HYPERSONIC INFLIGHT TRAJECTORY SCATTER (HITS) CN CODE USER'S MANUAL

ROBERT G. BELLAIRE THEODORE O. GUSTAFSON

APRIL 1976

FINAL REPORT

PREPARED BY

AVCO CORPORATION
201 LOWELL STREET
WILMINGTON, MA 01887

CONTRACT NO.
DAAA09-74-C-2026-P00004

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

PREPARED FOR

RESEARCH DIRECTORATE
GENERAL THOMAS J. RODMAN LABORATORY
ROCK ISLAND ARSENAL
ROCK ISLAND, ILLINOIS 61201
The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.
This report is a User's Manual for the HITS computer code. The HITS code statistically evaluates projectile dispersion, which is the dispersion associated with the in-flight behavior of a gun launched projectile. Projectile dispersion is a fundamental limit on overall weapon system effectiveness. The code is an automated computer-based analytic procedure for computing crossrange and downrange dispersion errors and budgets and sensitivity.
20. Abstract (Concl'd)

HITS evaluates projectile dispersion by either analytic or Monte Carlo methods employing trajectory equations. The trajectory equations are closed form approximations to the six degrees of freedom equations of motion. The closed form equations presume the projectile does not experience transonic flow conditions during any phase of flight. This limits the present HITS code to this type of projectile (referred to here as "hypervelocity", with no other connotation intended). The trajectory equations have been subjected to extensive testing. In all cases they were accurate to within engineering tolerances. These equations have potential application to operational fire control computers, the computation of firing tables, and any other situation requiring rapid, accurate, low-cost trajectory determination.

Since this report is a User's Manual, the emphasis is placed on providing the information necessary to operate the code and interpret the results. Detailed input/output information is summarized in tables featuring step-by-step cook-book instructions. Example problems are discussed. Appendices present the theoretical and programming fine points; as well as the complete program listing.
FOREWORD

This User's Manual was prepared by Avco Systems Division, 201 Lowell Street, Wilmington, Massachusetts 01887, for the U. S. Army Armament Command, Gen. T. J. Rodman Laboratory, Rock Island Arsenal, Rock Island Illinois 61201, with Mr. William P. Wohlford of the Research Directorate as contract monitor.

The principal authors appreciate the contributions and suggestions offered by their colleagues: P. Crimi, R. Flaherty, P. Fote, R. Kingsley, W. Reinecke, and R. Torrani.

The authors wish to thank the many Rock Island personnel, and particularly Mr. William P. Wohlford and Mr. Charles S. Hicks, who provided Avco with much timely and relevant data and many sound suggestions.

The authors also wish to thank Mrs. L. Kiley, Mrs. J. Di Pietro, Mrs. P. Ford and Miss M. Smith for their careful and conscientious typing, Mrs. H. Cassell for the many equations, Miss A. Gaudeat and Mr. R. Gaumond for their excellent art work, and Mr. E. Schurmann and Mr. L. Sacks for their editorial assistance.
ABSTRACT

This report is a User's Manual for the HITS computer code. The HITS code statistically evaluates "projectile dispersion," which is the dispersion associated with the inflight behavior of a gun launched projectile. Projectile dispersion is a fundamental limit on overall weapon system effectiveness. The code is an automated computer-based analytic procedure for computing crossrange and downrange dispersion error budgets and sensitivity coefficients to the sources of projectile dispersion. The error budgets statistically define projectile dispersion at a level conducive to interpretation and comprehensive understanding.

HITS evaluates projectile dispersion by either analytic or Monte Carlo methods employing trajectory equations. The trajectory equations are closed form approximations to the six degrees of freedom equations of motion. The closed form equations presume the projectile does not experience transonic flow conditions during any phase of flight. This limits the present HITS code to this type of projectile (referred to here as "hypervelocity", with no other connotation intended). The trajectory equations have been subjected to extensive testing. In all cases they were accurate to within engineering tolerances. These equations have potential application to operational fire control computers, the computation of firing tables, and any other situation requiring rapid, accurate, low-cost trajectory determination.

Since this report is a User's Manual, the emphasis is placed on providing the information necessary to operate the code and interpret the results. Detailed input/output information is summarized in tables featuring step-by-step cook-book instructions. Example problems are discussed. Appendices present the theoretical and programming fine points, as well as the complete program listing.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>ENGINEERING PERSPECTIVE</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Projectile Dispersion Sources</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Weapon System Effectiveness Implications</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Dispersion Analysis Methodology</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Code Limitations</td>
<td>8</td>
</tr>
<tr>
<td>1.5</td>
<td>Report Overview</td>
<td>11</td>
</tr>
<tr>
<td>2.0</td>
<td>OVERVIEW</td>
<td>13</td>
</tr>
<tr>
<td>2.1</td>
<td>Code Structure</td>
<td>13</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Input Processor</td>
<td>15</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Statistical Processor</td>
<td>15</td>
</tr>
<tr>
<td>2.1.3</td>
<td>Projectile Trajectory Module</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Analysis Options</td>
<td>16</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Single Trajectory</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Range Check</td>
<td>16</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Analytical Statistical</td>
<td>17</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Monte Carlo</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Trajectory Equations</td>
<td>17</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Closed Form Rationale</td>
<td>17</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Summary Description</td>
<td>18</td>
</tr>
<tr>
<td>2.4</td>
<td>Related Topics</td>
<td>21</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Crossrange Dispersion Indices</td>
<td>21</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Correlation Analysis - An Example</td>
<td>22</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Target-Fixed Crossrange Dispersion</td>
<td>25</td>
</tr>
<tr>
<td>3.0</td>
<td>INPUT DESCRIPTION</td>
<td>29</td>
</tr>
<tr>
<td>3.1</td>
<td>Statistical Processor Control Card (Card Group 1)</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>Input Processor Control Cards (Card Group 2)</td>
<td>34</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Cont'd)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1 Card Group 2 Assigned Qualities</td>
<td>34</td>
</tr>
<tr>
<td>3.2.2 Trajectory Variable Code Numbers</td>
<td>39</td>
</tr>
<tr>
<td>3.3 Optional Input Processor Control Cards (Card Group 3)</td>
<td>39</td>
</tr>
<tr>
<td>3.4 Error Source Model Encoding</td>
<td>54</td>
</tr>
<tr>
<td>4.0 OUTPUT INTERPRETATION</td>
<td>61</td>
</tr>
<tr>
<td>4.1 Single Trajectory Mode</td>
<td>61</td>
</tr>
<tr>
<td>4.2 Range Check Mode</td>
<td>71</td>
</tr>
<tr>
<td>4.3 Analytical Statistical Mode</td>
<td>84</td>
</tr>
<tr>
<td>4.4 Monte Carlo Mode</td>
<td>106</td>
</tr>
<tr>
<td>5.0 SUMMARY</td>
<td>139</td>
</tr>
<tr>
<td>APPENDIX A - ELEMENTS OF PROBABILITY THEORY</td>
<td>141</td>
</tr>
<tr>
<td>A.1 Random Phenomenon and Probabilities</td>
<td>141</td>
</tr>
<tr>
<td>A.1.1 Fundamental Concepts</td>
<td>141</td>
</tr>
<tr>
<td>A.1.2 Rules of Combination</td>
<td>142</td>
</tr>
<tr>
<td>A.1.3 Conditional Probabilities</td>
<td>144</td>
</tr>
<tr>
<td>A.2 Random Variables</td>
<td>145</td>
</tr>
<tr>
<td>A.2.1 Distribution Functions</td>
<td>145</td>
</tr>
<tr>
<td>A.2.2 Joint Distribution Functions</td>
<td>147</td>
</tr>
<tr>
<td>A.2.3 Statistical Independence</td>
<td>148</td>
</tr>
<tr>
<td>A.3 Averages</td>
<td>148</td>
</tr>
<tr>
<td>A.3.1 Expectation</td>
<td>149</td>
</tr>
<tr>
<td>A.3.2 Moments</td>
<td>150</td>
</tr>
<tr>
<td>A.3.3 Correlation</td>
<td>151</td>
</tr>
<tr>
<td>A.3.4 Confidence Limits</td>
<td>153</td>
</tr>
<tr>
<td>A.4 Typical Distributions</td>
<td>154</td>
</tr>
<tr>
<td>A.4.1 Uniformly Distributed Random Variable (TYPE = 3)</td>
<td>154</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS (Cont'd)

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.4.2</td>
<td>Gaussian Distributed Random Variable (TYPE = 2)</td>
<td>156</td>
</tr>
<tr>
<td>A.4.3</td>
<td>Jointly Gaussian Random Variables</td>
<td>159</td>
</tr>
<tr>
<td>A.4.4</td>
<td>Rayleigh Distributed Random Variable</td>
<td>165</td>
</tr>
<tr>
<td>A.5</td>
<td>Random Variable Generation</td>
<td>166</td>
</tr>
<tr>
<td>A.5.1</td>
<td>Uniformly Distributed Variables (TYPE = 3)</td>
<td>166</td>
</tr>
<tr>
<td>A.5.2</td>
<td>Gaussian Distributed Variables (TYPE = 2)</td>
<td>167</td>
</tr>
<tr>
<td>A.5.3</td>
<td>Arbitrarily Distributed Variable (TYPE = 4)</td>
<td>168</td>
</tr>
<tr>
<td>A.6</td>
<td>Monte Carlo Inaccuracies</td>
<td>170</td>
</tr>
<tr>
<td>A.6.1</td>
<td>Grouping Errors</td>
<td>170</td>
</tr>
<tr>
<td>A.6.2</td>
<td>Sampling Errors</td>
<td>173</td>
</tr>
<tr>
<td>B</td>
<td>APPENDIX B - HITS COMPUTER CODE DESCRIPTION</td>
<td>183</td>
</tr>
<tr>
<td>B.1</td>
<td>Statistical Processor Computational Procedures</td>
<td>183</td>
</tr>
<tr>
<td>B.1.1</td>
<td>Analytical Statistical Calculations</td>
<td>183</td>
</tr>
<tr>
<td>B.1.2</td>
<td>Monte Carlo Calculations</td>
<td>190</td>
</tr>
<tr>
<td>B.2</td>
<td>Code Architecture</td>
<td>194</td>
</tr>
<tr>
<td>B.2.1</td>
<td>Data Storage</td>
<td>194</td>
</tr>
<tr>
<td>B.2.2</td>
<td>Input Processor Functions</td>
<td>195</td>
</tr>
<tr>
<td>B.2.3</td>
<td>Statistical Processor Functions</td>
<td>198</td>
</tr>
<tr>
<td>B.2.4</td>
<td>HITS Subroutines</td>
<td>203</td>
</tr>
<tr>
<td>B.2.5</td>
<td>Histogram Computer Plots</td>
<td>203</td>
</tr>
<tr>
<td>C</td>
<td>APPENDIX C - CLOSED FORM TRAJECTORY EQUATIONS</td>
<td>213</td>
</tr>
<tr>
<td>C.1</td>
<td>Introduction</td>
<td>213</td>
</tr>
<tr>
<td>C.2</td>
<td>Particle Trajectory</td>
<td>214</td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS (Cont'd)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.2.1 Analytical Development</td>
<td>214</td>
</tr>
<tr>
<td>C.2.2 Verification</td>
<td>224</td>
</tr>
<tr>
<td>C.3 Crossrange Perturbations</td>
<td>224</td>
</tr>
<tr>
<td>C.3.1 Analytical Development</td>
<td>224</td>
</tr>
<tr>
<td>C.3.2 Verification</td>
<td>238</td>
</tr>
<tr>
<td>C.4 Downrange Perturbations</td>
<td>238</td>
</tr>
<tr>
<td>C.4.1 Analytical Development</td>
<td>238</td>
</tr>
<tr>
<td>C.4.2 Verification</td>
<td>246</td>
</tr>
</tbody>
</table>

### APPENDIX D - HITS PROGRAM LISTING

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.1 MAIN Program</td>
<td>250</td>
</tr>
<tr>
<td>D.2 Subroutine ARTLU</td>
<td>254</td>
</tr>
<tr>
<td>D.3 Function A2</td>
<td>255</td>
</tr>
<tr>
<td>D.4 Function A2F</td>
<td>256</td>
</tr>
<tr>
<td>D.5 Block Data</td>
<td>257</td>
</tr>
<tr>
<td>D.6 Subroutine CHKCA</td>
<td>258</td>
</tr>
<tr>
<td>D.7 Subroutine CHKIN</td>
<td>259</td>
</tr>
<tr>
<td>D.8 Subroutine CNVRT</td>
<td>261</td>
</tr>
<tr>
<td>D.9 Subroutine CROSS</td>
<td>262</td>
</tr>
<tr>
<td>D.10 Subroutine CROSSF</td>
<td>265</td>
</tr>
<tr>
<td>D.11 Subroutine CVFET</td>
<td>268</td>
</tr>
<tr>
<td>D.12 Subroutine DOABC</td>
<td>269</td>
</tr>
<tr>
<td>D.13 Subroutine D0234</td>
<td>271</td>
</tr>
<tr>
<td>D.14 Subroutine EXTRA</td>
<td>282</td>
</tr>
<tr>
<td>D.15 Subroutine FC2987</td>
<td>283</td>
</tr>
<tr>
<td>D.16 Function FEXP</td>
<td>285</td>
</tr>
<tr>
<td>D.17 Subroutine FILLA</td>
<td>286</td>
</tr>
<tr>
<td>D.18 Subroutine FILLIN</td>
<td>287</td>
</tr>
<tr>
<td>D.19 Subroutine GQCOR</td>
<td>288</td>
</tr>
<tr>
<td>D.20 Subroutine GQUC</td>
<td>289</td>
</tr>
<tr>
<td>D.21 Subroutine G1795</td>
<td>290</td>
</tr>
<tr>
<td>D.22 Subroutine G2440</td>
<td>291</td>
</tr>
<tr>
<td>D.23 Subroutine G2987</td>
<td>292</td>
</tr>
<tr>
<td>D.24 Subroutine HSTG1</td>
<td>295</td>
</tr>
<tr>
<td>D.25 Subroutine HSTG2</td>
<td>296</td>
</tr>
<tr>
<td>D.26 Subroutine HYSPLIT</td>
<td>297</td>
</tr>
<tr>
<td>Subroutine/Function</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------</td>
<td>------</td>
</tr>
<tr>
<td>INCARD</td>
<td>298</td>
</tr>
<tr>
<td>INCON</td>
<td>301</td>
</tr>
<tr>
<td>INCONF</td>
<td>305</td>
</tr>
<tr>
<td>INITIL</td>
<td>309</td>
</tr>
<tr>
<td>INLHST</td>
<td>311</td>
</tr>
<tr>
<td>LOADER</td>
<td>312</td>
</tr>
<tr>
<td>MCRL</td>
<td>315</td>
</tr>
<tr>
<td>MOVEUP</td>
<td>317</td>
</tr>
<tr>
<td>NCOV</td>
<td>318</td>
</tr>
<tr>
<td>NMEAN</td>
<td>319</td>
</tr>
<tr>
<td>NVAR</td>
<td>320</td>
</tr>
<tr>
<td>N2DARX</td>
<td>321</td>
</tr>
<tr>
<td>NPO</td>
<td>322</td>
</tr>
<tr>
<td>N2D</td>
<td>323</td>
</tr>
<tr>
<td>PICK1</td>
<td>324</td>
</tr>
<tr>
<td>PRNTMC</td>
<td>325</td>
</tr>
<tr>
<td>QCOR</td>
<td>327</td>
</tr>
<tr>
<td>RANDOM</td>
<td>328</td>
</tr>
<tr>
<td>RANDU</td>
<td>329</td>
</tr>
<tr>
<td>RW2987</td>
<td>330</td>
</tr>
<tr>
<td>SAMPLE</td>
<td>332</td>
</tr>
<tr>
<td>SPRNTT</td>
<td>334</td>
</tr>
<tr>
<td>STDDEV</td>
<td>336</td>
</tr>
<tr>
<td>STOREC</td>
<td>337</td>
</tr>
<tr>
<td>TG1795</td>
<td>338</td>
</tr>
<tr>
<td>TG2440</td>
<td>339</td>
</tr>
<tr>
<td>TG2987</td>
<td>340</td>
</tr>
<tr>
<td>TRAJT</td>
<td>343</td>
</tr>
<tr>
<td>TRAJX</td>
<td>346</td>
</tr>
<tr>
<td>TRAJX</td>
<td>348</td>
</tr>
<tr>
<td>TVX</td>
<td>350</td>
</tr>
<tr>
<td>T4NTV</td>
<td>351</td>
</tr>
<tr>
<td>VXT</td>
<td>352</td>
</tr>
<tr>
<td>WIND</td>
<td>353</td>
</tr>
<tr>
<td>CO(C</td>
<td>354</td>
</tr>
<tr>
<td>ZATAN2</td>
<td>355</td>
</tr>
<tr>
<td>ZZ</td>
<td>356</td>
</tr>
<tr>
<td>DISTRIBUTION</td>
<td>357</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Weapon System Effectiveness</td>
</tr>
<tr>
<td>1-2</td>
<td>HITS Overview</td>
</tr>
<tr>
<td>2-1</td>
<td>HITS Flow Diagram</td>
</tr>
<tr>
<td>2-2</td>
<td>Trajectory Equations Computational Procedure</td>
</tr>
<tr>
<td>2-3</td>
<td>Gaussian Scale Factors</td>
</tr>
<tr>
<td>2-4</td>
<td>Intercept Geometry</td>
</tr>
<tr>
<td>2-5</td>
<td>Crossrange Sensitivity to Downrange Dispersion</td>
</tr>
<tr>
<td>3-1</td>
<td>IBM-029 Keypunch Control Card for Card Group 1</td>
</tr>
<tr>
<td>3-2</td>
<td>IBM-029 Keypunch Control Cards for Card Group 2</td>
</tr>
<tr>
<td>3-3</td>
<td>IBM-029 Keypunch Control Cards for Card Group 3</td>
</tr>
<tr>
<td>3-4</td>
<td>Encoded Error Source Model</td>
</tr>
<tr>
<td>4-1</td>
<td>Single Trajectory Mode Check Problem Input</td>
</tr>
<tr>
<td>4-2</td>
<td>Single Trajectory Mode Check Problem Output</td>
</tr>
<tr>
<td>4-3</td>
<td>Range Check Mode Check Problem Input</td>
</tr>
<tr>
<td>4-4</td>
<td>Range Check Mode Check Problem Output</td>
</tr>
<tr>
<td>4-5</td>
<td>Analytical Statistical Mode Check Problem Input</td>
</tr>
<tr>
<td>4-6</td>
<td>Analytical Statistical Mode Check Problem Output</td>
</tr>
<tr>
<td>4-7</td>
<td>Monte Carlo Mode Check Problem Input</td>
</tr>
<tr>
<td>4-8</td>
<td>Monte Carlo Mode Check Problem Output</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES (CONT'D)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>The Venn Diagram</td>
<td>143</td>
</tr>
<tr>
<td>A-2</td>
<td>Chebyshev's Lower Bound</td>
<td>155</td>
</tr>
<tr>
<td>A-3</td>
<td>Gaussian Density Function</td>
<td>157</td>
</tr>
<tr>
<td>A-4</td>
<td>A Comparison of Two Gaussian Density Functions With Different Standard Deviations</td>
<td>158</td>
</tr>
<tr>
<td>A-5</td>
<td>Two Probability Density Functions With Identical Means and Variances</td>
<td>160</td>
</tr>
<tr>
<td>A-6</td>
<td>Gaussian Scale Factors</td>
<td>163</td>
</tr>
<tr>
<td>A-7</td>
<td>Von Neumann Rejection Sampling Illustration</td>
<td>169</td>
</tr>
<tr>
<td>A-8</td>
<td>Expected Error in Sample Mean in Percentage of Standard Deviation</td>
<td>180</td>
</tr>
<tr>
<td>A-9</td>
<td>Expected Error in Sample Variance Value</td>
<td>181</td>
</tr>
<tr>
<td>B-1</td>
<td>HITS Flow Diagram</td>
<td>184</td>
</tr>
<tr>
<td>B-2</td>
<td>Coordinates for Derivative Evaluations</td>
<td>191</td>
</tr>
<tr>
<td>B-3</td>
<td>Internal Data Transfers</td>
<td>196</td>
</tr>
<tr>
<td>B-4</td>
<td>HITS Subroutine Linking Diagram</td>
<td>208</td>
</tr>
<tr>
<td>B-5</td>
<td>Projectile Trajectory Module Subroutine Linking Diagram</td>
<td>209</td>
</tr>
<tr>
<td>B-6</td>
<td>Histogram Title Array</td>
<td>211</td>
</tr>
<tr>
<td>B-7</td>
<td>Histogram Storage Array ZDATAD (I,J,*)</td>
<td>212</td>
</tr>
<tr>
<td>C-1</td>
<td>Trajectory Coordinate System</td>
<td>215</td>
</tr>
<tr>
<td>C-2</td>
<td>Typical Ballistic Coefficient Variation With Mach Number</td>
<td>223</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES (CONCL'D)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-3</td>
<td>Trajectory Comparison</td>
<td>225</td>
</tr>
<tr>
<td>C-4</td>
<td>Projectile Body Fixed Coordinate System</td>
<td>228</td>
</tr>
<tr>
<td>C-5</td>
<td>Lift Force Components in the Trajectory Frame</td>
<td>234</td>
</tr>
<tr>
<td>C-6</td>
<td>Lift Induced Dispersion as a Function of Roll Rate</td>
<td>240</td>
</tr>
<tr>
<td>C-7</td>
<td>Downrange Forces Due to Angle of Attack</td>
<td>242</td>
</tr>
</tbody>
</table>
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Sources of Projectile Dispersion</td>
<td>2</td>
</tr>
<tr>
<td>1-2</td>
<td>Error Source Model</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Crossrange Dispersion at Nominal Time (Nominal Range = 10 Kft)</td>
<td>9</td>
</tr>
<tr>
<td>2-1</td>
<td>Probabilities of Occurrence</td>
<td>23</td>
</tr>
<tr>
<td>3-1</td>
<td>Statistical Processor Control Card (Card Group 1)</td>
<td>30</td>
</tr>
<tr>
<td>3-2</td>
<td>Input Processor Control Cards (Card Group 2)</td>
<td>35</td>
</tr>
<tr>
<td>3-3</td>
<td>HITS Trajectory Parameter List</td>
<td>41</td>
</tr>
<tr>
<td>3-4</td>
<td>Optional Input Processor Control Cards (Card Group 3)</td>
<td>55</td>
</tr>
<tr>
<td>3-5</td>
<td>Error Source Model Code Numbers</td>
<td>57</td>
</tr>
<tr>
<td>4-1</td>
<td>Monte Carlo Mode Output Interpretation</td>
<td>134</td>
</tr>
<tr>
<td>A-1</td>
<td>Probabilities of Occurrence</td>
<td>164</td>
</tr>
<tr>
<td>B-1</td>
<td>Time and Storage Requirements</td>
<td>199</td>
</tr>
<tr>
<td>B-2</td>
<td>HITS Subroutine List</td>
<td>204</td>
</tr>
<tr>
<td>C-1</td>
<td>List of Symbols</td>
<td>216</td>
</tr>
<tr>
<td>C-2</td>
<td>Test Case Projectile Properties</td>
<td>226</td>
</tr>
<tr>
<td>C-3</td>
<td>Trajectory Model Verification</td>
<td>239</td>
</tr>
</tbody>
</table>
1.0 ENGINEERING PERSPECTIVE

The Hypervelocity Inflight Trajectory Scatter (HITS) computer code is an automated procedure for evaluating "projectile dispersion", which is the dispersion associated with the in-flight behavior of a gun launched projectile. This report documents the code in the format of a User's Manual. Primary emphasis is placed on the information necessary to operate the code and interpret the results. Theoretical aspects and computational details are treated in the appendices.

