COMPARISON OF ABSORPTION AND RADIATION
BOUNDARY CONDITIONS USING A TIME-DOMAIN
THREE-DIMENSIONAL FINITE DIFFERENCE
ELECTROMAGNETIC COMPUTER CODE

THESIS
Clifford F. Williford
Captain, USAF

AFIT/GE/ENG/85D-53

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

Wright-Patterson Air Force Base, Ohio
COMPARISON OF ABSORPTION AND RADIATION BOUNDARY CONDITIONS USING A TIME-DOMAIN THREE-DIMENSIONAL FINITE DIFFERENCE ELECTROMAGNETIC COMPUTER CODE

THESIS

Clifford F. Williford
Captain, USAF

Approved for public release; distribution unlimited
COMPARISON OF ABSORPTION AND RADIATION BOUNDARY CONDITIONS
USING A TIME-DOMAIN THREE-DIMENSIONAL FINITE DIFFERENCE
ELECTROMAGNETIC COMPUTER CODE

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

Clifford F. Williford, B.S.E.E
Captain, USAF

December 1985

Approved for public release; distribution unlimited
Acknowledgments

In accomplishing the task outlined in this thesis I have had a great deal of help from many people. I am indebted to Lt Jim Hebert for creating the thesis topic and quite adequately filling the job of mentor. I appreciate also my thesis advisor, almost Dr. (Lt) Randy Jost. His patience, guidance, and support were welcomed. This was especially needed in my case since I completely changed my thesis topic at the end of June. I owe much to my thesis sponsor, AFWAL/FIESL. Hopefully, this does not include the $35,000 of computer time I used! Last, but most of all, I thank my wife Lori and my sons, Chip and Brandon. My family has put up with many lost family hours, well above and beyond the call of understanding.

Clifford F. Williford
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>ii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>ix</td>
</tr>
<tr>
<td>List of Symbols</td>
<td>x</td>
</tr>
<tr>
<td>Abstract</td>
<td>xii</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Lightning</td>
<td>3</td>
</tr>
<tr>
<td>Lightning and Aircraft</td>
<td>3</td>
</tr>
<tr>
<td>Electromagnetic Computer Codes</td>
<td>6</td>
</tr>
<tr>
<td>Background</td>
<td>7</td>
</tr>
<tr>
<td>Problem</td>
<td>11</td>
</tr>
<tr>
<td>Scope</td>
<td>11</td>
</tr>
<tr>
<td>Assumptions</td>
<td>12</td>
</tr>
<tr>
<td>2. Theory</td>
<td>13</td>
</tr>
<tr>
<td>Finite Difference Form of Maxwell's Equations</td>
<td>16</td>
</tr>
<tr>
<td>Field Locations</td>
<td>20</td>
</tr>
<tr>
<td>Three-Dimensional Finite Difference Equations</td>
<td>22</td>
</tr>
<tr>
<td>3. Boundary Conditions</td>
<td>31</td>
</tr>
<tr>
<td>Absorption Boundary Condition</td>
<td>32</td>
</tr>
<tr>
<td>Radiation Boundary Condition</td>
<td>33</td>
</tr>
<tr>
<td>4. Modeling the F-16</td>
<td>37</td>
</tr>
<tr>
<td>Problem Space</td>
<td>38</td>
</tr>
<tr>
<td>Field Locations</td>
<td>41</td>
</tr>
<tr>
<td>Cell Size Choices Made for the F-16</td>
<td>42</td>
</tr>
<tr>
<td>5. Program Details</td>
<td>50</td>
</tr>
<tr>
<td>Basic Code</td>
<td>50</td>
</tr>
<tr>
<td>Boundary Conditions</td>
<td>51</td>
</tr>
<tr>
<td>Sample Point Locations</td>
<td>51</td>
</tr>
<tr>
<td>Source Function</td>
<td>54</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6. Analysis</td>
<td>57</td>
</tr>
<tr>
<td>Waveform Appearance</td>
<td>68</td>
</tr>
<tr>
<td>Program Considerations</td>
<td>70</td>
</tr>
<tr>
<td>Conclusion</td>
<td>71</td>
</tr>
<tr>
<td>7. Summary and Recommendations</td>
<td>73</td>
</tr>
<tr>
<td>Appendix A: Updated Modified Rymes' Code</td>
<td>76</td>
</tr>
<tr>
<td>Appendix B: Rymes' Code with Radiation Boundary</td>
<td>95</td>
</tr>
<tr>
<td>Conditions</td>
<td></td>
</tr>
<tr>
<td>Appendix C: Sensor Plots</td>
<td>121</td>
</tr>
<tr>
<td>Bibliography</td>
<td>162</td>
</tr>
<tr>
<td>VITA</td>
<td>170</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Three-dimensional Field Representation</td>
<td>15</td>
</tr>
<tr>
<td>2.</td>
<td>Differencing Points</td>
<td>19</td>
</tr>
<tr>
<td>3.</td>
<td>Decentralizing Mesh in One-dimension</td>
<td>21</td>
</tr>
<tr>
<td>4.</td>
<td>Location of Difference Fields</td>
<td>23</td>
</tr>
<tr>
<td>5.</td>
<td>Finite Conductivity at Boundary Edges</td>
<td>32</td>
</tr>
<tr>
<td>6.</td>
<td>Basic Grid Block</td>
<td>39</td>
</tr>
<tr>
<td>7.</td>
<td>Field Locations with the Decentralizing Grid System</td>
<td>41</td>
</tr>
<tr>
<td>8.</td>
<td>F-16 Fighting Falcon</td>
<td>43</td>
</tr>
<tr>
<td>9.</td>
<td>Gridding of the F-16, Side View</td>
<td>44</td>
</tr>
<tr>
<td>10.</td>
<td>Gridding of the F-16, Front View</td>
<td>45</td>
</tr>
<tr>
<td>11.</td>
<td>Gridding of the F-16, Top View</td>
<td>46</td>
</tr>
<tr>
<td>12.</td>
<td>Three-dimensional Drawing of F-16 as Blocks</td>
<td>47</td>
</tr>
<tr>
<td>13.</td>
<td>Photograph of F-16 Block Model</td>
<td>49</td>
</tr>
<tr>
<td>14.</td>
<td>Sensor Locations on a Block F-16 Model</td>
<td>53</td>
</tr>
<tr>
<td>15.</td>
<td>Expanded Plot of Source (First 250 nanoseconds)</td>
<td>55</td>
</tr>
<tr>
<td>16.</td>
<td>Normalized Double Exponential Source Function</td>
<td>56</td>
</tr>
<tr>
<td>17.</td>
<td>Sensor One, H-field, Right Wing</td>
<td>58</td>
</tr>
<tr>
<td>18.</td>
<td>Sensor Two, H-field, Nose</td>
<td>59</td>
</tr>
<tr>
<td>19.</td>
<td>Sensor Three, H-field, Engine Burner Can</td>
<td>60</td>
</tr>
<tr>
<td>20.</td>
<td>Sensor Four, H-field, Forward Fuselage Bottom</td>
<td>61</td>
</tr>
<tr>
<td>21.</td>
<td>Sensor Five, H-field, Rear Fuselage Bottom</td>
<td>62</td>
</tr>
<tr>
<td>22.</td>
<td>Sensor Six, H-field, Vertical Stabilizer</td>
<td>63</td>
</tr>
<tr>
<td>23.</td>
<td>Sensor Seven, H-field, Left Wing</td>
<td>64</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>24. Sensor Eight, E-field, Right Wing</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>25. Sensor Nine, E-field, Middle Fuselage Top</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>26. Sensor Ten, E-field, Left Wing</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>27. Sensor One, H-field, Right Wing, $H_x(18,10,6)$</td>
<td>122</td>
<td></td>
</tr>
<tr>
<td>28. Sensor One, H-field, Right Wing, $H_x(18,10,6)$</td>
<td>123</td>
<td></td>
</tr>
<tr>
<td>29. Sensor One, H-field, Right Wing, $H_x(18,10,6)$</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>30. Sensor One, H-field, Right Wing, $H_x(18,10,6)$</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>31. Sensor Two, H-field, Nose, $H_z(4,10,13)$</td>
<td>126</td>
<td></td>
</tr>
<tr>
<td>32. Sensor Two, H-field, Nose, $H_z(4,10,13)$</td>
<td>127</td>
<td></td>
</tr>
<tr>
<td>33. Sensor Two, H-field, Nose, $H_z(4,10,13)$</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td>34. Sensor Two, H-field, Nose, $H_z(4,10,13)$</td>
<td>129</td>
<td></td>
</tr>
<tr>
<td>35. Sensor Three, H-field, Engine Burner Can, $H_z(24,11,13)$</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>36. Sensor Three, H-field, Engine Burner Can, $H_z(24,11,13)$</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>37. Sensor Three, H-field, Engine Burner Can, $H_z(24,11,13)$</td>
<td>132</td>
<td></td>
</tr>
<tr>
<td>38. Sensor Three, H-field, Engine Burner Can, $H_z(24,11,13)$</td>
<td>133</td>
<td></td>
</tr>
<tr>
<td>39. Sensor Four, H-field, Forward Fuselage Bottom, $H_z(10,4,14)$</td>
<td>134</td>
<td></td>
</tr>
<tr>
<td>40. Sensor Four, H-field, Forward Fuselage Bottom, $H_z(10,4,14)$</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>41. Sensor Four, H-field, Forward Fuselage Bottom, $H_z(10,4,14)$</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>42. Sensor Four, H-field, Forward Fuselage Bottom, $H_z(10,4,14)$</td>
<td>137</td>
<td></td>
</tr>
<tr>
<td>43. Sensor Five, H-field, Rear Fuselage Bottom, $H_z(19,4,14)$</td>
<td>138</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>44. Sensor Five, H-field, Rear Fuselage Bottom, $H_z(19,4,14)$</td>
<td>139</td>
<td></td>
</tr>
<tr>
<td>45. Sensor Five, H-field, Rear Fuselage Bottom, $H_z(19,4,14)$</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>46. Sensor Five, H-field, Rear Fuselage Bottom, $H_z(19,4,14)$</td>
<td>141</td>
<td></td>
</tr>
<tr>
<td>47. Sensor Six, H-field, Vertical Stabilizer, $H_x(22,18,14)$</td>
<td>142</td>
<td></td>
</tr>
<tr>
<td>48. Sensor Six, H-field, Vertical Stabilizer, $H_x(22,18,14)$</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>49. Sensor Six, H-field, Vertical Stabilizer, $H_x(22,18,14)$</td>
<td>144</td>
<td></td>
</tr>
<tr>
<td>50. Sensor Six, H-field, Vertical Stabilizer, $H_x(22,18,14)$</td>
<td>145</td>
<td></td>
</tr>
<tr>
<td>51. Sensor Seven, H-field, Left Wing, $H_x(18,10,23)$</td>
<td>146</td>
<td></td>
</tr>
<tr>
<td>52. Sensor Seven, H-field, Left Wing, $H_x(18,10,23)$</td>
<td>147</td>
<td></td>
</tr>
<tr>
<td>53. Sensor Seven, H-field, Left Wing, $H_x(18,10,23)$</td>
<td>148</td>
<td></td>
</tr>
<tr>
<td>54. Sensor Seven, H-field, Left Wing, $H_x(18,10,23)$</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td>55. Sensor Eight, E-field, Right Wing, $E_y(17,11,10)$</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>56. Sensor Eight, E-field, Right Wing, $E_y(17,11,10)$</td>
<td>151</td>
<td></td>
</tr>
<tr>
<td>57. Sensor Eight, E-field, Right Wing, $E_y(17,11,10)$</td>
<td>152</td>
<td></td>
</tr>
<tr>
<td>58. Sensor Eight, E-field, Right Wing, $E_y(17,11,10)$</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>59. Sensor Nine, E-field, Middle Fuselage Top, $E_y(15,13,14)$</td>
<td>154</td>
<td></td>
</tr>
<tr>
<td>60. Sensor Nine, E-field, Middle Fuselage Top, $E_y(15,13,14)$</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>61. Sensor Nine, E-field, Middle Fuselage Top, $E_y(15,13,14)$</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>62. Sensor Nine, E-field, Middle Fuselage Top, $E_y(15,13,14)$</td>
<td>157</td>
<td></td>
</tr>
<tr>
<td>63. Sensor Ten, E-field, Left Wing, $E_y(17,11,18)$</td>
<td>158</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>64. Sensor Ten, E-field, Left Wing, $E_y(17,11,18)$</td>
<td>159</td>
<td></td>
</tr>
<tr>
<td>65. Sensor Ten, E-field, Left Wing, $E_y(17,11,18)$</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>66. Sensor Ten, E-field, Left Wing, $E_y(17,11,18)$</td>
<td>161</td>
<td></td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Ten Sampled Points</td>
<td>52</td>
</tr>
<tr>
<td>2. CDC Cyber 845 Computer Run Data</td>
<td>71</td>
</tr>
</tbody>
</table>
List of Symbols

A ... generic quantity, 'A' = a vector, otherwise a magnitude in the subscripted direction (could be E-field or H-field).

ABC ... Absorption Boundary Condition.

c ... speed of light \((3 \times 10^8\) meters/second).

det ... determinant of the matrix which follows.

E-field (E) electric field (volts/meter), underbar indicates a vector quantity, subscript indicates magnitude in subscripted direction.

f(a) ... 'f' is a function of 'a'.

f[i,j,k] 'f' value evaluated at the point i,j,k.

x,y,z .. magnitude in a particular direction as defined by the Cartesian coordinate system.

\(\hat{x},\hat{y},\hat{z}\) .. unit vector in direction as defined by Cartesian coordinate system.

h ... small distance (or increment) as used in the definition of a derivative.

H-field (H) magnetic field (ampere/meter), underbar indicates a vector quantity, subscript indicates magnitude in subscripted direction.

I ... current (ampere).

i,j,k .. number designating a location in a three-dimensional grid space in a Cartesian coordinate system relative to x,y,z.

J ... current density (ampere/meter\(^2\)).

N ... integer number of number of cells in problem space.

r ... radius (meter).

RBC ... Radiation Boundary Condition.

t ... time (seconds).

< ... left side less than right side.
3DFD ... three-dimensional finite-difference computer code by Rymes'.

\[ \nabla \times \] curl operator.

\[ \nabla \cdot \] divergence operator.

\( \epsilon \) ... (epsilon) permittivity (farad/meter).

\( \mu \) ... (mu) permeability (henry/meter).

\( \rho \) ... (rho) electric charge density (coulomb/meter\(^3\)).

\( \sigma \) ... (sigma) conductivity (mho/meter)

\( \Delta \) ... (delta) small increment as in a derivative.

(#) ... general reference to entire reference # found in the bibliography.

(#:##) . reference from page ## of bibliography reference #.

(##;##) . general reference from reference # and ##.

(##;##;a) reference #, page ## and general reference 'a'.

{ } ... standing alone defines a point in the problem space (e.g., \{i,j,k\}).

\( \partial \) ... partial derivative

\( \pi \) ... (pi) = 3.14
Abstract

The Three-Dimensional Finite Difference (3DFD) computer code is compared using Absorption Boundary Conditions (ABS) versus Radiation Boundary Conditions (RBC). This comparison is made when the 3DFD code is used to study the interaction of lightning with an aircraft. The 3DFD computer code is a modified version of Rymes' 3DFD. The aircraft modeled for the paper is an F-16 'Fighting Falcon'. The ABC used simulates an infinite free-space by setting the conductivity of the boundary space to that of distilled water, to "absorb" the outgoing electromagnetic waves. The RBC simulates free-space by assigning the boundary fields to a previously calculated value. The value is calculated with a parabolic interpolation of three previous field values, which are offset in space. Therefore, the calculated value is also extrapolated to account for the time delay and position change. The results of incorporating RBC were dramatic. The ten locations sampled for the test showed marked improvement in the waveforms when using RBC's. Depending on the purpose of the analysis, this improved waveform output may be overshadowed by the 25% increase in CPU time that is needed for the more sophisticated RBC.
1. Introduction

This thesis compares Radiation Boundary Conditions (RBC) with Absorption Boundary Conditions (ABC) in a time-domain three-dimensional finite difference computer code (3DFD). In this application, the 3DFD computer code is used to analyze lightning’s electromagnetic interaction with an F-16 aircraft. The major tasks accomplished during this thesis effort include:

a. The F-16 'Fighting Falcon' aircraft was electromagnetically modeled, and implemented in a modified version of the Rymes' 3DFD code (16).

b. The 3DFD code with its original absorption boundary conditions was updated, corrected and run for a typical nose-to-tail lightning strike (Appendix 1) (48). The fields calculated were recorded at 10 monitor locations on the F-16 aircraft (Appendix 3).

c. The 3DFD code was then modified to implement the radiation boundary conditions (Appendix 2) (59).

d. The 3DFD code with the new radiation boundary conditions was run for the same nose-to-tail lightning strike on the F-16
geometric model. The fields calculated were sampled and recorded at the same 10 locations during both runs (Appendix 3).

e. The results of the two computer runs for the first two microseconds were compared (Chapter 6). The comparisons were based on one complete run with the absorption boundary conditions and one run with the radiation boundary conditions. The comparisons were made for the same 10 sample points on the F-16 (Appendix 3).

The F-16 was geometrically modeled and implemented into computer code in a manner suitable for analysis by a modified version of the Rymes' 3DFD computer code (48; 16). The Rymes' 3DFD code was modified and used by Lt Hebert at the Air Force Wright Aeronautical Laboratories Atmospheric Electricity Hazards Group (AFWAL/FIESL), Wright-Patterson AFB, Ohio (16). Hebert implemented the code with absorption boundary conditions, and modeled a CV-580 aircraft for his study (16). During this thesis effort radiation boundary conditions similar to those discussed in a paper by Kunz and Lee were developed, encoded, and written into the modified 3DFD code (27). The computer codes were run for a direct lightning strike, nose-to-tail, on an F-16 aircraft. The results are compared and presented in this thesis.
Lightning

Lightning brings to mind a flash of light in the sky, a loud clapping sound and rumbling thunder. This flash of light and the associated sounds are the discharging and neutralization of the atmosphere's large charge centers from one cloud to another, or from clouds to the earth (61:1). Lightning in this text is to be thought of as a high current electric discharge which has a path length measured in kilometers (61:1). Aircraft in the presence of these highly charged areas can become part of the high current channel (35:90-1). In fact, Mazur believes the aircraft itself triggers lightning (35:90-2). Shaeffer agrees with this, except with the qualification that a highly charged air mass, one which is conducive to a lightning discharge, must already exist before the aircraft can trigger the lightning discharge (50:67).

Lightning and Aircraft

Lightning poses a possible catastrophic threat to any aircraft (11:i). Lightning strikes account for more than one-half of the total USAF weather-related aircraft mishaps (36:vi). Since little can be done to prevent an aircraft from being struck by lightning, aircraft protection for the critical components is of utmost importance (45). Critical
components, such as flight control computers, fuel systems, and structurally critical airframe/propulsion units, are vulnerable to electrical disruption and/or mechanical failure (45). As electronic complexity has increased, the size and weight have decreased. However, the probability of lightning strikes has remained relatively constant (40:8). It is interesting to note that Pierce concludes, using basically the same logic, that the lightning hazard to aircraft operation is increasing (42:17). Therefore, the importance of characterizing lightning and its effects is of growing importance, as indicated by the number of articles written on the subject (8; 23; 31; 35; 36; 40; 42; 43; 46; 47; 49; 50). An aircraft, such as the F-16, with a "fly-by-wire" flight control system, were it unprotected, could be rendered inoperable by a lightning strike (1:1).

The starting point in characterizing lightning's effects with a computer program is to model the aircraft's outer skin. Studying the propagation of the electromagnetic energy from the entry to the exit point, as it redistributes on the surface of the aircraft's fuselage, is the first goal (34:2,16). Determining this redistribution can lead to coupling the surface currents to the interior components. The final goal is understanding lightning's effect on the sensitive electronic components, and then protecting sensitive components from these effects. An accurate computer code could be a valuable tool in designing
protection against lightning's effects. It would allow protection to be designed and tested before the production stage of new aircraft development.

