficient computing power to support interactive calculation of a set of equations which solve for the magnetic and electric fields as well as the vector potential of an arbitrary current source.

END

5-87

DTIC