This chapter addresses the following. Section 1.1 identifies the sources of projectile dispersion and, thereby, more clearly defines the term. Section 1.2 discusses the relationship of projectile dispersion to overall weapon system effectiveness. The dispersion analysis methodology suggested by the code is illustrated by example in Section 1.3. Section 1.4 summarizes the limitations of the code. An overview of the report is presented in Section 1.5. The overall objective of this chapter is to enable the engineer to quickly determine the utility of HITS with respect to his particular problem.

1.1 Projectile Dispersion Sources

The HITS computer code quantifies projectile dispersion. Projectile dispersion is defined here to be the dispersion associated with in-flight behavior which cannot be compensated by fire control. (HITS does not compute the dispersion attributable to fire control. However, it does include in the projectile dispersion calculation the effects of fire control corrections for in-flight phenomena.) Thus, the sources of projectile dispersion enumerated in this section are those factors which influence in-flight behavior and cause the fire control predicted trajectory to differ from the true flight path.

Table 1-1 lists the sources of projectile dispersion. The left hand column contains the physical mechanisms through which the contributing factors of the right hand column act. There are four general categories: (1) initial conditions at the muzzle, (2) projectile mass properties, (3) projectile aerodynamic characteristics, and (4) atmospheric effects. The sources of Table 1-1 cause dispersion only insofar as the fire control computer cannot anticipate their effects and apply corrective action. For instance, winds are a source of projectile dispersion if the weapon system does not measure them. If it

1 "Hypervelocity" is used throughout this report to indicate the limitation of the present HITS code to projectiles whose velocity does not become transonic at any point along the trajectory. No other connotation is intended.
Table 1-1 Sources of Projectile Dispersion

<table>
<thead>
<tr>
<th>ERROR SOURCE</th>
<th>TYPICAL CONTRIBUTING FACTORS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Conditions:</strong></td>
<td></td>
</tr>
<tr>
<td>Velocity Vector</td>
<td>Sabot Separation</td>
</tr>
<tr>
<td>Angle of Attack</td>
<td>In-Barrel Dynamics</td>
</tr>
<tr>
<td>Angular Rates</td>
<td>Blast Effects</td>
</tr>
<tr>
<td></td>
<td>Barrel Vibrations</td>
</tr>
<tr>
<td></td>
<td>Tipoff Rates</td>
</tr>
<tr>
<td></td>
<td>Gun Slew Rates</td>
</tr>
<tr>
<td><strong>Inertial Characteristics:</strong></td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>Manufacturing Tolerances</td>
</tr>
<tr>
<td>Moments of Inertia</td>
<td></td>
</tr>
<tr>
<td>Physical Dimensions</td>
<td></td>
</tr>
<tr>
<td>C. G. Asymmetries</td>
<td></td>
</tr>
<tr>
<td><strong>Aerodynamic Characteristics:</strong></td>
<td>Manufacturing Tolerances</td>
</tr>
<tr>
<td>Static Coefficients</td>
<td></td>
</tr>
<tr>
<td>Dynamic Coefficients</td>
<td>Ablations Effect on the</td>
</tr>
<tr>
<td>Spin Rate</td>
<td>Ballistic Coefficient</td>
</tr>
<tr>
<td>Trip Angle of Attack</td>
<td>Coefficient Measurement</td>
</tr>
<tr>
<td>Static Margin</td>
<td>Errors</td>
</tr>
<tr>
<td><strong>Atmospheric Effects:</strong></td>
<td></td>
</tr>
<tr>
<td>Winds</td>
<td>Temporal and Spatial</td>
</tr>
<tr>
<td>Density</td>
<td>Atmospheric Variations</td>
</tr>
</tbody>
</table>


does, only the wind measurement error causes projectile dispersion. Although not mentioned in Table 1-1, simplifying assumptions incorporated in the fire control trajectory to minimize computational requirements are also considered sources of projectile dispersion. The HITS code accounts for each of these factors.

1.2 Weapon System Effectiveness Implications

This section discusses the role projectile dispersion plays in determining weapon system effectiveness with implications to projectile design procedures. The discussion opens with a brief overview of the factors which affect weapon system effectiveness.

Figure 1-1 is a block diagram of a modern gun weapon system. It indicates the various factors that influence weapon system effectiveness, as defined by the probability of kill. Referring to Figure 1-1, the engagement scenario shown on the left hand side establishes the geometry of the encounter. Target motion dynamics limit target motion to physically possible rates and accelerations. The acquisition sensor continually measures the target coordinates and passes the information to the fire control computer. Fire control's basic jobs are to solve the intercept geometry, align the gun, and signal the gunner at the appropriate time. Its calculations take into account not only the target coordinates but also information concerning the inflight behavior of the projectile, winds, atmospheric density, gun orientation, and various system models describing target dynamic constraints, gun mount rate and orientation limits, etc. Fire control issues commands to the gun mount servo system to point the gun. The servo holds the gun on the target while awaiting the command to fire. After the round has been fired, the projectile trajectory relative to the true target position determines the point of closest approach, that is the miss distance. If a hit is scored, the location of the hit and the projectile terminal ballistics determine the probability of kill.

Figure 1-1 Weapon System Effectiveness
Projectile dispersion affects overall weapon system effectiveness. The accuracy with which the fire control computer can predict the true flight path of the projectile poses a fundamental limitation on weapon system effectiveness. Even if the exact position of the target relative to the gun were known and the target could be tracked with infinite precision, the inability of fire control to predict the flight path exactly would cause the projectile to miss the target. Thus, projectile dispersion is a fundamental limit on weapon system effectiveness because it defines a level of accuracy which cannot be improved upon by refinements of other elements of the weapon system. The HITS code quantitatively defines this limit.

HITS may also be used as a projectile design tool. Since all projectiles exhibit dispersion to a greater or lesser extent and this can limit overall weapon system effectiveness, it would be prudent to include the evaluation of projectile dispersion in all projectile design studies. The HITS code is an analytic tool for performing this evaluation inexpensively. Furthermore, HITS can compute a single trajectory under a given set of conditions or under a cyclic permutation of parameters. The calculation is quick, accurate, and inexpensive. Thus, HITS brings to design activities, additional capabilities that go beyond the assessment of projectile dispersion.

Although projectile dispersion poses a fundamental limit on weapon system effectiveness, it is not always clear what impact projectile dispersion will have on overall weapon system effectiveness. Returning to Figure 1-1, it is clear that overall weapon system effectiveness ultimately depends on how well the gun system operates as a whole. There is a synergistic effect which makes the system better than the sum of its parts, since one subsystem can be designed to compensate for the errors of another. For instance, fire control could be designed to monitor and correct for gun pointing errors, as indicated by the feedback loop shown in Figure 1-1. Thus, fire control can point the gun better than an analysis of the gun-mount servo system would indicate was possible. For this reason, it is potentially misleading to forecast overall weapon system effectiveness from an analysis of any single subsystem (e.g., a projectile dispersion analysis). Thus, judgment must be used in interpreting projectile dispersion assessments produced by the HITS code in terms of overall weapon system effectiveness.
1.3 Dispersion Analysis Methodology

The HITS computer code suggests a methodology for determining the susceptibility of projectile designs to projectile dispersion. This section demonstrates the two step method with an illustrative example. The following paragraphs illustrate the information required, the role of the HITS code, and the format of the results.

Table 1-1 lists the sources of projectile dispersion. By analyzing the contributing factors, statistics may be assigned to each source which quantify the uncertainty as illustrated in Table 1-2: the error source model. Displayed in the left hand column of Table 1-2 are the sources of projectile dispersion. The third column contains the nominal values for the projectile properties. The second column defines the distribution of variations about the nominal. The uncertainties are given in the fourth column: the standard deviations or one-sigma \(1-\sigma\) values.\(^1\) The standard deviations quantify the level of uncertainty in the information supplied to fire control. The error source model may be even more detailed. The basic error source model of Table 1-2 could be expanded to include correlations between the elements of the error source model, such as projectile weight and muzzle velocity. Whatever the level of detail, construction of the error source model concludes the first step in evaluating projectile dispersion via the HITS code.

The second and last step is to use the HITS code to evaluate the projectile dispersion corresponding to the error source model as suggested by the schematic block diagram of Figure 1-2. The code reads the error source model as data and proceeds to quantify projectile dispersion in the form of an error budget. The code contains two sets of closed form trajectory equations: one represents the true trajectory and the other represents fire control. A Statistical Processor manipulates the trajectory equations in either an analytical or Monte Carlo fashion to determine the dispersion statistics. The results of the calculations are most conveniently summarized in the format of a projectile dispersion error budget as indicated in Figure 1-2. Any simplifying assumptions incorporated in an actual fire control trajectory are input to the code and taken into account.

The HITS code can construct error budgets in both cross-range and downrange directions as well as evaluate three dimensional dispersion indices such as the Spherical Error Probable (SEP). Error budgets may be constructed at either nominal time or nominal range. In this respect HITS is very flexible and can compute most every statistic customarily used to describe dispersion.

\(^1\) Rayleigh/Uniform uncertainties are specified by the mean magnitude given in the third column.
Table 1-2 Error Source Model

<table>
<thead>
<tr>
<th>ERROR SOURCE</th>
<th>STATISTICAL DISTRIBUTION</th>
<th>MEAN VALUE</th>
<th>STANDARD DEVIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>• INITIAL CONDITIONS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• VELOCITY VECTOR</td>
<td>Gaussian</td>
<td>11,000 ft/sec</td>
<td>1/3%</td>
</tr>
<tr>
<td>- MAGNITUDE</td>
<td>Gaussian</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ORIENTATION</td>
<td>Rayleigh/Uniform*</td>
<td>0.0005 deg</td>
<td></td>
</tr>
<tr>
<td>• INERTIAL ORIENTATION</td>
<td>Rayleigh/Uniform*</td>
<td>0.1 deg</td>
<td></td>
</tr>
<tr>
<td>- ATTITUDE RATE</td>
<td>Rayleigh/Uniform*</td>
<td>55 rad/sec</td>
<td></td>
</tr>
<tr>
<td>• PHYSICAL CHARACTERISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• WEIGHT</td>
<td>Gaussian</td>
<td>0.11 lbs</td>
<td>1.0%</td>
</tr>
<tr>
<td>• MOMENTS OF INERTIA</td>
<td>Gaussian</td>
<td>1.234 x 10^-6 slug-ft^2</td>
<td>1 2/3%</td>
</tr>
<tr>
<td>- AXIAL</td>
<td>Gaussian</td>
<td>2.110 x 10^-5 slug-ft^2</td>
<td>1 2/3%</td>
</tr>
<tr>
<td>- PITCH</td>
<td>Gaussian</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PHYSICAL DIMENSIONS</td>
<td>Uniform</td>
<td>3.068 x 10^-3 ft^2</td>
<td>2/3%</td>
</tr>
<tr>
<td>- REFERENCE AREA</td>
<td>Uniform</td>
<td>0.31 ft</td>
<td>1/3%</td>
</tr>
<tr>
<td>- LENGTH</td>
<td>Uniform</td>
<td>0.0625 ft</td>
<td>1/3%</td>
</tr>
<tr>
<td>- BASE DIAMETER</td>
<td>Uniform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• AERODYNAMIC CHARACTERISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• STATIC COEFFICIENTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• DRAG VARIATION</td>
<td>Gaussian</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- VELOCITY</td>
<td>Gaussian</td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>- ANGLE OF ATTACK</td>
<td>Gaussian</td>
<td>1.0 1/deg^2</td>
<td>1.0%</td>
</tr>
<tr>
<td>- NORMAL FORCE</td>
<td>Gaussian</td>
<td>1.9767 1/deg</td>
<td>2.0%</td>
</tr>
<tr>
<td>• DYNAMIC COEFFICIENTS</td>
<td>Gaussian</td>
<td>-7.5 (none)</td>
<td>20.0%</td>
</tr>
<tr>
<td>- PITCH DAMPING</td>
<td>Gaussian</td>
<td>0.0 (none)</td>
<td>0.0</td>
</tr>
<tr>
<td>- MAGNUS MOMENT</td>
<td>Gaussian</td>
<td>400 rad/sec</td>
<td>1.0%</td>
</tr>
<tr>
<td>• SPIN RATE</td>
<td>Gaussian</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• TRIM ANGLE OF ATTACK</td>
<td>Rayleigh/Uniform*</td>
<td>0.1 deg</td>
<td></td>
</tr>
<tr>
<td>• STATIC MARGIN</td>
<td>Gaussian</td>
<td>6.2% length</td>
<td>1.0% of length</td>
</tr>
<tr>
<td>• ATMOSPHERIC EFFECT:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• CONSTANT WINDS</td>
<td>Rayleigh/Uniform*</td>
<td>11.0 ft/sec</td>
<td></td>
</tr>
<tr>
<td>• DENSITY VARIATIONS</td>
<td>Gaussian</td>
<td>2.378 x 10^-3 slug-ft^3</td>
<td>1.0%</td>
</tr>
</tbody>
</table>

*Denotes a Rayleigh distribution of magnitude with a Uniform 360° distribution in orientation.

**Drag variation with velocity closely approximated by \( C_{x_{D}} = C_{x_{\infty}} + \frac{K_D}{V^2} \) which is fit to two data points \((C_{x_1}, V_1) = (0.03585, 16,740)\) and \((C_{x_2}, V_2) = (0.11951, 3906)\). Uncertainty in \(C_{x_1}\) and \(C_{x_2}\).
Figure 1-2 HITS Overview
Table 1-3 presents the error budget generated by HITS for crossrange dispersion at nominal time corresponding to the error source model of Table 1-2. The error budget presents both the dispersion and the sensitivity coefficient associated with each error source. This identifies the larger contributors and those with significant potential. The total dispersion is given in the lower right hand corner. Table 1-3 illustrates the level of detail and types of information available from HITS. This is precisely the level of detail the engineer needs to assess the situation and make decisions. Supplemental information concerning the dispersion probability distribution can be developed if desired.

In summary, the methodology suggested by the HITS code consists of (1) constructing a comprehensive error source model and (2) processing it to obtain a detailed error budget. The error budget contains the information necessary to evaluate a projectile with respect to projectile dispersion program objectives and/or trade-off rival candidate designs. The method could also be employed in conjunction with ballistic range tests devoted to dispersion assessment with the objective of deriving additional insight. Whatever the motive, HITS provides a scientific, systematic method of evaluating projectile dispersion.

Presentation of the error source model and error budget, Tables 1-2 and 1-3, requires comment concerning their generality. The error source model contains implicit assumptions concerning (1) manufacturing tolerances on the projectile and barrel, (2) the amount of testing employed to determine the aerodynamic characteristics, and (3) the presumed accuracy of wind and density measurements. The error budget reflects these assumptions and further assumes the fire control trajectory algorithm contains no simplifying approximations. Thus, the error source model and error budget presented here are only for the purposes of illustration. They are not necessarily representative of any weapon system in the inventory, under procurement, or in development.

1.4 Code Limitations

This section states the limitations of the HITS code. Most of the items listed below are not theoretical limitations. Some are assumptions made for modeling convenience, while others are merely computer requirements. In any case, they are listed
Table 1-3  Crossrange Dispersion at Nominal Time  
(Nominal Range = 10 Kft)

<table>
<thead>
<tr>
<th>ERROR SOURCE</th>
<th>UNCERTAINTY</th>
<th>SENSITIVITY (MRAD/°)</th>
<th>1- σ ERROR (MRAD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIAL CONDITIONS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• VELOCITY VECTOR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>◦ MAGNITUDE</td>
<td>1/3%</td>
<td>0.001913</td>
<td>0.0006377</td>
</tr>
<tr>
<td>◦ ORIENTATION</td>
<td>0.0005 deg</td>
<td>13.91</td>
<td>0.006957</td>
</tr>
<tr>
<td>• INITIAL ORIENTATION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>◦ ATTITUDE</td>
<td>0.1 deg</td>
<td>0.01551</td>
<td>0.001551</td>
</tr>
<tr>
<td>◦ ATTITUDE RATE</td>
<td>55.0 rad/sec</td>
<td>0.02346</td>
<td>1.290</td>
</tr>
<tr>
<td>PHYSICAL CHARACTERISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• WEIGHT</td>
<td>1.0%</td>
<td>0.009808</td>
<td>0.009808</td>
</tr>
<tr>
<td>• MOMENTS OF INERTIA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>◦ AXIAL</td>
<td>1 2/3%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>◦ PITCH</td>
<td>1 2/3%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>• PHYSICAL DIMENSIONS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>◦ REFERENCE AREA</td>
<td>2/3%</td>
<td>0.003439</td>
<td>0.002292</td>
</tr>
<tr>
<td>◦ LENGTH</td>
<td>1/3%</td>
<td>0.01274</td>
<td>0.004248</td>
</tr>
<tr>
<td>◦ BASE DIAMETER</td>
<td>1/3%</td>
<td>0.00002225</td>
<td>0.00007418</td>
</tr>
<tr>
<td>AERODYNAMIC CHARACTERISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• STATIC COEFFICIENTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>◦ DRAG VARIATION EFFECTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>◦ VELOCITY</td>
<td>1.0%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>◦ ANGLE OF ATTACK</td>
<td>1.0%</td>
<td>0.0000007418</td>
<td>0.0000007418</td>
</tr>
<tr>
<td>◦ NORMAL FORCE</td>
<td>2.0%</td>
<td>0.0008093</td>
<td>0.001619</td>
</tr>
<tr>
<td>• DYNAMIC COEFFICIENTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>◦ PITCH DAMPING</td>
<td>20.0%</td>
<td>0.00001131</td>
<td>0.0002261</td>
</tr>
<tr>
<td>◦ MAGNUS MOMENT</td>
<td>0.0%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>• SPIN RATE</td>
<td>1.0%</td>
<td>0.0007241</td>
<td>0.0007241</td>
</tr>
<tr>
<td>• TRIM ANGLE OF ATTACK</td>
<td>0.1 deg</td>
<td>0.8103</td>
<td>0.08103</td>
</tr>
<tr>
<td>• STATIC MARGIN</td>
<td>1% L</td>
<td>0.2183</td>
<td>0.2183</td>
</tr>
<tr>
<td>ATMOSPHERIC EFFECTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• CONSTANT WINDS</td>
<td>11 ft/sec</td>
<td>0.01538</td>
<td>0.1692</td>
</tr>
<tr>
<td>• DENSITY VARIATIONS</td>
<td>1.0%</td>
<td>0.003429</td>
<td>0.003429</td>
</tr>
</tbody>
</table>

RSS TOTAL DISPERSION 1.322
below to aid the engineer in determining whether HITS is applicable to a particular problem and, if so, facilitate installation. The limitations are:

- The projectile drag model assumes the projectile does not experience transonic flow conditions at any time from launch to impact. A transonic condition would cause appreciable error only if it persisted over a significant portion of the flight time. This assumption can be overcome by additional code development. However, the code as presently configured is limited to "hypervelocity" projectiles.

- Muzzle blast, sabot separation and in-barrel dynamic effects are combined and represented by equivalent velocity and attitude perturbations at the muzzle. Ballistic range data reduction methods typically combine these effects in this manner.

- Wind gusts and density variations, that is fluctuations along the trajectory, are assumed to be zero. Covariance analysis using available wind gust spectra suggest wind gusts are a negligible source of dispersion for typical projectiles. On the other hand, steady winds are known to be a significant source and are accounted for by the HITS code.

- When operated in the Monte Carlo mode, HITS cannot assess the effects of error source model correlations. A program modification would be required. Correlation effects may, however, be evaluated via the Analytical Statistical mode.

- The HITS code is written in FORTRAN IV and requires approximately 175K bytes of computer memory. The card deck is punched in the EBCDIC format.

The random number generator (Subroutine RANDU) will work properly only on a 32 bit word computer, such as an IBM-360. The subroutine can be easily modified. However, some care must be exercised in performing the modification to insure the generated random sequences will reproduce those of the original code on a 32 bit word computer. Otherwise, the full instructional benefits of the example problems presented here will not be achieved, since it will not be possible to reproduce the Monte Carlo simulations.

- The code contains an interface for automatic computer plots. Since plotting software is hardware specific, this capability is not fully developed. The interface is thoroughly documented.

- The code was designed to facilitate real-time interactive mode operation from remote keyboard terminals. This capability is latent in the code, but not an operational reality.

1.5 Report Overview

Since this report is a User's Manual for the HITS code, the emphasis is on providing the information necessary to operate the code and interpret the results. The text presents the day-to-day necessities required to perform dispersion analyses. The appendices discuss the theoretical and programming aspects.

Chapter 2 presents an overview of the HITS code directed to the user. The objective is to acquaint the analyst quickly with the code. The basic code structure and options are described. The trajectory equations, which are the heart of the code, are discussed from a functional point of view. Chapter 2 concludes with a collection of topics peripheral to HITS, but pertinent to dispersion analysis.

Detailed input information is supplied in Chapter 3. A quick-reference step-by-step format is employed. Tables are used to define all variables and present all relevant information; comments direct the user to sections of this report containing related indepth discussions. Chapter 3 closes with an illustrative example demonstrating the encoding of the error source model of Table 1-2.
Chapter 4 presents four input-output example problems. Since
the computer printed output is self-explanatory, the initial im-
pression might be that these examples are superfluous. They are
not. They illustrate the output format and provide numerical
check problems. The accompanying text stresses interpretation.
Thus, users are encouraged to reproduce these results to famil-
iarize themselves with HITS and to verify the code at their
facility.

A brief summary is presented in Chapter 5. The ante-
cedants of the HITS code and related analyses are discussed.

The appendices relate the theoretical and programming
aspects of the HITS code. Appendix A is a summary discussion
of the relevant aspects of probability theory and statistics.
Appendix B discusses programming fine points. Appendix C math-
eptically develops the trajectory equations. Appendix D con-
tains the complete program listing.
2.0 OVERVIEW

The purpose of this chapter is to discuss the salient aspects of the HITS code. The objective is to present the user with a clear picture of the essence and function of the code as a whole. To this end, analytical and programming fine points are glossed over in an effort to place maximum emphasis on the big picture. Detailed "how to do it" information is presented in later chapters. Theoretical analyses and programming information are presented in the appendices.

Since projectile dispersion is most meaningfully stated in a statistical sense, the HITS code quantifies it in statistical terms. Thus, familiarity with the basic concepts of probability and statistics is essential to (1) the preparation of the engineering formulation of the HITS input, (2) a thorough understanding of the algorithms employed by HITS, and (3) the interpretation of the HITS output. Appendix A reviews the most pertinent elements of probability and statistics with the objective of supplying this understanding. HITS can be operated by personnel unfamiliar with probability and statistics. Given the information presented in Chapter 3, they can encode the formulated problem and obtain the desired numerical results.

Section 2.1 discusses the structure of the code. Analysis options are enumerated in Section 2.2. The close-form trajectory equations, which are the heart of the HITS code, are described in Section 2.3. Section 2.4 discusses three topics which are peripheral to HITS, but pertinent to dispersion analysis.

2.1 Code Structure

A flow diagram illustrating the basic structure of the HITS code is shown in Figure 2-1. As suggested by the figure, HITS basic function is to process an error source model to obtain a projectile dispersion error budget. This section describes the three computational modules that perform this task: (1) the Input Processor, (2) the Statistical Processor, and (3) the Projectile Trajectory Module. The code contains facilities for accessing a Time Phased Data Base for updating projectile characteristics under direction from a remote interactive computer terminal; however, this capability is not an operational reality at the present time. Sections 2.1.1 and 2.1.2 discuss the function of the Input and Statistical Processors which were taken from previously developed software. Section 2.1.3 functionally describes the Projectile Trajectory Module, which was developed specifically for dispersion assessment.
Figure 2-1  HITS Flow Diagram
2.1.1 Input Processor

The Input Processor reads error source model information from the input data stream. Two operations are performed once all the data has been read: (1) the input data is checked against a master list to determine whether crucial information is missing, and (2) the experimenter is informed as to the completeness of the stated error model. Missing data is automatically filled in with preset values.