All-metal aircraft are shielded from many of lightning's effects, as the all-metal skin presents a Faraday cage type enclosure for the electronics bay (19; 45; 52; 58). Recent efforts to use the new, lighter, advanced composite and thermoplastic materials in aircraft reduces this shielding protection (10:1). As early as 1952, Burkley was looking at plywood, plastics, laminated fiberglass, and other non-conductive materials to be used as aircraft structural components (5). Burkley was studying the effects of, and the necessary protection needed from, the high current surges associated with lightning strikes on these non-conductive materials (5). Examples and techniques of various protection schemes can be found in numerous sources. Sommer of Boeing Company, Weinstock of McDonnell Aircraft Company, and Fisher of General Electric are just three of the many authors publishing in this area of aircraft lightning protection (51; 63; 14).
Electromagnetic Computer Codes

Electromagnetic computer codes allow the engineer to characterize and study the electromagnetic interaction with aircraft. The engineer can then design protection against those effects which cause electronic upset or damage. As previously stated, the use of advanced composite materials and sensitive microelectronics in more modern aircraft makes them more susceptible to the effects of lightning strikes. This increased susceptibility demands more accurate electromagnetic modeling (46:220). This modeling is to be used in the development of lightning protection. Accurate models, when used in design for lightning protection, allow the use of optimal lightning protection. This protection is optimal in that excessive lightning protection would add weight that is not in proportion to the risk reduction gained (63:34-1).

While accurate, these models must also be flexible enough to handle new lightning channel models. A recent inflight lightning characterization program has shown that intracloud lightning attachment is even more severe than previously estimated, with faster risetimes and higher charge transfers (19:1). This demonstrates the need for flexible analysis systems which can be easily modified to incorporate new findings without completely reworking each study.
Many electromagnetic computer codes are available to approximate a solution to Maxwell's equations (2; 13; 28; 34; 60). One type of code solves Maxwell's equations in the integral form based upon Harrington's Method of Moments, such as Auckland's study of an F-14 (15; 4). Another type of code solves Maxwell's equations in their differential form using the Finite Difference Method of approximation, such as Holland's "THREDE" (20; 21; 22). The time-domain finite difference electromagnetic code approximation of Maxwell's equations was used during this thesis effort. In a study by Longmire of Mission Research Corporation, it is stated that the finite difference method is the fastest way of solving Maxwell's equations (33:2341). And, in a recent article, Mei states that he believes the finite difference method is the most adaptable to the virtual memory systems of minicomputers which are in growing use in industry (37:1145).

Background

In the study of lightning's interaction with aircraft, it is often necessary to have a computer model for analytical comparisons. Hebert has successfully modified a finite difference electromagnetic code written by M.D. Rymes of Electromagnetic Applications, Inc., to analyze a CV-580 lightning strike aircraft test-bed (48; 16). Other
modifications to Rymes 3DFD code and an expanded user's guide are detailed in Hebert's report (16). This modified 3DFD code presently uses absorption boundary conditions (Appendix 1), which cause the boundaries to artificially reflect unabsorbed electromagnetic energy. This gives less than desired accuracy, as the reflected waves return to the aircraft's surface at later time steps (55:626).

3DFD is a finite difference formulation of the time-domain electromagnetic-field problem. This code's major advantage is that it does not require the memory or the time needed to invert the matrices encountered in the MOM codes (33:2340). As Mur states it, though, the finite difference method has as its major problem the limited size of the problem space (41:377). It must be remembered that it is the problem space size that is the problem, not the size of the object inside. For example, an aircraft as large as a B-52 has been studied using 3DFD (21). A Cartesian finite difference method uses a rectangular problem space, totally enclosing the object to be studied (e.g., aircraft). The object to be studied is defined in the problem space by assigning values of permittivity and conductivity to each component of the total electric field (E-field), which describes the geometry of the object. This geometry has the restriction of being described by a limited number of predetermined rectangular shapes, which make up the problem space in the Cartesian coordinate system.
The computer code solves Maxwell's time dependent curl equations, which in turn solve the boundary conditions for the object in a "natural way" (62:397). The code progresses through the problem space in time steps using an algorithm originally developed by Yee (64). Since an infinite problem space cannot be defined on the computer, difficulties arise when the propagating wave reaches the problem space boundary. These boundaries cause reflections unless they are modified to account for the ideal analytical situation of free space. Thus, additional algorithms are needed to account for the radiation conditions. Taylor was the first to implement Yee's algorithm. He used absorption boundary conditions to account for the previously-mentioned reflections at the problem space boundaries (59:585).

Yee originally started with "hard" lattice truncation (another way of expressing the boundary condition) (64). Hard lattice truncation is defined as forcing the outside boundary of the problem space to be a perfect conductor. This is done by assigning the tangential E-field the value of zero at the boundary (55:626). This is also known as "tin can" boundary conditions, as the problem space is totally enclosed by perfectly conducting metal boundaries (16:22).

The two methods that are examined in this thesis are both referred to as either "soft" lattice truncation methods, or "soft" boundary conditions. These are the
absorption boundary condition and the radiation boundary condition. The absorption boundary condition is accomplished by assigning increasing values of conductivity to the cells of the problem space near the boundaries, thereby effectively chipping away at the E-field, a small amount at a time (55:626). The other method, the radiation boundary condition, reduces the magnitude of the E-field by a factor of $1/r$, which is the radiation condition characteristic of an electromagnetic wave propagating in free-space (39:41). Here 'r' is the radial distance from the origin of a centrally located coordinate system.

The finite difference codes have been found to agree, within 1 dB and 1 lattice cell, with known analytical and experimental quantities (54:202). Due to the accuracy and efficiency of Yee's basic algorithm, Taflove, Kunz, Umashankar, Merewether, Fisher, Mei, and others have published many papers on combined methods, hybrids, curved surfaces, and other modifications, making this a very competitive method for a numerical solution of Maxwell's equations (14; 26; 37; 38; 56; 62).
Problem

First, an electromagnetic model of an F-16 aircraft had to be implemented on the computer for a new study using the present modified code. During the computer runs using the absorption boundary conditions (ABC), sample data from several points was stored for further study. This required the correct geometrical modeling of the aircraft into the three-dimensional finite difference problem space. Next, the code was modified to incorporate the radiation boundary conditions (which were believed to limit previous reflections). Then the code was run with the same F-16 model, sampling the same points as before. The final task was to compare the results of both computer runs and analyze the findings.

Scope

This study was limited to developing a geometrically correct electromagnetic model of the F-16 using the subroutine AIRPLN from an AFWAL/FIESL technical report by IlLt Hebert (16). The subroutine AIRPLN, modeling an F-16, was run with the present modified computer code for a nose-to-tail lightning strike. Samples were recorded at 10 locations (7 H-fields and 3 E-fields) for predetermined orientations. The code was then changed to incorporate the
radiation boundary conditions (RBC). The modified version of the code with radiation boundary conditions was run with the same subroutine: AIRPLN. The same 10 sample points were recorded and the results were analyzed.

Assumptions

The assumptions were that the AFWAL/FIESL technical report, and previous studies which determined that RBC would be more accurate than ABC, were correct (16). An isotropic, linear, and homogeneous medium was considered. The region of interest was considered source-free except for the injected lightning strike.
2. Theory

The three-dimensional finite difference code uses the time-domain differential form of Maxwell's equations (64:302). The E-field and H-field magnitudes in each of the three vector directions that define the Cartesian coordinate system are calculated using these equations (64). The calculations are made while stepping through the grid problem space $(N_xN_yN_z)$ at a particular time. It is necessary to increment the time step after each complete pass through the grid space calculating the E-field or H-field. All units are in the MKS system unless otherwise specified. The form of Maxwell's equations for isotropic material used here is (24:361)

\[
\frac{\partial E}{\partial t} = \nabla \times H - \sigma E \tag{2.2}
\]

\[
\frac{\partial H}{\partial t} = -\nabla \times E \tag{2.1}
\]

\[
\nabla \cdot (\varepsilon E) = \rho \tag{2.3}
\]

\[
\nabla \cdot (\mu H) = 0 \tag{2.4}
\]

where
These are point relationships \((J = \sigma E)\). In the finite difference code, the term 'decentralizing mesh' is often used to refer to the gridding of the problem space. This is not to be confused with the mesh relationship of the integral form of Maxwell's equations.

A necessary process in understanding the finite difference code is to develop the three-dimensional finite difference form of Maxwell's equations. This development will be using a central differencing estimation for a derivative (32:163). The first operation in the process is to put Maxwell's equations in a finite difference form. Then one must both understand where the fields are located in the problem space, and comprehend the makeup of the problem space itself. In particular, the single point reference system references fields located on three sides of the rectangular box. A point in the three-dimensional grid, and the associated fields are shown in Figure 1. Finally, one must put all of this together and express the three-dimensional finite difference equations in a form useful to algorithm development. A source-free region is considered for the development of the algorithm.
Since the computer memory available is a limited quantity, some boundary must be placed on the problem space in order to arrive at a meaningful solution. Each author has his own style of implementing boundary conditions. The two compared in this thesis are often referred to as "soft" boundary conditions, or "soft" lattice truncation conditions. The first to be used is the absorption boundary condition which Rymes implemented in 3DFD (48). The second boundary condition is the radiation boundary condition used by Kunz (29).
Finite Difference Form of Maxwell's Equations

The finite difference form of Maxwell's equations uses Eq (2.1) thru Eq (2.4), manipulated in the Cartesian coordinate system, to arrive at a convenient algorithmic form for programming.

In the Cartesian coordinate system, the electric and magnetic field vectors are as found in Thiele's notation (53:11):

\[
\vec{A} = A_x \hat{a}_x + A_y \hat{a}_y + A_z \hat{a}_z
\]  

(2.5)

where 'A' is equal to the magnitude of the E-field or H-field component and 'a' is the unit vector in the x, y, or z direction.

The curl of A in the Cartesian coordinate system is expressed as follows (24:178):

\[
\text{curl } \vec{A} = \nabla \times \vec{A} = \det \begin{vmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \\ \partial_x & \partial_y & \partial_z \\ A_x & A_y & A_z \end{vmatrix}
\]  

(2.6)

When the determinant (det) of Eq (2.6) is found, it is
expressed as (59:556):

\[
\text{curl } A = \left( \frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z} \right) a_x + \left( \frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x} \right) a_y + \left( \frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y} \right) a_z \tag{2.7}
\]

In general, 'A' may also be a function of time.

Substituting E or H for A into Eq (2.7), replacing the curls in Eq (2.1) and Eq (2.2) with the results from Eq (2.7), and lastly separating the vector components yields:

\[
\begin{align*}
\mu \frac{\partial H_x}{\partial t} &= -\frac{\partial E_z}{\partial y} + \frac{\partial E_y}{\partial z} \quad (2.8a) \\
\mu \frac{\partial H_y}{\partial t} &= -\frac{\partial E_x}{\partial z} + \frac{\partial E_z}{\partial x} \quad (2.8b) \\
\mu \frac{\partial H_z}{\partial t} &= -\frac{\partial E_y}{\partial x} + \frac{\partial E_x}{\partial y} \quad (2.8c) \\
\varepsilon \frac{\partial E_x}{\partial t} + \sigma E_x &= \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \quad (2.9a) \\
\varepsilon \frac{\partial E_y}{\partial t} + \sigma E_y &= \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \quad (2.9b) \\
\varepsilon \frac{\partial E_z}{\partial t} + \sigma E_z &= \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \quad (2.9c)
\end{align*}
\]
These equations are now in a form convenient for use in the algorithm of the finite difference computer code.

The derivatives in Eq (2.8) and Eq (2.9) are replaced by a finite difference approximation. The exact definition of a derivative using forward differencing is (25:289; 7:297):

\[
\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} = \frac{f(x+h) - f(x)}{h} \quad (2.10)
\]

The finite differencing approximation used in this work is the central differencing approximation defined by (7:298):

\[
\frac{df(x)}{dx} \approx \frac{f(x+h/2) - f(x-h/2)}{h} \quad (2.11)
\]

and is illustrated graphically in Figure 2.
By using a Taylor series approximation of Eq (2.11) and Eq (2.12), the errors can be compared (32; 16). The result is that the central differencing approximation is a second order approximation as opposed to forward differencing, which is a first order approximation (32:297, 298). Both of the codes (Appendix A and Appendix B) were run using the central differencing approximations for each derivative.
Field Locations

The points \((i,j,k)\) which describe the location of the fields in each cell space are reference points on a decentralizing mesh scheme. This decentralizing mesh scheme is best described by first considering the one-dimensional decentralizing mesh (Figure 3). Note that the E-fields and H-fields are not co-located. This displacement keeps one from knowing the E-field and H-field components at the same point in space. The fields are only known at 1/2 the differential distance \((\Delta x/2,\Delta y/2,\Delta z/2)\) in any particular coordinate system direction. Also note that the E-field and H-field are not known at the same time, but that they are separated by 1/2 the time step \((\Delta t/2)\).
To form the derivatives required in Eq (2.8) and Eq (2.9), the dH/dt and dE/dt are needed. As demonstrated graphically in Figure 3, these values are available using this decentralizing mesh concept.
Three-Dimensional Finite Difference Equations

The Three-Dimensional Finite Difference Equations can be derived by combining the differential form of Maxwell's equations, the definition of curl in the Cartesian coordinate system, and the central differencing approximation to a derivative.

Magnetic-Field Algorithm Development. An example in terms of $H_x$ would combine Eq (2.8a) and Eq (2.11) at a point specified as:

$$\{(i-1/2)\Delta x, j\Delta y, k\Delta z, (n-1/2)\Delta t\}$$

and yield:

$$H_x((i-1/2)\Delta x, j\Delta y, k\Delta z, n\Delta t) - H_x((i-1/2)\Delta x, j\Delta y, k\Delta z, (n-1)\Delta t)$$

$$\Delta t$$

$$= \frac{E_y((i-1/2)\Delta x, j\Delta y, (k+1/2)\Delta z, (n-1/2)\Delta t)}{\mu\Delta z}$$

$$- \frac{E_y((i-1/2)\Delta x, j\Delta y, (k-1/2)\Delta z, (n-1/2)\Delta t)}{\mu\Delta z}$$

$$- \frac{E_z((i-1/2)\Delta x, (j+1/2)\Delta y, k\Delta z, (n-1/2)\Delta t)}{\mu\Delta y}$$

$$+ \frac{E_z((i-1/2)\Delta x, (j-1/2)\Delta y, k\Delta z, (n-1/2)\Delta t)}{\mu\Delta y}$$

(2.12)

In using this equation, an approximation for $H_x$ can be obtained by knowing $E_y$ and $E_z$ at $1/2$ a space increment on
either side of $H_x$, at $\Delta t/2$ earlier and $H_x$ one $\Delta t$ earlier. Figure 4 illustrates these relationships. Taflove and Umashankar have shown that this approximation is within the accuracy of computer calculations (54; 62).

![Diagram](image)

**Figure 4. Location of Difference Fields**

Note: The origin is centered at $((i-1/2)\Delta x, j\Delta y, k\Delta z)$, at the time $n\Delta t$.

Next, the E-field can be determined in a similar manner $\Delta t/2$ later at $1/2$ a space increment in the x direction. The program steps through the decentralizing mesh in half-space increments and half-time increments until the entire grid space is covered ($NxNxN$).
Similar modifications transform Eq (2.8b) and Eq (2.8c) into

\[
H_y [iAx, (j-1/2)Ay, kAz, nAt] - H_y [iAx, (j-1/2)Ay, kAz, (n-1/2)At] \Delta t
= \frac{E_z [(i+1/2)Ax, (j-1/2)Ay, kAz, (n-1/2)At]}{\mu \Delta x} - \frac{E_z [(i-1/2)Ax, (j-1/2)Ay, kAz, (n-1/2)At]}{\mu \Delta x} + \frac{E_x [iAx, (j-1/2)Ay, (k+1/2)Az, (n-1/2)At]}{\mu \Delta z} - \frac{E_x [iAx, (j-1/2)Ay, (k-1/2)Az, (n-1/2)At]}{\mu \Delta z}
\]

(2.13)

at point location \( [iAx, (j-1/2)Ay, kAz, (n-1/2)At] \) and

\[
H_z [iAx, jAy, (k-1/2)Az, nAt] - H_z [iAx, jAy, (k-1/2)Az, (n-1/2)At] \Delta t
= \frac{E_x [iAx, (j+1/2)Ay, (k-1/2)Az, (n-1/2)At]}{\mu \Delta y} - \frac{E_x [iAx, (j-1/2)Ay, (k-1/2)Az, (n-1/2)At]}{\mu \Delta y} + \frac{E_y [(i+1/2)Ax, jAy, (k-1/2)Az, (n-1/2)At]}{\mu \Delta x} - \frac{E_y [(i-1/2)Ax, jAy, (k-1/2)Az, (n-1/2)At]}{\mu \Delta x}
\]

(2.14)
Electric Field Algorithm Development. Operating on the E-field in Eq (2.9a) at grid point

\[(i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z, n\Delta t\]

\(E_x\) is first needed at \(t = (n+1/2)\Delta t\). Using the average of \(E_x\) at \(\Delta t/2\), before and after \(t\),

\[
E_x(n\Delta t) = \frac{E_x((n+1/2)\Delta t) + E_x((n-1/2)\Delta t)}{2}
\]  

(2.15)

substituting in the space coordinates, one finds

\[
E_x((i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z, n\Delta t)
\]

\[
= \frac{E_x((i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z, (n+1/2)\Delta t)}{2}
\]  

(2.16)

\[
+ \frac{E_x((i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z, (n-1/2)\Delta t)}{2}
\]

and substituting into Eq (2.9a), then expanding the right side as was done in Eq (2.8), the result is
\[
\frac{(\epsilon/\Delta t + \sigma/2)}{E_x}\{(i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z, (n+1/2)\Delta t\} - \frac{(\epsilon/\Delta t - \sigma/2)}{E_x}\{(i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z, (n-1/2)\Delta t\}
\]

\[
= \frac{H_z\{(i-1)\Delta x, j\Delta y, (k-1/2)\Delta z, n\Delta t\}}{\Delta y} - \frac{H_z\{(i-1)\Delta x, (j-1)\Delta y, (k-1/2)\Delta z, n\Delta t\}}{\Delta y} + \frac{H_y\{(i-1)\Delta x, (j-1/2)\Delta y, (k-1)\Delta z, n\Delta t\}}{\Delta z} + \frac{H_y\{(i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z, n\Delta t\}}{\Delta z}
\]

(2.17)

Now sigma, the conductivity, is in the equation. This is the area, that if one were dealing with something other than a perfect conductor, changes in the conductivity could be made. In the data base, one can express the conductivity at each grid point. In the program it is averaged in this manner:

\[
\sigma_x = \sigma\{(i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z\}
\]

(2.18)

\[
= \frac{\sigma\{(i-1/2)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z\}}{2} + \frac{\sigma\{(i-3/2)\Delta x, (j-3/2)\Delta y, (k-1/2)\Delta z\}}{2}
\]

26
And epsilon, the permittivity is

$$\varepsilon_x = \varepsilon \{(i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z\} \quad (2.19)$$

Similar operations for $E_y$ at

$$[(i-1/2)\Delta x, (j-1)\Delta y, (k-1/2)\Delta z, n\Delta t]$$

yields

$$(\varepsilon/\Delta t + \sigma/2)(E_y \{(i-1/2)\Delta x, (j-1)\Delta y, (k-1/2)\Delta z, (n+1/2)\Delta t\})$$

$$- (\varepsilon/\Delta t - \sigma/2)(E_y \{(i-1/2)\Delta x, (j-1)\Delta y, (k-1/2)\Delta z, (n-1/2)\Delta t\})$$

$$= \frac{H_x \{(i-1/2)\Delta x, (j-1)\Delta y, (k+1)\Delta z, n\Delta t\}}{\Delta z}$$

$$- \frac{H_x \{(i-1/2)\Delta x, (j-1)\Delta y, (k-1)\Delta z, n\Delta t\}}{\Delta z}$$

$$- \frac{H_z \{(i-1)\Delta x, (j-1)\Delta y, (k-1)\Delta z, n\Delta t\}}{\Delta x}$$

$$+ \frac{H_z \{(i-1)\Delta x, (j-1)\Delta y, (k-1)\Delta z, n\Delta t\}}{\Delta x} \quad (2.20)$$

and for $E_z$ at $$[(i-1/2)\Delta x, (j-1/2)\Delta y, (k-1)\Delta z, n\Delta t]$$
\[
\begin{align*}
\left( \frac{\epsilon}{\Delta t} + \sigma/2 \right) & \left( E_z \right)_{(i-1/2, j, k, (n+1/2) \Delta t)} \\
- \left( \frac{\epsilon}{\Delta t} - \sigma/2 \right) & \left( E_z \right)_{(i-1/2, j, k, (n-1/2) \Delta t)} \\
= & \frac{H_y \left[ (i, j-1/2, k, n \Delta t) \right]}{\Delta x} \\
- & \frac{H_y \left[ (i, j, k-1, n \Delta t) \right]}{\Delta x} \\
+ & \frac{H_x \left[ (i-1/2, j, k, n \Delta t) \right]}{\Delta y} \\
+ & \frac{H_x \left[ (i-1/2, j, k, n \Delta t) \right]}{\Delta y}
\end{align*}
\]

Finally, in rearranging the electric field equation and the magnetic field equation, one arrives at a form which is useful in understanding the relationship between various fields and the code. For example:

\[
H_x \left[ i, j, k, n \Delta t \right] = H_x \left[ i, j, k, (n-1) \Delta t \right] \\
+ \left( \frac{\Delta t}{\mu} \right) \left( E_y \right)_{(i, j, (k+1/2), (n-1/2) \Delta t)} \\
- \frac{E_y \left[ i, j, (k-1/2), (n-1/2) \Delta t \right]}{\Delta z} \\
- \frac{E_z \left[ i, (j+1/2), k, (n-1/2) \Delta t \right]}{\Delta y} \\
+ \frac{E_z \left[ i, (j-1/2), k, (n-1/2) \Delta t \right]}{\Delta y}
\]

(2.22)
and

\[
E_x[i, j, k, (n+1/2)\Delta t] = E_x[i, j, k, (n-1/2)\Delta t] \left( \frac{\varepsilon/\Delta t - \sigma/2}{\varepsilon/\Delta t + \sigma/2} \right) + \left( \frac{1}{(\varepsilon/\Delta t + \sigma/2)} \right) \frac{H_z[i, (j+1/2), k, n\Delta t]}{\Delta y} \text{(2.23)}
\]

\[
- \frac{H_z[i, (j-1/2), k, n\Delta t]}{\Delta y} + \frac{H_y[i, j, (k+1/2), n\Delta t]}{\Delta z} + \frac{H_y[i, j, (k-1/2), n\Delta t]}{\Delta z}
\]

In this manner, the \( E_x \) can be calculated knowing the \( H \)-field \( \Delta t/2 \) earlier at the four adjacent points in the orthogonal plane, \( \Delta y/2 \) and \( \Delta z/2 \) away, and the \( E_x \) \( \Delta t/2 \) earlier at the same point. A similar statement can be made about \( H_x \), Eq (2.23).

Thus, at every time step (\( \Delta t/2 \)), a complete set of either \( E \) or \( H \) fields is calculated for the entire problem space. This process is continued in time until a steady state response is reached.

For computational stability, \( \Delta t \) must satisfy the Courant stability condition (27:329; 64:303),
\[ \Delta t < \frac{1}{V_{\text{max}} (\Delta x^{-2} + \Delta y^{-2} + \Delta z^{-2})^{1/2}} \] (2.24)

where \( V_{\text{max}} \) is the maximum velocity of propagation in the medium. Eq. (2.24) means that the increment of the time steps must be smaller than the time it takes the wave to travel. This keeps the magnitude of the field finite differences small, so that the solution is stable. In other words, the finite differences are small because the passage of time was small enough to allow only small changes in magnitudes.
3. **Boundary Conditions**

Boundary conditions are necessary because space in general is infinite, but the computer has to have a finite number of values to make its calculations. An infinite space is simulated by truncating the problem space with carefully chosen boundary conditions. Examining Eq (2.22) and (2.23), it can be shown (as in Hebert's development of the propagation in one direction) that some type of boundary condition is absolutely necessary for the finite difference code (16). Without boundary conditions, the fields at the problem space boundary would be undefined and would not allow the calculation of the next field when stepping through the space. The end result is a decreasing data base of past fields from which to calculate the subsequent solutions.

There are numerous boundary conditions imposed by different authors in the algorithms of the finite difference codes. The first author that developed the finite difference algorithm, Yee, simply forced the outer problem space boundary to be a perfect conductor (64). This causes 'noise' in the solution, due to the reflections off the boundary (55:626). Others use a combination of Yee's, "hard" boundary conditions and "soft" ones, which reduce the reflection problem (55:626). The two "soft" boundary conditions considered are the absorption and the radiation boundary conditions.
Absorption Boundary Condition

The conductivity of the problem space is set to zero in the center, but at the outer edges it is set to some small value, such as that of distilled water (Figure 5). This has the effect of "absorbing" the fields to the point that the reflected waves from the outermost boundary will not perturb the solution of the fields in the center of the problem space.

\[
\begin{array}{c|c}
\sigma = 0 & \sigma = 10^{-4} \\
\hline
\cdot & \cdot \\
+ & + \\
\cdot & \cdot \\
\end{array}
\]

Figure 5. Finite Conductivity at Boundary Edges

This absorption boundary condition requires approximately one wavelength's distance to the boundary to simulate a low reflectivity surface (55:626). This additional size added onto the problem space greatly increases the computer memory.
and time for calculations (55:626). This particular boundary condition was used in Rymes' code (48).

**Radiation Boundary Condition**

The radiation boundary condition uses a far-field approximation and the radiation condition to emulate an ideal infinite problem space. The general form of the radiation condition was originally introduced by Merewether (38:41). Bayless and Turkel have shown that the radiation boundary condition is a very valid mathematical technique (4). The form of the radiation boundary condition is: some function of time \( f(t-r/c) \) divided by radial distance, \( r \), from the center of the problem space (38:41).

\[
E = \frac{f(t-r/c)}{r} \tag{3.1}
\]

where

- \( f \) = causal vector function
- \( t \) = time
- \( c \) = speed of light
- \( r \) = large distance from the center of the test object

The function, 'f', used by Merewether will be the one incorporated into the modified Rymes' code (38:42). The fields at the boundary are found by parabolic interpolation.
in time, of the fields that are one cell in from the outer boundary, at \( n\Delta t \), \((n-1)\Delta t \) and \((n+1)\Delta t \). This is expressed as \( E_z \) at the outer boundary point \((i,j,k)\), maximum \(j\)-plane boundary, in the \(y\)-direction as:

\[
E_z^{n+1}(i,j,k) = \frac{R_z(i,j-1,k)}{2R_z(i,j,k)} f[\theta,E(t)]
\]  

(3.2)

where

\[
f[\theta,E(t)] = \theta_{zy}(\theta_{zy} - 1)E_z^{n-1}(i,j-1,k) + 2(1 - \theta_{zy}^2)E_z^n(i,j-1,k) + \theta_{zy}(\theta_{zy} + 1)E_z^{n+1}(i,j-1,k)
\]

\[
R_z(i,j,k) = \left(x(i)^2 + y(j)^2 + z(k)^2\right)^{1/2}
\]

\[
\theta_{zy} = 1 - \frac{R_z(i,j,k) - R_z(i,j-1,k)}{c\Delta t}
\]

and the \(x,y,z\) position coordinates for the outer boundaries are such that the coordinate system's origin is in the center of the problem space. The 'c' here is taken to be the speed of light in free space, and \( \Delta t \) is the time elapsed since last grid position. Similarly, \( E_x \) can be expressed for this boundary. Two components, for each of the remaining five sides of the problem space, are also expressed in the same way. Eq (3.2) works for a constant
cell size problem space. More elaborate expressions are required for an expanding cell size grid (20:2419).

Implementing Eq.(3.2) into code turned out to be a major task. In fact, half the time spent on this thesis was devoted to simply understanding this radiation boundary condition. Researching the radiation boundary condition's origin, and then implementing it into the 3DFD computer code, was a much larger task than originally estimated. References were not found which specifically covered a non-uniform problem space. A non-uniform problem space in this text is one with three different size sides to each element's rectangular box. A major clue was found in a report by Merewether and Fisher (39:74). Looking at Eq.(3.2), the expression for \( \theta \) contains a '1' and a \( c \Delta t \). The expression directly coded into FORTRAN gave disastrous results.

This direct programming caused the calculated field magnitudes to blow up. In looking into the problem, it became obvious that \( c \Delta t \) had to be different for each direction considered. This was taken care of by realizing that \( c \Delta t \) was the distance traveled in the increment of time in a particular direction. This could simply be replaced by the space increment dimension over which the derivative was being considered. Note that the '1' now needs to represent an integer one higher than the quantity being subtracted away in the expression for \( \theta \) (7:120). If this last
precaution is not taken, the sign of $\Theta$ varies and the parabolic interpolation is incorrect. Detailed expressions can be found in Appendix B.

A smoother waveform is the expected result when using the radiation boundary conditions instead of the absorption boundary conditions. The reflected waves which are caused by the discontinuities at the changes in problem space conductivity, are expected to be reduced. Because of this, the interaction of the reflected waves should be greatly reduced or eliminated. In general, the reflective waves can be thought of as destructive and constructive interference of the output/resulting waveform. This interference is believed to be the cause of the abnormal magnitude variations. Therefore, reducing this destructive and constructive interference caused by the reflected waves should result in a smoother-appearing output waveform.
4. Modeling the F-16

The F-16 'Fighting Falcon' was chosen as the aircraft to be modeled in this thesis. The F-16 was chosen because of the current lightning susceptibility testing being conducted. The thesis sponsor, AFWAL/FIESL, was performing lightning strike tests on the new LANTIRN navigation pod installed on a F-16 (9). Another reason for the choice of the F-16 was due to its being the first operational fighter with a "fly-by-wire" flight control system. The high interest in the survivability of the "fly-by-wire" system after a lightning strike caused many studies to be performed, supplemented with many on-going efforts to protect this system (6). A large data base was therefore available for follow-on comparisons and studies. First, the problem space will be described in general. Then the choices which were made to model the F-16 in the time-domain finite difference problem space, will be explained. The last area to be covered will be the location of the ten sample points.
Problem Space

The problem space is made up of a rectangular area subdivided into 19,683 rectangular cells. The positions within this problem space are defined in terms of the Cartesian coordinate system. This particular choice of coordinate systems is well-suited to the finite difference formulation previously mentioned. The space is divided into rectangular grids referenced by integers such as \((i,j,k) = (1,1,1)\). The particular length of each grid is chosen to enhance the description of the geometry of the subject object. Each of the three directions is divided into the same number of grids, although this is not required. The best way to describe this is by an example. The program used for this thesis incorporates a 27X27X27 (=19,683) grid space. In meters, the x-direction increment is 0.69 meters, y-direction is 0.22 meters, and the z-direction is 0.45 meters (Figure 6). Since the grids remain a uniform size in a given direction, this makes the overall physical size of the grid space 18.63, 5.94, and 12.15 meters respectively, in the x,y,z directions.
Figure 6. Basic Grid Block

The constraint of memory size and processing time are the main limiting factors for the number of grid divisions used (e.g., 27X27X27). Each of the 19,683 grid points has six fields associated with it. After one complete run of the program through the problem space, 118,098 fields have been calculated. This run through the problem space occurs at every time increment (Δt/2). The time step in this thesis was 0.366667 nanoseconds. A valid run may include up to two microseconds of data. More time results in large numerical errors (13:50). Two microseconds corresponds to 5455 complete cycles through the problem space. Clearly, this is already amounting to large amounts of memory and
central processor (CPU) time. When the amount of calculations per cycle is considered, the problem becomes evident.

The finest detail needed to be represented on the model determines the increment size in each direction. Complete descriptions can be found on the problem space in most finite difference users' manuals (18; 20; 29; 48; 60). These same users' manuals will contain information on implementing the most efficient dimensions for the problem space and element blocks. The 27 cubed problem space was used in comparing both the radiation and absorption boundary conditions. The aircraft was allowed to take up a large portion of the problem space.

The remainder of the problem space left a minimal number of rectangular blocks beyond the aircraft dimensions. The blocks outside the aircraft's dimensions are used for the boundary conditions. Using a minimum number of blocks is not a good practice in general, but was intentionally done to exercise the boundary conditions under a worst case scenario. In particular, only three blocks were used to implement the boundary conditions. The aircraft's maximum dimensions were extended into an aircraft grid space 21X21X21, centered in the 27X27X27 problem space. This leaves the 6 blocks (3 on each face of the box) in each direction for use in implementing the boundary conditions.
Field Locations

Each corner of a rectangular box in the problem space defines a set of fields (Figure 7). The 6 fields described are the 3 electric fields and the 3 magnetic fields in each vector direction. The fields are only defined by these points; they are not actually located there. The E-fields defined by point \((i,j,k)\) are perpendicular and centered on the three sides of the rectangular box, not touching the defining point. The H-fields lie on the edges of these same sides most distant from the defining point. The directional nature of these 6 fields is as established by the coordinate system used (Figure 7).

Figure 7. Field Locations with the Decentralizing Grid System
Cell Size Choices Made for the F-16

The F-16 is a very smooth, aerodynamically designed aircraft. This aerodynamic design consists of many curved surfaces (Figure 8). These curved surfaces presented a challenge to model with the rectangular blocks of the Cartesian coordinate system in the problem space. The F-16 has an overall length of 15.09 meters, width of 9.45 meters, and a height of 4.6 meters (Figure 8) (9:FO-3). The x coordinate is associated with the length of the aircraft (Figure 9). The x direction increment of 0.69 meters was chosen as a compromise to better model the geometrical structure of the wings in the xz plane (Figure 9). The y coordinate is associated with the height of the aircraft (Figure 10). The y direction increment of 0.22 meters was selected primarily to model the wing root area of the F-16 (Figure 10). The z coordinate is associated with the width of the aircraft (Figure 11). The z direction increment of 0.45 meters was chosen to model the detail of the speed brake and the base of the vertical stabilizer. The result of these choices was a good geometrical model which matches the F-16's predominant details (Figure 12).
Figure 8. F-16 Fighting Falcon
Figure 9. Gridding of the F-16, Side View
Figure 10. Gridding of the F-16, Front View
Figure 11. Gridding of the F-16, Top View
The aircraft is described to the computer in subroutine 'AIRPLN' (Appendix A). The description of the geometry of the aircraft is accomplished by defining the tangential E-
fields of the surfaces. The building of the model (Figure 13) was an extremely important step in defining the location of the tangential surface fields. Each field must be assigned by grid space location and magnitude. The tangential surface E-fields, surface normal H-fields and all interior fields are set equal to zero in the case of a perfectly conducting surface. Refer to Appendixes A and B for listings of all of the computer codes used, including the subroutine AIRPLN. A useful device for detailing the field locations was a clear plexiglas cube, with the field locations accordingly marked such as Figure 1. Another technique used in modeling the F-16 was to take the wood and metal model and position the xz-planes, or layers, of the mock-up on a scaled grid plane. This enabled actually seeing where the fields were physically located. The coding of the subroutine AIRPLN and understanding the field locations was greatly simplified with this technique.
Figure 13. Photograph of F-16 Block Model
5. Program Details

The modified Rymes' code was reviewed in great detail. Some changes were made to the modified code before running the F-16 with the absorption boundary conditions. The code was then modified to include the radiation boundary conditions. The sensor locations were selected and encoded in subroutines 'EADV' and 'HADV'. Lastly, the source was updated to reflect parameters recently measured in flight (8; 17; 47; 49).

Basic Code

The assumptions made about the modified Rymes' 3DFD code were basically sound. In reviewing the modified code, only a few minor omissions and corrections were made. The results after complete computer runs on the F-16 model were as expected. The changes made to the modified code with the absorption boundary conditions enhanced the results (Appendix C). A completely updated and corrected copy of the code is in Appendix A.
Boundary Conditions

After running the code with the absorption boundary condition, the radiation boundary condition was implemented. The specific listing can be found in Appendix B, subroutine 'RADBC' and 'HADV'. A significant amount of time was spent in transforming the rather simple-appearing parabolic interpolation of (Eq 3.2) into Fortran 77 code. The basic problem was caused by a lack of details in the literature. The missing details were definitions of time increments in relation to the three different dimensions of the basic problem space block (Figure 6). In particular, the calculation of \( \Theta \) is very sensitive to the definition of \( c \Delta t \) (Eq 3.2).

Sample Point Locations

Ten locations were selected as typical for field sensor layouts on the aircraft. Similar locations were used in a CV-580 flight test conducted by AFWAL (16). Any of the three orientations of either the E-field or the H-field could have been selected at any point in the problem space for monitoring. The selections made are detailed in Table 1.
TABLE 1. Ten Sampled Points

<table>
<thead>
<tr>
<th>Sensor #</th>
<th>Type H or E</th>
<th>Location Description</th>
<th>Direction Component</th>
<th>Coordinates (i,j,k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H-field</td>
<td>right wing</td>
<td>x</td>
<td>(18,10,6)</td>
</tr>
<tr>
<td>2</td>
<td>H-field</td>
<td>nose</td>
<td>z</td>
<td>(4,10,13)</td>
</tr>
<tr>
<td>3</td>
<td>H-field</td>
<td>engine feathers</td>
<td>z</td>
<td>(24,11,13)</td>
</tr>
<tr>
<td>4</td>
<td>H-field</td>
<td>bottom fuselage</td>
<td>z</td>
<td>(10,4,14)</td>
</tr>
<tr>
<td>5</td>
<td>H-field</td>
<td>bottom fuselage</td>
<td>z</td>
<td>(19,4,14)</td>
</tr>
<tr>
<td>6</td>
<td>H-field</td>
<td>vertical stab</td>
<td>x</td>
<td>(22,18,14)</td>
</tr>
<tr>
<td>7</td>
<td>H-field</td>
<td>left wing</td>
<td>x</td>
<td>(18,10,23)</td>
</tr>
<tr>
<td>8</td>
<td>E-field</td>
<td>right wing</td>
<td>y</td>
<td>(17,11,10)</td>
</tr>
<tr>
<td>9</td>
<td>E-field</td>
<td>top fuselage</td>
<td>y</td>
<td>(15,13,14)</td>
</tr>
<tr>
<td>10</td>
<td>E-field</td>
<td>left wing</td>
<td>y</td>
<td>(17,11,18)</td>
</tr>
</tbody>
</table>

The sensor locations are shown in Figure 14. Appendix A, subroutine 'EADV' and 'HADV' show the code used to sample these points at each time step. This sampling will be found under 'sample points of interest' at the end of both subroutines.
Figure 14. Sensor Locations on a Block F-16 Model
Source Function

A couple of the changes made to the code were in the source function. The "transparent" source function utilized in the modified Rymes' code caused some unnatural growth in the response (19). The so-called "transparent" source added the scattered fields to the source fields, resulting in an ever-increasing source function. Modifications made in the code appearing in Appendix A corrected this growth and gave much better results. The risetime (10% to 90% of the maximum amplitude) was changed to 100 nanoseconds (Figure 15). This was found, by AFWAL measurements, to be more characteristic of cloud-to-cloud lightning strikes (8:128). The normalized source function is plotted in Figure 16 out to the 50% falloff point, which occurs at 20.4 microseconds.