The input data must define each element of the error source model (be it aerodynamic coefficient, physical dimension, or whatever) either as a constant or as a random variable. Random variables must be defined as to distribution type, mean or nominal value, and standard deviation (i.e., the 1-σ uncertainty). Gaussian, Rayleigh and uniform random variables are particularly easy to input, but any distribution is accepted. The input information is passed to the Statistical Processor.

2.1.2 Statistical Processor

The Statistical Processor derives dispersion statistics by referring to the Projectile Trajectory Module, as suggested by Figure 2-1. Each reference is the analytic equivalent of a ballistic range shot fired under the conditions specified by the Trajectory Module inputs. The Statistical Processor manipulates the Trajectory Module inputs in accord with the error source model uncertainties. The shot-by-shot dispersion returned by the Trajectory Module is compiled by the Statistical Processor to determine the dispersion error budget statistics.

2.1.3 Projectile Trajectory Module

The Projectile Trajectory Module consists of two sets of closed form parametric trajectory equations. The first computes the Fire Control predicted trajectory and the second simulates the true flight path or "Real World" Trajectory. The Trajectory Module returns the difference to the Statistical Processor.

The Fire Control equations are solved for the nominal time required to reach nominal range and the coordinates and velocity of the projectile at that time. This fixes the aim point in time as well as space. Typically, the parameters of the Fire Control equations are set equal to the error source model nominal values, so that the aim point is based on nominal muzzle velocity, weight, drag, etc. Simplifying assumptions found in operational
Fire Control computers can be simulated by judiciously selecting these parameters. Dispersion is computed relative to the aim point.

The Real World trajectory equations are solved for the actual projectile coordinates and velocity at nominal time and at nominal range. The distance between the projectile position at nominal time and the aim point defines dispersion for engagements with rapidly moving targets, where time of arrival is important. The displacement of the projectile position at nominal range from the aim point defines dispersion for engagements with slowly moving targets. The parameters of the Real World equations change from shot to shot and contain perturbations in accord with the error source model uncertainties. Thus, the dispersion returned to the Statistical Processor is consistent with the error source model and incorporates the effects of Fire Control simplifying assumptions.

2.2 Analysis Options

The Statistical Processor of Figure 2-1 was extracted from previously developed software. As a result, it has several desirable features unrelated to statistical dispersion assessment. The Statistical Processor has four modes of operation: (1) Single Trajectory, (2) Range Check, (3) Analytical Statistical, and (4) Monte Carlo. The first two modes are of primary interest when designing projectiles or analyzing various launch phenomena. The latter two modes are statistical methods for quantifying dispersion. The following subsections define the modes and suggest usages.

2.2.1 Single Trajectory

This mode computes the dispersion associated with a single shot at a given range. The Single Trajectory mode would be particularly useful in selecting nominal design parameters such as spin rate. A nominal trim lift would be input and the spin rate varied to minimize dispersion at maximum range.

2.2.2 Range Check

This mode evaluates the dispersion associated with a systematic variation of input parameters, thereby providing valuable design data. For instance, a Range Check on nominal range (5000, 6000, ---, 10,000 ft) would generate a complete trajectory. Alternatively, a Range Check on static margin (4, 5, and 6%) and spin rate (300, 400, and 500 rad/sec) would produce simulated shots with all possible combinations of static margin and spin rate. Comparison of the dispersions might suggest the optimal combination.
2.2.3 Analytical Statistical

The Analytical Statistical mode evaluates first and second order dispersion sensitivity coefficients by varying the trajectory inputs about the error source nominal. This mode computes, for example, the jump angle sensitivity to cross wind uncertainty; that is, the milliradians per ft/sec. The Analytical Statistical mode then uses the sensitivity coefficients to evaluate the average dispersion as well as the $1-\sigma$ uncertainty. The dispersion probability distribution is not determined. The analytic techniques employed permit rapid, low-cost dispersion assessment.

2.2.4 Monte Carlo

The Monte Carlo mode simulates ballistic range tests. It evaluates not only dispersion statistics (i.e., mean values and $\sigma$'s) but also the dispersion distribution. Numerous trajectories are computed using computer generated random numbers to simulate error source model uncertainties. Each simulated trajectory is comparable to a ballistic range shot, at a fraction of the cost. A test sequence is summarized by statistical indices and bar charts showing the number of times the projectiles had dispersions of say 0 to 0.25 milliradians, 0.25 to 0.50 milliradians, etc. The Monte Carlo mode gives the most complete dispersion picture.

2.3 Trajectory Equations

The Projectile Trajectory Module contains both Fire Control and Real World closed form trajectory equations. Both mechanize the theory presented in Appendix C. Section 2.3.1 discusses the rationale for the selection of closed form equations. An overview of the theory is presented in Section 2.3.2. Although the trajectory equations are used here to define dispersion, there are other potential applications, which include incorporation in operational Fire Control computers, the computation of firing tables, as well as any other situation requiring rapid, accurate, low-cost trajectory determination.

2.3.1 Closed Form Rationale

One approach to determining the Projectile Trajectory would be to perform full numerical simulations of the six degrees of freedom (6 DOF) differential equations of motion. However, since a Monte Carlo approach requires at least hundreds and possibly thousands of simulations in order to determine the dispersion
accurately, the comparatively long running times of 6 DOF simulations makes this approach impractical. Another approach is to develop approximate trajectory equations which can be rapidly evaluated, while at the same time include the effects of the major error sources. HITS uses the latter approach. The projectile equations of motion were simplified and the trajectory was determined in closed form. Perturbation equations are solved to refine the trajectory model. Six degrees of freedom simulations were conducted to authenticate the trajectory model approximations and are presented in Appendix C. In all cases the trajectory model checked out to within engineering tolerances.

2.3.2 Summary Description

The trajectory equations are composed of three parts, as illustrated in Figure 2-2. The first part is a particle trajectory which accounts for a large number of drag effects and the effect of constant velocity winds. The second part is a perturbation model which evaluates the effect of lift on crossrange dispersion. The final part is a perturbation model to account for the downrange effect of lift induced drag. The particle trajectory is evaluated first. Corrections for lifting effects are then computed via the perturbation equations and applied to the particle trajectory. The following paragraphs discuss the three components of the Trajectory Module.

Particle Trajectory

The particle trajectory equations shown in Figure 2-2 account for the downrange effects of muzzle velocity, projectile weight, drag, etc., as well as crossrange and downrange winds. The equations assume a zero angle of attack, a uniform density atmosphere, and the absence of gravity, and a variable drag coefficient. Classically, particle trajectories have approximated the drag coefficient as being independent of velocity. Although adequate for conventional muzzle velocities and short flight times, the drag coefficient model for hypervelocity projectiles must include the variation with velocity in order to avoid substantial errors, since the drag coefficient of typical hypervelocity projectiles (i.e., slender bodies) can vary by a factor of 3 to 4 over the velocity range of interest. The HITS particle trajectory models the drag coefficient as

\[ C_D = C_{D\infty} + \frac{K_D}{v^2} \]  

(2.3-1)
PROJECTILE MASS CHARACTERISTICS
PROJECTILE AERODYNAMIC PROPERTIES
INITIAL CONDITIONS AT LAUNCH

INPUT:

TRAJECTORY MODULE

PARTICLE TRAJECTORY EQUATIONS

PARTICLE TRAJECTORY

CROSS RANGE PERTURBATION EQUATIONS

ANGLE OF ATTACK HISTORY

DOWN RANGE PERTURBATION EQUATIONS

OUTPUT:
PROJECTILE TRAJECTORY IN TIME AND SPACE

Figure 2-2 Trajectory Equations Computational Procedure
which can also be expressed in terms of the ballistic coefficient \( \beta = \frac{W}{C_D A} \):

\[
\frac{1}{\beta} \frac{1}{\beta_0} \cdot \frac{K}{v^2}
\]

(2.3-2)

where \( C_{D00} \) and \( \beta_0 \) or \( \beta_\infty \) and \( K \) are empirically determined constants. This is an excellent model in the subsonic and supersonic velocity regimes. Ablative effects (i.e., tip recession and mass loss) can be approximated by making appropriate changes to the drag model parameters. Thus, including the drag variation with velocity opens the door to considering the effects of ablation.

**Crossrange Perturbation Equations**

This component of the Trajectory Model computes perturbations to the particle trajectory due to spurious lift forces caused by angle of attack oscillations and static trim angles. The angle of attack history is determined. The effect of a constant projectile spin rate is included. Velocity is assumed to be constant, so the crossrange error arising from transient roll resonance is not determined. Aerodynamic crossrange perturbation forces are significant only during the first few pitch oscillations. Since these occur near the muzzle while velocity is essentially constant, the constant velocity assumption is realistic. The crossrange perturbations are applied to the particle trajectory as a function of range. The particle trajectory includes the variation in velocity. The solution of these equations are applied as corrections to the particle trajectory as indicated in Figure 2-2.

**Downrange Perturbation Equations**

The downrange perturbation equations incorporate the effects of angle of attack variations on the drag coefficient. They include the effect of a constant spin rate. During the period when the projectile is oscillating, the perturbations to the velocity and flight time are determined by computing a mean drag coefficient with and without the angle of attack history as determined by the crossrange perturbation equations. The perturbations are applied as corrections to the velocity and flight time as determined by the (zero angle of attack) particle trajectory. The trajectory is carried beyond the point of angle of attack convergence by restarting the particle trajectory solution with the corrected velocity and time at convergence as initial conditions.
2.4 Related Topics

This section discusses three subjects which are peripheral to RITS, but pertinent to projectile dispersion analyses. Section 2.4.1 presents the various statistics customarily used to describe unsteady, unbiased, random dispersion analysis or related to error source models, as discussed in Section 2.4.2. Section 2.4.3 considers the effects of engagement geometry on target-fixed cross-range dispersion. Both of these topics has a bearing on the validity of the RITS code.

2.4.1 Cross-Range Dispersion Indices

Cross-range dispersion is an important measure of accuracy, particularly for engagements involving slowly moving targets and low-velocity weapons. Intuitively, cross-range dispersion is a measure of the size of the impact spread. This section defines the various statistics and presents conversion factors, which are useful to engineers in practical work.

- The percent standard deviation is denoted by \( P \).
- The jump angle, \( \gamma \), is the average of the impact point from the aim point, \( (x, y) \),

\[
P \left( \sqrt{x^2 + y^2} \right) = \ldots \quad \text{(2.4-1)}
\]

- The radius of the 90% circle is denoted \( R_{90} \).
- The Radial Standard Deviation (RSD) is the rms value of the impact point from the aim point, \( (x, y) \),

\[
\text{RSD} = \sqrt{\text{E}[x^2 + y^2]} \quad \text{(2.4-2)}
\]

where \( \text{E} \) denotes the ensemble average.

- The "jump angle", \( \gamma \), is the average displacement of the impact point from the aim point, \( (x, y) \),

\[
\gamma = \text{E} \left[ \sqrt{x^2 + y^2} \right] \quad \text{(2.4-3)}
\]

when the impact coordinates \( x \) and \( y \) are expressed as angular deflections.
There is a 50% probability the impact point coordinates relative to the aim point, \((x, y)\), will lie between two parallel lines which are equidistant from the origin and are separated by twice the Linear Error Probable (LEP).

The aim point is usually defined as the centroid of the impact points.

Since the components of crossrange dispersion can usually be assumed to have equal magnitude and be independent Gaussian random phenomenon, the statistical dispersion indices are proportional. The scale factors are presented in Figure 2-3. The HITS code evaluates the crossrange standard deviation, \(\sigma_{\text{CR}}\). The probability of occurrence associated with each index is given in Table 2-1.

2.4.2 Correlation Analysis - An Example

The error source model of Table 1-2 treats each source as an independent random phenomenon. Whereas this is a perfectly valid assumption in most instances, the potentiality exists that there are some subtle correlations (i.e., interrelationships) which could affect the dispersion statistics. The HITS code can evaluate the effects of correlation provided the correlation coefficient can be determined. This section illustrates analytic procedures for evaluating correlation coefficients from design information. The most direct approach would be to use ballistic range test data to evaluate the correlation. However, for the purposes of illustration, it is assumed this avenue is not open and the correlation must be evaluated from design information. The example is incomplete in the sense that no concrete conclusions are drawn. The objective is to illustrate that correlation coefficients are amenable to analyses based on physical considerations. The correlation coefficient is theoretically discussed in Appendix A.3.3.

Intuitively, muzzle velocity variations, \(\delta V\) and projectile weight variations, \(\delta W_p\), would be expected to be negatively correlated to some extent, since a heavier projectile should result in a lower muzzle velocity. These two sources are typically large contributors to downrange dispersion if they are considered uncorrelated. However, because a heavier projectile would be expected to slow down less, it might be suspected that when correlation is taken into account, weight and muzzle velocity effects compensate, and result in significantly less net downrange dispersion. Since they are likely to be correlated and their
Figure 2-3  Gaussian Scale Factors

Table 2-1  Probabilities of Occurrence

<table>
<thead>
<tr>
<th>Circle Radius</th>
<th>Probability of Occurrence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEP</td>
<td>20.3</td>
</tr>
<tr>
<td>$\sigma_{CR}$</td>
<td>39.3</td>
</tr>
<tr>
<td>CEP</td>
<td>50.0</td>
</tr>
<tr>
<td>$\bar{J}$</td>
<td>54.4</td>
</tr>
<tr>
<td>RSD</td>
<td>63.2</td>
</tr>
<tr>
<td>R80</td>
<td>80.0</td>
</tr>
</tbody>
</table>
correlation potentially could have a substantial effect on down-range dispersion, muzzle velocity and projectile weight variations are selected for the purpose of discussing analyses procedures.

Presuming the gun is a relatively uniform accelerator suggests an analysis of the kinetic energy imparted to the launch package might be the most direct approach to determining the correlation between projectile weight and muzzle velocity variations. (An analysis of the momentum imparted by launch might be a more fruitful approach in a launcher with an impulsive launch cycle.) The kinetic energy of the sabot/projectile is

\[ Ke = \frac{1}{2} \frac{W_p \cdot W_s}{g} v^2 \]  

(2.4-4)

where \( W_s \) is the sabot weight and \( g \) is the acceleration due to gravity. Viewing weight as the dependent variable, Eq. (2.4-4) can be linearized about the nominal launch conditions and solved for the variation in velocity.

\[ \frac{\delta v}{v} = \frac{1}{2} \left\{ \frac{\delta K_e}{K_e} \right\} \left[ \frac{\delta W_p}{W_p} \cdot \eta + (1-\eta) \frac{\delta W_s}{W_s} \right] \]  

(2.4-5)

where \( \eta \) is the design ratio of projectile weight to the total package weight. Equation (2.4-5) determines the muzzle velocity-projectile weight correlation coefficient. It is

\[ \rho W_p W_s = \frac{1}{2} \left\{ \frac{\sigma_{K_e}}{K_e} \right\} \left[ \frac{\alpha_{W_p}}{W_p} \cdot \eta + (1-\eta) \frac{\alpha_{W_s}}{W_s} \right] \]  

(2.4-6)

The correlation coefficient \( \rho W_p W_s \) is the correlation coefficient of the projectile and sabot weights. Since these are the result of different manufacturing processes, \( \rho W_p W_s \) is most probably zero. \( \sigma_{W_p} \) and \( \sigma_{V} \) are the standard deviations of the projectile weight and the muzzle velocity and are given by the basic error source model. \( \sigma_{K_e} \) is the standard deviation of the kinetic energy imparted to the launch package. Assuming the launch cycle is efficient, \( \sigma_{K_e} \) should be dominated by the variations in the chemical energy released by the propellant. (Other factors influencing \( \sigma_{K_e} \) would include heat absorbed by the gun, the energy lost by in-barrel balloting and friction, and the residual energy in the gun gases.) The launch package kinetic energy-projectile
weight correlation coefficient, $\rho_{\text{KeWp}}$, is an interesting one. At first these two quantities would appear to be unrelated ($\rho_{\text{KeWp}} = 0$). However, the projectile weight has an influence on the diameter of the projectile which in turn affects the tightness of the seal between the launch package and the barrel. By this argument an increased weight should cause more kinetic energy to be imparted to the launch package. Analysis of the projectile geometry and consideration of typical variations in material mass densities would determine the appropriate value of $\rho_{\text{KeWp}}$. All values would be substituted into Eq. (2.4-5) to determine the projectile weight-muzzle velocity correlation coefficient.

This example has illustrated the determination of correlation coefficients. They may be evaluated from ballistic range test data or design analyses. The analytical procedures are straightforward, systematic, and scientific and are based on the first principles of the physical sciences.

2.4.3 Target-Fixed Crossrange Dispersion

The HITS code defines projectile dispersion in a coordinate frame fixed with respect to the gun implacement. The dispersion must be transformed into a frame moving with the target in order to assess weapon system effectiveness. The conversion to target-fixed coordinates is usually performed during an engagement analysis and can be quite elaborate. This section presents a simplified procedure which allows the analyst to quickly determine the approximate target-fixed crossrange dispersion.

For the intercept geometry of Figure 2-4, the target-fixed crossrange dispersion in milliradians, $\sigma_{\text{TCD}}$, is given by

$$\sigma_{\text{TCD}} = \sqrt{\sigma_{\text{CR}}^2 + (S_{\text{C/D}} \sigma_{\text{DR}})^2}$$  \hspace{1cm} (2.4-7)

where $\sigma_{\text{CR}}$ and $\sigma_{\text{DR}}$ are the crossrange and downrange projectile dispersion standard deviations in milliradians and percent of range, respectively, and

$$S_{\text{C/D}} = \left| \frac{10 \sin \psi}{\frac{V_p}{V_t} + \cos \psi} \right|$$  \hspace{1cm} (2.4-8)
Figure 2-4 Intercept Geometry

Figure 2-5 Crossrange Sensitivity to Downrange Dispersion
where $V_p/V_t$ is the projectile to target velocity ratio in the vicinity of the target, and $\psi$ is the deflection angle, i.e., the angle between the target velocity vector and the bore sight.

The coefficient $SC/D$ is the equivalent crossrange dispersion sensitivity to downrange dispersion. The sensitivity coefficient is evaluated in Figure 2-5 for representative velocity ratios. Maximum sensitivity occurs at a deflection angle of

$$
\psi_{\text{max}} = 90^\circ + \sin^{-1}\left(\frac{V_t}{V_p}\right)
$$

The maximum sensitivity is

$$
(SC/D)_{\text{max}} = \frac{10}{\sqrt{(V_p/V_t)^2 - 1}}
$$

and can be used to perform a worst-case analysis.

This section has presented a simple approximate technique for converting projectile dispersion into a target-fixed coordinate frame. The results may be used in conjunction with target vulnerability, projectile terminal ballistics, and target lethality data to estimate the effect of projectile dispersion on weapon system effectiveness.
This chapter describes the order and content of the input cards. The input for a single case consists of three card groups. Section 3.1 defines the Statistical Processor Control Card (Card Group 1). Input Processor Control Cards (Card Group 2) are discussed in Section 3.2. The Optional Input Processor Control Cards (Card Group 3) are described in Section 3.3. The card groups are defined by cook-book tables that give the card format and options. Comments in the tables alert the user to related, in-depth discussions presented in other portions of the text. Section 3.4 illustrates encoding of an error source model.

The card groups for a single case must appear in the order suggested by the card group numbers. The card formats utilize fixed numeric fields to facilitate database operation. IBM-029 keypunch control cards are provided to simplify keypunching. All integer variables must be right-hand justified. Multiple cases may be stacked. Each case in the stack must be complete in itself and not simply changes from the prior case.

3.1 Statistical Processor Control Card (Card Group 1)

The Statistical Processor Control Card is the first card appearing in the deck describing a single case. This card directs the Statistical Processor to enter a specific mode of operation. Table 3-1 defines the format of the card and the various options. Figure 3-1 illustrates the associated IBM-029 keypunch control card.

The Statistical Processor Control Card (Card Group 1) described in Table 3-1, defines 13 variables. Only \( \text{IOPRNT} \) and \( \text{MCOPT} \) play significant roles and are discussed in depth. \( \text{IOPRNT} \) and \( \text{MCOPT} \) control the mode of the Statistical Processor as described in the following two paragraphs.

Referring to Table 3-1, the \( \text{IOPRNT} \) variable serves a dual purpose. When \( \text{IOPRNT} = 0 \), the Statistical Processor identifies the case as either a Single Trajectory or Range Check simulation, and relinquishes control to the Input Processor. The Input Processor determines the proper mode from the contents of Card Group 2. Whenever \( \text{IOPRNT} \geq 0 \), the Statistical Processor enters the Analytical Statistical mode and requests the Input Processor to supply the contents of Card Group 2.
Table 3-1 Statistical Processor Control Card (Card Group 1)

<table>
<thead>
<tr>
<th>COLUMN(S)</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IV</td>
<td>Dispersion analysis selector</td>
<td>• HITS has facilities to access (as yet undeveloped) alternate dispersion analysis procedures.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IV = 1 procedure described in this manual.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ISPRINT</td>
<td>Analytical Statistical mode calculation control. ISPRINT = 0 for Single Trajectory or Range Check modes. ISPRINT = 1 calculation stops after limiting cases. ISPRINT = 2 calculation stops after unmix first and second derivatives are evaluated. ISPRINT = 3 calculation stops after second order mean and first order variance are calculated. ISPRINT = 4 calculation stops after second order mixed partial derivatives are evaluated. ISPRINT = 5 calculation stops after second order mean and first order variance are calculated for correlated independent variables. ISPRINT = 6 calculation stops after second order variance calculation, assumes all independent variables are Gaussian. ISPRINT = 7 calculation stops after second order variance for correlated independent variables, assumes all independent variables are Gaussian.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• ISPRINT ≥ 3 for Monte Carlo mode. The number of TYPE = 2, 3, and 4 variables (K234) and the number of dependent variables (KD) must be limited so that KST(5) ≤ 5000 where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KST = KD(1+2K234), ISPRINT ≤ 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KST = KD(3+2K234), ISPRINT ≤ 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KST = KD(3+K234(2 + KD/2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Otherwise, the C array will be exceeded and computation will terminate abnormally.</td>
</tr>
</tbody>
</table>

*Card Group 3 required*
Table 3-1 Statistical Processor Control Card (Card Group 1) (cont'd)

<table>
<thead>
<tr>
<th>COLUMN(S)</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
</table>
| 7         | ISPRNT        | Trajectory Module output suppression code | • ISPRNT = 1 provides extensive information. Usually used only in Single Trajectory or Range Check mode.  
• ISPRNT = 0 recommended for all Monte Carlo mode and most Analytical Statistical mode operations. |
|           |               | ISPRNT = 0 trajectory module output suppressed |          |
|           |               | ISPRNT = 1 trajectory module output printed at each reference to trajectory module |          |
| 12        | MCPTP         | Monte Carlo mode control variable |          |
|           | MCPTP = 0     | no Monte Carlo experiments |          |
|           | MCPTP = 1     | Monte Carlo mode to be exercised |          |
| 17        | MCALC         | Monte Carlo control variable to bypass Trajectory module |          |
|           | MCALC = 0     | trajectory module specified by IV variable is used to evaluate all trajectories. |          |
|           | MCALC = 1, 2, 3 | a Taylor series approximation with (1) first order, (2) second order unmixed, and (3) full second order partial derivatives is used to approximate the full trajectory equations. |          |
| 21-22     | NCELL         | Number of histogram cells for each independent and dependent variable. | • See Appendix A.6.1 for guidelines. |
|           | NCELL ≤ 20    |                                          |          |
| 24-27     | NTRIAL        | Number of Monte Carlo experiments to be conducted | • See Appendix A.6.2 for guidelines. |
Table 3-1 Statistical Processor Control Card (Card Group 1) (concl’d)

<table>
<thead>
<tr>
<th>COLUMN(S)</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>29-32</td>
<td>IRJ1</td>
<td>Maximum number of Monte Carlo experiments to be rejected as outside internal ranges (predicted by Analytical Statistical mode) before computation is suspended.</td>
<td>IRJ1 should be large enough to allow for 10% MCELL rejections but small enough to abort simulations with gross input errors.</td>
</tr>
<tr>
<td>34-37</td>
<td>IRJ2</td>
<td>Maximum number of Monte Carlo experiments to be rejected as outside external ranges (TYPE = 8 variable) limits before computation is suspended.</td>
<td>See IRJ1 comment.</td>
</tr>
<tr>
<td>42</td>
<td>MCPPN</td>
<td>Monte Carlo mode Trajectory Module summary print control</td>
<td>Summary print should be suppressed if large numbers of Monte Carlo experiments are performed.</td>
</tr>
<tr>
<td></td>
<td>MCPPN</td>
<td>MCPPN = 0 suppress summary trajectory information for all Monte Carlo experiments.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MCPPN = 1 print summary information for each trajectory calculation.</td>
<td></td>
</tr>
<tr>
<td>44-47</td>
<td>IRRANG</td>
<td>Random number generator seed for Monte Carlo mode.</td>
<td>IRRANG should be a large odd positive integer such as 1201.</td>
</tr>
<tr>
<td>52</td>
<td>ISPLOT</td>
<td>Control variable for automatic histogram bar chart plots.</td>
<td>Plot facility is presently incomplete. Plot arrays are listed on the printed output. See Appendix B.2.5 for format.</td>
</tr>
<tr>
<td></td>
<td>ISPLOT</td>
<td>ISPLOT = 0 no plots</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISPLOT = 1 plot histograms</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>ISPLOT</td>
<td>Automatic histogram plot format control.</td>
<td>Presently has no effect on execution.</td>
</tr>
<tr>
<td></td>
<td>ISPLOT</td>
<td>ISPLOT = 0 generate individual plots.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISPLOT = 1 overlay internal and external range histograms.</td>
<td></td>
</tr>
</tbody>
</table>
The MC\textsc{opt} variable of Table 3-1 determines whether or not the Statistical Processor will enter the Monte Carlo mode after completing Analytical Statistical mode calculations. If MC\textsc{opt} = 1, the Statistical Processor automatically shifts into the Monte Carlo mode after the Analytical Statistical mode results have been used to set up the histograms.