Numerous programs, or researchers, use this double exponential as the source function, which is sometimes called the 'Bruce-Golde' model (8:36). This source function emulates a lightning strike's characteristic waveform (19). The source is attached to the aircraft with a surface current injection technique, as described by Kunz (30:1423). This surface current injection technique is basically accomplished by inserting an H-field into the grid space boundary conditions. The insertion of the H-field takes place around the point that the lightning attachment is being simulated. The H-field is approximated from the
relationship, \( H = I/(2r) \); where 'I' is the time-varying, analytical current model for the lightning channel desired, and 'r' is the radial distance from the point of desired injection to the nearest H-fields surrounding that point (24:165).

![Graph showing 10% to 90% risetime of 100 nanoseconds](image)

**Figure 15. Expanded Plot of Source (First 250 nanoseconds)**
Figure 16. Normalized Double Exponential Source Function
6. Analysis

The result of implementing the radiation boundary conditions appears to be an improvement over the absorption boundary conditions results. The absorption boundary condition results contain many fluctuations in amplitude that can be attributed to the boundary conditions (19; 55). When the waves travel and hit the discontinuity at the element rectangular cells which have increased conductivity, they reflect back with a discrete magnitude. These reflected waves cause the additional amplitude fluctuations not found with the radiation boundary condition. All ten sensors showed the same results: a general smoothing of the amplitude of the response on the aircraft's skin (Appendix C). All the field plots (Appendix C) had the same basic shape before and after the boundary conditions were changed. The first 300 nanoseconds of each field's response are plotted in this chapter (Figure 17 thru Figure 26). Each sensor's response is plotted on the same graph for both absorption boundary conditions and radiation boundary conditions. The absorption boundary condition plots are annotated with an '*'. The smoother response of the radiation boundary condition resembles the MIE solutions as calculated by Holland (20:206). Note that the disparity in the polarity of the E-field response (sensor 8 & 10) was possibly caused by a resonance effect in the wing region.
Figure 17. Sensor One, H-field, Right Wing
Figure 18. Sensor Two, H-field, Nose
Figure 19. Sensor Three, H-field, Engine Burner Can
Figure 20. Sensor Four, H-field, Forward Fuselage Bottom
Figure 21. Sensor Five, H-field, Rear Fuselage Bottom
Figure 22. Sensor Six, H-field, Vertical Stabilizer
Figure 23. Sensor Seven, H-field, Left Wing
Figure 24. Sensor Eight, E-field, Right Wing
Figure 25. Sensor Nine, E-field, Middle Fuselage Top
Figure 26. Sensor Ten, E-field, Left Wing
Waveform Appearance

The general appearance of the responses from the different boundary conditions was dramatic. The amplitude oscillation frequency of the radiation boundary condition response was less distinct than that of the absorption boundary condition. As Figure 17 thru Figure 26 display, the radiation condition gave a much smoother response. The overall response for the entire two microseconds of each of the computer runs was consistent. In each and every graph of the response, the radiation boundary condition curve was smoother and settled to a steady state response much quicker than that of the absorption boundary condition.

Note also that the amplitude, in general, of the radiation boundary curves are higher than those of the absorption boundary condition. This lower absorption amplitude was due in part to the source itself being absorbed. That is, the source actually goes through a couple of element blocks which have a finite conductivity. However, a small portion of the amplitude differences could be due to constructive and destructive interference of the reflected and surface waves.
The absorption boundary condition's H-field response of Figure 17 has an average period of 3.6 nanoseconds, which corresponds to a wavelength of 1.08 meters. The radiation boundary condition's H-field response for the same time span is different (Figure 17). It has a average period of 7.2 nanoseconds, which corresponds to a wavelength of 2.16 meters. Using the aircraft's length as the predominant resonance structure, the wavelength of the response calculated for radiation boundary conditions was a quarter wavelength multiple of the aircraft's length. The absorption boundary condition did not have this same relationship, as its multiple was in thirds of the aircraft's length. This is only a general conclusion. Time did not permit taking the major surface dimensions of each sensor's local area, and correlating the calculated field response's frequency to the major local surface dimensions. But, the result is as expected; that is, the major geometrical object's characteristic wavelength will dominate. More time would have permitted looking at the problem space's dominant cavity frequency, and its contribution to the response curve's shape.
Program Considerations

Implementing the radiation boundary conditions has many effects. One positive effect was just discussed, that of smoothing the response waveform. Other effects, such as increased storage and run time, are not desirable. The amount of storage for running the program of Appendix A with its absorption boundary condition is doubled when the radiation boundary condition is implemented. An additional 363 lines of code were added to the program, to implement the radiation boundary condition. The worst effect was the CPU time per 100 nanoseconds of program time increased by 25% for the radiation boundary condition over the absorption boundary condition. Table 2 contains some details from the CDC Cyber 845 after the first 100 nanoseconds of data was calculated. No claim is made that any code optimization for run time, speed or memory utilization was attempted. However these times are comparable to those given in Eriksen's report (13:50). This is a first attempt at a comparison on the same code with only the boundary condition changed. The similarity of the plots at each sensor alone, support the validity of the program as modified with radiation boundary conditions.
TABLE 2. CDC Cyber 845 Computer Run Data

<table>
<thead>
<tr>
<th></th>
<th>Absorption Boundary Condition</th>
<th>Radiation Boundary Condition</th>
<th>Percent Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time elapsed within program</td>
<td>100 ns</td>
<td>100 ns</td>
<td>-</td>
</tr>
<tr>
<td>SRU's CDC 845</td>
<td>8388.6</td>
<td>11534.3</td>
<td>37.5</td>
</tr>
<tr>
<td>CPU seconds execution time</td>
<td>3471.4</td>
<td>4341.2</td>
<td>25.1</td>
</tr>
<tr>
<td>Number cycles through program</td>
<td>272</td>
<td>272</td>
<td>-</td>
</tr>
<tr>
<td>CPU cost @ $320/hour</td>
<td>$308.57</td>
<td>$385.88</td>
<td>25.1</td>
</tr>
<tr>
<td>Cost per cycle (0.1833 ns)</td>
<td>$1.13</td>
<td>$1.42</td>
<td>25.1</td>
</tr>
</tbody>
</table>

Conclusion

The smoother response curves of the radiation boundary condition are quite clear. Assuming this smoothing is due to ridding the solution of unwanted resonances by changing the boundary conditions, a more accurate result has been accomplished. But, as with all things, more study needs to be done before real confidence can be placed in this computer solution. Checks were run on both of the computer programs to verify them. Techniques, such as running the codes without sources, or without an object in the problem space, were performed. The results of the checks of the
program proved that it worked properly in those aspects. But more experimentation needs to be performed. On the surface, the radiation boundary conditions give response curves of the same general shape as found in lightning strike characterization papers. This comparison is made, in general, with other calculation methods of response curve generation. A lot of questions yet to be answered have to do with whether or not the increased cost of this program is justified. Can the analysis be accomplished, with the results of the absorption boundary conditions, to the desired accuracy? Which degree of accuracy is needed? What is the decision point for using one boundary condition over the other?

In comparing the response curves side-by-side, it is noted that the general wave patterns for each sensor are the same. Therefore, implementing the radiation boundary condition did not alter the basic wave pattern. And the radiation boundary condition did smooth the waveforms. This tends to back up the premise that the spurious reflections caused by the absorption boundary condition's changes in conductivity, are seemingly removed by the radiation boundary condition. The objective of ridding the output of spurious reflections seems to have been accomplished.
7. **Summary and Recommendations**

The major objectives of this study have been successfully completed. The code for modeling an F-16 into the Rymes' 3DFD computer code was written and implemented successfully. The 3DFD computer code was run with absorption boundary conditions for two microseconds of response time. The 3DFD code was then rewritten to incorporate the radiation boundary conditions, and run with the radiation boundary conditions for two microseconds of response time. During each complete run, ten points on the surface of the simulated aircraft were sampled. These samples are in the form of magnetic field responses and electric field responses.

The responses for each sample point were compared for the different boundary conditions. The radiation boundary condition produced a much smoother response, apparently accomplishing the task set forth by the sponsor, AFWAL/PIESL. That is, the radiation boundary conditions removed many of the unwanted variations in the amplitude. The major drawbacks of implementing the radiation boundary conditions over the absorption boundary conditions were discussed. The 25% increase in the cost of running the radiation boundary condition program was a major disadvantage. However, the overall results verified that
the radiation boundary conditions do remove unwanted problem space reflections. The sponsor (AFWAL/FIESL) was very pleased with the results.

Future projects to parallel this effort could be numerous. Comparisons should be made of 3DFD output data with measured, actual inflight lightning strikes. Another possibility, would be exploring in greater detail the result of this study for more subtle effects of the different boundary conditions that were not readily apparent. Or researchers could take on the very challenging task of optimizing the code for minimum run time and maximum efficiency in storage use. A detailed examination should be made to explain why the E-fields of sensors 8 and 10, seem to have opposite polarities for the different boundary conditions. An interesting technique would be to use the aircraft symmetry in such a manner that only half the problem need be solved, which, in theory, would reduce the expense by half. Researching combinations of boundary conditions would also be of great interest. One possibility would be to allow a small amount of absorption in the outermost cell, in combination with the radiation condition. Another possibility would be to create variations in the cell dimensions in a given direction, as opposed to adding extra grid cells (which cause cost increases or loss of details in the modeling of the object being studied).
Many studies have been performed in the area of Finite Difference Codes, as attested to by the many listings in the bibliography. However, few real comparisons of the varied techniques are available in any detail. Much can be accomplished with this code. The 3DFD is very flexible and allows cell-by-cell assignment of material properties. This flexibility makes this code an issue of possibly great interest in the study of composite aircraft materials in many electromagnetic pulse environments.

Future implementation on specialized parallel processing computers, and/or the CRAY super computer, would greatly reduce the major disadvantage of the 3DFD code, which is the CPU runtime. Of course, the increased cost of the use of these newer computers must be weighed against the benefits. The last suggestion that will be made in this thesis is that an interactive modeling system be developed for electromagnetic codes in general. Modeling an aircraft is a very time-consuming task. Cooperation with the aeronautical engineers and their detailed drawing capabilities would open new frontiers for the electromagnetic community. The ability to take the aeronautical engineer's detailed structural drawing and converting them into electromagnetic models by software, would be a great stride forward for any electromagnetic computer code utilization.
Updated Modified Rymes' Code

This appendix contains the Modified Rymes' Code, with absorption boundary conditions, written by Hebert (16). The code listing is complete as implemented and runs on a CDC Cyber 845. Improvements are included which correct minor errors in the code mechanics, and an improved source is also implemented. The code compiles on the Cyber 845 using FTN5, and runs with no problem when suppressing ANSI errors. The compile time is approximately 2.603 CPU seconds. The run time is 3,472 CPU seconds for 100 nanoseconds of sample time elapsed. The aircraft modeled in this code is an F-16 'Fighting Falcon'. This code will run as listed only for the first 150 nanoseconds. For continued runs, modifications must be made as annotated in the program. Additionally 'TAPE1' must be saved from the previous run.
PROGRAM ABC (OUTF15,TAPE=OUTF15,TAPE2)

THE MODIFIED RYMES 3-D CODE F16

COMMON /SET1/ MAXI, EPSL, SIGL, C, TMUL, DELT
COMMON /SET2/ ICAN, ICMI, ICPI, JCAN, JCMI, JCP1, KCAN, KCM1, KCP1
COMMON /SET3/ DELX, DELY, DELZ
COMMON /TIME/ TXN, TN
COMMON /ARRAYS/ A1(27,27), B1(39), A2(27,27), B2(39), A3(27,27), B3(39), A4(27,27), B4(39), A5(27,27), B5(39), A6(27,27), B6(39), A7(27,27), B7(39), A8(27,27), B8(39), A9(27,27), B9(39), A10(27,27), B10(39), A11(27,27), B11(39)
DIMENSION INDEX(190)
CALL OPENMS(2, INDEX, 198, 8)

READ INPUT DATA

CALL SETUP

ZERO THE FIELDS INITIALLY.
PROGRAM USES A MASTER STORAGE FILE TO STORE FIELDS AND CONDUCTIVITY
OF EACH LATTICE POINT. INDEX K GOES FROM 1 TO 27.

EX K
EY K+27
EZ K+54(27*2)
HX K+81(27*3)
HY K+108(27*4)
HZ K+125(27*5)
SIG K+162(27*6)

**********************************************************************************
**********************************************************************************

TO REMOVE THE ZEROING OF TAPE 1
FOR CONTINUED RUNS BEYOND THE INITIAL Tmax STOPPING TIME
DO 100 I=1,27
DO 100 J=1,27
100 A1(I,J)=.0
DO 200 K=1,189
200 CALL WRITES(2,A1(I,J),768,K,0)

**********************************************************************************
**********************************************************************************

PRESET THE CONDUCTIVITY ARRAY

CALL SIGSET

TIME STEP LOOP
DO 1000 N=1,NMAX
CALL APERT

77
COMPUTE TIME CONSTANTS

FOR CONTINUED RUNS BEYOND Tmax

ADD THE STOPPING TIME FROM PREVIOUS RUNS TO TIM AND TEN PLUS 1/2 OF DELTA T

EXAMPLE

AFTER FIRST RUN

TIN=DELTA*(FLOAT(N)-1.0)+1.501383E-7
TEN=DELTA*(FLOAT(N)-.5)+1.501383E-7

ADVANCE THE H-FIELDS BY Z-PLANES

CALL HADV

ADVANCE THE E-FIELDS BY Z-PLANES

CALL EADV

CONTINUE

END OF TIME LOOP

CALL CLOSMS(2)

STOP

END

SUBROUTINE SETUP

CODE USES A UNIFORM GRID.

COMMON /SET1/ NMAX, EPSI, EPSR, NGRID, IDATA, DELT
COMMON /SET2/ ICPI, ICHI, ICPI, JCPI, JCM1, JCPI, KCPI, KCPI, KCPI
COMMON /SET3/ DELX, DELY, DELZ

INPUT DATA PROVIDED BY USER.

DATA C, EPSI, EPSR, IMAG, SIGMA, 3.DES, 8.8542E-12, 1.0, 1.2566E-6, 1.E-4/
DATA ICPI, JCPI, KCPI/27,27,27/
DATA DELX, DELY, DELZ/0.025, 0.025, 0.025/
DATA Tmax/1.5E-7/

ICAN=ICPI-1
ICH1=ICAN-1
JCPI=ICAN-1
JCPI=ICAN-1
KCPI=KCPI-1
KCMI=KCAN-1

VERIFY THAT ICPI, JCPI, KCPI ARE ODD.

IF(ICHAN.EQ.2*ICHAN/2). AND. JCHAN.EQ.2*JCHAN/2). AND.
+KCPI.EQ.2*KCHAN/2). GO TO 1

PRINT 1, 'THE NUMBER OF GRID POINTS MUST BE ODD.'

STOP

CONTINUE

DELT IS THE TIME INCREMENT (IT SATISFIES COURANT CONDITION).

NMAX IS THE TOTAL NUMBER OF TIME INCREMENTS FIELDS WOULD BE ADVANCED.

DELT=MIN(MIN(DELX,DELY,DELT)/(2.*C)

NMAX=MAX/DELT

EPST=EPSO+EPS/DELT

TMUO=DELT/XMU

RETURN

END

SUBROUTINE SIGSET

SIG(1,J) AT KI=K+162 IS THE CONDUCTIVITY AT THE POINT (X(I),Y(J),Z(K)).

X(I)=(I-.5)*DELX, Y(J)=(J-.5)*DELY, Z(K)=(K-.5)*DELZ

COMMON /ARRAYS/ SIG(27,27),BI(39)

COMMON /SETI/ NMAX,EPST,SIGO,C,TMUO,DELX

COMMON /SET2/ ICHAN,ICPI,JCPI,JCHAN,KCHAN,KCPI,KCMI

PRESSET THE CONDUCTIVITY TO 1E-4 EVERYWHERE IN SPACE.

DO 10 I=1,ICHAN
   DO 10 J=1,JCHAN
10 SIG(I,J)=SIGO

NOW WRITE THE SIG ARRAY TO MASS STORAGE.

DO 20 K=1,KCHAN
   KI=K+162
20 CALL WRIT1IS(2,SIG(I,1),768,KI)

THEN ESTABLISH FREE SPACE CONDITIONS

ICM2=ICAN-2

JCM2=JCHAN-2

KCM2=KCHAN-2

DO 30 I=3,ICM2
   DO 30 J=3,JCM2
30 SIG(I,J)=0.0

NOW OVERWRITE THIS PORTION OF THE SIG ARRAY

DO 40 K=3,KCM2
40 CALL WRITS(2, SIG1(1, 1), 768, K1, 1)
RETURN
END

SUBROUTINE EADV

C
C THIS SUBROUTINE ADVANCES THE E-FIELDS TWO Z-PLANES AT A TIME.
C RECORDS OF H-FIELDS FOR TWO SUCCESSIVE Z-PLANES MUST BE KEPT IN
C ORDER TO TAKE DERIVATIVES IN THE Z-DIRECTION.
C RECORDS OF SIG FOR TWO SUCCESSIVE Z-PLANES MUST BE KEPT IN ORDER TO TAKE
C AVERAGES IN THE Z-DIRECTION.
C
COMMON /SET1/ NMAX, FMST, SIGO, C, TMUD, DELT
COMMON /SET2/ ICAN, ICMI, ICPI, JCAN, JCMI, JCP1, KCAN, KCMI, KCPI
COMMON /SET3/ DELX, DELY, DELZ
COMMON /TIME/ TEN, TIXH
 COMMON /ARRAYS/ EX(27, 27), B1(39), EY(27, 27), B2(39), EZ(27, 27),
                      EXB(27, 27), HXB(27, 27), HX(27, 27), H2(39),
                      EXA(27, 27), B3(39), EXB(27, 27), B4(39),
                      EYB(27, 27), B5(39), EY(27, 27), B6(39),
                      EZB(27, 27), B7(39), EZA(27, 27), B8(39),
                      EZA(27, 27), B9(39),
                      SISA(27, 27), SIB(39), SICA(27, 27), SIC(39),
                      SIGA(27, 27), SIGB(27, 27), SIG(39),
                      *SISA(27, 27), SIB(39), SICA(27, 27), SIC(39),

C ADVANCE E-FIELDS BY Z-PLANES
C READ IN FIRST H-PLANES
C
CALL READHMS(2, HXA(1, 1), 768, 82)
CALL READHMS(2, HYA(1, 1), 768, 189)

C DO FOR ALL Z-PLANES
C
MMB=8
MMA=9
MMB=10
DO 1000 KLOOP=1, KCAN, 2
K=KLOOP
KPI=K+1
K2=K+27
K3=K+54
K4=K+82
K5=K+109
K6=K+135
K7=K+162

C READ IN SECOND H-PLANES AND OLD E-FIELDS
C
CALL READHMS(2, EX(1, 1), 768, K)
CALL READHMS(2, EY(1, 1), 768, K2)
CALL READHMS(2, EZ(1, 1), 768, K3)
CALL READHMS(2, HXB(1, 1), 768, K4)
CALL READHMS(2, HYB(1, 1), 768, K5)
CALL READHMS(2, HZA(1, 1), 768, K6)
CALL READHMS(2, SIGA(1, 1), 768, K7)

80
DO 200 J=1,JCAN
   JP1=J+1
DO 200 I=2,ICAN
   SIG(X,Y,Z)=.5*(SIG(X-DELX,Y,Z)+SIG(X,Y,Z))
C
   SIG(X,Y,0)=.5*(SIG(X,Y-DELY,Z)+SIG(X,Y,Z))
   SIG(X,Y,Z0)=.5*(SIG(X,Y,Z-DELZ)+SIG(X,Y,Z))
C
   SIG2=.25*(SIG(I,J)+SIG(I-1,J))
   A=EPST-SIG2
   B=I./(EPST+SIG2)
   EX(I,J)=(A*EX(I,J)+HZ(I,JPI)-HZ(I,J))/DELY -
   + (HY(I,J)-HY(I,J))/DELZ)*B
C
200 CONTINUE
   DO 300 I=1,ICAN
      IP1=I+1
   DO 300 J=2,JCAN
      SIG2=.25*(SIG(I,J-1)+SIG(I,J))
      A=EPST-SIG2
      B=I./(EPST+SIG2)
      EY(I,J)=(A*EY(I,J)+HXB(I,J)-HXA(I,J))/DELZ -
      + (HZ(I,JPI)-HZ(I,J))/DELZ)*B
300 CONTINUE
C
   WRITE NEW EX,EY PLANES TO MASS STORAGE
C
   CALL WRITMS(2,EX(I,1),768,K1,1)
   CALL WRITMS(2,EY(I,1),768,K2,1)
   IF (K.EQ.1) GO TO 450
   DO 400 I=1,ICAN
      IP1=I+1
   DO 400 J=2,JCAN
      JP1=J+1
      SIG2=.25*(SIG(I,J)+SIG(I-1,J))
      A=EPST-SIG2
      B=I./(EPST+SIG2)
      EZ(I,J)=(A*EZ(I,J)+HY(I,JPI)-HY(I,J))/DELY -
      + (HZ(I,JPI)-HZ(I,J))/DELY)*B
400 CONTINUE
C
   WRITE NEW EZ PLANE TO MASS STORAGE
C
   CALL WRITMS(2,EZ(I,1),768,K3,1)
C
   SAMPLE POINTS OF INTEREST(ODD K PLANES)
C
   111 FORMAT (14,2(I1,1PIE11.4))
   112 FORMAT (14,2(I1,1PIE11.4),2(I))
C
430 CONTINUE
K=KPI
K1=K1+1
K2=K2+1
K3=K3+1
K4=K4+1
K5=K5+1
K6=K6+1
K7=K7+1

C READ IN SECOND OLD E-PLANES AND NEXT H-PLANES OFER A PLAN

CALL READMS(2, EX(I, J), 768, K)
CALL READMS(2, EY(I, J), 768, K2)
CALL READMS(2, EZ(I, J), 768, K3)
CALL READMS(2, HX(I, J), 768, K4)
CALL READMS(2, HY(I, J), 768, K5)
CALL READMS(2, HZ(I, J), 768, K6)
CALL READMS(*, SIGB(I, J), 768, K7)
DO 600 J = 1, ICAN
   JP1 = J + 1
   DO 600 I = 2, ICAN
      A = EPST - SIG2
      B = 1. / (EPST + SIG2)
      EX(I, J) = (A*EX(I, J) + (HX(I, J) - HYB(I, J))/DELZ -
                  + (HYB(IP1, J) - HYB(I, J))/DELZ)*B
   CONTINUE
DO 700 I = 1, ICAN
   IP1 = I + 1
   DO 700 J = 2, ICAN
      A = EPST - SIG2
      B = 1. / (EPST + SIG2)
      EY(I, J) = (A*EY(I, J) + (HX(I, J) - HYB(I, J))/DELZ -
                  + (HYB(IP1, J) - HYB(I, J))/DELZ)*B
   CONTINUE
DO 800 I = 1, ICAN
   JP1 = J + 1
   DO 800 J = 1, ICAN
      A = EPST - SIG2
      B = 1. / (EPST + SIG2)
      EZ(I, J) = (A*EZ(I, J) + (HYB(IP1, J) - HYB(I, J))/DELX -
                  + (HYB(IP1, J) - HYB(I, J))/DELX)*B
   CONTINUE

C WRITE NEW EX, EY, EZ PLANES TO MASS STORAGE

CALL WRITMS(2, EX(I, J), 768, K, 1)
CALL WRITMS(2, EY(I, J), 768, K2, 1)
CALL WRITMS(2, EZ(I, J), 768, K3, 1)
MICROCOPY RESOLUTION TEST CHART

Resolution test chart for microcopy.
SAMPLE POINTS OF INTEREST (EVEN K PLANES)

IF (K.EQ.10) WRITE (4,111) MNI,TEN,EY(17,11)
IF (K.EQ.11) WRITE (4,112) MNI,TEN,EY(17,11)

IF (K.EQ.14) THEN
   WRITE (4,111) MNI,TEN,EY(15,13)
   ELSE
      END IF

1000 CONTINUE

ZERO THE ELECTRIC FIELD WITHIN THE METAL COMPONENTS OF AIRCRAFT.

CALL AIRPLN
RETURN
END

SUBROUTINE HADV

THIS SUBROUTINE ADVANCES THE H-FIELDS TWO Z-PLANES AT A TIME.
RECORDS OF E-FIELDS FOR TWO SUCCESSIVE PLANES MUST BE KEPT IN ORDER TO TAKE
DERIVATIVES IN THE Z-DIRECTION.

COMMON /SET1/ NMAX,EPST,SlBO,C,TMU,DELT
COMMON /SET2/ ICAN,ICMI,ICPI,JCAN,JCMI,JCPI,KCAN,KCMI,KCP1
COMMON /SET3/ DELX,DELY,DELFZ
COMMON /TIME/ TEN,THN
COMMON /ARRAYS/ EXA(27,27),EXB(27,27),Ey(27,27),B2(39),
                 EYB(27,27),B3(39),EYB(27,27),B4(39),E2A(27,27),B5(39),
                 E2B(27,27),B6(39),H(27,27),B7(39),HY(27,27),B8(39),
                 HZ(27,27),B9(39)

DO FOR ALL Z-PLANES
   MNI=1
   MNI2=2
   MNI3=3
   MNI4=4
   MNI5=5
   MNI6=6
   MNI7=7

DO 900 KLOOP=1,KCNI,2
   K=KLOOP
   KM1=K-1
   K2=K+27
   K3=K+54
   K4=K+81
   K5=K+108
   K6=K+135
   CALL READMS(2,EXB(11,11,768,K)
   CALL READMS(2,EYB(11,11,768,K2)
   CALL READMS(2,HZ(11,11,768,K6)
IF (K.EQ.1) GO TO 356
CALL READS(2,EIB(I,J),768,K3)
CALL READS(2,HX(I,J),768,K4)
CALL READS(2,HY(I,J),768,K5)
DO 200 J=2,JCN
DO 200 I=1,JCN
HX(I,J)=HX(I,J)+TMUD*((EYB(I,J)-EYA(I,J))/DELY -
+ (EIB(I,J)-EIB(I,J-1))/DELY)
200 CONTINUE
DO 300 I=1,JCN
IP=I+1
DO 300 J=1,JCN
HY(IP,J)=HY(IP,J)+TMUD*((EIB(IP,J)-EIB(I,J))/DELX -
+ (EIB(IP,J)-EIB(IP,J))/DELZ)
300 CONTINUE
C C APPROXIMATES H FIELDS IN PLANE PERPENDICULAR TO WIRE
C BY H=1/2*PI*R.
C SOURCE FUNCTION, ODD K REFERENCE PLANES, HY
C CURRENT PULSE MOVING IN X DIRECTION ATTACHES TO THE NOSE OF THE AIRPLANE
C AT (X,Y,Z)=(0,10,14), EXITS BY THE TAIL AT (X,Y,Z)=(22,10,14).
C AT TIME T=0.0 PULSE IS LOCATED AT X0(8).
C IF(K.NE.15) GO TO 1
DO 10 I=2,3
TPULSE=TIN-69*(FLOAT(I)-3.1)/C
IF(TPULSE.LE.8.0) GO TO 10
C X=EXP(-3.5E0A*TPULSE)-EXP(-2.625E7*TPULSE)
C HYS=7.07E3*X
HY(I,J)=-HYS
10 CONTINUE
DO 30 I=25,26
TPULSE=TIN-69*(FLOAT(I)-3.0)/C
IF(TPULSE.LE.0.0) GO TO 30
X=EXP(-3.5E4*TPULSE)-EXP(-2.625E7*TPULSE)
HYS=7.07E3*X
HY(I,10)=-HYS
30 CONTINUE
C C WRITE NEW HX, HY PLANES TO MASS STORAGE
C CALL WRITMS(2,HX(I,J),768,K4,1)
CALL WRITMS(2,HY(I,J),768,K5,1)
350 CONTINUE
DO 400 I=1,JCN
IP=I+1
DO 400 J=1,JCN
JP=J+1
HZ(IP,JP)=HZ(IP,JP)+TMUD*((EIB(IP,JP)-EIB(IP,J))/DELY -
+ (EIB(IP,JP)-EIB(IP,J))/DELY)
CONTINUE
C WRITE NEW Hz PLANE TO MASS STORAGE
C CALL WRHMS(2,Hz(1,1),168,K6,1)
C SAMPLE POINTS OF INTEREST (ODD K PLANES)
C IF (K.EQ.13) WRITE (4,111) MW2,TMN,Hz(1,10)
IF (K.EQ.13) WRITE (4,111) WN3,TMN,Hz(24,11)
IF (K.EQ.23) WRITE (4,111) MW7,TMN,Hx(18,10)
C 112 FORMAT (14,2(1X,IPEII.I.4),2(/))
111 FORMAT (14,2(1X,IPEII.I.4))
C K=K+1
K1=M-1
K2=K2+1
K3=K3+1
K4=K4+1
K5=K5+1
K6=K6+1
C READ IN NEXT PLANES OF E AND H
C CALL READMS(2,E(1,1),768,K1)
CALL READMS(2,E(1,1),768,K2)
CALL READMS(2,E(1,1),768,K3)
CALL READMS(2,E(1,1),768,K4)
CALL READMS(2,H(1,1),768,K5)
CALL READMS(2,H(1,1),768,K6)
DO 600 J=1,JCM1
JP1=J,P1
DO 600 I=1,ICM1
HI(I,JP1)=HI(I,JP1)+TMU*K*(E(1,1,JP1)-E(1,1,JP1))/DEL1 -
+(E(1,1,JP1)-E(1,1,JP1))/DEL1)
600 CONTINUE
DO 700 I=1,ICM1
IPI=I,P1
DO 700 J=1,JCM1
HY(IPI,J)=HY(IPI,J)+TMU*K*(E(1,1,J)-E(1,1,J))/DELX -
+(E(1,1,J)-E(1,1,J))/DELX)
700 CONTINUE
C SOURCE FUNCTION, EVEN K REFERENCE PLANES, HY
C IF (K.NE.1A) GO TO 2
DO 20 I=2,3
TPULSE=THN-.69*(FLOAT(I)-3)/C
IF (TPULSE.LE.0.0) GO TO 20
X*EXP(-3.5*TPULSE)-EXP(-2.625E7*TPULSE)
20 CONTINUE
HYS = 7.07E-3

20 CONTINUE
DO 40 I = 25, 26
TPULSE = THN - 69 * (FLOAT(I) - 3.)/C
IF (TPULSE .LE. 0.0) GO TO 40
X = EXP(-3.5E4 * TPULSE) - EXP(-2.62E7 * TPULSE)
HYS = 7.07E-3

30 CONTINUE
DO 80 I = 25, 26
TPULSE = THN - 69 * (FLOAT(I) - 3.)/C
IF (TPULSE .LE. 0.0) GO TO 80
X = (EXP(-3.5E4 * TPULSE) - EXP(-2.62E7 * TPULSE))
HY(1, I) = X
HY(1, 10) = X
HYS = 7.07E-3

DO 80 I = 1, 141
HZ(I, 1) = HYS
HZ(I, 10) = HYS
CONTINUE
80 CONTINUE

C
C SOURCE FUNCTION, EVEN K REFERENCE PLANE, HZ
C
IF (K .NE. 14) GO TO 4
DO 60 I = 2, 3
TPULSE = THN - 69 * (FLOAT(I) - 3.)/C
IF (TPULSE .LE. 0.0) GO TO 60
X = (EXP(-3.5E4 * TPULSE) - EXP(-2.62E7 * TPULSE))
HZ = 1.447E+4
HZ(1, 1) = HZ
HZ(1, 10) = HZ
CONTINUE
60 CONTINUE

DO 80 I = 25, 26
TPULSE = THN - 69 * (FLOAT(I) - 3.)/C
IF (TPULSE .LE. 0.0) GO TO 80
X = (EXP(-3.5E4 * TPULSE) - EXP(-2.62E7 * TPULSE))
HZ = 1.447E+4
HZ(1, 1) = HZ
HZ(1, 10) = HZ
CONTINUE
80 CONTINUE

CALL WRITMS(2, HZ(1, 1), 768, K4, 1)
CALL WRITMS(2, HZ(1, 1), 768, K5, 1)
CALL WRITMS(2, HZ(1, 1), 768, K6, 1)

C
C SAMPLE POINTS OF INTEREST (EVEN K PLANES)
C
IF (K .EQ. 14) THEN
WRITE (4, 111) NH, THN, HZ(10, 4)
WRITE (4, 111) N4, THN, HZ(19, 4)
WRITE (4, 111) N6, THN, HZ(22, 18)
ELSE
END IF
C
IF (K.EQ.6) WRITE (4,111) XI, THI, H(18,10)

900 CONTINUE
RETURN
END
SUBROUTINE APERT
RETURN
END
SUBROUTINE AIRPLN

C ZEROS THE ELECTRIC FIELDS WITHIN THE F16 AIRCRAFT AND
C TANGENTIAL ELECTRIC FIELD COMPONENTS AT THE SURFACE
C
COMMON /ARRAYS/ FLD (27,27), B1 (39)
C
C EX ZER OED
C
DO 1000 K=3,25
CALL READS (2,FLD(1,1),768,K)
C WING
IF (K.NE.3) GO TO 40
FLD(10,9) = 0.0
FLD(19,9) = 0.0
40 IF (K.NE.4) GO TO 50
DO 41 I = 17, 19
41 FLD(I,9) = 0.0
50 IF (K.LT.5 OR K.GT.6) GO TO 70
DO 51 I = 16, 19
51 FLD(I,9) = 0.0
70 IF (K.LT.7 OR K.GT.8) GO TO 90
DO 71 I = 15, 19
71 FLD(I,9) = 0.0
DO 71 J = 9, 10
71 FLD(I,J) = 0.0
90 IF (K.NE.9) GO TO 100
DO 91 I = 14, 19
91 FLD(I,9) = 0.0
DO 91 J = 9, 10
91 FLD(I,J) = 0.0
100 IF (K.NE.10) GO TO 110
DO 101 I = 13, 19
101 FLD(I,9) = 0.0
DO 101 J = 9, 10
101 FLD(I,J) = 0.0
110 IF (K.NE.11) GO TO 120
DO 111 I = 11, 23
111 FLD(I,9) = 0.0
DO 111 J = 9, 10
111 FLD(I,J) = 0.0
DO 112 I = 11, 22
112
112   FLD(1,0)=0.0
C MAIN FUSELAGE
120   IF (K.LT.12,OR.K.GT.15) GO TO 126
   DO 121 I=1,19
   DO 121 J=5,12
121   FLD(1,J)=0.0
C FORAYS FUSELAGE
C
   DO 122 I=6,9
   DO 122 J=8,10
122   FLD(1,J)=0.0
   DO 123 I=6,9
   DO 123 J=11,12
123   FLD(1,J)=0.0
   FLD(7,11)=0.0
C REAR FUSELAGE
   DO 124 I=20,22
124   FLD(1,6)=0.0
   DO 125 I=20,23
   DO 125 J=7,12
125   FLD(1,J)=0.0
126   IF (K.NE.12,OR.K.NE.15) GO TO 130
   DO 127 I=19,20
   DO 127 J=3,4
127   FLD(1,J)=0.0
   FLD(20,5)=0.0
130   IF (K.LT.13,OR.K.GT.14) GO TO 135
   FLD(24,9)=0.0
   FLD(24,10)=0.0
C NOSE AND COCKPIT
   DO 131 I=4,5
   DO 131 J=8,9
131   FLD(1,J)=0.0
   FLD(5,10)=0.0
   FLD(6,11)=0.0
   FLD(7,12)=0.0
   DO 132 I=8,11
   DO 132 J=13,14
132   FLD(1,J)=0.0
   FLD(12,13)=0.0
   FLD(13,13)=0.0
   FLD(9,15)=0.0
C RIGHT VERTICAL STABILIZER
C
135   IF (K.NE.13) GO TO 140
   DO 136 I=19,23
   DO 136 J=13,14
136   FLD(1,J)=0.0
   FLD(17,13)=0.0
   FLD(18,13)=0.0
140 IF (K.NE.14) GO TO 160
   DO 141 I=18,23
   DO 141 J=13,14
141 FLD(I,J)=0.0
   FLD(17,23)=0.0
   DO 142 I=20,23
   DO 142 J=15,19
142 FLD(I,J)=0.0
   FLD(20,15)=0.0
   FLD(20,16)=0.0
   DO 143 I=22,23
   DO 143 J=20,22
143 FLD(I,J)=0.0
   FLD(24,21)=0.0
   FLD(24,22)=0.0
   DO 144 I=23,24
   DO 144 J=23,24
144 FLD(I,J)=0.0
150 IF (K.NE.16) GO TO 170
   DO 151 I=11,22
   DO 151 J=8,10
151 FLD(I,J)=0.0
   FLD(23,9)=0.0
   FLD(23,10)=0.0
170 IF (K.NE.17) GO TO 180
C   WING
   DO 171 I=13,19
   DO 171 J=9,10
171 FLD(I,J)=0.0
C   C   RIGHT HORIZONTAL STABILIZER
   C
   DO 172 I=21,24
172 FLD(I,9)=0.0
180 IF (K.NE.18) GO TO 190
   DO 181 I=14,19
   DO 181 J=9,10
181 FLD(I,J)=0.0
   DO 182 I=22,24
182 FLD(I,9)=0.0
190 IF (K.LT.19 OR K.GT.20) GO TO 210
   DO 191 I=15,19
   DO 191 J=9,10
191 FLD(I,J)=0.0
   FLD(23,9)=0.0
   FLD(24,9)=0.0
210 IF (K.LT.21 OR K.GT.22) GO TO 230
   DO 211 I=16,19
211 FLD(I,9)=0.0
230 IF (K.NE.23) GO TO 240
   DO 231 I=17,19
231 FLD(I,9)=0.0
240 IF (K.NE.24) GO TO 1000
    FLD(18,9) = 0.0
    FLD(19,9) = 0.0
  C STORE EX FIELDS BACK IN TO MASS STORAGE
1000 CALL WRITNS(2,FLD(1,1),768,K,1)
  C ZERO EV
    DO 2000 K=3,24
      L(K)=27
    CALL READNS (2,FLD(1,1),768,L)
    IF (K.LT.7.OR.K.GT.10) GO TO 1110
    DO 1081 J=14,19
1081   FLD(1,10) = 0.0
    IF (K.NE.9) GO TO 1100
    FLD(13,10) = 0.0
1100   IF (K.NE.10) GO TO 1110
    FLD(12,10) = 0.0
    FLD(13,10) = 0.0
1110   IF (K.NE.11) GO TO 1120
    DO 1111 I=10,22
    DO 1111 J=9,10
1111   FLD(1,I) = 0.0
    FLD(23,10) = 0.0
  C MAIN FUSELAGE
1120   IF (K.LT.13.OR.K.GT.15) GO TO 1130
    DO 1121 I=9,19
    DO 1121 J=6,12
1121   FLD(I,J) = 0.0
  C FWD FUSELAGE
    DO 1122 I=5,8
    DO 1122 J=9,10
1122   FLD(I,J) = 0.0
    DO 1123 I=7,8
    DO 1123 J=11,12
1123   FLD(I,J) = 0.0
    FLD(6,11) = 0.0
  C REAR FUSELAGE
    DO 1124 I=20,22
    DO 1124 J=7,12
1124   FLD(I,J) = 0.0
    DO 1125 J=8,12
1125   FLD(23,10) = 0.0
1130   IF (K.LT.13.OR.K.GT.14) GO TO 1126
    FLD(24,10) = 0.0
1126   IF (K.NE.12.OR.K.NE.15) GO TO 1131
    DO 1127 I=10,20
    DO 1127 J=4,5
1127   FLD(I,J) = 0.0
    FLD(20,6) = 0.0
1131   IF (K.LT.13.OR.K.GT.14) GO TO 1135
    FLD(3,9) = 0.0
    FLD(4,9) = 0.0
    FLD(4,10) = 0.0
FLD(5,11)=0.0
FLD(6,12)=0.0
DO 1132 I=7,13
1132  FLD(I,13)=0.0
DO 1133 I=7,11
1133  FLD(I,14)=0.0
FLD(8,15)=0.0
FLD(9,15)=0.0
IF (K.NE.13) GO TO 1140
DO 1135 I=16,23
1135  FLD(I,13)=0.0
DO 1136 I=18,23
1136  FLD(I,14)=0.0
IF (K.NE.14) GO TO 1160
DO 1140 I=22,23
DO 1141 J=13,24
1141  FLD(I,J)=0.0
DO 1142 I=21,24
1142  FLD(24,J)=0.0
DO 1143 I=20,22
1143  FLD(21,J)=0.0
DO 1144 I=17,19
1144  FLD(I,19)=0.0
DO 1145 I=19,21
1145  FLD(I,21)=0.0
IF (K.LT.17 .OR. K.GT.19) GO TO 2090
DO 1147 I=14,19
1147  FLD(I,16)=0.0
C  STORE EY BACK TO MASS STORAGE
2000  CALL WRMS(2,FLD(1,1),768,L,1)
C  EZ ZEROED
DO 3000 K=3,24
L=K+54
CALL READMS(2,FLD(1,1),768,L)
C WING TIP
   IF (K.NE.4) GO TO 2050
   DO 2041  I=17,19
2041   FLD(I,9)=0.0
C PLANE
2050   IF (K.NE.5) GO TO 2060
   DO 2051  I=16,19
2051   FLD(I,9)=0.0
C WING
2060   IF (K.LT.6.OR.K.GT.7) GO TO 2080
   DO 2061  I=15,19
2061   FLD(I,9)=0.0
C RIGHT HAND STABILIZER
C
2081   IF (K.NE.8) GO TO 2090
   DO 2081  I=14,19
2081   FLD(I,9)=0.0
C
2090   IF (K.NE.9) GO TO 2100
   DO 2091  I=13,19
2091   FLD(I,9)=0.0
C
2100   IF (K.NE.10) GO TO 2110
   DO 2101  I=12,19
2101   FLD(I,9)=0.0
C
2110   IF (K.NE.11) GO TO 2120
   DO 2111  I=11,19
2111   FLD(I,9)=0.0
C
2120   IF (K.NE.12) GO TO 2130
   DO 2121  I=10,19
2121   FLD(I,9)=0.0
C
2130   IF (K.LT.13.OR.K.GT.15) GO TO 2140
   DO 2131  I=9,19
2131   FLD(I,9)=0.0
C
2140   IF (K.NE.17) GO TO 2150
   DO 2141  I=8,19
2141   FLD(I,9)=0.0
C
2150   IF (K.NE.16) GO TO 2160
   DO 2151  I=7,19
2151   FLD(I,9)=0.0
DO 2133 J=11,12
  FLD(11,J)=0.0
  FLD(16,11)=0.0
  DO 2134 I=20,22
  DO 2134 J=6,12
  FLD(11,J)=0.0
  DO 2135 J=7,12
  FLD(23,J)=0.0
  IF (K.NE.14) GO TO 2160
  FLD(24,9)=0.0
  FLD(24,10)=0.0
  DO 2141 I=3,4
  DO 2141 J=8,9
  FLD(11,J)=0.0
  FLD(4,10)=0.0
  FLD(5,11)=0.0
  FLD(6,12)=0.0
  DO 2142 I=7,11
  DO 2142 J=13,14
  FLD(11,J)=0.0
  FLD(12,13)=0.0
  FLD(13,13)=0.0
  FLD(18,15)=0.0
  FLD(9,15)=0.0
  DO 2143 I=16,23
  FLD(11,13)=0.0
  DO 2144 J=18,23
  FLD(11,14)=0.0
2160  IF (K.NE.16) GO TO 2170
  DO 2161 I=10,22
  DO 2161 J=6,10
  FLD(11,J)=0.0
  FLD(23,9)=0.0
  FLD(23,10)=0.0
2170  IF (K.NE.17) GO TO 2180
  DO 2171 I=12,24
  FLD(11,9)=0.0
  DO 2172 J=12,19
  FLD(11,10)=0.0
2180  IF (K.NE.18) GO TO 2190
  DO 2181 I=13,19
  DO 2181 J=9,18
  FLD(11,J)=0.0
  DO 2182 I=21,24
  FLD(11,9)=0.0
2190  IF (K.LT.19 OR K.GT.20) GO TO 2210
  DO 2191 I=14,19
  DO 2191 J=9,10
  FLD(11,J)=0.0
  DO 2192 I=22,24
  FLD(11,9)=0.0
2210  IF (K.LT.21 OR K.GT.22) GO TO 2230
DO 2211 I=15,19
2211 FLD(1,9)=0.0
2230 IF (K.NE.23) GO TO 2240
    DO 2231 I=15,19
2231 FLD(1,9)=0.0
2240 IF (K.NE.24) GO TO 3000
    DO 2241 I=17,19
2241 FLD(1,9)=0.0
C
C STORE EZ BACK TO MASS STORAGE
C
3000 CALL WRITMS (2,FLD(1,1),758,L,1)
RETURN
END
Appendix B