3.2 Input Processor Control Cards (Card Group 2)

The Input Processor Control Cards appear second in the deck describing a single case. Each card directs the Input Processor to assign qualities to a trajectory variable. These qualities affect the calculations performed as described in Section 3.2.1. The trajectory variables are defined in Section 3.2.2.

3.2.1 Card Group 2 Assigned Qualities

Each Group 2 card defines two parameters: C\textsc{ode}\# and TYPE. C\textsc{ode}\# (i.e., code number) identifies the trajectory variable to which the qualities determined by the value of TYPE are to be assigned. This is one of the more interesting aspects of the HITS code. There are two basically different TYPE specifications that can be assigned. They are described in the succeeding paragraph. It is assumed the user wishes to determine the effect of projectile weight variations on downrange dispersion at nominal time, for the purposes of discussion.

Whenever $1 \leq \text{TYPE} \leq 5$, the variable is assigned the distinction of being an independent variable, that is, an independent variable whose value is determined by input data rather than preset (or default) values. In the example, projectile weight is the independent variable and would be assigned a TYPE between 1 and 5 depending on whether it is to be treated as a deterministic constant (different than the preset), a range check variable, or a random variable. Whenever $7 \leq \text{TYPE} \leq 8$, the variable is defined to be a dependent variable, that is, a dependent variable for which printed output is to be produced. In the example, downrange dispersion at nominal time would be declared a TYPE = 7 or TYPE = 8 variable, since the user desires to have the dispersion displayed. Thus, the C\textsc{ode}\# identifies the variable and TYPE determines what is to be done with it. All independent variables not mentioned in Card Group 2 are given their preset values. All dependent variables not mentioned are computed but not printed.

A Group 2 card may require more information than just the C\textsc{ode}\# and TYPE to completely define the variable. Table 3-2
<table>
<thead>
<tr>
<th>COLUMN(S)</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
</table>
| 3-5       | CODE#         | Independent/dependent variable identification number. | • See Table 3-3 for definitions.  
• Group 2 Cards may appear in any order.  
• Integer variable  
★ Negative value (e.g. "-1") required to indicate end of Card Group 2.  
• CODE# is an address in the SE array. |
| 7         | TYPE          | Statistical characterization variable for independent and dependent variables. | • Every Group 2 Card must define TYPE.  
• Integer variable  
• Maximum of three TYPE = 1 variables per Card Group 2 data set.  
• Minimum of one and a maximum of ten dependent variables (TYPE = 7 and TYPE = 8) per Card Group 2 data set.  
★ Additional data is required for TYPE = 1 variables.  
• Card format: 5 floating point fields of length 13 (beginning in column 1) per card.  
• Use multiple cards if required.  
• NT is defined later in this table.  

Independent Variables:  
TYPE = 1 Range Check variable to be parametrically varied over the NT values on the immediately succeeding card(s).
Table 3-2 Input Processor Control Cards (Card Group 2) (cont'd)

<table>
<thead>
<tr>
<th>COLUMN(S)</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>TYPE (cont'd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TYPE = 2</td>
<td>A Gaussian random variable</td>
<td>Gaussian random variables (TYPE = 2) are discussed in Appendices A.4.2 and A.4.3.</td>
</tr>
<tr>
<td></td>
<td>TYPE = 3</td>
<td>A uniformly distributed random variable.</td>
<td>Uniformly distributed random variables (TYPE = 3) are discussed in Appendix A.4.1.</td>
</tr>
<tr>
<td></td>
<td>TYPE = 4</td>
<td>Arbitrarily distributed random variable. Immediately succeeding cards must define the probability density function at NT points.</td>
<td>Additional data required for arbitrarily distributed (TYPE = 4) random variables.</td>
</tr>
</tbody>
</table>

- Card format: 5 floating point fields of length 11 each (beginning in column 1) per card.
- Multiple cards will be required.
- Probability Density Function (PDF) represented by two paired tables. Ordinate values (i.e., PDF values) must appear first. Corresponding abscissa values must appear second with abscissa table beginning on a new card.
- PDF assumed to be zero outside range of abscissa table, so use enough points.
- Abscissa table must contain equally spaced values.
- NT is defined later in this table.
<table>
<thead>
<tr>
<th>COLUMN(s)</th>
<th>DESCRIPTION</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>TYPE = 5</td>
<td>VALUE</td>
</tr>
<tr>
<td></td>
<td>TYPE = 6</td>
<td>VALUE</td>
</tr>
<tr>
<td></td>
<td>TYPE = 7</td>
<td>VALUE</td>
</tr>
<tr>
<td></td>
<td>TYPE = 8</td>
<td>VALUE</td>
</tr>
</tbody>
</table>

**Dependent Variables:**

- TYPE = 5 variables simply override preset (default) values.
- TYPE = 6 should not be used.
- Histogram INTERNAL RANGE and USER RANG are identical.
- Histogram INTERNAL RANGE determined by Analytical Statistical mode.
- Histogram USER RANG determined by user supplied values.
- Interpretation given in Appendix A.3 for TYPE = 2 and TYPE = 3 variables.

**Floating point variable variables:**

- Floating point variable.
- Floating point variable.
<table>
<thead>
<tr>
<th>COLUMN(S)</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>22-33</td>
<td>TDL (cont'd)</td>
<td>TYPE = 2 TDL is the incremental length used in computing the partial derivatives.</td>
<td>• For TYPE = 2, TDL may be set equal to TDL = 3*STDEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYPE = 3 TDL defines the variable as being uniformly distributed on the interval (VALUE-TDL, VALUE + TDL)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYPE = 8 TDL defines the USERS RANGES as (VALUE-TDL, VALUE + TDL)</td>
<td></td>
</tr>
<tr>
<td>35-46</td>
<td>STDEV</td>
<td>Standard deviation (σ) for some statistically defined variables.</td>
<td>• Floating point variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYPE = 1, 4, 5, 7, 8 STDEV not required. May be left blank.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYPE = 2 STDEV is the standard deviation of the Gaussian random variable.</td>
<td>• Interpretation given in Appendix A.3 for TYPE = 2 and TYPE = 3 variables.</td>
</tr>
</tbody>
</table>
|           |               | TYPE = 3 STDEV is the standard deviation of the uniformly distributed random variable. | • For a TYPE = 3 variable uniformly distributed on the interval (VALUE-TDL, VALUE + TDL), STDEV must be set equal to $STDEV = \frac{TDL}{\sqrt{3}}$
|           |               | | $= 0.577350*TDL$ |
| 48-50     | NT            | Table length for additional data on immediately succeeding cards. | • Integer variable. |
|           |               | TYPE = 2, 3, 5, 7, 8 NT not required. May be left blank. | • Succeeding cards required only for TYPE = 1 and TYPE = 4 variables. |
|           |               | TYPE = 1 NT is the number of values over which systematic variation is to occur. |      |
|           |               | TYPE = 4 NT is the number of pairs of points used to represent the probability density function. | • For TYPE = 4, NT is the number of ordinate-abscissa pairs. |
explains the details. In Table 3-2 hash marks are used to delimit the extent of the comments made in the right hand column. Comments identified by stars command action on the part of the user. Group 2 cards may be arranged in any order. IBM-029 keypunch control cards for Group 2 are presented in Figure 3-2.

3.2.2 Trajectory Variable Code Numbers

HITS uses "code numbers" to identify variables in input data cards and the printed output. CODE# in Card Group 2 is just the first occurrence of code numbers in the discussion. This section discusses Table 3-3 which catalogs the code numbers.

The code number of a variable is its address in the HITS' active storage array (i.e., the $OE$ array). Table 3-3 lists the code numbers and their associated FORTRAN names, preset values, units, and analysis names. The code numbers also link the HITS variables to the closed form trajectory equations of Appendix C via Table C-1. An asterisk attached to a FORTRAN name denotes a potential independent variable. All others may be defined as dependent variables. The presets are the nominal values of the Table 1-2 error source model.

Referring to Table 3-3, the code numbers from 1 to 100 define Fire Control trajectory variables. Code numbers from 101 to 200 are the Real World trajectory variables. Variables with code numbers from 201 to 300 are system controls. These are particularly important since they can greatly reduce the amount of input data and the number of computer runs. Note that variable 203, (i.e., IFC) distinguishes between Real World and Fire Control trajectory solutions. Variables 301 to 400 are measures of dispersion. Variables 401 to 500 are computed trajectory quantities. Code numbers from 501 to 600 denote variables defining the terminal conditions of the Real World and Fire Control trajectories.

3.3 Optional Input Processor Control Cards (Card Group 3)

Optional Input Processor Control Cards (Card Group 3) appear third (last) in the deck describing a single case. Card Group 3 may not appear when the Statistical Processor is operated in either the Single Trajectory or Range Check modes. It has no effect on the Monte Carlo mode. Only the Analytical Statistical mode is affected by Group 3 data.

Each Group 3 card directs the Analytical Statistical Processor to treat two of the statistically defined independent
NOTE: All Toggle Switches up

First Card

Drum Card

(b) Additional Data Control Cards

NOTE: All Toggle Switches up

First Card

Drum Card

(a) Basic Group 2 Control Cards

Figure 3-2 IBM-029 Keypunch Control Cards for Card Group 2
<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VOF*</td>
<td>11,000.</td>
<td>ft/sec</td>
<td>F.C.(1) Nozzle Velocity</td>
</tr>
<tr>
<td>2</td>
<td>WXF*</td>
<td>0.</td>
<td>ft/sec</td>
<td>F.C. Steady Winds:</td>
</tr>
<tr>
<td>3</td>
<td>WZF*</td>
<td>0.</td>
<td>ft/sec</td>
<td>Down Range</td>
</tr>
<tr>
<td>4</td>
<td>RHØF*</td>
<td>2.378x10^-3</td>
<td>slugs-ft³</td>
<td>Horiz. Cross Wind</td>
</tr>
<tr>
<td>5</td>
<td>CXLF*</td>
<td>3.585x10^-2</td>
<td>--</td>
<td>F.C. Projectile Drag Model</td>
</tr>
<tr>
<td>6</td>
<td>VLF*</td>
<td>1.674x10^4</td>
<td>ft/sec</td>
<td>$C_X = C_{X\infty} + K/V^2$</td>
</tr>
<tr>
<td>7</td>
<td>CX2F*</td>
<td>1.195x10^-1</td>
<td>--</td>
<td>Fit to</td>
</tr>
<tr>
<td>8</td>
<td>V2F*</td>
<td>3.906x10^3</td>
<td>ft/sec</td>
<td>$C_{X2F}&lt;C_{XLF}$</td>
</tr>
<tr>
<td>9</td>
<td>XNF*</td>
<td>10,000</td>
<td>ft</td>
<td>$V_{X2F}&gt;V_{XLF}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>CNAF*</td>
<td>1.9767</td>
<td>1/rad</td>
<td>F.C. Aerodynamic Effects:</td>
</tr>
<tr>
<td>11</td>
<td>SMF*</td>
<td>0.062</td>
<td>--</td>
<td>$C_N_a$</td>
</tr>
<tr>
<td>12</td>
<td>CMQF*</td>
<td>-7.5</td>
<td>--</td>
<td>Static Margin</td>
</tr>
<tr>
<td>13</td>
<td>CMPAF*</td>
<td>0.</td>
<td>--</td>
<td>$C_{m_q}$</td>
</tr>
<tr>
<td>14</td>
<td>ATRMSF*</td>
<td>0.</td>
<td>deg</td>
<td>Magnus Moment $C_{m_{pa}}$</td>
</tr>
<tr>
<td>15</td>
<td>BTRMSF*</td>
<td>0.</td>
<td>deg</td>
<td>Static Trim ($C_m=0$)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>THETOF*</td>
<td>0.</td>
<td>deg</td>
<td>$\theta_0$, vertical Plane</td>
</tr>
<tr>
<td>17</td>
<td>PSI0F*</td>
<td>0.</td>
<td>deg</td>
<td>$\psi_0$, Horizontal Plane</td>
</tr>
<tr>
<td>18</td>
<td>TDP0F*</td>
<td>0.</td>
<td>rad/sec</td>
<td>$\delta_0$, $\phi_0$, $\chi_0$</td>
</tr>
<tr>
<td>19</td>
<td>PD0F*</td>
<td>0.</td>
<td>rad/sec</td>
<td></td>
</tr>
</tbody>
</table>

*Denotes Potential Independent Variable
(1) Indicates Fire Control Parameter
<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>GAMOF*</td>
<td>0.</td>
<td>deg</td>
<td>F.C. Initial Attitude (Cont'd)</td>
</tr>
<tr>
<td>21</td>
<td>AZOF*</td>
<td>0.</td>
<td>deg</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>AF*</td>
<td>3.068x10^-3</td>
<td>ft^2</td>
<td>F.C. Projectile Physical Characteristics:</td>
</tr>
<tr>
<td>23</td>
<td>ELLF*</td>
<td>3.1x10^-1</td>
<td>ft</td>
<td>A - Base Area (πD^2/4)</td>
</tr>
<tr>
<td>24</td>
<td>DF*</td>
<td>6.25x10^-2</td>
<td>ft</td>
<td>L - Length</td>
</tr>
<tr>
<td>25</td>
<td>WF*</td>
<td>1.1x10^-1</td>
<td>lbs</td>
<td>D - Base Diameter</td>
</tr>
<tr>
<td>26</td>
<td>AIXF*</td>
<td>1.235x10^-6</td>
<td>slugs-ft^2</td>
<td>W - Weight</td>
</tr>
<tr>
<td>27</td>
<td>AIFY*</td>
<td>2.110x10^-5</td>
<td>slugs-ft^2</td>
<td>I_x - Roll Moment of Inertia</td>
</tr>
<tr>
<td>28</td>
<td>PF*</td>
<td>400.</td>
<td>rad/sec</td>
<td>I_y - Pitch Moment of Inertia</td>
</tr>
<tr>
<td>29</td>
<td>ALPCOF*</td>
<td>0.5</td>
<td>deg</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>CAAF*</td>
<td>0.</td>
<td>1/deg^2</td>
<td></td>
</tr>
</tbody>
</table>

*Potential Independent Variable
<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>VO*</td>
<td>11,000</td>
<td>ft/sec</td>
<td>R.W. Muzzle Velocity</td>
</tr>
<tr>
<td>102</td>
<td>WX*</td>
<td>0.</td>
<td>ft/sec</td>
<td>R.W. Steady Winds Down Range</td>
</tr>
<tr>
<td>103</td>
<td>WZ*</td>
<td>0.</td>
<td>ft/sec</td>
<td>Horiz. Cross Wind</td>
</tr>
<tr>
<td>104</td>
<td>RHø*</td>
<td>2.378x10^-3</td>
<td>slugs-ft³</td>
<td>R.W. Atmospheric Density</td>
</tr>
<tr>
<td>105</td>
<td>CX1*</td>
<td>3.585x10^-2</td>
<td></td>
<td>R.W. Projectile Drag Model</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$C_X = C_{X∞} + K/V^2$</td>
</tr>
<tr>
<td>106</td>
<td>V1*</td>
<td>1.674x10^4</td>
<td>ft/sec</td>
<td>Fit to</td>
</tr>
<tr>
<td>107</td>
<td>CX2*</td>
<td>1.195x10^-1</td>
<td></td>
<td>$CX2 &lt; CX1$</td>
</tr>
<tr>
<td>108</td>
<td>V2*</td>
<td>3.906x10^3</td>
<td>ft/sec</td>
<td>VX2 &gt; VX1</td>
</tr>
<tr>
<td>109</td>
<td>XN*</td>
<td>10,000</td>
<td>ft</td>
<td>R.W. Nominal Range</td>
</tr>
<tr>
<td>110</td>
<td>CNA*</td>
<td>1.9767</td>
<td>l/rad</td>
<td>R.W. Aerodynamic Effects:</td>
</tr>
<tr>
<td>111</td>
<td>SM*</td>
<td>0.062</td>
<td></td>
<td>• C_{Na}</td>
</tr>
<tr>
<td>112</td>
<td>CMQ*</td>
<td>-7.5</td>
<td></td>
<td>• Static Margin</td>
</tr>
<tr>
<td>113</td>
<td>CMPA*</td>
<td>0.</td>
<td></td>
<td>• C_{mq}</td>
</tr>
<tr>
<td>114</td>
<td>ATRMS*</td>
<td>0.</td>
<td>deg</td>
<td>• Magnus Moment C_{mpa}</td>
</tr>
<tr>
<td>115</td>
<td>BTRMS*</td>
<td>0.</td>
<td>deg</td>
<td>• Static Trim $C_{m=0}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-Angle of Attack</td>
</tr>
<tr>
<td>116</td>
<td>THET0*</td>
<td>0.</td>
<td>deg</td>
<td>R.W. Initial Attitude</td>
</tr>
<tr>
<td>117</td>
<td>PSI0*</td>
<td>0.</td>
<td>deg</td>
<td>$θ_o$ Vertical Plane</td>
</tr>
<tr>
<td>118</td>
<td>TDST0*</td>
<td>0.</td>
<td>rad/sec</td>
<td>$ψ_o$ Horizontal Plane</td>
</tr>
<tr>
<td>119</td>
<td>FDST0*</td>
<td>0.</td>
<td>rad/sec</td>
<td>$δ_o$</td>
</tr>
</tbody>
</table>

*Potential Independent Variable
(1) Denotes Real World Parameter
<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>GAMO*</td>
<td>0.</td>
<td>deg</td>
<td>R.W. Initial Attitude (Cont'd)</td>
</tr>
<tr>
<td>121</td>
<td>AZO*</td>
<td>0.</td>
<td>deg</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>A*</td>
<td>3.068x10^-3</td>
<td>ft²</td>
<td>R.W. Projectile Physical Characteristics</td>
</tr>
<tr>
<td>123</td>
<td>ELL*</td>
<td>3.1x10^-1</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>D*</td>
<td>6.25x10^-2</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>W*</td>
<td>1.1x10^-1</td>
<td>lbs</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>AIX*</td>
<td>1.235x10^-6</td>
<td>slugs-ft²</td>
<td>I_x - Roll Moment of Inertia</td>
</tr>
<tr>
<td>127</td>
<td>AIY*</td>
<td>2.110x10^-5</td>
<td>slugs-ft²</td>
<td>I_y - Pitch Moment of Inertia</td>
</tr>
<tr>
<td>128</td>
<td>P*</td>
<td>400.</td>
<td>rad/sec</td>
<td>R.W. Spin Rate</td>
</tr>
<tr>
<td>129</td>
<td>ALPCON*</td>
<td>0.5</td>
<td>deg</td>
<td>R.W. Angle of Attack Oscillation Convergence Criterion</td>
</tr>
<tr>
<td>130</td>
<td>CAA*</td>
<td>0.</td>
<td>1/deg²</td>
<td>R.W. Second Order Drag Effect</td>
</tr>
<tr>
<td>131</td>
<td>XBART*</td>
<td>0.</td>
<td>ft</td>
<td>Externally Supplied SEP Center</td>
</tr>
<tr>
<td>132</td>
<td>YBART*</td>
<td>0.</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>ZBART*</td>
<td>0.</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>YBARX*</td>
<td>0.</td>
<td>ft</td>
<td>Externally Supplied CEP Center</td>
</tr>
<tr>
<td>135</td>
<td>ZBARX*</td>
<td>0.</td>
<td>ft</td>
<td></td>
</tr>
</tbody>
</table>

*Potential Independent Variable
<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>I</td>
<td>0.</td>
<td>--</td>
<td>System Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IFCRW=1 F.C. Parameters are set equal to R.W. nominal values</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IFCRW=0 F.C. Parameters determined solely by inputs</td>
</tr>
<tr>
<td>202</td>
<td>IFCRC*</td>
<td>0.</td>
<td>--</td>
<td>System Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IFCRC=1 F.C. Parameters are set equal to R.W. values for each Range Check Combination</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IFCRC=0 F.C. Parameters determined solely by inputs.</td>
</tr>
<tr>
<td>203</td>
<td>IFC</td>
<td>1.</td>
<td>--</td>
<td>Internal Sequencing Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IFC = 1 F.C. Solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IFC = 0 R.W. Solution</td>
</tr>
<tr>
<td>204</td>
<td>IDUMP*</td>
<td>0.</td>
<td>--</td>
<td>System Print Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IDUMP = 1 Print lots of intermediate data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IDUMP = 0 Print summary data only</td>
</tr>
<tr>
<td>205</td>
<td>ICONV*</td>
<td>-10.</td>
<td>--</td>
<td>Trajectory Module Iteration Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TOL = 10.ICONV</td>
</tr>
<tr>
<td>206</td>
<td>ICNCL*</td>
<td>1.</td>
<td>--</td>
<td>Trajectory Aerodynamic Control Parameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ICNCL = 1 $C_L$ used in crossrange perturbation. Recommended for most applications.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ICNCL = 0 $C_N$ used in crossrange perturbation.</td>
</tr>
</tbody>
</table>

* Potential Independent Variable
<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>301</td>
<td>XBARTI</td>
<td>--</td>
<td>ft</td>
<td>Internally Computed SEP Center (\delta x_t) (Displacements from Fire Control Nominal Aim Point)</td>
</tr>
<tr>
<td>302</td>
<td>YBARTI</td>
<td>--</td>
<td>ft</td>
<td>(\delta y_T)</td>
</tr>
<tr>
<td>303</td>
<td>ZBARTI</td>
<td>--</td>
<td>ft</td>
<td>(\delta z_T)</td>
</tr>
<tr>
<td>304</td>
<td>YBARXI</td>
<td>--</td>
<td>ft</td>
<td>Internally Computed CEP Center (\delta y_x) (Displacements from Fire Control Nominal Aim Point)</td>
</tr>
<tr>
<td>305</td>
<td>ZBARXI</td>
<td>--</td>
<td>ft</td>
<td>(\delta z_x)</td>
</tr>
<tr>
<td>306</td>
<td>DXT</td>
<td>--</td>
<td>ft</td>
<td>Projectile Position at Nominal Time (\delta x_t) (Displacement from Fire Control Nominal Aim Point)</td>
</tr>
<tr>
<td>307</td>
<td>DYT</td>
<td>--</td>
<td>ft</td>
<td>(\delta y_t)</td>
</tr>
<tr>
<td>308</td>
<td>DZT</td>
<td>--</td>
<td>ft</td>
<td>(\delta z_t)</td>
</tr>
<tr>
<td>309</td>
<td>DXY</td>
<td>--</td>
<td>ft</td>
<td>Projectile Position at Nominal Range (\delta y_x) (Displacement from Fire Control Nominal Aim Point)</td>
</tr>
<tr>
<td>310</td>
<td>DZX</td>
<td>--</td>
<td>ft</td>
<td>(\delta z_x)</td>
</tr>
<tr>
<td>311</td>
<td>DVXT</td>
<td>--</td>
<td>ft/sec</td>
<td>Velocity Difference at Nominal Time (\delta V_{xt}) (Difference from F.C. Value at Nominal Time)</td>
</tr>
<tr>
<td>312</td>
<td>DVYT</td>
<td>--</td>
<td>ft/sec</td>
<td>(\delta V_{yt})</td>
</tr>
<tr>
<td>313</td>
<td>DVZT</td>
<td>--</td>
<td>ft/sec</td>
<td>(\delta V_{zt})</td>
</tr>
<tr>
<td>314</td>
<td>DVXX</td>
<td>--</td>
<td>ft/sec</td>
<td>Velocity Difference at Nominal Range (\delta V_{xx}) (Difference from F.C. Value at Nominal Time)</td>
</tr>
<tr>
<td>315</td>
<td>DVYX</td>
<td>--</td>
<td>ft/sec</td>
<td>(\delta V_{yx})</td>
</tr>
<tr>
<td>316</td>
<td>DVZX</td>
<td>--</td>
<td>ft/sec</td>
<td>(\delta V_{zx})</td>
</tr>
<tr>
<td>317</td>
<td>DT</td>
<td>--</td>
<td>sec</td>
<td>(\delta t) Time to Nominal Range less F.C. Nominal Time to Nominal Range</td>
</tr>
</tbody>
</table>
Table 3-3 HITS Trajectory Parameter List (Cont'd)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>318</td>
<td>RADT&lt;sup&gt;+&lt;/sup&gt;</td>
<td>--</td>
<td>ft</td>
<td>Radial Displacement from Externally Supplied SEP Center</td>
</tr>
<tr>
<td>319</td>
<td>RADX&lt;sup&gt;+&lt;/sup&gt;</td>
<td>--</td>
<td>ft</td>
<td>Radial Displacement from Externally Supplied CEP Center</td>
</tr>
<tr>
<td>320</td>
<td>RH&lt;sub&gt;0&lt;/sub&gt;T&lt;sup&gt;(1)&lt;/sup&gt;</td>
<td>--</td>
<td>ft</td>
<td>Radial Displacement from Internally Computed SEP Center</td>
</tr>
<tr>
<td>321</td>
<td>RH&lt;sub&gt;0&lt;/sub&gt;X&lt;sup&gt;(2)&lt;/sup&gt;</td>
<td>--</td>
<td>ft</td>
<td>Radial Displacement from Internally Computed CEP Center</td>
</tr>
<tr>
<td>322</td>
<td>DX&lt;sub&gt;T&lt;/sub&gt;I</td>
<td>--</td>
<td>ft</td>
<td>Projectile Position at Nominal Time (\delta_{x_{ti}}) (Displacement from Internally Computed SEP Center)</td>
</tr>
<tr>
<td>323</td>
<td>DY&lt;sub&gt;T&lt;/sub&gt;I</td>
<td>--</td>
<td>ft</td>
<td>(\delta_{y_{ti}})</td>
</tr>
<tr>
<td>324</td>
<td>DZ&lt;sub&gt;T&lt;/sub&gt;I</td>
<td>--</td>
<td>ft</td>
<td>(\delta_{z_{ti}})</td>
</tr>
<tr>
<td>325</td>
<td>DX&lt;sub&gt;E&lt;/sub&gt;T</td>
<td>--</td>
<td>ft</td>
<td>Projectile Position at Nominal Time (\delta_{x_{te}}) (Displacement from Externally Supplied SEP Center)</td>
</tr>
<tr>
<td>326</td>
<td>DY&lt;sub&gt;E&lt;/sub&gt;T</td>
<td>--</td>
<td>ft</td>
<td>(\delta_{y_{te}})</td>
</tr>
<tr>
<td>327</td>
<td>DZ&lt;sub&gt;E&lt;/sub&gt;T</td>
<td>--</td>
<td>ft</td>
<td>(\delta_{z_{te}})</td>
</tr>
<tr>
<td>328</td>
<td>DYX&lt;sub&gt;I&lt;/sub&gt;X</td>
<td>--</td>
<td>ft</td>
<td>Projectile Position at Nominal Range (\delta_{y_{xi}}) (Displacement from Internally Computed CEP Center)</td>
</tr>
<tr>
<td>329</td>
<td>DZX&lt;sub&gt;I&lt;/sub&gt;X</td>
<td>--</td>
<td>ft</td>
<td>(\delta_{z_{xi}})</td>
</tr>
<tr>
<td>330</td>
<td>DYX&lt;sub&gt;E&lt;/sub&gt;E</td>
<td>--</td>
<td>ft</td>
<td>Projectile Position at Nominal Range (\delta_{y_{xe}}) (Displacement from Externally Supplied CEP Center)</td>
</tr>
<tr>
<td>331</td>
<td>DZX&lt;sub&gt;E&lt;/sub&gt;X</td>
<td>--</td>
<td>ft</td>
<td>(\delta_{z_{xe}})</td>
</tr>
<tr>
<td>332</td>
<td>DXD&lt;sub&gt;Y&lt;/sub&gt;T</td>
<td>--</td>
<td>ft&lt;sup&gt;2&lt;/sup&gt;</td>
<td>Correlations of Projectile Position at Nominal Time (\delta_{x_{t}} \delta_{y_{t}}) (Displacements from Fire Control Nominal)</td>
</tr>
<tr>
<td>333</td>
<td>DXD&lt;sub&gt;D&lt;/sub&gt;DT</td>
<td>--</td>
<td>ft&lt;sup&gt;2&lt;/sup&gt;</td>
<td>(\delta_{y_{t}} \delta_{z_{t}}) Aim Point</td>
</tr>
<tr>
<td>334</td>
<td>DYD&lt;sub&gt;D&lt;/sub&gt;DT</td>
<td>--</td>
<td>ft&lt;sup&gt;2&lt;/sup&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup> TYPE = 8 only
<sup>(1)</sup> Requires Code Numbers 306, 307, and 308 be TYPE = 7 or TYPE = 8.
<sup>(2)</sup> Requires Code Numbers 309 and 310 be TYPE = 7 or TYPE = 8.
<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>335</td>
<td>DYDZX</td>
<td>--</td>
<td>ft²</td>
<td>Correlation at Nominal Range $\delta y_x \delta z_x$ (Displacements from F.C. Nominal Aim Point)</td>
</tr>
<tr>
<td>336</td>
<td>DXDYTE</td>
<td>--</td>
<td>ft²</td>
<td>Correlations about Internally Computed SEP Center at Nominal Time $\delta x_{ti} \delta y_{ti}$ (Displacements from Internally Computed SEP Center)</td>
</tr>
<tr>
<td>337</td>
<td>DXDZM</td>
<td>--</td>
<td>ft²</td>
<td>Correlations about Internally Computed CEP Center at Nominal Range $\delta y_{xi} \delta z_{xi}$ (Displacements from Internally Computed CEP Center)</td>
</tr>
<tr>
<td>338</td>
<td>DYDZM</td>
<td>--</td>
<td>ft²</td>
<td>Correlations about Externally Supplied SEP Center at Nominal Time $\delta x_{te} \delta y_{te}$ (Displacements from Externally Supplied SEP Center)</td>
</tr>
<tr>
<td>339</td>
<td>DYDZXI</td>
<td>--</td>
<td>ft²</td>
<td>Correlations about Externally Supplied CEP Center at Nominal Range $\delta y_{xe} \delta z_{xe}$ (Displacements from Externally Supplied CEP Center)</td>
</tr>
<tr>
<td>Code Number</td>
<td>FORTRAN Name</td>
<td>Preset</td>
<td>Units</td>
<td>Remarks</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>--------</td>
<td>-------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>400</td>
<td>ALPHA</td>
<td>--</td>
<td>rad</td>
<td>$a (t_N)$ or $a(x_N)$</td>
</tr>
<tr>
<td>401</td>
<td>ATST</td>
<td>--</td>
<td>rad</td>
<td>$\mathcal{A} (t_N)$ or $a(x_N)$</td>
</tr>
<tr>
<td>402</td>
<td>ALPTRM</td>
<td>--</td>
<td>rad</td>
<td>Rolling Trim $a$</td>
</tr>
<tr>
<td>403</td>
<td>ADPHDO</td>
<td>--</td>
<td>rad/sec</td>
<td>$\beta$</td>
</tr>
<tr>
<td>404</td>
<td>B</td>
<td>--</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>405</td>
<td>BETTRM</td>
<td>--</td>
<td>rad</td>
<td>Rolling Trim $\beta$</td>
</tr>
<tr>
<td>406</td>
<td>BETA0</td>
<td>--</td>
<td>rad</td>
<td>$\beta_0$</td>
</tr>
<tr>
<td>407</td>
<td>BETADO</td>
<td>--</td>
<td>rad/sec</td>
<td>$\dot{\beta}_0$</td>
</tr>
<tr>
<td>408</td>
<td>BETA</td>
<td>--</td>
<td>rad</td>
<td>$\beta (t_N)$ or $\beta(x_N)$</td>
</tr>
<tr>
<td>409</td>
<td>ALPMAX</td>
<td>--</td>
<td>rad</td>
<td>$\mathcal{A}_{\text{UPPER}}$</td>
</tr>
<tr>
<td>410</td>
<td>ALPMIN</td>
<td>--</td>
<td>rad</td>
<td>$\mathcal{A}_{\text{LOWER}}$</td>
</tr>
<tr>
<td>411</td>
<td>CAYD</td>
<td>--</td>
<td>(ft/sec)$^2$</td>
<td>$K_D$</td>
</tr>
<tr>
<td>412</td>
<td>CD8</td>
<td>--</td>
<td>--</td>
<td>$C_{D_{\infty}}$</td>
</tr>
<tr>
<td>413</td>
<td>CMA</td>
<td>--</td>
<td>l/rad</td>
<td>$C_{M_a}$</td>
</tr>
<tr>
<td>414</td>
<td>CMTHTD</td>
<td>--</td>
<td>sec</td>
<td>$C_{M_\delta}$</td>
</tr>
<tr>
<td>415</td>
<td>CMTHTA</td>
<td>--</td>
<td>sec</td>
<td>$C_{M_{\phi a}}$</td>
</tr>
<tr>
<td>416</td>
<td>CAY1</td>
<td>--</td>
<td>rad</td>
<td>$K_1$</td>
</tr>
<tr>
<td>417</td>
<td>CAY2</td>
<td>--</td>
<td>rad</td>
<td>$K_2$</td>
</tr>
<tr>
<td>418</td>
<td>CDA0B</td>
<td>--</td>
<td>--</td>
<td>$\overline{C_D} \ (\mathcal{A} = 0)$</td>
</tr>
<tr>
<td>419</td>
<td>CDAB</td>
<td>--</td>
<td>--</td>
<td>$\overline{C_D}$</td>
</tr>
</tbody>
</table>
Table 3-3 HITS Trajectory Parameter List (Cont'd)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>FORTRAN Name</th>
<th>Preset</th>
<th>Units</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>420</td>
<td>DELTT</td>
<td>--</td>
<td>sec</td>
<td>$\delta t$</td>
</tr>
<tr>
<td>421</td>
<td>DELV</td>
<td>--</td>
<td>ft/sec</td>
<td>$\delta V$</td>
</tr>
<tr>
<td>422</td>
<td>DELT</td>
<td>--</td>
<td>sec</td>
<td>$\gamma_t$</td>
</tr>
<tr>
<td>423</td>
<td>DELX</td>
<td>--</td>
<td>ft</td>
<td>$\Delta x$</td>
</tr>
<tr>
<td>424</td>
<td>DYDXO</td>
<td>--</td>
<td>rad</td>
<td>$\gamma_o$</td>
</tr>
<tr>
<td>425</td>
<td>DZDXO</td>
<td>--</td>
<td>rad</td>
<td>$\gamma_{z_0}$</td>
</tr>
<tr>
<td>426</td>
<td>DYDTO</td>
<td>--</td>
<td>ft/sec</td>
<td>$V_y(0)$</td>
</tr>
<tr>
<td>427</td>
<td>DZDTO</td>
<td>--</td>
<td>ft/sec</td>
<td>$V_z(0)$</td>
</tr>
<tr>
<td>428</td>
<td>DELW</td>
<td>--</td>
<td>1/sec</td>
<td>$\Delta \omega$</td>
</tr>
<tr>
<td>429</td>
<td>DELLAM</td>
<td>--</td>
<td>1/sec</td>
<td>$\Delta \lambda$</td>
</tr>
<tr>
<td>430</td>
<td>EYEP</td>
<td>--</td>
<td>sec$^2$</td>
<td>$\delta$</td>
</tr>
<tr>
<td>431</td>
<td>EMP</td>
<td>--</td>
<td>sec</td>
<td>$m'$</td>
</tr>
<tr>
<td>432</td>
<td>EOLT</td>
<td>--</td>
<td>--</td>
<td>$e^\lambda_o t'$</td>
</tr>
<tr>
<td>433</td>
<td>EDLT</td>
<td>--</td>
<td>--</td>
<td>$e^{\Delta \lambda_o t'}$</td>
</tr>
<tr>
<td>434</td>
<td>F</td>
<td>--</td>
<td>1/ft</td>
<td>$f$</td>
</tr>
<tr>
<td>435</td>
<td>H</td>
<td>--</td>
<td>ft/sec$^2$</td>
<td>$h$</td>
</tr>
<tr>
<td>436</td>
<td>PSIDO0</td>
<td>--</td>
<td>rad/sec</td>
<td>$\dot{\psi}_o$</td>
</tr>
<tr>
<td>437</td>
<td>PHI0</td>
<td>--</td>
<td>rad</td>
<td>$\phi_o$</td>
</tr>
<tr>
<td>438</td>
<td>PHI1</td>
<td>--</td>
<td>rad</td>
<td>$\phi_1$</td>
</tr>
<tr>
<td>439</td>
<td>PHI2</td>
<td>--</td>
<td>rad</td>
<td>$\phi_2$</td>
</tr>
<tr>
<td>Code Number</td>
<td>FORTRAN Name</td>
<td>Preset</td>
<td>Units</td>
<td>Remarks</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>--------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>440</td>
<td>PSI0</td>
<td>--</td>
<td>rad</td>
<td>$\psi_o$</td>
</tr>
<tr>
<td>441</td>
<td>R1</td>
<td>--</td>
<td>rad</td>
<td>$R_1$</td>
</tr>
<tr>
<td>442</td>
<td>R2</td>
<td>--</td>
<td>rad</td>
<td>$R_2$</td>
</tr>
<tr>
<td>443</td>
<td>R3</td>
<td>--</td>
<td>rad</td>
<td>$R_3$</td>
</tr>
<tr>
<td>444</td>
<td>R4</td>
<td>--</td>
<td>rad</td>
<td>$R_4$</td>
</tr>
<tr>
<td>445</td>
<td>RTRIM</td>
<td>--</td>
<td>--</td>
<td>$R_{TRIM}$</td>
</tr>
<tr>
<td>446</td>
<td>TC</td>
<td>--</td>
<td>sec</td>
<td>$t_c$</td>
</tr>
<tr>
<td>447</td>
<td>TCO</td>
<td>--</td>
<td>sec</td>
<td>$t_{c0}$</td>
</tr>
<tr>
<td>448</td>
<td>TS</td>
<td>--</td>
<td>sec</td>
<td>$t*$</td>
</tr>
<tr>
<td>449</td>
<td>TGO</td>
<td>--</td>
<td>sec</td>
<td>$t_{G0}$</td>
</tr>
<tr>
<td>450</td>
<td>TP</td>
<td>--</td>
<td>sec</td>
<td>$t'$</td>
</tr>
<tr>
<td>451</td>
<td>THETA0</td>
<td>--</td>
<td>rad</td>
<td>$\delta_{o}$</td>
</tr>
<tr>
<td>452</td>
<td>THETDO</td>
<td>--</td>
<td>rad/sec</td>
<td>$\theta_{o}$</td>
</tr>
<tr>
<td>453</td>
<td>TGL</td>
<td>--</td>
<td>--</td>
<td>$TOL = 10^{ICONV}$</td>
</tr>
<tr>
<td>454</td>
<td>VC</td>
<td>--</td>
<td>ft/sec</td>
<td>$V_c$</td>
</tr>
<tr>
<td>455</td>
<td>VCO</td>
<td>--</td>
<td>ft/sec</td>
<td>$V_{c0}$</td>
</tr>
<tr>
<td>456</td>
<td>VYA</td>
<td>--</td>
<td>ft/sec</td>
<td>$V_{ya}$</td>
</tr>
<tr>
<td>457</td>
<td>VZA</td>
<td>--</td>
<td>ft/sec</td>
<td>$V_{za}$</td>
</tr>
<tr>
<td>458</td>
<td>WX</td>
<td>--</td>
<td>ft/sec</td>
<td>$W_x$</td>
</tr>
<tr>
<td>459</td>
<td>WZ</td>
<td>--</td>
<td>ft/sec</td>
<td>$W_z$</td>
</tr>
<tr>
<td>Code Number</td>
<td>FORTRAN Name</td>
<td>Preset</td>
<td>Units</td>
<td>Remarks</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>--------</td>
<td>---------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>460</td>
<td>W0</td>
<td>--</td>
<td>sec⁻¹</td>
<td>$\omega_0$</td>
</tr>
<tr>
<td>461</td>
<td>W1</td>
<td>--</td>
<td>sec⁻¹</td>
<td>$\omega_1$</td>
</tr>
<tr>
<td>462</td>
<td>W2</td>
<td>--</td>
<td>sec⁻¹</td>
<td>$\omega_2$</td>
</tr>
<tr>
<td>463</td>
<td>WL2</td>
<td>--</td>
<td>sec⁻²</td>
<td>$2(\omega_0^2 + \Delta \lambda^2)$</td>
</tr>
<tr>
<td>464</td>
<td>XLAM0</td>
<td>--</td>
<td>sec⁻¹</td>
<td>$\lambda_0$</td>
</tr>
<tr>
<td>465</td>
<td>XLAM1</td>
<td>--</td>
<td>sec⁻¹</td>
<td>$\lambda_1$</td>
</tr>
<tr>
<td>466</td>
<td>XLAM2</td>
<td>--</td>
<td>sec⁻¹</td>
<td>$\lambda_2$</td>
</tr>
<tr>
<td>467</td>
<td>XNU1</td>
<td>--</td>
<td>rad</td>
<td>$\nu_1$</td>
</tr>
<tr>
<td>468</td>
<td>XNU2</td>
<td>--</td>
<td>rad</td>
<td>$\nu_2$</td>
</tr>
<tr>
<td>469</td>
<td>XG</td>
<td>--</td>
<td>ft</td>
<td>$x_G$</td>
</tr>
<tr>
<td>470</td>
<td>XMU</td>
<td>--</td>
<td>--</td>
<td>$\mu$</td>
</tr>
<tr>
<td>471</td>
<td>XJAY</td>
<td>--</td>
<td>rad</td>
<td>$J_{A_Y}$</td>
</tr>
<tr>
<td>472</td>
<td>XJAZ</td>
<td>--</td>
<td>rad</td>
<td>$J_{A_Z}$</td>
</tr>
<tr>
<td>473</td>
<td>XJA</td>
<td>--</td>
<td>rad</td>
<td>$J_A$</td>
</tr>
<tr>
<td>474</td>
<td>YA</td>
<td>--</td>
<td>ft</td>
<td>$\psi_a$</td>
</tr>
<tr>
<td>475</td>
<td>ZA</td>
<td>--</td>
<td>ft</td>
<td>$Z_a$</td>
</tr>
<tr>
<td>476</td>
<td>ALPH0</td>
<td>--</td>
<td>rad</td>
<td>$\alpha_0$</td>
</tr>
<tr>
<td>Code Number</td>
<td>FORTRAN Name</td>
<td>Preset</td>
<td>Units</td>
<td>Remarks</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>--------</td>
<td>---------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>500</td>
<td>TW</td>
<td>--</td>
<td>sec</td>
<td>F.C. Established Nominal Time at Nominal Range</td>
</tr>
<tr>
<td>501</td>
<td>XTFc</td>
<td>--</td>
<td>ft</td>
<td>F.C. Computed Position at Nominal Time</td>
</tr>
<tr>
<td>502</td>
<td>YTFc</td>
<td>--</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>503</td>
<td>ZTFc</td>
<td>--</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>504</td>
<td>VXTfc</td>
<td>--</td>
<td>ft/sec</td>
<td></td>
</tr>
<tr>
<td>505</td>
<td>VYTFc</td>
<td>--</td>
<td>ft/sec</td>
<td></td>
</tr>
<tr>
<td>506</td>
<td>VZTFc</td>
<td>--</td>
<td>ft/sec</td>
<td></td>
</tr>
<tr>
<td>507</td>
<td>XT</td>
<td>--</td>
<td>ft</td>
<td>Projectile Position at Nominal Time</td>
</tr>
<tr>
<td>508</td>
<td>YT</td>
<td>--</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>509</td>
<td>ZT</td>
<td>--</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>510</td>
<td>VXr</td>
<td>--</td>
<td>ft/sec</td>
<td>Projectile Velocity at Nominal Time</td>
</tr>
<tr>
<td>511</td>
<td>VYr</td>
<td>--</td>
<td>ft/sec</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>VZr</td>
<td>--</td>
<td>ft/sec</td>
<td></td>
</tr>
<tr>
<td>513</td>
<td>YR</td>
<td>$\uparrow$</td>
<td>ft</td>
<td>Projectile Position at Nominal Range</td>
</tr>
<tr>
<td>514</td>
<td>ZR</td>
<td>--</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>515</td>
<td>VXr</td>
<td>--</td>
<td>ft</td>
<td>Projectile Velocity at Nominal Range</td>
</tr>
<tr>
<td>516</td>
<td>VYr</td>
<td>--</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>517</td>
<td>VZr</td>
<td>--</td>
<td>ft</td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>TX</td>
<td>--</td>
<td>sec</td>
<td>Projectile Time to Reach Nominal Range</td>
</tr>
</tbody>
</table>
variables as being correlated. Thus each variable appearing in Card Group 3 must also appear in Card Group 2 with a TYPE specification of 2, 3, or 4. Table 3-4 defines the format of Group 3 cards and provides pertinent details. The comments denoted by stars command action on the part of the user. The IBM-029 key-punch control cards are presented in Figure 3-3. Group 3 cards may appear in any order.

3.4 Error Source Model Encoding

This section illustrates the encoding of the error source model of Table 1-2. Primary attention is focused on Card Group 2. Maximum instructional benefit will be achieved by the user who works the problem independently and refers to the text simply for verification.

The first task is to identify the code numbers (i.e., trajectory variable names), corresponding to line items in the error source model. Table 3-5 is a complete listing arrived at by cross referencing the error source model, Table 1-2, and the Parameter List, Table 3-3. These are Real World trajectory variables. Each of the Rayleigh/Uniform distribution sources are two-dimensional and, as a result, are described by two trajectory variables.