Rymes' Code with Radiation Boundary Conditions

This appendix contains a complete listing of the modified Rymes' code with the radiation boundary conditions implemented. The compile time is 3.722 CPU seconds. The run time for the first 100 nanoseconds was 4342 CPU seconds. The F-16 is modeled in subroutine AIRPLN, as it is in the previous appendix. The major changes include expansion of the memory required and a complete rewrite of subroutine SIGSET, now renamed RADBC. The code listing is complete as implemented, and runs on a CDC Cyber 845. Improvements are included which correct minor errors in code mechanics, and also an improved source is implemented. The code compiles on the Cyber 845 using FTN5, and runs with no problem when suppressing ANSI errors. Continued runs require the changes annotated in the code in addition to saving 'TAPE1' from the previous run.
PROGRAM RBC (OUTF16,TAPE6=OUTF16,TAPE 1)

C THE MODIFIED RYNES 3-D CODE F16
C
C TO INCLUDE RADIATION BOUNDARY CONDITIONS DEVELOPED
C DEVELOPED BY MEREWETHER AND INCORPORATED INTO THIS
C CODE BY WILLFORD AND HEBERT
C
COMMON /SET1/ NMAX,EPST,S190,C,TMU,DELT
COMMON /SET2/ ICAN,ICMI,ICPI,ICAN,ICMI,ICPI,ICAN,ICMI,ICPI
COMMON /SET3/ DELX,DELY,DELZ
COMMON /TIME/ TEN,TIN
COMMON /ARRAYS/ A1(27,27),B1(39),A2(27,27),B2(39),A3(27,27),B3(39),A4(27,27),B4(39),A5(27,27),B5(39),A6(27,27),B6(39),A7(27,27),B7(39),A8(27,27),B8(39),A9(27,27),B9(39),A10(27,27),B10(39),A11(27,27),B11(39),A12(27,27),B12(39),

C
COMMON /ARRAYS/ A13(27,27),B13(39),A14(27,27),B14(39),A15(27,27),B15(39),A16(27,27),B16(39),A17(27,27),B17(39),A18(27,27),B18(39),A19(27,27),B19(39),A20(27,27),B20(39)

C
COMMON /SET/ INDEX(352)
C
C CALL OPENSS1(INDEX,352,0)
C
C
C READ INPUT DATA
C
C CALL SETUP
C
C ZERO THE FIELDS INITIALLY.
C PROGRAM USES A MASTER STORAGE FILE TO STORE FIELDS AND CONDUCTIVITY
C OF EACH LATTICE POINT. INDEX K GOES FROM 1 TO 27.
C
C EX K
C EY K+27
C EZ K+54(27*2)
C HX K+81(27*3)
C HY K+108(27*4)
C HZ K+135(27*5)
C SIG K+162(27*6)
C
C HXME K+189(27*7)
C HYME K+216(27*8)
C HZME K+243(27*9)
C CI K+270(27*10)
C C2 K+297(27*11)
C C3 K+324(27*12)
C
C DELETE FROM HERE TO STOP DELETE**********
C TO REMOVE THE ZEROING OF TAPE1
C FOR CONTINUED RUNS BEYOND THE
C INITIAL MAX STOPING TIME
C******************************************************************************
  DO 100 I=1,27
  DO 100 J=1,27
  100  A(I,J)=0.0
C******************************************************************************
  DO 200 K=1,351
C******************************************************************************
  200  CALL WRITNS(I,A(I,1),758,K,0)
C******************************************************************************
C*******STOP DELETE HERE GO TO THN= **********
C******************************************************************************
C  PRESET THE CONDUCTIVITY ARRAY
C  CALL RADBC
C  TIME STEP LOOP
C  DO 1000 N=1,NMAX
    CALL APERT
C  COMPUTE TIME CONSTANTS
C******************************************************************************
  FOR CONTINUED RUNS BEYOND TMAX************
  ADD THE STOPPING TIME FROM PREVIOUS
  RUNS TO THN AND TEN PLUS 1/2 OF DELTA T
C******************************************************************************
  EXAMPLE
  AFTER FIRST RUN
  THN=DELT*(FLOAT(N)-1.0)+1.00183E-7
  TEN=DELT*(FLOAT(N)-.5)+1.00183E-7
C******************************************************************************
  THN=DELT*(FLOAT(N)-1.0)
  TEN=DELT*(FLOAT(N)-.5)
C  ADVANCE THE H-FIELDS BY Z-PLANES
C  CALL HADV
C  ADVANCE THE E-FIELDS BY Z-PLANES
C  CALL EADV
  1000  CONTINUE
C  END OF TIME LOOP
C  CALL CLOSES(I)
  STOP
END
SUBROUTINE SETUP
C CODE USES A UNIFORM GRID.

COMMON /SET1/ IMAX, EPSI, SIG0, CTMU, DELT
COMMON /SET2/ ICAN, ICMM, ICPI, ICAN, ICM1, JCP1, KCAN, KCPI
COMMON /SET3/ DELI, DELY, DELZ

C*********
C
C
C

C INPUT DATA PROVIDED BY USER.
C
C*********
C SET SIGD=0.0 TO STOP ABSORPTION BC
DATA C, EPSI, EPSR, XMU, SIGD/3.0E8, 0.8542E-12, 1.0, 1.2566E-6, 0.0/
C*********
DATA ICPI, JCP1, KCPI/27, 27, 27/
DATA DELI, DELY, DELZ/69, 22, 45/
DATA TMAX/1.E-7/
C
ICAN=ICPI-1
ICMM=ICAN-1
JCP1=JCP1-1
JCM1=JCM1-1
KCPI=KCPI-1
KCPI1=KCPI-1

C VERIFY THAT ICPI, JCP1, KCPI ARE ODD.
C
IF (ICAN .EQ. 2*ICAN/2) .AND. JCP1 .EQ. 2*JCP1/2 .AND. +KCPI .EQ. 2*KCPI/2)
GO TO 1
PRINT *, ' THE NUMBER OF GRID POINTS MUST BE ODD ' STOP
1 CONTINUE

C DELT IS THE TIME INCREMENT (IT SATISFIES COURANT CONDITION).
C NMAX IS THE TOTAL NUMBER OF TIME INCREMENTS FIELDS WOULD BE ADVANCED.
C
DELT=MIN1(DEL1, DELY, DELZ)/(2.*C)
NMAX=TMAX/DELT
EPSI_EPSR_EPSR/DELT
TMU=DEL1/XMU
RETURN
END
SUBROUTINE RADBC
C
SIG1(I, J) AT K1=K+162 IS THE CONDUCTIVITY AT THE POINT (X(I), Y(J), Z(K)).
C
X(I)=(I-.5)*DEL1, Y(J)=(J-.5)*DELY, Z(K)=(K-.5)*DELZ
C
C**********
REAL RN(27, 27), RNH(27, 27), THETA, Rcomma
COMMON /ARRAYS/ SIG127, SIG1, B1139)
+A13(27, 27), B13(39), A14(27, 27), B14(39), C1(27, 27), B15(39),
+ C16(27, 27), B16(39), C3(27, 27), B17(39), A18(27, 27), B18(39),
+ A19(27, 27), B19(39), A20(27, 27), B20(39)
COMMON /SET3/ DELX, DELY, DELZ

C

C
****

COMMON /SET1/ MMAX, EPS1, S180, C, TMUD, DELT
COMMON /SET2/ ICAN, ICMI, ICP1, ICAN1, ICPI, ICAN1, ICP1, ICAN1, KIC1, KIC1

C

C
PRESET THE CONDUCTIVITY HERE IF NEEDED

C

C
CALCULATE GEOMETRIC CONSTANTS FOR RADOC

C

C
FINDING CENTER OF SHIFTED COORDINATE SYSTEM
IX0=ICAN/2+1
IY0=ICAN/2+1
IZ0=ICAN/2+1

C

DO 50 K=1, ICAN
   K11=K*270
   K12=K*297
   K13=K*324

C

CALL READMS (1, CI1(1, 1), 768, K11)
CALL READMS (1, CI1(1, 1), 768, K12)
CALL READMS (1, CI1(1, 1), 768, K13)
IF (K.EQ.2 .OR. K.EQ.ICAN) THEN

C

FACE
KN1=IABS(K-IZ0)
KNP=KN1+1
DO 60 I=1, ICAN
   DO 70 J=1, ICAN

C

IN=IABS(1-IX0)
JN=IABS(J-IY0)

C

RN1(I,J)=SORT ( (FLOAT(IN)*DELX)**2.0+(FLOAT(JN)*DELY)**2.0
     *(FLOAT(KN)*DELZ)**2.0)
RN2(I,J)=SORT ( (FLOAT(IN)*DELX)**2.0+(FLOAT(JN)*DELY)**2.0
     *(FLOAT(KNP)*DELZ)**2.0)
THETA=FIX((RN1(I,J)-RN1(I,J))/(DELZ)+0.5))-
   ((RN1(I,J)-RN1(I,J))/(DELZ))

C

RCOM=RN1(I,J)/(2.0*RN1(I,J))

C

C1(I,J)=RCOM*THETA*(THETA-1.0)
C2(I,J)=RCOM*2.0*(1.0-(THETA**2.0))
C3(I,J)=RCOM*THETA*(THETA+1.0)

99
ELSE
C SIDES OF K-PLANES
KN=IABS(K-120)
C
ICM2=ICAN-2
JCM2=JCAN-2
C SIDES OF I=2 & ICAN
DO 80 I=2,ICAN,ICM2
DO 90 J=1,JCAN
C
IN=IABS(1-IX0)
INP=IN+1
JN=ABS(J-IYO)
C
RN(I,J)=SORT((FLOAT(IN)*DELXI)**2.0+(FLOAT(JN)*DELY)**2.0)*
*(FLOAT(KN)*DELTZ)**2.0)
RNP(I,J)=SORT((FLOAT(INP)*DELXI)**2.0+(FLOAT(JNP)*DELY)**2.0)*
*(FLOAT(KNP)*DELTZ)**2.0)
C
THETA=IFIX((RNP(I,J)-RN(I,J))/(DELY)+0.5)
RCOND=RN(I,J)/2.0*RNP(I,J)
C
C1(I,J)=RCOND*(THETA*(THETA-1.0))
C2(I,J)=RCOND*2.0*(1.0-(THETA**2.0))
C3(I,J)=RCOND*(THETA*(THETA+1.0))
C
90 CONTINUE
80 CONTINUE
C SIDES AT J=2 & JCAN
DO 100 J=2,JCAN,JCM2
DO 110 I=1,JCAN
C
IN=IABS(1-IX0)
JN=ABS(J-IYO)
JNP=JN+1
C
RN(I,J)=SORT((FLOAT(IN)*DELXI)**2.0+(FLOAT(JN)*DELY)**2.0)*
*(FLOAT(KN)*DELTZ)**2.0)
RNP(I,J)=SORT((FLOAT(INP)*DELXI)**2.0+(FLOAT(JNP)*DELY)**2.0)*
*(FLOAT(KNP)*DELTZ)**2.0)
C
THETA=IFIX((RNP(I,J)-RN(I,J))/(DELY)+0.5)
RCOND=RN(I,J)/2.0*RNP(I,J)
C
C1(I,J)=RCOND*(THETA*(THETA-1.0))
C2(I,J)=RCOND*2.0*(1.0-(THETA**2.0))
C3(I,J)=RCOND*(THETA*(THETA+1.0))
C
100
CONTINUE
CC
END IF
CC
CALL WRiMS(I,C1(I,1),768,K11,1)
CALL WRiMS(I,C2(I,1),768,K12,1)
CALL WRiMS(I,C3(I,1),768,K13,1)
CC
CONTINUE
CC
RETURN
END
SUBROUTINE EADV
CC
THIS SUBROUTINE ADVANCES THE E-FIELDS TWO Z-PLANES AT A TIME.
CC
RECORDS OF H-FIELDS FOR TWO SUCCESSIVE Z-PLANES MUST BE KEPT IN
CC
ORDER TO TAKE DERIVATIVES IN THE Z-DIRECTION.
CC
RECORDS OF SIG FOR TWO SUCCESSIVE Z-PLANES MUST BE KEPT IN ORDER TO TAKE
CC
AVERAGES IN THE Z-DIRECTION.
CC
COMMON /SET1/MAX,EPST,SIG0,C,TWO,DELT
COMMON /SET2/JCM,JCM1,JCPI,JCM1,JCM1,JCM1,KDM1,KDM1
COMMON /SET3/DELT,DELT,DELT
COMMON /TIME/TEN,TEN
COMMON /ARRAYS/EZ(27,27),B1(39),EY(27,27),B2(39),EZ(27,27),
+3B3(39),HZA(27,27),B4(39),HIB(27,27),B5(39),HYA(27,27),B6(39),
+H1B(27,27),B7(39),HZA(27,27),B8(39),H1B(27,27),B9(39),
+SIGA(27,27),B10(39),SIGB(27,27),B11(39)
CC
ADVANCE E-FIELDS BY Z-PLANES
CC
READ IN FIRST H-PLANES
CC
CALL READS(I,HZA(I,1),768,82)
CALL READS(I,HYA(I,1),768,82)
CC
DO FOR ALL Z-PLANES
CC
MK=8
MN=9
MN=10
DO 1000 KLOOP=1,KCM,2
K=KLOOP
KPI=K+1
K2=K+2
K3=K+4
K4=K+8
K5=K+109
K6=K+135
101
K7=K+162

C READ IN SECOND H-PLANES AND OLD E-FIELDS
C
CALL READS(1,EX(1,1),768,K)
CALL READS(1,EY(1,1),768,K2)
CALL READS(1,EZ(1,1),768,K3)
CALL READS(1,HX(1,1),768,K4)
CALL READS(1,HY(1,1),768,K5)
CALL READS(1,HZ(1,1),768,K6)
CALL READS(1,SIGA(1,1),768,K7)

C DO 200 J=1,JCAN
JP1=J+1
DO 200 I=2,ICAN

C SIG(IX,Y,Z)=.5*(SIG(I-DELXIY,Z)+SIG(I,Y,Z))
C SIG(I,Y,Z)=.5*(SIG(I,Y-DELY,Z)+SIG(I,Y,Z))
C SIG(I,Y,120)=.5*(SIG(I,Y,Z-DELZ)+SIG(I,Y,Z))
C
SIG2=-.25*(SIGA(I,J)+SIGA(I-1,J))
A=EPS-SIG2
B=1./(EPS+SIG2)
EX(I,J)=(A*EX(I,J)+HZA(I,J))/DELY -
+(HYB(I,J)-HZB(I,J))/DELZ)*B