With the code numbers in hand, Card Group 2 may be constructed by reference to Table 3-2. For instance:

- Muzzle velocity (CODE# = 101) is a Gaussian distributed uncertainty (TYPE = 2). The nominal or mean value is

\[
\text{VALUE} = 1.1 \times 10^4 \text{ ft/sec}
\]

The standard deviation is

\[
\text{STDEV} = (1/3\%) (1.1 \times 10^4) = 3.666 \times 10^1 \text{ ft/sec}
\]

and

\[
\text{TOL} = 3 \times \text{STDEV} = 1.1 \times 10^2 \text{ ft/sec}
\]

- Projectile reference area (CODE# = 122) is a uniformly distributed uncertainty (TYPE = 3) with a nominal or mean value of

\[
\text{VALUE} = 3.068 \times 10^{-3} \text{ ft}^2
\]
### Table 3-4  Optional Input Processor Control Cards (Card Group 3)

<table>
<thead>
<tr>
<th>COLUMN(S)</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
</table>
| 3-5       | CODE#1        | Code number for first variable of correlated pair. | • See Table 3-3 for definitions.  
• Group 3 cards may appear in any order.  
• Integer variable  
★ Negative value (e.g. "-1") required to indicate end of Card Group 3.  
• CODE#1 is an address in the ΘE array. |
| 8-10      | CODE#2        | Code number for second variable of correlated pair. Of necessity CODE#2 ≠ CODE#1 | • See Table 3-3 for definitions.  
• Integer variable  
• CODE#2 is an address in the ΘE array. |
| 12-23     | RHØ           | The correlation coefficient between the variables defined by CODE#1 and CODE#2. Of necessity $-1 \leq RHØ \leq 1$ | • Floating point variable.  
• See Appendix A.3.3 for discussion.  
• The covariance matrix, Eq. (A.4-15), formed from the correlation coefficients is positive definite as theoretically required. |
NOTE: All Toggle Switches up

Figure 3-3 IBM-029 Keypunch Control Cards for Card Group 3
Table 3-5 Error Source Model Code Numbers

<table>
<thead>
<tr>
<th>ERROR SOURCE</th>
<th>STATISTICAL DISTRIBUTIONS</th>
<th>CODE NUMBER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• INITIAL CONDITIONS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• VELOCITY VECTOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- MAGNITUDE</td>
<td>GAUSSIAN</td>
<td>101</td>
</tr>
<tr>
<td>- ORIENTATION</td>
<td>RAYLEIGH/UNIFORM*</td>
<td>120 &amp; 121</td>
</tr>
<tr>
<td>• INERTIAL ORIENTATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ATTITUDE</td>
<td>RAYLEIGH/UNIFORM*</td>
<td>116 &amp; 117</td>
</tr>
<tr>
<td>- ATTITUDE RATE</td>
<td>RAYLEIGH/UNIFORM*</td>
<td>118 &amp; 119</td>
</tr>
<tr>
<td>• PHYSICAL CHARACTERISTICS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• WEIGHT</td>
<td>GAUSSIAN</td>
<td>125</td>
</tr>
<tr>
<td>• MOMENTS OF INERTIA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- AXIAL</td>
<td>GAUSSIAN</td>
<td>126</td>
</tr>
<tr>
<td>- PITCH</td>
<td>GAUSSIAN</td>
<td>127</td>
</tr>
<tr>
<td>• PHYSICAL DIMENSIONS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- REFERENCE AREA</td>
<td>UNIFORM</td>
<td>122</td>
</tr>
<tr>
<td>- LENGTH</td>
<td>UNIFORM</td>
<td>123</td>
</tr>
<tr>
<td>- BASE DIAMETER</td>
<td>UNIFORM</td>
<td>124</td>
</tr>
<tr>
<td>• AERODYNAMIC CHARACTERISTICS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• STATIC COEFFICIENTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• DRAG VARIATION EFFECTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VELOCITY**</td>
<td>GAUSSIAN</td>
<td>105 &amp; 107</td>
</tr>
<tr>
<td>ANGLE OF ATTACK</td>
<td>GAUSSIAN</td>
<td>130</td>
</tr>
<tr>
<td>NORMAL FORCE</td>
<td>GAUSSIAN</td>
<td>110</td>
</tr>
<tr>
<td>• DYNAMIC COEFFICIENTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- PITCH DAMPING</td>
<td>GAUSSIAN</td>
<td>112</td>
</tr>
<tr>
<td>- MAGNUS MOMENT</td>
<td>GAUSSIAN</td>
<td>113</td>
</tr>
<tr>
<td>• SPIN RATE</td>
<td>GAUSSIAN</td>
<td>128</td>
</tr>
<tr>
<td>• TRIM ANGLE OF ATTACK</td>
<td>RAYLEIGH/UNIFORM*</td>
<td>114 &amp; 115</td>
</tr>
<tr>
<td>• STATIC MARGIN</td>
<td>GAUSSIAN</td>
<td>111</td>
</tr>
<tr>
<td>• ATMOSPHERIC EFFECTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• CONSTANT WINDS</td>
<td>RAYLEIGH/UNIFORM*</td>
<td>102 &amp; 103</td>
</tr>
<tr>
<td>• DENSITY VARIATIONS</td>
<td>GAUSSIAN</td>
<td>104</td>
</tr>
</tbody>
</table>

*Denotes a Rayleigh distribution of magnitude with a uniform 360° distribution in orientation.

**Drag variation with velocity closely approximated by $C_{x_0} = C_{x_\infty} + \frac{K_D}{V^2}$ which is fit to two data points $(C_{x_1}, V_1) = (0.03585, 16.740)$ and $(C_{x_2}, V_2) = (0.11951, 3906.)$. Uncertainty in $C_{x_1}$ and $C_{x_2}$. 

-57-
The standard deviation is

$$STDEV = \frac{2/3\%}{3.068 \times 10^{-3}} = 2.045 \times 10^{-5} \text{ ft}^2$$

and

$$T\sigma L = \sqrt{3} \times STDEV = 3.543 \times 10^{-5} \text{ ft}^2$$

- Constant winds (Code Numbers 102 and 103) are Rayleigh distributed in magnitude and uniformly distributed in direction. This type of distribution is modeled as two independent Gaussian random variables (TYPE = 2). There is no average wind so

$$VALUE = 0.0$$

As discussed in Appendix A.4.4, the equivalent Gaussian standard deviation is given by

$$STDEV = \frac{J}{1.2533}$$

where $J$ is the mean magnitude given in the mean value column of the error source model, so

$$STDEV = \frac{11}{1.2533} = 8.777 \text{ ft/sec}$$

and

$$T\sigma L = 3 \times STDEV = 2.633 \times 10 \text{ ft/sec}$$

These and the other error source Group 2 cards are shown in Figure 3-4.

The error source model of Table 1-2 presumed Fire Control made no simplifying assumptions in the prediction of the projectile trajectory. Thus, since Fire Control would always make use of the known nominal values, the first Group 2 card in Figure 3-4 sets a system control (CODE# = 201) to initialize the Fire Control trajectory parameters at the nominal values of the Real World trajectory by declaring it a TYPE = 5 equal to 1. This considerably simplifies the input. Even though CODE# = 201 denotes an integer variable, VALUE is input as a floating point number. Nominal range (CODE# = 109) is entered as a TYPE = 5 constant value, as shown in Figure 3-4. The second order drag effect variable (CODE# = 130) appears on two Group 2 cards. The latter prevails. The dependent variables (TYPE = 7), force
<table>
<thead>
<tr>
<th>Card Group 1</th>
<th>Card Group 2</th>
<th>Card Group 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6</td>
<td>1 2 3 4 5 6</td>
<td>1 2 3 4 5 6</td>
</tr>
<tr>
<td>2 3 4 5 6 7</td>
<td>2 3 4 5 6 7</td>
<td>2 3 4 5 6 7</td>
</tr>
<tr>
<td>3 4 5 6 7 8</td>
<td>3 4 5 6 7 8</td>
<td>3 4 5 6 7 8</td>
</tr>
<tr>
<td>4 5 6 7 8 9</td>
<td>4 5 6 7 8 9</td>
<td>4 5 6 7 8 9</td>
</tr>
<tr>
<td>5 6 7 8 9 0</td>
<td>5 6 7 8 9 0</td>
<td>5 6 7 8 9 0</td>
</tr>
</tbody>
</table>

Figure 3-4 Encoded Error Source Model
the statistics of dispersion at nominal time to be calculated and printed. Card Group 2 is closed by the required card with a negative code number.

Card Group 2 is now complete. The basic error source model has been encoded and attention turns to selecting various HITS options. The Group 1 card is constructed by referring to Table 3-1. An Analytical Statistical mode calculation is requested by the Group 1 card in Figure 3-4. Card Group 3 calls for muzzle velocity (CODE#1 = 101) and projectile weight (CODE#2 = 125) variations to be treated as negatively correlated. The Group 3 card was constructed in accord with Table 3-4. Card Group 3 concludes with the mandatory CODE#1 = -1 card.

Exercising HITS with the case input of Figure 3-4 would evaluate the total dispersion at nominal time due to all twenty error sources acting in consort. If it were desired to evaluate the dispersion due to a single error source, the case input would be generated by (1) duplicating the cards shown in Figure 3-4, and (2) discarding the Group 2 cards pertaining to other error sources. Muzzle velocity and projectile weight effects would be treated as a single source, since they are declared correlated. An error budget similar to Table 1-3 could be constructed by repeating this process for each error source. The resulting case inputs could be stacked and submitted simultaneously.
4.0 OUTPUT INTERPRETATION

This chapter presents four example problems which demonstrate the basic modes of operation of the HITS code. The user is encouraged to reproduce these results to verify that the code has been properly installed and to gain familiarity.

4.1 Single Trajectory Mode

This problem demonstrates the evaluation of single-shot dispersion using the HITS code. This mode is of interest to projectile designers and exterior ballisticians.

Figure 4-1 lists the input cards. The first card is the Group 1 card defined by Table 3-1. All other cards belong to Card Group 2 discussed in Table 3-2. Only TYPE = 5 and TYPE = 7 variables may appear in Single Trajectory mode simulations.

The results of the HITS simulation are shown in Figure 4-2. Frame 'a' is the cover sheet which identifies the code and separates the results of stacked cases. Frames 'b' through 'f' verify the inputs. Frame 'b' is the transcription of the Group 1 card. Frame 'c' lists the Card Group 2 data. Frames 'd', 'e', and 'f' are cross reference indices linking variable names and code numbers. The objective is to minimize the need to refer to Table 3-3 for variable definitions. Frame 'd' gives the FORTRAN names and code numbers of the dependent variables (TYPE = 7 and TYPE = 8). Frame 'e' gives the same information for the deterministic constants whose values have been changed from the presets (TYPE = 5 variables). All variables not changed by the input sequence are listed in Frame 'f'.

The results of calculations presented in Figure 4-2g and h. This is the information passed across the Trajectory Module interface. This level of information may be requested at any time by setting ISPRNT = 1 in Card Group 1 (see Table 3-1), although it can result in excessive output. The formats of Frames 'g' and 'h' are identical. At the top are the Fire Control parameters. These are followed by the Real World parameters, System Controls, and Computed Quantities. Table 3-3 gives the definitions of all quantities. Frame 'g' summarizes the Fire Control calculation to establish the coordinates of the nominal aim point and the nominal time-to-nominal range. The Fire Control summary is distinguished from the Real World summary by the value of the IFC System Control (IFC = 1 → Fire Control, IFC = 0 → Real World). Frame 'h' is the Real World trajectory. Values presented under the Computed Quantities heading quantify dispersion.
Figure 4-1 Single Trajectory Mode Check Problem Input

```
1  0.1
206 5.00000000000
105 5.03565397650
107 5.14341590610
111 5.05
120 5.0001
126 5.53810000000-6
127 5.78660000000-5
201 51.0
109 52085.2
116 510.0
318 7
319 7
320 7
321 7
306 7
307 7
308 7
309 7
310 7
-1
```
### INPUT VARIABLE SPECIFICATIONS

<table>
<thead>
<tr>
<th>CODE NUMBER</th>
<th>VARIABLE TYPE</th>
<th>NOMINAL VALUE</th>
<th>TOLERANCE</th>
<th>STANDARD DEVIATION</th>
<th>SUBSEQUENT POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>206</td>
<td>5</td>
<td>0.0</td>
<td>±0.0</td>
<td>±0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>109</td>
<td>5</td>
<td>3.55E+00</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>107</td>
<td>5</td>
<td>1.43242E+01</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>111</td>
<td>4</td>
<td>9.00000E+02</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>120</td>
<td>5</td>
<td>-1.00000E+04</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>126</td>
<td>5</td>
<td>5.31E+00</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>127</td>
<td>5</td>
<td>7.86E+00</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>221</td>
<td>5</td>
<td>1.00000E+00</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>199</td>
<td>5</td>
<td>2.00E+00</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>110</td>
<td>5</td>
<td>1.00000E+01</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>319</td>
<td>7</td>
<td>0.0</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>320</td>
<td>7</td>
<td>0.0</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>321</td>
<td>7</td>
<td>0.0</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>324</td>
<td>7</td>
<td>0.0</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>327</td>
<td>7</td>
<td>0.0</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>329</td>
<td>7</td>
<td>0.0</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>339</td>
<td>7</td>
<td>0.0</td>
<td>±6.0</td>
<td>±6.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Figure 4-2c Single Trajectory Mode Check Problem Output
### VariablesReset by Input Sequence

<table>
<thead>
<tr>
<th>Variable</th>
<th>Code Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 REAL WORLD</td>
<td>CODE NUMBER 105</td>
</tr>
<tr>
<td>C2 REAL WORLD</td>
<td>CODE NUMBER 107</td>
</tr>
<tr>
<td>X1 REAL WORLD</td>
<td>CODE NUMBER 109</td>
</tr>
<tr>
<td>X2 REAL WORLD</td>
<td>CODE NUMBER 111</td>
</tr>
<tr>
<td>THF REAL WORLD</td>
<td>CODE NUMBER 113</td>
</tr>
<tr>
<td>GAMO REAL WORLD</td>
<td>CODE NUMBER 116</td>
</tr>
<tr>
<td>AIX REAL WORLD</td>
<td>CODE NUMBER 120</td>
</tr>
<tr>
<td>AVY REAL WORLD</td>
<td>CODE NUMBER 121</td>
</tr>
<tr>
<td>IFCRM SYSTEM CONTROL</td>
<td>CODE NUMBER 201</td>
</tr>
<tr>
<td>TCMCL SYSTEM CONTROL</td>
<td>CODE NUMBER 206</td>
</tr>
</tbody>
</table>

---

**Figure 4-2e**  Single Trajectory Mode Check Problem Output
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>W0 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>U7 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>D7 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>C1 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>V1 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>GY FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>V4 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CNA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CN2 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CNIA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CLNS FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>REDS FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>TNR1 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>TR10 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>POSTO FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>GD3 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>LA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CALL FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>D1 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>D2 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>AX FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATV FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ALPON FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CAA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>C4 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>C5 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>AX FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>YI FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>V7 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>V10 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CNA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CNA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CNA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH1 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH2 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH3 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH4 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH5 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH6 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH7 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH8 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH9 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH10 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH11 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH12 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH13 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH14 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH15 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH16 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH17 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH18 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH19 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH20 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH21 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH22 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH23 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH24 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH25 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ALPON FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CAA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>C4 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>C5 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>AX FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>YI FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>V7 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>V10 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CNA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CNA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>CNA FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH1 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH2 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH3 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH4 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH5 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH6 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH7 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH8 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH9 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH10 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH11 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH12 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH13 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH14 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH15 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH16 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH17 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH18 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH19 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH20 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH21 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH22 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH23 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH24 FINE CONTROL</td>
<td>0</td>
</tr>
<tr>
<td>ATH25 FINE CONTROL</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4-2f Single Trajectory Mode Check Problem Output
4.2 Range Check Mode

This problem demonstrates the evaluation of dispersion for a systematic permutation of projectile characteristics. This mode could be used to define a trajectory as a function of range. The Range Check mode is of interest to projectile designers and exterior ballisticians.

Figure 4-3 illustrates the input deck for a Range Check mode simulation. The first card is the Group 1 card defined by Table 3-1. All other cards belong to Card Group 2 discussed in Table 3-2. Only TYPE = 1, 5, and 7 variables may appear in a Range Check Card Group 2. There must be at least one TYPE = 1 variable followed by the Range Check values. There must be at least one TYPE = 7 card.

The printed output generated by HITS is illustrated in Figure 4-4. Frame 'a' is the cover sheet. Frames 'b' through 'i' document the input. Frames 'b' and 'c' were discussed in Section 4.1. Frame 'd' lists the Range Check (TYPE = 1) variables. Frames 'e' through 'h' were discussed in Section 4.1. As indicated in Frame 'h', the System Control IFCRC is left at its preset value of zero. Thus, the Fire Control parameters remain at their default values for all Range Check combinations and do not vary with each combination. If a trajectory were being generated as a function of range it would be necessary to set IFCRC = 1 (see Table 3-3 for details). Frame 'i' lists the Range Check combinations. Since there are two TYPE = 1 variables with three values each, HITS will evaluate \(3 \times 3 = 9\) trajectories.

Frames 'j' and 'k' summarize the results of computation. The summary presents the nine combinations in a sequence of blocks with the variable appearing last in Card Group 2 varying most rapidly. The TYPE = 1 independent variables (IND. VAR.) values appear first, immediately followed by the TYPE = 7 dependent variable values. The code numbers are linked to FORTRAN names by the lists presented in Frames 'd' and 'e'.

-71-
Figure 4-3 Range Check Mode Check Problem Input
Figure 4-4a. Range Check Mode Check Problem Output
### SIMULATION INPUT SUMMARY ###

**SINGLE-CASE, RANGE-CHECK, AND STATISTICAL PROCESSOR CONTROLS**

<table>
<thead>
<tr>
<th>IF</th>
<th>IOPUNT</th>
<th>ISPANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**MONTE CARLO CONTROLS**

<table>
<thead>
<tr>
<th>WCOPT</th>
<th>MCALC</th>
<th>MCCF</th>
<th>NTRIAL</th>
<th>TRNNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IRI</th>
<th>IRIJ2</th>
<th>MCRTN</th>
<th>TRNNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IOPLOT</th>
<th>ISPLOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Figure 4-4b** Range Check Mode Check Problem Output
### INPUT VARIABLE SPECIFICATIONS

<table>
<thead>
<tr>
<th>CODE NUMBER</th>
<th>VARIABLE TYPE</th>
<th>NOMINAL VALUE</th>
<th>TOLERANCE</th>
<th>STANDARD DEVIATION</th>
<th>SUBSEQUENT POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>206</td>
<td>5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>123</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>123</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>105</td>
<td>5</td>
<td>3.58000-02</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>107</td>
<td>5</td>
<td>1.42440-01</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>5</td>
<td>5.00000-02</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>318</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>319</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>320</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>321</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>306</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>307</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>328</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>319</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>310</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 4-4c** Range Check Mode Check Output
### DEPENDENT VARIABLES

<table>
<thead>
<tr>
<th>Variable</th>
<th>Code Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: T Statistical Variable</td>
<td>319</td>
</tr>
<tr>
<td>RAX Statistical Variable</td>
<td>319</td>
</tr>
<tr>
<td>RH0Y Statistical Variable</td>
<td>320</td>
</tr>
<tr>
<td>RH0X Statistical Variable</td>
<td>321</td>
</tr>
<tr>
<td>DAX Statistical Variable</td>
<td>306</td>
</tr>
<tr>
<td>DYT Statistical Variable</td>
<td>307</td>
</tr>
<tr>
<td>D2Y Statistical Variable</td>
<td>308</td>
</tr>
<tr>
<td>D2X Statistical Variable</td>
<td>309</td>
</tr>
<tr>
<td>D2X Statistical Variable</td>
<td>310</td>
</tr>
</tbody>
</table>

---

**Figure 4-4e** Range Check Mode Check Problem Output
### Variables Reset by Input Sequence

<table>
<thead>
<tr>
<th>Variable</th>
<th>Code Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>CY1 REAL WORLD</td>
<td>105</td>
</tr>
<tr>
<td>CY2 REAL WORLD</td>
<td>107</td>
</tr>
<tr>
<td>SM REAL WORLD</td>
<td>111</td>
</tr>
<tr>
<td>ICNGL SYSTEM CONTROL</td>
<td>20A</td>
</tr>
</tbody>
</table>

---

**Figure 4-4f** Range Check Mode Check Problem Output
### VARIABLES ASSIGNED PRESET VALUES ###

<table>
<thead>
<tr>
<th>Variable</th>
<th>Code Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>WZ FIRE CONTROL</td>
<td>CODE NUMBER 3</td>
</tr>
<tr>
<td>C H O FIRE CONTROL</td>
<td>CODE NUMBER 4</td>
</tr>
<tr>
<td>C A 1 FIRE CONTROL</td>
<td>CODE NUMBER 5</td>
</tr>
<tr>
<td>V I FIRE CONTROL</td>
<td>CODE NUMBER 6</td>
</tr>
<tr>
<td>C A 2 FIRE CONTROL</td>
<td>CODE NUMBER 7</td>
</tr>
<tr>
<td>V 2 FIRE CONTROL</td>
<td>CODE NUMBER 8</td>
</tr>
<tr>
<td>X N FIRE CONTROL</td>
<td>CODE NUMBER 9</td>
</tr>
<tr>
<td>C N A FIRE CONTROL</td>
<td>CODE NUMBER 10</td>
</tr>
<tr>
<td>C M FIRE CONTROL</td>
<td>CODE NUMBER 11</td>
</tr>
<tr>
<td>C H O FIRE CONTROL</td>
<td>CODE NUMBER 12</td>
</tr>
<tr>
<td>C M A FIRE CONTROL</td>
<td>CODE NUMBER 13</td>
</tr>
<tr>
<td>A T H N S FIRE CONTROL</td>
<td>CODE NUMBER 14</td>
</tr>
<tr>
<td>R T FIRE CONTROL</td>
<td>CODE NUMBER 15</td>
</tr>
<tr>
<td>T H E O FIRE CONTROL</td>
<td>CODE NUMBER 16</td>
</tr>
<tr>
<td>R S T D FIRE CONTROL</td>
<td>CODE NUMBER 17</td>
</tr>
<tr>
<td>P D T O FIRE CONTROL</td>
<td>CODE NUMBER 18</td>
</tr>
<tr>
<td>G A N D FIRE CONTROL</td>
<td>CODE NUMBER 19</td>
</tr>
<tr>
<td>A T O FIRE CONTROL</td>
<td>CODE NUMBER 20</td>
</tr>
<tr>
<td>E L L FIRE CONTROL</td>
<td>CODE NUMBER 21</td>
</tr>
<tr>
<td>E FIRE CONTROL</td>
<td>CODE NUMBER 22</td>
</tr>
<tr>
<td>W FIRE CONTROL</td>
<td>CODE NUMBER 23</td>
</tr>
<tr>
<td>A X FIRE CONTROL</td>
<td>CODE NUMBER 24</td>
</tr>
<tr>
<td>A Y FIRE CONTROL</td>
<td>CODE NUMBER 25</td>
</tr>
<tr>
<td>A L P C O N FIRE CONTROL</td>
<td>CODE NUMBER 26</td>
</tr>
<tr>
<td>C A A FIRE CONTROL</td>
<td>CODE NUMBER 27</td>
</tr>
<tr>
<td>V O FIRE CONTROL</td>
<td>CODE NUMBER 28</td>
</tr>
<tr>
<td>W X REAL WORLD</td>
<td>CODE NUMBER 29</td>
</tr>
<tr>
<td>WZ REAL WORLD</td>
<td>CODE NUMBER 30</td>
</tr>
<tr>
<td>V I REAL WORLD</td>
<td>CODE NUMBER 31</td>
</tr>
<tr>
<td>C H O REAL WORLD</td>
<td>CODE NUMBER 32</td>
</tr>
<tr>
<td>C M A REAL WORLD</td>
<td>CODE NUMBER 33</td>
</tr>
<tr>
<td>A T H N S REAL WORLD</td>
<td>CODE NUMBER 34</td>
</tr>
<tr>
<td>B T R N S REAL WORLD</td>
<td>CODE NUMBER 35</td>
</tr>
<tr>
<td>T H E O REAL WORLD</td>
<td>CODE NUMBER 36</td>
</tr>
<tr>
<td>R S T D REAL WORLD</td>
<td>CODE NUMBER 37</td>
</tr>
<tr>
<td>P D T O REAL WORLD</td>
<td>CODE NUMBER 38</td>
</tr>
<tr>
<td>G A N D REAL WORLD</td>
<td>CODE NUMBER 39</td>
</tr>
<tr>
<td>A T O REAL WORLD</td>
<td>CODE NUMBER 40</td>
</tr>
<tr>
<td>E L L REAL WORLD</td>
<td>CODE NUMBER 41</td>
</tr>
<tr>
<td>N A X REAL WORLD</td>
<td>CODE NUMBER 42</td>
</tr>
<tr>
<td>W FIRE CONTROL</td>
<td>CODE NUMBER 43</td>
</tr>
<tr>
<td>V FIRE CONTROL</td>
<td>CODE NUMBER 44</td>
</tr>
<tr>
<td>A X FIRE CONTROL</td>
<td>CODE NUMBER 45</td>
</tr>
<tr>
<td>A Y FIRE CONTROL</td>
<td>CODE NUMBER 46</td>
</tr>
<tr>
<td>A L P C O N REAL WORLD</td>
<td>CODE NUMBER 47</td>
</tr>
<tr>
<td>C A A REAL WORLD</td>
<td>CODE NUMBER 48</td>
</tr>
<tr>
<td>V O REAL WORLD</td>
<td>CODE NUMBER 49</td>
</tr>
<tr>
<td>W X REAL WORLD</td>
<td>CODE NUMBER 50</td>
</tr>
<tr>
<td>WZ REAL WORLD</td>
<td>CODE NUMBER 51</td>
</tr>
<tr>
<td>V I REAL WORLD</td>
<td>CODE NUMBER 52</td>
</tr>
<tr>
<td>C H O REAL WORLD</td>
<td>CODE NUMBER 53</td>
</tr>
<tr>
<td>C M A REAL WORLD</td>
<td>CODE NUMBER 54</td>
</tr>
<tr>
<td>A T H N S REAL WORLD</td>
<td>CODE NUMBER 55</td>
</tr>
<tr>
<td>B T R N S REAL WORLD</td>
<td>CODE NUMBER 56</td>
</tr>
<tr>
<td>T H E O REAL WORLD</td>
<td>CODE NUMBER 57</td>
</tr>
<tr>
<td>R S T D REAL WORLD</td>
<td>CODE NUMBER 58</td>
</tr>
<tr>
<td>P D T O REAL WORLD</td>
<td>CODE NUMBER 59</td>
</tr>
<tr>
<td>G A N D REAL WORLD</td>
<td>CODE NUMBER 60</td>
</tr>
<tr>
<td>A T O REAL WORLD</td>
<td>CODE NUMBER 61</td>
</tr>
<tr>
<td>E L L REAL WORLD</td>
<td>CODE NUMBER 62</td>
</tr>
<tr>
<td>N A X REAL WORLD</td>
<td>CODE NUMBER 63</td>
</tr>
<tr>
<td>W FIRE CONTROL</td>
<td>CODE NUMBER 64</td>
</tr>
<tr>
<td>V FIRE CONTROL</td>
<td>CODE NUMBER 65</td>
</tr>
<tr>
<td>A X FIRE CONTROL</td>
<td>CODE NUMBER 66</td>
</tr>
<tr>
<td>A Y FIRE CONTROL</td>
<td>CODE NUMBER 67</td>
</tr>
<tr>
<td>A L P C O N REAL WORLD</td>
<td>CODE NUMBER 68</td>
</tr>
<tr>
<td>C A A REAL WORLD</td>
<td>CODE NUMBER 69</td>
</tr>
<tr>
<td>V O REAL WORLD</td>
<td>CODE NUMBER 70</td>
</tr>
<tr>
<td>W X REAL WORLD</td>
<td>CODE NUMBER 71</td>
</tr>
<tr>
<td>WZ REAL WORLD</td>
<td>CODE NUMBER 72</td>
</tr>
<tr>
<td>V I REAL WORLD</td>
<td>CODE NUMBER 73</td>
</tr>
<tr>
<td>C H O REAL WORLD</td>
<td>CODE NUMBER 74</td>
</tr>
<tr>
<td>C M A REAL WORLD</td>
<td>CODE NUMBER 75</td>
</tr>
<tr>
<td>A T H N S REAL WORLD</td>
<td>CODE NUMBER 76</td>
</tr>
<tr>
<td>B T R N S REAL WORLD</td>
<td>CODE NUMBER 77</td>
</tr>
<tr>
<td>T H E O REAL WORLD</td>
<td>CODE NUMBER 78</td>
</tr>
<tr>
<td>R S T D REAL WORLD</td>
<td>CODE NUMBER 79</td>
</tr>
<tr>
<td>P D T O REAL WORLD</td>
<td>CODE NUMBER 80</td>
</tr>
<tr>
<td>G A N D REAL WORLD</td>
<td>CODE NUMBER 81</td>
</tr>
<tr>
<td>A T O REAL WORLD</td>
<td>CODE NUMBER 82</td>
</tr>
<tr>
<td>E L L REAL WORLD</td>
<td>CODE NUMBER 83</td>
</tr>
<tr>
<td>N A X REAL WORLD</td>
<td>CODE NUMBER 84</td>
</tr>
<tr>
<td>W FIRE CONTROL</td>
<td>CODE NUMBER 85</td>
</tr>
<tr>
<td>V FIRE CONTROL</td>
<td>CODE NUMBER 86</td>
</tr>
<tr>
<td>A X FIRE CONTROL</td>
<td>CODE NUMBER 87</td>
</tr>
<tr>
<td>A Y FIRE CONTROL</td>
<td>CODE NUMBER 88</td>
</tr>
<tr>
<td>A L P C O N REAL WORLD</td>
<td>CODE NUMBER 89</td>
</tr>
<tr>
<td>C A A REAL WORLD</td>
<td>CODE NUMBER 90</td>
</tr>
</tbody>
</table>

Figure 4-4g Range Check Mode Check Problem Output
<table>
<thead>
<tr>
<th>CODE NUMBER</th>
<th>VALUE 1</th>
<th>VALUE 2</th>
<th>VALUE 3</th>
<th>VALUE 4</th>
<th>VALUE 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>0.00000-02</td>
<td>1.00000-01</td>
<td>1.50000-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>2.50000-01</td>
<td>3.10000-01</td>
<td>3.50000-01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4-4i  Range Check Mode Check Problem Output
Figure 4-4k  Range Check Mode Check Problem Output
4.3 Analytical Statistical Mode

This problem demonstrates the analytical evaluation of dispersion statistics. This mode is of interest to projectile designers and those concerned with the implications of projectile dispersion to overall weapon system effectiveness.

Figure 4-5 illustrates the input deck for an Analytical Statistical mode simulation. The first card is the Group 1 card defined by Table 3-1. All other cards belong to Group 2 discussed in Table 3-2. Only TYPE = 2, 3, 4, 5, and 7 cards may appear in an Analytical Statistical mode simulation. There must be at least one TYPE = 2, 3, or 4 and one TYPE = 7 variable. Figure 4-5 illustrates the TYPE = 4 variable input format. The additional data defining the probability density function must immediately follow the TYPE = 4 Group 2 card.

Figure 4-6 illustrates the Analytical Statistical mode printed output. Frames 'a' through 'h' have been described previously. Frames 'i' and 'j' list the TYPE = 4 probability density functions. Frames 'k' and 'l' give the Fire Control (IFC = 1) and Real World (IFC = 0) solutions for the nominal conditions. Frame 'm' is a self-explanatory warning which appears only when code numbers 320 and 321 are declared TYPE = 7.

Frames 'n' and 'o' of Figure 4-6 present the nominal case independent and dependent variable values, respectively. Whenever the warning doesn't appear, Frame 'o' is followed by the LIMITING CASES output illustrated in Figure 4-6, Frame 'oo', presented at the conclusion of Figure 4-6. As shown in Frame 'oo', each independent variable is incrementally varied by the TOLERANCE amount about the nominal given by VALUE. TOLERANCE is added to get the first set of dependent variable values and subtracted to get the second. Return to the example problem at Frame 'p'. These are the values of the partial derivatives of the dependent variables with respect to the independent. Independent variable code numbers are listed on the left. Dependent variable code numbers are listed above. 1ST and 2ND indicate the first and second unmixed partial derivatives. The second order mixed partial derivatives are printed next as illustrated in Frame 'q'. These are the second order partial derivatives of the dependent variables listed above with respect to the independent variables listed to the left. The partial derivatives define the sensitivity coefficients customarily listed in an error budget.
Frame 'r' of Figure 4-6 presents the dispersion statistics. This is the culmination of the Analytical Statistical mode. The heading summarizes the level of the calculation as determined by the IPRNT variable in Card Group 1. The mean values, variances, and standard deviations of the dependent variables are printed. Standard deviations are printed a second time with greater numerical precision.