200 CONTINUE
DO 300 I=1,ICAN
JP1=J+1
DO 300 J=2,JCAN
SIG2=-.25*(SIGA(I,J-1)+SIGA(I,J))
A=EPS-SIG2
B=1./(EPS+SIG2)
EY(I,J)=(A*EY(I,J)+HXB(I,J))/DELY -
+(HZB(I,J)-HZA(I,J))/DELZ)*B

300 CONTINUE

C WRITE NEW EX,EY PLANES TO MASS STORAGE
C
CALL WRITMS(1,EX(1,1),768,K1)
CALL WRITMS(1,EY(1,1),768,K2,1)
IF (K.EQ.1660 TO 450)
DO 400 I=1,ICAN
JP1=J+1
DO 400 J=1,JCAN
JP1=J+1
SIG2=-.25*(SIGA(I,J)+SIGA(I,J))
A=EPS-SIG2
B=1./(EPS+SIG2)
EZ(I,J)=(A*EZ(I,J)+HZA(1,J))/DELY -
+(HZB(1,J)-HZB(I,J))/DELZ)*B

400 CONTINUE

C

102
WRITE NEW EZ PLANE TO MASS STORAGE

CALL WRTHS(I,EX(I,1),768,K3,1)

SAMPLE POINTS OF INTEREST (ODD K PLANES)

111  FORMAT (I4,2(I,1,IP1E11.4))
112  FORMAT (I4,2(I,1,IP1E11.4),2(/))

450  CONTINUE
   K=K1
   KPI=K+1
   K2=K2+1
   K3=K3+1
   K4=K4+1
   K5=K5+1
   K6=K6+1
   K7=K7+1

READ IN SECOND OLD E-PLANES AND NEXT H-PLANES OVER A PLANE

CALL READMS(I,EX(I,1),768,K)
CALL READMS(I,EY(I,1),768,K2)
CALL READMS(I,EZ(I,1),768,K3)
CALL READMS(I,HA(I,1),768,K4)
CALL READMS(I,HA(I,1),768,K5)
CALL READMS(I,HB(I,1),768,K6)
CALL READMS(I,SEG(I,1),768,K7)
DO 600 J=1,ICAN
   JPI=J+1
   DO 600 I=2,ICAN
   SIGB=25*(SIGB(I,J)+SIGB(I-1,J))
   A=EPS1*SIG2
   B=1./(EPS1*SIG2)
   EX(I,J)=(A*EX(I,J)+HB(1,JPI)-HB(I,J))/DELZ -
               + (HYA(I,J)+HB(I,J))/DELZ*B
600  CONTINUE
   DO 700 I=1,ICAN
      IPI=I+1
   DO 700 J=2,ICAN
      SIGB=25*(SIGB(I,J-1)+SIGB(I,J))
      A=EPS1*SIG2
      B=1./(EPS1*SIG2)
      EY(I,J)=(A*EY(I,J)+HA(I,J)-HIB(I,J))/DELZ -
               + (HB(1,I,J)+HB(I,J))/DELX*B
700  CONTINUE
   DO 800 I=1,ICAN
      IPI=I+1
   DO 800 J=1,ICAN
      JPI=J+1
      SIGB=25*(SIGB(I,J)+SIGA(I,J))
800  CONTINUE
A=EPST-SIG2
B=I.(EPSTSIG2)

\[ E_{i, j} = (A^2 E_{i, j}) + (H_Y B_{i, j}) - H_Y B_{i, j} \times (D_{i, j}) \times E_{i, j} \times D_{i, j} \times E_{i, j} \times D_{i, j} \]

800 CONTINUE
C
C WRITE NEW EX, EY, EZ PLANES TO MASS STORAGE
C
CALL WRITMS(I, EX(I, I), 768, K, 1)
CALL WRITMS(I, EY(I, I), 168, K2, 1)
CALL WRITMS(I, EZ(I, I), 768, K3, 1)
C
SAMPLE POINTS OF INTEREST (EVEN K PLANES)
C
IF (K.EQ.10) WRITE (6, 111) MN8, TEN, EY(17, 11)
IF (K.EQ.10) WRITE (6, 112) MN10, TEN, EY(17, 11)
C
IF (K.EQ.14) THEN
WRITE (6, 111) MN9, TEN, E1(15, 11)
ELSE
END IF
C
1000 CONTINUE
C
ZERO THE ELECTRIC FIELD WITHIN THE METAL COMPONENTS OF AIRCRAFT.
C
CALL AIRPLN
RETURN
END
SUBROUTINE HADV
C
THIS SUBROUTINE ADVANCES THE H-FIELDS TWO Z-PLANES AT A TIME.
C
RECORDS OF E-FIELDS FOR TWO SUCCESSIVE PLANES MUST BE KEPT IN ORDER TO TAKE
C DERIVATIVES IN THE Z-DIRECTION.
C
***
C ALSO MOST MODIFICATIONS FOR THE RADIATION
C BOUNDARY CONDITIONS WERE PLACED HERE
C
***
C
COMMON /SET1/ NAUX, EPST, SIG2, C, TMUD, DELT
COMMON /SET2/ ICAN, IDM, JCP, JCM, KCAN, KCP, KCM, KCP
COMMON /SET3/ DELT, DELZ, DELF
COMMON /TIME/ TEN, TAH
COMMON /ARRAYS/ E1A(27, 27), B1(39), E1B(27, 27), B2(39),
+E19(27, 27), B3(39), E1B(27, 27), B4(39), E1A(27, 27), B5(39),
+E1B(27, 27), B6(39), H1(27, 27), B7(39), H1(27, 27), B8(39),
+H2(27, 27), B9(39),
C
***
+A1(27, 27), B10(39), A11(27, 27), B11(39), H1M(27, 27), B12(39),
+H1M(27, 27), B13(39), H1M(27, 27), B14(39), C1(27, 27), B15(39),
+C2(27, 27), B16(39), C3(27, 27), B17(39), H1M(27, 27), B18(39),
+H1M(27, 27), B19(39), H1M(27, 27), B20(39)
REAL HITM1(27,27), HYTM1(27,27), HZTM1(27,27)

C*********
C
C DO FOR ALL Z-PLANES
MN1=1
MN2=2
MN3=3
MN4=4
MN5=5
MN6=6
MN7=7
C
DO 900 KLOOP=1,KCAN,2
K=KLOOP
KMI=K-1
K2=K+27
K3=K+54
K4=K+81
K5=K+108
K6=K+135
C*********
K10=K+243
K11=K+270
K12=K+297
K13=K+324
CALL READS (1,HITM2(1,1),768,K10)
CALL READS (1,HITM2(1,1),768,K11)
CALL READS (1,HITM2(1,1),768,K12)
CALL READS (1,HITM2(1,1),768,K13)
C*********
CALL READS(1,EYB(1,1),768,K)
CALL READS(1,EYB(1,1),768,K2)
CALL READS(1,EZB(1,1),768,K6)
IF (K.EQ.1) GO TO 350
C*********
K8=K+109
K9=K+216
CALL READS(1,HITM2(1,1),768,K8)
CALL READS(1,HITM2(1,1),768,K9)
C*********
CALL READS(1,EYB(1,1),768,K3)
CALL READS(1,HX(1,1),768,K4)
CALL READS(1,HY(1,1),768,K5)
DO 200 J=2,ICAN
DO 200 I=1,ICAN
C*********
HITM(1,1)=HX(1,1)
C*********
HX(1,1)=HX(1,1)+TMUD*([EYB(1,1)-EYB(1,1)]/DELZ -
+([EZB(1,1)-EZB(1,1)]/DELY)
APPROXIMATES H FIELDS IN PLANE PERPENDICULAR TO WIRE

SOURCE FUNCTION, ODD K REFERENCE PLANES, H

CURRENT PULSE MOVING IN X DIRECTION ATTACHES TO THE NOSE OF THE AIRPLANE

AT (X,Y,Z)=(3,9,15), EXITS BY THE TAIL AT (X,Y,Z)=(25,10,15).

AT TIME T=0.0 PULSE IS LOCATED AT X(3).

IF(K.NE.15)60 TO 1
DO 10 I=2,3,1
T_PULSE=13N.-69*(FLOAT(I)-3.1)/C
IF(T_PULSE.LE.0.0)60 TO 10

I=EXP(-3.5E4*T_PULSE)-EXP(-2.625E7*T_PULSE)

HYS=7.07E3*X
HY(I,9)=-HYS

10 CONTINUE
DO 30 I=25,26,1
T_PULSE=13N.-69*(FLOAT(I)-3.0)/C
IF(T_PULSE.LE.0.0)60 TO 30

I=EXP(-3.5E4*T_PULSE)-EXP(-2.625E7*T_PULSE)

HYS=7.07E3*X
HY(I,10)=-HYS

30 CONTINUE

SIDE WHEN I=1 IF CPI
ICM2=ICM-2
DO 310 I=2,ICM,ICM2
DO 320 J=1,ICAN
IF (I.EQ.2) THEN
10=1
ELSE
10=ICAN+1
END IF
HYMLE(I,J)=HYMLE(I,J)
IF (J.GE.2) THEN
HITMLE(I,J)=HITMLE(I,J)
ELSE

106
END IF

320 CONTINUE

310 CONTINUE
C SIDES WHEN J=1 & JC1
JC2=JC2+1
DO 330 J=2,JC2,JC1
DO 340 I=1,ICAN
    IF (J.EQ.2) THEN
      JB=1
    ELSE
      JB=JC1+1
    END IF

340 CONTINUE

330 CONTINUE
HZ(1,1)=HZ(2,1)
HZ(1,27)=HZ(2,27)
HZ(27,1)=HZ(27,2)
HZ(27,7)=HZ(27,26)

C****

C WRITE NEW HX, HY PLANES TO MASS STORAGE

C
CALL WRITMS(I,HX(I,1),768,K4,1)
CALL WRITMS(I,HX(I,1),768,K5,1)

C****
CALL WRITMS(I,HY(I,1),768,K6,1)
CALL WRITMS(I,HX2(I,1),768,K8,1)
CALL WRITMS(I,HY2(I,1),768,K9,1)
CALL WRITMS(I,HX2(I,1),768,K10,1)

C****

350 CONTINUE
DO 400 I=1,ICM1
  IPI=I+1
DO 400 J=1,JC2
  JPI=J+1

C****
HZT(IPI,JPI)=HZ(IPI,JPI)

C****
HZ(IPI,JPI)=HZ(IPI,JPI)+TMX*((EXB(IPI,JPI)-EXB(IPI,JPI))/DELY -
  +(EYB(IPI,JPI)-EYB(IPI,JPI))/DELX)

400 CONTINUE
C SIDES I=1 & JC1
ICM2=ICAN-2
DO 610 I=2,ICAN,ICM2
DO 629 J-2, JCAN
   IF (J.EQ.2) THEN
     IB=1
   ELSE
     IB=ICAN+1
   END IF

C

   HZ(1B, J)=C1(I, J)+HZM2(I, J)+C2(I, J)+HZM1(I, J)+C3(I, J)+HZ(1, J)
   HZM2(I, J)=HZM1(I, J)

620 CONTINUE

C

610 CONTINUE

   JCAN=JCAN-2

C SIDES J=1 & JCP1
 DO 630 J=2, JCAN, JCM2
 DO 640 I=2, ICAN
   IF (J.EQ.2) THEN
     JB=1
   ELSE
     JB=ICAN+1
   END IF

C

640 CONTINUE

630 CONTINUE

   HZ(1, J)=HZ(2, 1)
   HZ(1, 27)=HZ(2, 27)
   HZ(27, 1)=HZ(27, 2)
   HZ(27, 27)=HZ(27, 26)

C WRITE NEW HZ PLANES TO MASS STORAGE

C

CALL WRITMS(1, HZ(1, 1), 768, K6, 1)
CALL WRITMS(1, HZM2(1, 1), 768, K10, 1)

C

C SAMPLE POINTS OF INTEREST (ODD K PLANES)

C

IF (K.EQ.13) WRITE(6, 111) M12, TAN, HZ(4, 10)
IF (K.EQ.13) WRITE(6, 111) M3, TAN, HZ(24, 11)
IF (K.EQ.23) WRITE(6, 111) M7, TAN, HZ(18, 10)

C

111 FORMAT (14, 2,1X, 1PE11.4))

C

K=K+1
K1=K-1
K2=K2+1
K3=K3+1
K4=K4+1
K5=K5+1
K6=K6+1

C

K8=K8+1
K9=K9+1

100
K10=K+243
K11=K+270
K12=K+297
K13=K+324
CALL READMS1(I,HXMI1(I,1),768,K1)
CALL READMS1(I,HYMI1(I,1),768,K2)
CALL READMS1(I,HZMI1(I,1),768,K3)
CALL READMS1(I,HX(I,1),768,K4)
CALL READMS1(I,HY(I,1),768,K5)
CALL READMS1(I,HZ(I,1),768,K6)
DO 680 J=1,ICAN
   JPI=J+1
   DO 680 I=1,ICAN
      HXTM1(I,JPI)=HX(I,JPI)
      HX(I,JPI)=HX(I,JPI)+TM1*(EYA(I,JPI)-EYB(I,JPI))/DELZ -
      *(EZA(I,JPI)-EZA(I,JPI)/DELY)
   CONTINUE
      DO 700 J=1,ICAN
         [JPI]=J+1
         DO 700 I=1,ICAN
            HXTM1(I,JPI)=HX(I,JPI)
            HX(I,JPI)=HX(I,JPI)+TM1*(EYA(I,JPI)-EYB(I,JPI))/DELZ -
            *(EZA(I,JPI)-EZA(I,JPI)/DELY)
            CONTINUE
   CONTINUE
      DO 700 J=1,ICAN
         [JPI]=J+1
         DO 700 I=1,ICAN
            HXTM1(I,JPI)=HX(I,JPI)
            HX(I,JPI)=HX(I,JPI)+TM1*(EYA(I,JPI)-EYB(I,JPI))/DELZ -
            *(EZA(I,JPI)-EZA(I,JPI)/DELY)
            CONTINUE
   CONTINUE
HYTM(IPI,JPI)=HY(IPI,JPI)

C********
HY(IPI,JPI)=HY(IPI,JPI)+TMUDD*(EXA(IPI,JPI)-EXA(IPI,JPI))/DELX -
+ (EXA(IPI,JPI)-EXB(IPI,JPI))/DELY)

700 CONTINUE
C
C SOURCE FUNCTION, EVEN K REFERENCE PLANES, HY
C
IF (K.NE.14) GO TO 2
DO 20 I=2,2,1
TPULSE=TPULSE-.69*FLOAT(I)-3.)/C
IF (TPULSE.LE.0.0) GO TO 20
X=EXP(-3.5E+TPULSE)*EXP(-2.6E7*TPULSE)
HY(IPI,JPI)=7.0763/X
HY(IPI,JPI)=HY(IPI,JPI)
20 CONTINUE
DO 40 I=25,25,1
TPULSE=TPULSE-.69*FLOAT(I)-3.)/C
IF (TPULSE.LE.0.0) GO TO 40
X=EXP(-3.5E+TPULSE)*EXP(-2.6E7*TPULSE)
HY(IPI,JPI)=7.0763/X
HY(IPI,JPI)=HY(IPI,JPI)
40 CONTINUE
2 CONTINUE
DO 80 I=1,1,1
IPI=1+1
DO 80 J=1,1,1
JPI=J+1
C******
HZTM(IPI,JPI)=HZ(IPI,JPI)
C******
HZ(IPI,JPI)=HZ(IPI,JPI)+TMUDD*(EXA(IPI,JPI)-EXA(IPI,JPI))/DELX -
+ (EXA(IPI,JPI)-EYA(IPI,JPI))/DELX)
800 CONTINUE
C
C SOURCE FUNCTION, EVEN K REFERENCE PLANES, HZ
C
IF (K.NE.14) GO TO 4
DO 60 I=2,2,1
TPULSE=TPULSE-.69*FLOAT(I)-3.)/C
IF (TPULSE.LE.0.0) GO TO 60
X=EXP(-3.5E+TPULSE)*EXP(-2.6E7*TPULSE)
HIS=1.447E4/X
HZ(IPI,JPI)=HIS
HZ(IPI,JPI)=HZ(IPI,JPI)
60 CONTINUE
DO 80 I=25,25,1
TPULSE=TPULSE-.69*FLOAT(I)-3.)/C
IF (TPULSE.LE.0.0) GO TO 80
X=EXP(-3.5E+TPULSE)*EXP(-2.6E7*TPULSE)
HIS=1.447E4/X
HZ(IPI,JPI)=HIS
HZ(IPI,JPI)=HZ(IPI,JPI)
800 CONTINUE

110
HZ(1,1) = HZS
80 CONTINUE
4 CONTINUE
C IF (K.EQ.2 OR K.EQ.KCAN) THEN
   DO 210 I = 2, ICAN
     DO 220 J = 2, JCAN
       RXTM2(I,J) = RXTM1(I,J)
       RXHIM2(I,J) = RXHMI(I,J)
     END IF
     RXKMI(I,B,J) = RXKMI(I,1)
220 CONTINUE
210 CONTINUE
C EDGES OF FACES
   ICM2 = ICAN - 2
   DO 410 I = 2, ICAN, ICM2
     DO 420 J = 2, JCAN
       IF (I.EQ.2) THEN
         IB = 1
       ELSE
         IB = ICAN + 1
       END IF
       RXKM(IB,J) = RXKM(I,J)
420 CONTINUE
410 CONTINUE
   JCM2 = JCAN - 2
   DO 430 J = 2, JCAN, JCM2
     DO 440 I = 2, ICAN
       IF (J.EQ.2) THEN
         JB = 1
       ELSE
         JB = JCAN + 1
       END IF
       RXHMI(I,J,B) = RXHMI(I,J)
440 CONTINUE
430 CONTINUE
   CALL WRTMS(1, RXKM(I,1), 768, KMIX, 1)
   CALL WRTMS(1, RXHMI(I,1), 768, KMIX, 1)
   CALL WRTMS(1, RXKMI(I,1), 768, KMIX, 1)
   ELSE
     END IF
     ICM2 = ICAN - 2
C SIDED I = 1 & ICPI
   DO 230 I = 2, ICAN, ICM2
     DO 240 J = 1, JCAN
       IF (I.EQ.2) THEN
         IB = 1
       ELSE
         IB = ICAN + 1
111
END IF
HYTM2(I,J)=HYTM1(I,J)
IF (I .GE. 2) THEN
HZTM2(I,J)=HZTM1(I,J)
ELSE
END IF
240 CONTINUE
230 CONTINUE
JCM2=JCA1-2
DO 256 J=2,JCM,JCM2
DO 260 I=1,ICAN
IF (J.EQ.2) THEN
JB=1
ELSE
JB=JCA1+1
END IF
HZTM2(I,J)=HZTM1(I,J)
IF (1 .GE. 2) THEN
HZTM2(I,J)=HZTM1(I,J)
ELSE
END IF
260 CONTINUE
250 CONTINUE
HZ(I,1)=HZ(2,1)
HZ(I,27)=HZ(27,2)
HZ(27,1)=HZ(27,2)
HZ(27,27)=HZ(27,26)
CALL WRITMS (I,HXTM2(I,1),768,K8,1)
CALL WRITMS (I,HXTM2(I,1),768,K9,1)
CALL WRITMS (I,HXTM2(I,1),768,K16,1)
CALL WRITMS (I,HXTM2(I,1),768,K18,1)
C********
CALL WRITMS (I,HX(I,1),768,K4,1)
CALL WRITMS (I,HX(I,1),768,K5,1)
CALL WRITMS (I,HX(I,1),768,K6,1)
C
C SAMPLE POINTS OF INTEREST (EVEN K PLANES)
C
IF (K.EQ.14) THEN
WRITE (6,111) MMN,TMN,HXI,4)
WRITE (6,111) MMN5,TMN,HXI,4)
WRITE (6,111) MMN6,TMN,HXI,4)
ELSE
END IF
C
IF (K.EQ.6) WRITE (6,111) MMN, TMN, HX(18,10)
C
900 CONTINUE
RETURN
END
SUBROUTINE AIRPT
RETURN
END
SUBROUTINE AIRPLN

C ZEROES THE ELECTRIC FIELDS WITHIN THE F16 AIRCRAFT AND
C TANGENTIAL ELECTRIC FIELD COMPONENTS AT THE SURFACE
C
COMMON /ARRAYS/ FLD (27,27),B1 (39)
C
C EX ZERED
C
DO 1000 K=3,25
CALL READS (1,FLD(1,1),768,K)
C WING
   IF (K.NE.3) GO TO 40
   FLD(10,9)=0.0
   FLD(19,9)=0.0
40 IF (K.NE.4) GO TO 50
   DO 41 I=17,19
   FLD(1,9)=0.0
41 IF (K.LT.5.OR.K.GT.6) GO TO 70
   DO 51 I=16,19
   FLD(1,9)=0.0
51 IF (K.LT.7.OR.K.GT.8) GO TO 90
   DO 71 I=15,19
   DO 71 J=9,10
   FLD(I,J)=0.0
    FLD(23,9)=0.0
    FLD(24,9)=0.0
90 IF (K.NE.9) 60 TO 100
   DO 91 I=14,19
   DO 91 J=9,10
91 FLD(I,J)=0.0
   DO 92 I=22,24
   FLD(1,9)=0.0
92 IF (K.NE.10) 60 TO 110
   DO 101 I=11,19
   DO 101 J=9,10
101 FLD(I,J)=0.0
   DO 102 I=21,24
   FLD(I,9)=0.0
102 IF (K.NE.11) 60 TO 120
   DO 111 I=11,23
   DO 111 J=9,10
111 FLD(I,J)=0.0
   DO 112 I=11,22
   FLD(I,8)=0.0
112 IF (K.LT.12.OR.K.GT.15) GO TO 126
   DO 121 I=10,19
   FLD(I,8)=0.0
C MAIN FUSELAGE
120 IF (K.LT.12.OR.K.GT.15) GO TO 126
   DO 121 I=10,19
   FLD(I,8)=0.0
121 FLD(I,8)=0.0
122 FLD(I,8)=0.0
C
DO 121 J=5,12
  121 FLD(1,J)=0.0
C
C FORWARD FUSELAGE
C
DO 122 I=6,9
  122 FLD(I,J)=0.0
DO 123 I=6,9
  123 FLD(J,11)=0.0
C REAR FUSELAGE
DO 124 I=20,22
  124 FLD(I,6)=0.0
DO 125 I=20,23
  125 J=7,12
C
DO 126 IF (K.NE.12 .OR. K.NE.15) GO TO 130
  126 DO 127 I=19,20
  127 J=3,4
  127 FLD(1,J)=0.0
  127 FLD(20,5)=0.0
C
DO 130 IF (K.LT.13 .OR. K.GT.14) GO TO 135
  130 FLD(4,9)=0.0
  130 FLD(24,10)=0.0
C
NOSE AND COCKPIT
DO 131 I=4,5
  131 J=8,9
  131 FLD(1,J)=0.0
  131 FLD(15,10)=0.0
  131 FLD(6,11)=0.0
  131 FLD(7,12)=0.0
  131 DO 132 I=8,11
  132 J=13,14
  132 FLD(11,J)=0.0
  132 FLD(12,13)=0.0
  132 FLD(13,13)=0.0
  132 FLD(19,15)=0.0
C
C RIGHT VERTICAL STABILIZER
C
DO 135 IF (K.NE.13) GO TO 140
  135 DO 136 I=19,23
  136 J=13,14
  136 FLD(I,J)=0.0
  136 FLD(17,13)=0.0
  136 FLD(18,13)=0.0
  136 DO 140 I=10,23
  140 J=13,14
  140 FLD(I,J)=0.0
FLD(17,23)=0.0
DO 142 I=20,23
DO 142 J=15,19
142  FLD(1,J)=0.0
FLD(20,15)=0.0
FLD(20,16)=0.0
DO 143 I=22,23
DO 143 J=20,22
143  FLD(1,J)=0.0
FLD(24,21)=0.0
FLD(24,22)=0.0
DO 144 I=23,24
DO 144 J=23,24
144  FLD(1,J)=0.0
160  IF (K.NE.16) GO TO 170
DO 161 I=11,22
DO 161 J=6,18
161  FLD(1,J)=0.0
FLD(23,9)=0.0
FLD(23,10)=0.0
170  IF (K.NE.17) GO TO 180
C WING
DO 171 I=13,19
DO 171 J=9,19
171  FLD(1,J)=0.0
C C RIGHT HORIZONTAL STABALIZER
C
DO 172 I=21,24
172  FLD(1,9)=0.0
180  IF (K.NE.18) GO TO 190
DO 181 I=14,19
DO 181 J=9,18
181  FLD(1,J)=0.0
DO 182 I=22,24
182  FLD(1,9)=0.0
190  IF (K.LT.19.OR.K.GT.20) GO TO 210
DO 191 I=15,19
DO 191 J=9,19
191  FLD(1,J)=0.0
FLD(23,9)=0.0
FLD(24,9)=0.0
210  IF (K.LT.21.OR.K.GT.22) GO TO 230
DO 211 I=16,19
211  FLD(1,9)=0.0
230  IF (K.NE.23) GO TO 240
DO 231 I=17,19
231  FLD(1,9)=0.0
240  IF (K.NE.24) GO TO 1000
FLD(18,9)=0.0
FLD(19,9)=0.0
C STORE EX FIELDS BACK IN TO MASS STORAGE
1000 CALL WRITMS(1,FLD(1,1),',68,5,1)
C ZERO EY
DO 2000 K=3,24
  L=K+27
  CALL READMS (1,FLD(1,1),768,L)
  IF (K.LT.7.OR.K.GT.110) GO TO 1110
  DO 1081 I=14,19
1081 FLD(I,10)=0.0
  IF (K.NE.9) GO TO 1100
  FLD(13,10)=0.0
1100 IF (K.NE.10) GO TO 1110
  FLD(12,10)=0.0
  FLD(13,10)=0.0
1110 IF (K.NE.11) GO TO 1120
  DO 1111 I=10,22
  DO 1111 J=9,10
1111 FLD(I,J)=0.0
  FLD(23,10)=0.0
C MAIN FUSELAGE
1120 IF (K.LT.12.OR.K.GT.15) GO TO 1130
  DO 1121 I=9,19
  DO 1121 J=6,12
1121 FLD(I,J)=0.0
C FWD FUSELAGE
  DO 1122 I=5,8
  DO 1122 J=9,10
1122 FLD(I,J)=0.0
  DO 1123 I=7,8
  DO 1123 J=11,12
1123 FLD(I,J)=0.0
  FLD(6,11)=0.0
C REAR FUSELAGE
  DO 1124 I=20,22
  DO 1124 J=7,12
1124 FLD(I,J)=0.0
  DO 1125 I=8,12
1125 FLD(23,J)=0.0
1130 IF (K.LT.13.OR.K.GT.14) GO TO 1126
1126 FLD(24,10)=0.0
1126 IF (K.NE.12.OR.K.NE.15) GO TO 1131
  DO 1127 I=10,20
  DO 1127 J=4,5
1127 FLD(I,J)=0.0
1127 FLD(20,6)=0.0
1130 IF (K.LT.13.OR.K.GT.14) GO TO 1135
1131 FLD(3,9)=0.0
  FLD(4,9)=0.0
  FLD(14,10)=0.0
  FLD(15,11)=0.0
  FLD(16,12)=0.0
  DO 1132 I=7,13
1132 FLD(I,13)=0.0
1133  FLD(1,14)=0.0
        FLD(8,15)=0.0
        FLD(19,15)=0.0
1135  IF (K.NE.13) GO TO 1140
        DO 1136 I=16,23
1136  FLD(1,13)=0.0
        DO 1137 I=18,23
1137  FLD(1,14)=0.0
1140  IF (K.NE.14) GO TO 1160
        DO 1141 I=22,23
        DO 1141 J=13,24
        FLD(I,J)=0.0
        DO 1142 J=20,22
1142  FL D(24,J)=0.0
        DO 1143 J=29,22
        FLD(21,J)=0.0
1143  FLD(21,J)=0.0
        DO 1144 J=17,19
        DO 1144 I=20,21
        FLD(I,J)=0.0
        DO 1145 J=19,21
        DO 1145 J=15,16
1145  FLD(I,J)=0.0
        DO 1146 J=17,21
1146  FLD(I,J)=0.0
        DO 1147 I=16,21
        FLD(I,13)=0.0
1160  IF (K.NE.16) GO TO 1170
        DO 1161 I=18,22
        DO 1161 J=9,10
1161  FLD(I,J)=0.0
        FLD(23,10)=0.0
C WING
117  IF (K.NE.17) GO TO 1180
        DO 1171 I=12,19
1171  FLD(I,10)=0.0
1180  IF (K.NE.18) GO TO 1190
        DO 1181 I=13,19
1181  FLD(I,10)=0.0
119  IF (K.LT.19. OR. K.GT.29) GO TO 2000
        DO 1191 I=14,19
1191  FLD(I,10)=0.0
C STORE EY BACK TO MASS STORAGE
2000  CALL WRITMS11,FLD(1,11),768,L,11
C EZ ZERGED
        DO 3000 K=3,24
        L=K+54
        CALL READMS11,FLD(1,11),768,L
C WING TIP
        IF (K.NE.4) GO TO 2050
        DO 2041 I=17,19
1141  FLD(I,9)=0.0
2041  IF (K.NE.5) GO TO 2050
2050  IF (K.NE.5) GO TO 2050

117
DO 2051 I=16,19
FLD(I,9)=.0
C PLANE
2060 IF (K.LT.6.OR.K.GT.7) GO TO 2080
DO 2061 I=15,19
FLD(I,9)=.0
C WING
2080 IF (K.NE.8) GO TO 2090
DO 2081 J=9,10
DO 2081 I=14,19
FLD(I,J)=.0
C RIGHT HAND STABILIZER
C DO 2082 I=22,24
FLD(I,9)=.0
2090 IF (K.NE.9) GO TO 2100
DO 2091 I=14,19
DO 2091 J=9,10
FLD(I,J)=.0
DO 2092 I=22,24
FLD(I,9)=.0
2100 IF (K.NE.10) GO TO 2110
DO 2101 I=13,19
DO 2101 J=9,10
FLD(I,J)=.0
2102 I=21,24
FLD(I,9)=.0
2110 IF (K.NE.11) GO TO 2120
DO 2111 I=12,24
FLD(I,9)=.0
DO 2112 I=12,19
FLD(I,10)=.0
2120 IF (K.NE.12) GO TO 2130
DO 2121 I=10,22
DO 2121 J=8,10
FLD(I,J)=.0
FLD(23,9)=.0
FLD(23,10)=.0
2130 IF (K.LT.13.OR.K.GT.15) GO TO 2140
DO 2131 I=9,19
DO 2131 J=5,12
FLD(I,J)=.0
DO 2132 I=5,8
DO 2132 J=8,10
FLD(I,J)=.0
DO 2133 I=7,8
DO 2133 J=11,12
FLD(I,J)=.0
DO 2134 I=28,22
DO 2134 J=6,12
2134   FLD(I,J)=0.0
      DO 2135 J=7,12
2135   FLD(23,J)=0.0
      IF (K.NE.14) GO TO 2160
      FLD(24,9)=0.0
      FLD(24,10)=0.0
      DO 2141 I=3,4
      DO 2141 J=9,9
2141   FLD(I,J)=0.0
      FLD(14,10)=0.0
      FLD(15,11)=0.0
      FLD(16,12)=0.0
      DO 2142 I=7,11
      DO 2142 J=13,14
2142   FLD(I,J)=0.0
      FLD(12,13)=0.0
      FLD(13,13)=0.0
      FLD(18,15)=0.0
      FLD(19,15)=0.0
      DO 2143 I=16,23
2143   FLD(I,13)=0.0
      DO 2144 I=18,23
2144   FLD(I,14)=0.0
      IF (K.NE.16) GO TO 2170
      DO 2161 I=10,22
      DO 2161 J=8,10
2161   FLD(I,J)=0.0
      FLD(23,9)=0.0
      FLD(23,10)=0.0
2170   IF (K.NE.17) GO TO 2180
      DO 2171 I=12,24
2171   FLD(I,9)=0.0
      DO 2172 I=12,19
2172   FLD(I,10)=0.0
      IF (K.NE.18) GO TO 2190
      DO 2181 I=13,19
      DO 2181 J=9,10
2181   FLD(I,J)=0.0
      DO 2182 I=21,24
2182   FLD(I,9)=0.0
2190   IF (K.LT.19.OR.K.GT.20) GO TO 2210
      DO 2191 I=14,19
      DO 2191 J=9,10
2191   FLD(I,J)=0.0
      DO 2192 I=22,24
2192   FLD(I,9)=0.0
2210   IF (K.LT.21.OR.K.GT.22) GO TO 2230
      DO 2211 I=15,19
2211   FLD(I,9)=0.0
2230   IF (K.NE.23) GO TO 2240
      DO 2231 I=16,19
2231   FLD(I,9)=0.0

119
2240 IF (K.NE.24) GO TO 3000
   DO 2241 I=17,19
2241   FLD(I,9)=0.0
C
C STORE E2 BACK TO MASS STORAGE
C
3000   CALL WRITMS (I,FLD(I,1),768,L,1)
      RETURN
   END
END OF FILE
Appendix C

Sensor Plots

This appendix contains graphs of the ten sensors for both the radiation and absorption boundary conditions. Each page contains one graph for the absorption boundary conditions and one graph for the radiation boundary condition. Each graph is made up of 1000 data points calculated from the corresponding programs in Appendix A or B. The points were connected with a natural cubic spline. The only differences in the two codes are the boundary conditions as annotated in the program. Both plots on the same page are for identical time periods so that side-by-side comparisons could be made. The programs were run for results out to 2.5 microseconds. The graphs in this appendix only reflect the first 1.0 microseconds. The graphs beyond 1.0 microseconds were only continuations of the same trends as previous graphs, so they were deleted for brevity. Also, Absorption and Radiation Boundary Conditions are abbreviated ABC and RBC respectively on the plots.
Figure 27. Sensor One, H-field, Right Wing, $H_x(18.10.16)$
Figure 28. Sensor One, H-field, Right Wing, $H_x(18,10,6)$
Figure 29. Sensor One, H-field, Right Wing, $H_x(18,10,6)$
Figure 30. Sensor One, H-field, Right Wing, $H_x(18,10,6)$
Figure 31. Sensor Two, B-field, Nose, H (4,10,13)
Figure 32. Sensor Two, H-field, Nose, Hx (4,10,13)
Figure 33. Sensor Two, H-field, Nose, $H_z(4,10,13)$
Figure 34. Sensor Two, H-field, Nose, $H_z(4,10,13)$
Figure 35. Sensor Three, H-field, Engine Burner Can, H_{2}(24/11/13)
Figure 36. Sensor Three, H-field, Engine Burner Can, $H_z(24,11,13)$
Figure 39. Sensor Four, H-field, Forward Fuselage Bottom, Hz (10^4, 4, 14)
Figure 40. Sensor Four, H-field, Forward Fuselage Bottom, H<sub>z</sub>(10,4,14)
Figure 41. Sensor Four, H-field, Forward Fuselage Bottom, $H_z(10,4,14)$
Figure 42. Sensor Four, H-field, Forward Fuselage Bottom, $H_z(10,4,14)$
Figure 45. Sensor Five, H-field, Rear Fuselage Bottom, H_{Z}(19,4,14)
Figure 46. Sensor Five, H-field, Rear Fuselage Bottom, $H_z(19,4,14)$
Figure 47. Sensor Six, H-field, Vertical Stabilizer, $H_x(22,18,14)$
Figure 48. Sensor Six, H-field, Vertical Stabilizer, $H_x(22,18,14)$
Figure 49. Sensor Six, H-field, Vertical Stabilizer, $H_x(22,18,14)$
Figure 50. Sensor Six, H-field, Vertical Stabilizer, $B_x (22, 18, 14)$

(a) ABC

(b) RBC
Figure 51. Sensor Seven, H-field, Left Wing, $H_x(18,10,23)$
Figure 52. Sensor Seven, H-field, Left Wing, $H_x(18,10,23)$
Figure 53. Sensor Seven, H-field, Left Wing, $H_x(18,10,23)$
Figure 54. Sensor Seven, H-field, Left Wing, $H_x(18,10,23)$
Figure 57. Sensor Eight, E-field, Right Wing, $E_y(17, 11, 10)$
Figure 58. Sensor Eight, E-field, Right Wing, $E_y(17,11,10)$
Figure 59. Sensor Nine, E-field, Middle Fuselage Top, $E_y(15,13,14)$
Figure 60. Sensor Nine, E-field, Middle Fuselage Top, $E_y(15,13,14)$
Figure 61. Sensor Nine, E-field, Middle Fuselage Top, $E_y(15,13,14)$
Figure 62. Sensor Nine, E-field, Middle Fuselage Top, E_y (15, 13, 14)
Figure 63. Sensor Ten, E-field, Left Wing, $E_y(17,11,18)$
Figure 64. Sensor Ten, E-field, Left Wing, $E_y(17,11,18)$
Figure 65. Sensor Ten, E-field, Left Wing, E_y (17/11/18)
Figure 66. Sensor Ten, E-field, Left Wing, $E_y(17,11,18)$
Bibliography


162


Captain Clifford F. Williford was born on 12 July 1950 in Norfolk, Virginia. He graduated from high school (Frederick Military Academy, Portsmouth, Virginia) in 1968. He attended Virginia Polytechnic Institute for one year, then transferred to Old Dominion University. On 13 February 1970, Capt Williford enlisted in the Air Force. In December of 1973, he was selected for the Airman Education and Commissioning Program (AECP). He attended Virginia Polytechnic Institute and State University from January 1974 till March 1976, where he received his Bachelor of Science in Electrical Engineering. Upon graduation, he attended the Flight Screening Program in Hondo, Texas, and then attended Officer Training School, Medina Annex, Lackland AFB, Texas. After commissioning, he attended Undergraduate Pilot Training at Williams AFB, and earned his Pilots' wings in July 1977. His first assignment after UPT was RAF Lakenheath, where Capt. Williford flew the F-111F. He was a Pilot Weapons Systems Officer in the 492nd TFS, and later an Aircraft Commander in the 493rd TFS. Following his return to the States, Capt. Williford served as an Aircraft Commander, Flight Instructor and Flight Examiner in the 522 TFS until entering the School of Engineering, Air Force Institute of Technology, in June 1984.

Permanent address: 1926 E. Ocean View Ave.
Norfolk, Virginia 23503
Title: COMPARISON OF ABSORPTION AND RADIATION BOUNDARY CONDITIONS USING A TIME DOMAIN THREE-DIMENSIONAL FINITE-DIFFERENCE ELECTROMAGNETIC COMPUTER CODE

Thesis Chairman: Randy J. Jost, Lt, USAF
Instructor of Electrical Engineering
Abstract

The Three-Dimensional Finite Difference (3DFD) computer code is compared using Absorption Boundary Conditions (ABS) versus Radiation Boundary Conditions (RBC). This comparison is made when the 3DFD code is used to study the interaction of lightning with an aircraft. The 3DFD computer code is a modified version of Rymes' 3DFD. The aircraft modeled for the paper is an F-16 'Fighting Falcon'. The ABC used simulates an infinite free-space by setting the conductivity of the boundary space to that of distilled water, to "absorb" the outgoing electromagnetic waves. The RBC simulates free-space by assigning the boundary fields to a previously calculated value. The value is calculated with a parabolic interpolation of three previous field values, which are offset in space. Therefore, the calculated value is also extrapolated to account for the time delay and position change. The results of incorporating RBC were dramatic. The ten locations sampled for the test showed marked improvement in the waveforms when using RBC's. Depending on the purpose of the analysis, this improved waveform output may be overshadowed by the 25% increase in CPU time that is needed for the more sophisticated RBC.
END

FILMED

3-86

DTIC