Had any of the independent variables been declared correlated by Card Group 3, Frame 'r' would have been printed assuming no correlation. The following two frames would document the effect of the correlation specified by Card Group 3. The dispersion would be presented in the format of Frame 'r'. Printing the dispersion statistics with and without correlation make it possible to determine the size of the effect from a single simulation.
```
Figure 4-5 Analytical Statistical Mode Check Problem Input

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>5</td>
<td>0.00000000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>2</td>
<td>0.00000000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>2</td>
<td>0.00000000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.004430000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.017520000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.053980000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.0000000-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.352030000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.398900000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.352030000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.241900000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.129500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.241900000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.129500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.035800000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.143400000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.050000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

-86-
Figure 4-6a Analytical Statistical Mode Check Problem Output
### ***SIMULATION INPUT SUMMARY***

<table>
<thead>
<tr>
<th>SINGLE-CASE, RANGE-CHECK, AND STATISTICAL PROCESSOR CONTROLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV = 1  IPRINT = 6  ISPRINT = 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MONTE-CARLO CONTROLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCOPY = 0  NCALC = 0  NCELL = 0  NTRIAL = 0</td>
</tr>
<tr>
<td>NRJ = 0  INJ = 0  McPRINT = 0  TRANO = 0</td>
</tr>
<tr>
<td>IOPLOT = 0  I_SPLIT = 0</td>
</tr>
</tbody>
</table>

Figure 4-6b  Analytical Statistical Mode Check Problem Output
### STOCHASTIC INDEPENDENT VARIABLES

<table>
<thead>
<tr>
<th>Variable</th>
<th>Code Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theta Real World</td>
<td>116</td>
</tr>
<tr>
<td>Psi0 Real World</td>
<td>117</td>
</tr>
<tr>
<td>ELL Real World</td>
<td>123</td>
</tr>
<tr>
<td>W Real World</td>
<td>125</td>
</tr>
</tbody>
</table>

---

**Figure 4-6d Analytical Statistical Mode Check Problem Output**
<table>
<thead>
<tr>
<th>Variable Description</th>
<th>Code Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHOY Statistical Variable</td>
<td>329</td>
</tr>
<tr>
<td>RH0X Statistical Variable</td>
<td>321</td>
</tr>
<tr>
<td>OXY Statistical Variable</td>
<td>306</td>
</tr>
<tr>
<td>OX0 Statistical Variable</td>
<td>307</td>
</tr>
<tr>
<td>O2T Statistical Variable</td>
<td>309</td>
</tr>
<tr>
<td>O2Y Statistical Variable</td>
<td>309</td>
</tr>
<tr>
<td>O2X Statistical Variable</td>
<td>310</td>
</tr>
</tbody>
</table>

Figure 4-6e Analytical Statistical Mode Check Problem Output
### CODE NUMBER 123 PROBABILITY DENSITY FUNCTION

<table>
<thead>
<tr>
<th>POINT NUMBER</th>
<th>COORDINATE VALUE</th>
<th>PROBABILITY DENSITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.30000-01</td>
<td>4.43000-03</td>
</tr>
<tr>
<td>2</td>
<td>3.45000-01</td>
<td>1.75200-02</td>
</tr>
<tr>
<td>3</td>
<td>3.40000-01</td>
<td>1.26500-02</td>
</tr>
<tr>
<td>4</td>
<td>3.45000-01</td>
<td>1.26500-02</td>
</tr>
<tr>
<td>5</td>
<td>3.50000-01</td>
<td>3.58000-02</td>
</tr>
<tr>
<td>6</td>
<td>3.60000-01</td>
<td>3.58000-02</td>
</tr>
<tr>
<td>7</td>
<td>3.70000-01</td>
<td>3.58000-02</td>
</tr>
<tr>
<td>8</td>
<td>3.80000-01</td>
<td>1.75200-02</td>
</tr>
<tr>
<td>9</td>
<td>3.90000-01</td>
<td>1.75200-02</td>
</tr>
<tr>
<td>10</td>
<td>3.90000-01</td>
<td>4.43000-03</td>
</tr>
<tr>
<td>11</td>
<td>3.90000-01</td>
<td>4.43000-03</td>
</tr>
<tr>
<td>12</td>
<td>3.90000-01</td>
<td>4.43000-03</td>
</tr>
<tr>
<td>13</td>
<td>3.90000-01</td>
<td>4.43000-03</td>
</tr>
</tbody>
</table>

COPY AVAILABLE TO DOC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION

Figure 4-61 Analytical Statistical Mode Check Problem Output
<table>
<thead>
<tr>
<th>POINT NUMBER</th>
<th>COORDINATE</th>
<th>PROBABILITY DENSITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.52000E-01</td>
<td>4.43000E-03</td>
</tr>
<tr>
<td>2</td>
<td>1.54000E-01</td>
<td>4.75200E-03</td>
</tr>
<tr>
<td>3</td>
<td>1.54000E-01</td>
<td>4.69600E-03</td>
</tr>
<tr>
<td>4</td>
<td>1.54000E-01</td>
<td>3.41600E-03</td>
</tr>
<tr>
<td>5</td>
<td>1.54000E-01</td>
<td>5.20300E-03</td>
</tr>
<tr>
<td>6</td>
<td>1.54000E-01</td>
<td>3.49900E-03</td>
</tr>
<tr>
<td>7</td>
<td>1.54000E-01</td>
<td>1.62000E-03</td>
</tr>
<tr>
<td>8</td>
<td>1.54000E-01</td>
<td>5.52000E-03</td>
</tr>
<tr>
<td>9</td>
<td>1.54000E-01</td>
<td>6.19000E-03</td>
</tr>
<tr>
<td>10</td>
<td>1.54000E-01</td>
<td>1.36000E-03</td>
</tr>
<tr>
<td>11</td>
<td>2.06000E-01</td>
<td>5.24000E-03</td>
</tr>
<tr>
<td>12</td>
<td>2.06000E-01</td>
<td>1.75200E-03</td>
</tr>
<tr>
<td>13</td>
<td>2.06000E-01</td>
<td>4.63000E-03</td>
</tr>
</tbody>
</table>

Figure 4-6j  Analytical Statistical Mode Check Problem Output
### Projectile Dispersion Case Summary

#### Fire Control Parameters

<table>
<thead>
<tr>
<th>V2</th>
<th>FT/SEC</th>
<th>XW</th>
<th>0.0</th>
<th>0.0</th>
<th>FT/SEC</th>
<th>PMG</th>
<th>2.6780-03 SL/FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK1</td>
<td>3.5650-02 (NONE)</td>
<td>V1</td>
<td>1.5740-04</td>
<td>FT/SEC</td>
<td>C22</td>
<td>1.9150-01 (NONE)</td>
<td>V2</td>
</tr>
<tr>
<td>CK2</td>
<td>1.2500-04</td>
<td>FT/SEC</td>
<td>0.0</td>
<td>CADL</td>
<td>6.2000-02 SEC/L</td>
<td>CHD</td>
<td>7.5500-03 (NONE)</td>
</tr>
<tr>
<td>CPRA</td>
<td>0.0</td>
<td>(NONE)</td>
<td>APRY</td>
<td>0.0</td>
<td>SEC</td>
<td>MTRPS</td>
<td>0.0</td>
</tr>
<tr>
<td>CPRA</td>
<td>0.0</td>
<td>(NONE)</td>
<td>TDCOT</td>
<td>0.0</td>
<td>SEC</td>
<td>PDPID</td>
<td>0.0</td>
</tr>
<tr>
<td>CPRA</td>
<td>0.0</td>
<td>(NONE)</td>
<td>A</td>
<td>3.1000-03 SO-FT</td>
<td>ELL</td>
<td>3.1000-03 FT</td>
<td>D</td>
</tr>
<tr>
<td>CPRA</td>
<td>0.0</td>
<td>(NONE)</td>
<td>A</td>
<td>1.0000-01 LPS</td>
<td>AIT</td>
<td>2.1100-05 SL-FT</td>
<td>P</td>
</tr>
<tr>
<td>CPRA</td>
<td>5.0000-01</td>
<td>DPC</td>
<td>0.0</td>
<td>LPSDE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Real World Parameters

<table>
<thead>
<tr>
<th>V1</th>
<th>1.5740-04</th>
<th>FT/SEC</th>
<th>0.0</th>
<th>FT/SEC</th>
<th>RAD</th>
<th>2.6780-03 SL/FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK1</td>
<td>1.5740-04</td>
<td>FT/SEC</td>
<td>0.0</td>
<td>FT/SEC</td>
<td>V1</td>
<td>3.9650-03 FT/SEC</td>
</tr>
<tr>
<td>CK2</td>
<td>1.2500-04</td>
<td>FT/SEC</td>
<td>0.0</td>
<td>FT/SEC</td>
<td>V2</td>
<td>7.5500-03 (NONE)</td>
</tr>
<tr>
<td>CPRA</td>
<td>0.0</td>
<td>(NONE)</td>
<td>APRY</td>
<td>0.0</td>
<td>SEC</td>
<td>MTRPS</td>
</tr>
<tr>
<td>CPRA</td>
<td>0.0</td>
<td>(NONE)</td>
<td>TDCOT</td>
<td>0.0</td>
<td>SEC</td>
<td>PDPID</td>
</tr>
<tr>
<td>CPRA</td>
<td>0.0</td>
<td>(NONE)</td>
<td>A</td>
<td>3.1000-03 SO-FT</td>
<td>ELL</td>
<td>3.1000-03 FT</td>
</tr>
<tr>
<td>CPRA</td>
<td>0.0</td>
<td>(NONE)</td>
<td>A</td>
<td>1.0000-01 LPS</td>
<td>AIT</td>
<td>2.1100-05 SL-FT</td>
</tr>
<tr>
<td>CPRA</td>
<td>5.0000-01</td>
<td>DPC</td>
<td>0.0</td>
<td>LPSDE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### IFCON

<table>
<thead>
<tr>
<th>YPAR1</th>
<th>0.0</th>
<th>FT</th>
<th>YPAR1</th>
<th>0.0</th>
<th>FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFCON</td>
<td>-10</td>
<td>(NONE)</td>
<td>IFSC</td>
<td>0.0</td>
<td>(NONE)</td>
</tr>
<tr>
<td>IFC</td>
<td>1</td>
<td>(NONE)</td>
<td>IFC</td>
<td>0</td>
<td>(NONE)</td>
</tr>
</tbody>
</table>

#### Systems Controls

<table>
<thead>
<tr>
<th>XPAR1</th>
<th>0.0</th>
<th>FT</th>
<th>YPAR1</th>
<th>0.0</th>
<th>FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFC</td>
<td>1</td>
<td>(NONE)</td>
<td>IFC</td>
<td>0</td>
<td>(NONE)</td>
</tr>
</tbody>
</table>

#### Completed Quantities

<table>
<thead>
<tr>
<th>XPAR1</th>
<th>0.0</th>
<th>FT</th>
<th>YPAR1</th>
<th>0.0</th>
<th>FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFC</td>
<td>1</td>
<td>(NONE)</td>
<td>IFC</td>
<td>0</td>
<td>(NONE)</td>
</tr>
</tbody>
</table>

---

**Figure 4-6k Analytical Statistical Mode Check Problem Output**

COPY AVAILABLE TO DOC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION
COPY AVAILABLE TO DDC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION

Figure 4-6m Analytical Statistical Mode Check Problem Output
### Derivatives

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
<th>7th</th>
<th>8th</th>
<th>9th</th>
</tr>
</thead>
<tbody>
<tr>
<td>115</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.04440-01</td>
<td>-0.47570-02</td>
<td>9.62500-02</td>
<td>7.77170-02</td>
<td>0.00000-17</td>
<td>0.0</td>
</tr>
<tr>
<td>117</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.47570-02</td>
<td>1.04440-01</td>
<td>7.77170-02</td>
<td>9.62500-02</td>
<td>0.00000-17</td>
<td>0.0</td>
</tr>
<tr>
<td>125</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.02990-04</td>
<td>0.00000-05</td>
<td>0.00000-05</td>
<td>0.00000-05</td>
<td>0.00000-05</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Figure 4-6p** Analytical Statistical Mode Check Problem Output
### 2nd Order Means and 1st Order Variances

<table>
<thead>
<tr>
<th></th>
<th>329</th>
<th>332</th>
<th>306</th>
<th>307</th>
<th>308</th>
<th>309</th>
<th>310</th>
</tr>
</thead>
<tbody>
<tr>
<td>Means</td>
<td>5.459E01</td>
<td>1.999E00</td>
<td>9.042E00</td>
<td>2.159E00</td>
<td>1.799E00</td>
<td>2.081E00</td>
<td>1.664E00</td>
</tr>
<tr>
<td>Variances</td>
<td>4.397E02</td>
<td>8.542E00</td>
<td>1.347E00</td>
<td>1.813E00</td>
<td>1.913E00</td>
<td>1.539E00</td>
<td>1.530E00</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>2.369E00</td>
<td>1.194E00</td>
<td>1.023E00</td>
<td>1.346E00</td>
<td>1.346E00</td>
<td>1.237E00</td>
<td>1.237E00</td>
</tr>
</tbody>
</table>

### Standard Deviations with Higher Precision

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.333E01</td>
<td>2.303E01</td>
<td>2.303E01</td>
<td>2.303E01</td>
<td>2.303E01</td>
<td>2.303E01</td>
<td>2.303E01</td>
<td></td>
</tr>
<tr>
<td>6.124E02</td>
<td>8.104E02</td>
<td>1.023E02</td>
<td>1.345E02</td>
<td>1.345E02</td>
<td>1.237E02</td>
<td>1.237E02</td>
<td></td>
</tr>
<tr>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td></td>
</tr>
<tr>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td>1.237E01</td>
<td></td>
</tr>
</tbody>
</table>

---

Figure 4-6r Analytical Statistical Mode Check Problem Output
### LIMITING CASES

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
<th>TOLERANCE</th>
<th>DEP (106)</th>
<th>DEP (207)</th>
<th>DEP (304)</th>
<th>DEP (310)</th>
<th>DEP (311)</th>
<th>DEP (312)</th>
<th>DEP (313)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>3.160000E-01</td>
<td>1.79E-03</td>
<td>-6.00440D-12</td>
<td>0.0</td>
<td>0.0</td>
<td>6.0</td>
<td>2.72850D-12</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>-V</td>
<td>-3.160000E-01</td>
<td>-1.79E-03</td>
<td>-6.00440D-12</td>
<td>0.0</td>
<td>0.0</td>
<td>6.0</td>
<td>2.72850D-12</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**NOTE:** This frame illustrates only the format of the LIMITING CASE printout. It was not generated by the input of Figure 4-5.

**Figure 4-600 Analytical Statistical Mode Check Problem Output**
4.4 Monte Carlo Mode

This problem demonstrates the Monte Carlo evaluation of dispersion statistics and probability distributions. The Monte Carlo mode determines dispersion by performing numerical experiments with the full non-linear trajectory equations rather than the Taylor series approximation used by the Analytical Statistical mode. Thus it is inherently more accurate, although large numbers of experiments would have to be conducted to realize the advantage. The number of experiments impacts cost. This mode is of interest to projectile designers and those interested in projectile dispersion statistics.

Figure 4-7 illustrates the input deck for a Monte Carlo mode simulation. The first card is the Group 1 card defined by Table 3-1. The Group 1 card sets MC$OPT = 1$, which forces the Statistical Processor into the Monte Carlo mode after using the results of the Analytical Statistical mode to set up the histogram boundaries. With the exception of the first card, all cards shown in Figure 4-7 belong to Group 2. Only TYPE = 2, 3, 4, 5, 7, and 8 cards may appear in a Monte Carlo simulation. There must be a minimum of one TYPE = 2, 3, or 4 and one TYPE = 7 or 8. Figure 4-7 illustrates the use of System Control IFCRM (CODE# = 201) to initialize the Fire Control parameters at the Real World nominals.

Figure 4-8 illustrates the printed output generated by a Monte Carlo mode simulation. Frames 'a' through 'p' are the results of the initial Analytical Statistical mode calculations. The user is referred to Section 4.3 for a discussion of this output. Subsequent to the Analytical Statistical calculations, INTERNAL RANGE histograms are set up for all TYPE = 7 and TYPE = 8 variables with the cell boundaries determined by the analytically computed mean values and standard deviations. USER RANGE histograms are also set up for all variables with the cell boundaries for the TYPE = 8 variables determined by the information supplied on the Group 2 cards, as described in Table 3-2. At this point the Monte Carlo experiments begin.

Figure 4-8, Frame 'q' and Frame 'r' illustrate the summary information printed for each of the Monte Carlo experiments. This sequence of experiments is generated by using a random number generator to simulate random variations in the projectile parameters consistent with the input statistics. A given random sequence may be repeated by reusing the value of IRANN$0$ on the Group 1 card (see Table 3-1 for instructions). A different sequence will result from every selection of IRANN$0$. The CASE variable in Frame 'q' counts the experiments. The first line,
labeled IND. VAR., are the stochastic independent variable values appearing in the order given by Frame 'd'. The line labeled DEP. VAR. are the corresponding values of the dependent variables arranged in the order of Frame 'e'. Even though only a small number of experiments (100) were conducted, the summary print required ten pages (only two are included here). Thus, the user should consider suppressing the summary print by setting MCPRNT = 0 in Card Group 1, whenever large numbers of experiments are conducted. The information in the summary print is used to update the histograms after each experiment but is not stored in any other way.

At the conclusion of the Monte Carlo experiments, the histogram information is printed. In general, there are two sets of histograms. The INTERNAL RANGE histograms are printed first and are usually followed by the USER RANGE histograms. Since the example problem input shown in Figure 4-7 contains no TYPE = 8 variables, the sample output of Figure 4-8 contains only INTERNAL RANGE histograms. There is no need to document the USER RANGE histograms separately because they are constructed in an identical fashion and are subject to the same interpretation. The only difference is in how the cell boundaries are set up, which has already been discussed. The histogram data for the example problem commences in Frame 's' of Figure 4-8. The first line of print defines (1) the total number of random experiments performed (SAMPLE), (2) the number of experiments which were rejected from the INTERNAL RANGE histogram set because at least one of the dependent variable histogram ranges were exceeded (INTERNAL REJECTS) and (3) the number of experiments rejected from the USER RANGE histogram set for the same reason. The latter is zero in the example because USER RANGE histograms were never set up. The histograms making up the INTERNAL RANGE histogram set are presented on a variable-by-variable basis in Frames 's' through 'y'. Independent variable histograms precede the dependent variable histograms. Generally, the block of data describing the variable, consists of (1) a line identifying the variable and statistics computed from the histogram, (2) a second line defining the number of experiments rejected from the set because of range limitations on this variable and Analytical Statistical mode statistics, and (3) multiple lines of print defining the histogram on a cell-by-cell basis. The columns define the cell boundaries, the number of occurrences (FREQ), and the probability density function (DIST. FUNCT.). Although the format of the block is the same for all, there are some differences in interpretation. Table 4-1 rigorously interprets each printed value. Hash marks delimit the extent of the comments contained in the right hand column of Table 4-1.
Figure 4-7 Monte Carlo Mode Check Problem Input
Figure 4-8a Monte Carlo Mode Check Problem Output
### SIMULATION INPUT SUMMARY ###

**SINGLE-CASE, RANGE-CHECK, AND STATISTICAL PROCESSOR CONTROLS**

<table>
<thead>
<tr>
<th>IV</th>
<th>IPRINT</th>
<th>ISPRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

**MONTE CARLO CONTROLS**

- MC0PT = 1
- MCALC = 0
- MCELL = 20
- MTRIAL = 100
- INU1 = 50
- INU2 = 50
- MPCPRINT = 1
- [RAMNO = 12001]

- IPLOT = 0
- ISPLOT = 0

---

**Figure 4-8b**: Monte Carlo Mode Check Problem Output
### INPUT VARIABLE SPECIFICATIONS

<table>
<thead>
<tr>
<th>CODE NUMBER</th>
<th>VARIABLE TYPE</th>
<th>NOMINAL VALUE</th>
<th>TOLERANCE VALUE</th>
<th>STANDARD DEVIATION</th>
<th>SUBSEQUENT POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>234</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>116</td>
<td>2</td>
<td>0.0</td>
<td>3.00000 00</td>
<td>1.00000 00</td>
<td>0.0</td>
</tr>
<tr>
<td>117</td>
<td>2</td>
<td>0.0</td>
<td>3.00000 00</td>
<td>1.00000 00</td>
<td>0.0</td>
</tr>
<tr>
<td>118</td>
<td>2</td>
<td>0.0</td>
<td>2.70000 01</td>
<td>9.00000 01</td>
<td>0.0</td>
</tr>
<tr>
<td>119</td>
<td>2</td>
<td>0.0</td>
<td>2.70000 01</td>
<td>9.00000 01</td>
<td>0.0</td>
</tr>
<tr>
<td>120</td>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>251</td>
<td>5</td>
<td>1.00000 00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>109</td>
<td>9</td>
<td>1.17300 03</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>209</td>
<td>5</td>
<td>6.00000 00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>305</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>307</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>308</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>320</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>317</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>322</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>323</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>324</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>310</td>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Figure 4-8c  Monte Carlo Mode Check Problem Output
### Stochastic Independent Variables

<table>
<thead>
<tr>
<th>Theta Real World</th>
<th>Code Number 110</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theta Real World</td>
<td>Code Number 117</td>
</tr>
<tr>
<td>Theta Real World</td>
<td>Code Number 118</td>
</tr>
<tr>
<td>Theta Real World</td>
<td>Code Number 119</td>
</tr>
</tbody>
</table>

Figure 4-8d Monte Carlo Mode Check Problem Input
### Variables Assigned Preset Values

<table>
<thead>
<tr>
<th>Variable</th>
<th>Code Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>CX1 FIRE CONTROL</td>
<td>5</td>
</tr>
<tr>
<td>V1 FIRE CONTROL</td>
<td>6</td>
</tr>
<tr>
<td>E1 FIRE CONTROL</td>
<td>7</td>
</tr>
<tr>
<td>V2 FIRE CONTROL</td>
<td>8</td>
</tr>
<tr>
<td>V3 FIRE CONTROL</td>
<td>9</td>
</tr>
<tr>
<td>C1A FIRE CONTROL</td>
<td>10</td>
</tr>
<tr>
<td>V4 FIRE CONTROL</td>
<td>11</td>
</tr>
<tr>
<td>SK FIRE CONTROL</td>
<td>12</td>
</tr>
<tr>
<td>CH5 FIRE CONTROL</td>
<td>13</td>
</tr>
<tr>
<td>ATM5 FIRE CONTROL</td>
<td>14</td>
</tr>
<tr>
<td>BTMS FIRE CONTROL</td>
<td>15</td>
</tr>
<tr>
<td>TEO5 FIRE CONTROL</td>
<td>16</td>
</tr>
<tr>
<td>P50 FIRE CONTROL</td>
<td>17</td>
</tr>
<tr>
<td>TD50 FIRE CONTROL</td>
<td>18</td>
</tr>
<tr>
<td>P510 FIRE CONTROL</td>
<td>19</td>
</tr>
<tr>
<td>G06 FIRE CONTROL</td>
<td>20</td>
</tr>
<tr>
<td>AZ6 FIRE CONTROL</td>
<td>21</td>
</tr>
<tr>
<td>EL6 FIRE CONTROL</td>
<td>22</td>
</tr>
<tr>
<td>V7 FIRE CONTROL</td>
<td>23</td>
</tr>
<tr>
<td>W8 FIRE CONTROL</td>
<td>24</td>
</tr>
<tr>
<td>AX9 FIRE CONTROL</td>
<td>25</td>
</tr>
<tr>
<td>A10 FIRE CONTROL</td>
<td>26</td>
</tr>
<tr>
<td>ADcon FIRE CONTROL</td>
<td>27</td>
</tr>
<tr>
<td>CAA FIRE CONTROL</td>
<td>28</td>
</tr>
<tr>
<td>VO FIRE CONTROL</td>
<td>29</td>
</tr>
<tr>
<td>WK REAL WORLD</td>
<td>30</td>
</tr>
<tr>
<td>WJ REAL WORLD</td>
<td>31</td>
</tr>
<tr>
<td>W5 REAL WORLD</td>
<td>32</td>
</tr>
<tr>
<td>RHO REAL WORLD</td>
<td>33</td>
</tr>
<tr>
<td>CX1 REAL WORLD</td>
<td>34</td>
</tr>
<tr>
<td>V1 REAL WORLD</td>
<td>35</td>
</tr>
<tr>
<td>V2 REAL WORLD</td>
<td>36</td>
</tr>
<tr>
<td>CH5 REAL WORLD</td>
<td>37</td>
</tr>
<tr>
<td>SK REAL WORLD</td>
<td>38</td>
</tr>
<tr>
<td>CH5 REAL WORLD</td>
<td>39</td>
</tr>
<tr>
<td>BTM5 REAL WORLD</td>
<td>40</td>
</tr>
<tr>
<td>V8 REAL WORLD</td>
<td>41</td>
</tr>
<tr>
<td>W9 REAL WORLD</td>
<td>42</td>
</tr>
<tr>
<td>AX10 FIRE CONTROL</td>
<td>43</td>
</tr>
<tr>
<td>G11 REAL WORLD</td>
<td>44</td>
</tr>
<tr>
<td>AZ12 REAL WORLD</td>
<td>45</td>
</tr>
<tr>
<td>A REAL WORLD</td>
<td>46</td>
</tr>
<tr>
<td>D REAL WORLD</td>
<td>47</td>
</tr>
<tr>
<td>W REAL WORLD</td>
<td>48</td>
</tr>
<tr>
<td>AX10 FIRE CONTROL</td>
<td>49</td>
</tr>
<tr>
<td>A11 REAL WORLD</td>
<td>50</td>
</tr>
<tr>
<td>ADcon REAL WORLD</td>
<td>51</td>
</tr>
<tr>
<td>A51 REAL WORLD</td>
<td>52</td>
</tr>
<tr>
<td>IFCON SYSTEM CONTROL</td>
<td>202</td>
</tr>
</tbody>
</table>

Figure 4-8g  Monte Carlo Mode Check Problem Output
### Projectile Dispersion Case Summary

<table>
<thead>
<tr>
<th>FIRE CONTROL PARAMETERS</th>
<th>REAL WORLD PARAMETERS</th>
<th>SYSTEMS CONTROLS</th>
<th>COMPILED QUANTITIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>V2 = 1,19000 04 FT/SEC</td>
<td>WX = 0.0 FT/SEC</td>
<td>IW = 0.0 FT/SEC</td>
<td>RM = 2,37800-03 SL/CFT</td>
</tr>
<tr>
<td>CT = 3,06000 04 (NONE)</td>
<td>V3 = 0.0 FT/SEC</td>
<td>WZ = 0.0 FT/SEC</td>
<td>RM = 2,37800-03 SL/CFT</td>
</tr>
<tr>
<td>X4 = 1,17700 03 FT</td>
<td>CNA = 1,07700 00 1/RAO</td>
<td>SW = 8,00000-02 100/L</td>
<td>RM = 2,37800-03 (NONE)</td>
</tr>
<tr>
<td>CWA = 0.0 INW</td>
<td>APM = 0.0 FT/SEC</td>
<td>PMP = 0.0 DEG</td>
<td>RM = 2,37800-03 (NONE)</td>
</tr>
<tr>
<td>PEI = 0.0 DEG</td>
<td>RUNTO = 0.0 FT/SEC</td>
<td>RT = 0.0 DEG</td>
<td>RM = 2,37800-03 (NONE)</td>
</tr>
<tr>
<td>A2 = 0.0 DEG</td>
<td>EL = 0.0 FT/SEC</td>
<td>RAR = 0.0 DEG</td>
<td>RM = 2,37800-03 (NONE)</td>
</tr>
<tr>
<td>ALPCON = 4,00000-01 DEG</td>
<td>CA = 0.0 FT/SEC</td>
<td>RFD = 0.0 DEG</td>
<td>RM = 2,37800-03 (NONE)</td>
</tr>
</tbody>
</table>

### ICCW

<table>
<thead>
<tr>
<th>IFCWC = 0 (NONE)</th>
<th>IFCRC = 0 (NONE)</th>
<th>IPM = 0 (NONE)</th>
<th>IDUMP = 0 (NONE)</th>
</tr>
</thead>
</table>

**Figure 4-8i Monte Carlo Mode Check Problem Output**
### PROJECTILE DISPERSION CASE SUMMARY

#### FIRE CONTROL PARAMETERS

<table>
<thead>
<tr>
<th>V0</th>
<th>1,10200 04 FT/SEC</th>
<th>Wx</th>
<th>0.0</th>
<th>PFT/SEC</th>
<th>0.0</th>
<th>RMH</th>
<th>2,17500 03 SL/FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCI</td>
<td>3,400 00 00 (NONE)</td>
<td>V1</td>
<td>1,10200 04 FT/SEC</td>
<td>Cxz</td>
<td>1,10200 01 (NONE)</td>
<td>V2</td>
<td>2,17500 03 FT/SEC</td>
</tr>
<tr>
<td>CHPA</td>
<td>5,600 00 (NONE)</td>
<td>CMA</td>
<td>1,10200 01 1/RAD</td>
<td>SM</td>
<td>6,20000 00 (NONE)</td>
<td>CSM</td>
<td>7,50000 00 (NONE)</td>
</tr>
<tr>
<td>FEE</td>
<td>5,600 00 (NONE)</td>
<td>TOX</td>
<td>0.0</td>
<td>DEX</td>
<td>0.0</td>
<td>DEG</td>
<td>0.0</td>
</tr>
<tr>
<td>W</td>
<td>1,10200 01 1/RAD</td>
<td>AI</td>
<td>1,12600 00 1/SL-FT</td>
<td>EY</td>
<td>0.0</td>
<td>DEG</td>
<td>0.0</td>
</tr>
<tr>
<td>ALFOC</td>
<td>6,00000 01 1/DEG</td>
<td>CA</td>
<td>0.0</td>
<td>1/5/DEG</td>
<td>1/5/DEG</td>
<td>1/5/DEG</td>
<td></td>
</tr>
</tbody>
</table>

#### REAL WORLD PARAMETERS

<table>
<thead>
<tr>
<th>V0</th>
<th>1,10200 04 FT/SEC</th>
<th>Wx</th>
<th>0.0</th>
<th>PFT/SEC</th>
<th>0.0</th>
<th>RMH</th>
<th>2,17500 03 SL/FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCI</td>
<td>3,400 00 00 (NONE)</td>
<td>V1</td>
<td>1,10200 04 FT/SEC</td>
<td>Cxz</td>
<td>1,10200 01 (NONE)</td>
<td>V2</td>
<td>2,17500 03 FT/SEC</td>
</tr>
<tr>
<td>CHPA</td>
<td>5,600 00 (NONE)</td>
<td>CMA</td>
<td>1,10200 01 1/RAD</td>
<td>SM</td>
<td>6,20000 00 (NONE)</td>
<td>CSM</td>
<td>7,50000 00 (NONE)</td>
</tr>
<tr>
<td>FEE</td>
<td>5,600 00 (NONE)</td>
<td>TOX</td>
<td>0.0</td>
<td>DEX</td>
<td>0.0</td>
<td>DEG</td>
<td>0.0</td>
</tr>
<tr>
<td>W</td>
<td>1,10200 01 1/RAD</td>
<td>AI</td>
<td>1,12600 00 1/SL-FT</td>
<td>EY</td>
<td>0.0</td>
<td>DEG</td>
<td>0.0</td>
</tr>
<tr>
<td>ALFOC</td>
<td>6,00000 01 1/DEG</td>
<td>CA</td>
<td>0.0</td>
<td>1/5/DEG</td>
<td>1/5/DEG</td>
<td>1/5/DEG</td>
<td></td>
</tr>
</tbody>
</table>

#### SYSTEM CONTROL

<table>
<thead>
<tr>
<th>IFP</th>
<th>-1 (NONE)</th>
<th>IFPC</th>
<th>(NONE)</th>
<th>IPIC</th>
<th>0 (NONE)</th>
<th>IDMP</th>
<th>0 (NONE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YWDR</td>
<td>2.0</td>
<td>FT</td>
<td>YWDR</td>
<td>0.0</td>
<td>FT</td>
<td>ZEAR</td>
<td>0.0</td>
</tr>
<tr>
<td>YWDR</td>
<td>2.0</td>
<td>FT</td>
<td>YWDR</td>
<td>0.0</td>
<td>FT</td>
<td>ZEAR</td>
<td>0.0</td>
</tr>
<tr>
<td>FAND</td>
<td>3.0</td>
<td>FT</td>
<td>DX</td>
<td>0.0</td>
<td>FT</td>
<td>DX</td>
<td>0.0</td>
</tr>
<tr>
<td>FAND</td>
<td>3.0</td>
<td>FT</td>
<td>DX</td>
<td>0.0</td>
<td>FT</td>
<td>DX</td>
<td>0.0</td>
</tr>
<tr>
<td>D</td>
<td>1.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
</tr>
<tr>
<td>D</td>
<td>1.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
</tr>
<tr>
<td>F</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
</tr>
<tr>
<td>F</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
</tr>
<tr>
<td>F</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
</tr>
<tr>
<td>F</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
</tr>
<tr>
<td>F</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
</tr>
<tr>
<td>F</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
<td>SEC</td>
<td>DAD</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### Figure 3-1 Monte Carlo Mode Check Problem Output
Figure 4-81 Monte Carlo Mode Check Problem Output
Figure 4-8m Monte Carlo Mode Check Problem Output
Figure 4-8n Monte Carlo Mode-Check Problem Output
<table>
<thead>
<tr>
<th>INCL. VARP</th>
<th>316</th>
<th>307</th>
<th>308</th>
<th>320</th>
</tr>
</thead>
<tbody>
<tr>
<td>115 117</td>
<td>0.0</td>
<td>2.66593E-17</td>
<td>4.81870E-19</td>
<td>0.0</td>
</tr>
<tr>
<td>115 118</td>
<td>-2.33340E-04</td>
<td>-1.37563E-18</td>
<td>2.94449E-19</td>
<td>0.0</td>
</tr>
<tr>
<td>116 119</td>
<td>0.3</td>
<td>9.85150E-19</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>117 118</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>117 119</td>
<td>-2.33340E-04</td>
<td>0.0</td>
<td>2.74130E-18</td>
<td>0.0</td>
</tr>
<tr>
<td>118 119</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPENDENT VARIABLES</th>
<th>317</th>
<th>322</th>
<th>323</th>
<th>324</th>
<th>319</th>
<th>319</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>3.0</td>
<td>2.69050E-17</td>
<td>4.81870E-19</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>2.28830E-09</td>
<td>-2.33340E-04</td>
<td>-1.37563E-18</td>
<td>2.94449E-19</td>
<td>2.28830E-09</td>
<td>-2.74580E-03</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>9.85150E-19</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>2.28830E-09</td>
<td>-2.33340E-04</td>
<td>0.0</td>
<td>2.74130E-18</td>
<td>2.28830E-09</td>
<td>-2.74580E-03</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4-8o Monte Carlo Mode Check Problem Output
### Second Order Means and Variances Assuming Gaussian Independent Variables

<table>
<thead>
<tr>
<th>Dependent Variables</th>
<th>306</th>
<th>307</th>
<th>308</th>
<th>320</th>
<th>317</th>
<th>322</th>
<th>323</th>
<th>324</th>
<th>318</th>
<th>510</th>
</tr>
</thead>
<tbody>
<tr>
<td>Means</td>
<td>-3.20140 02</td>
<td>0.0</td>
<td>1.67530-17</td>
<td>5.298780 00</td>
<td>3.73520-04</td>
<td>-3.89340 00</td>
<td>0.0</td>
<td>1.67530-17</td>
<td>1.09907-01</td>
<td>1.02220 24</td>
</tr>
<tr>
<td>Variances</td>
<td>1.43780 01</td>
<td>9.441160 00</td>
<td>0.441160 00</td>
<td>4.69450 00</td>
<td>1.32240-07</td>
<td>1.43780 01</td>
<td>9.441160 00</td>
<td>0.441160 00</td>
<td>1.32240-07</td>
<td>1.67310 03</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>7.79250 00</td>
<td>3.07600 00</td>
<td>3.07600 00</td>
<td>2.23190 00</td>
<td>3.63650-04</td>
<td>3.79340 00</td>
<td>3.07600 00</td>
<td>3.07600 00</td>
<td>3.63650-04</td>
<td>4.32900 01</td>
</tr>
</tbody>
</table>

### Standard Deviations with Highest Precision

<table>
<thead>
<tr>
<th></th>
<th>3.72620 00</th>
<th>3.4767817359174500 00</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.73500 00</td>
<td>3.675687047037422200 00</td>
<td></td>
</tr>
<tr>
<td>7.27400 00</td>
<td>3.675687047037422200 00</td>
<td></td>
</tr>
<tr>
<td>2.23190 00</td>
<td>2.231902732734010500 00</td>
<td></td>
</tr>
<tr>
<td>1.63450-04</td>
<td>1.63450-04</td>
<td></td>
</tr>
<tr>
<td>3.72620 00</td>
<td>3.72620 00</td>
<td></td>
</tr>
<tr>
<td>3.67620 00</td>
<td>3.079377047247022200 00</td>
<td></td>
</tr>
<tr>
<td>4.67620 00</td>
<td>4.079377047247022200 00</td>
<td></td>
</tr>
<tr>
<td>5.37420-04</td>
<td>5.37420-04</td>
<td></td>
</tr>
<tr>
<td>4.72620 00</td>
<td>4.32675917194675100 01</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 4-8p** Monte Carlo Mode Check Problem Output
### SUMMARY OF MONTE CARLO RANDOM EXPERIMENTS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2465D+02</td>
<td>-5.372D+01</td>
<td>2.647D+02</td>
<td>-5.683D+01</td>
<td>1.159D+03</td>
<td>-6.089D+00</td>
<td>-6.952D+00</td>
<td>-1.995D+00</td>
<td>1.086D+01</td>
<td>1.021D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3.473D+01</td>
<td>-2.657D+02</td>
<td>-2.613D+02</td>
<td>6.121D+00</td>
<td>1.138D+03</td>
<td>-7.853D+00</td>
<td>-6.837D+00</td>
<td>2.071D+00</td>
<td>1.067D+01</td>
<td>1.022D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-1.727D+01</td>
<td>1.077D+02</td>
<td>-2.573D+00</td>
<td>7.304D+00</td>
<td>1.669D+04</td>
<td>-2.157D+00</td>
<td>-2.234D+00</td>
<td>2.516D+00</td>
<td>1.097D+01</td>
<td>1.039D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-4.375D+01</td>
<td>-1.948D+02</td>
<td>8.952D+00</td>
<td>4.867D+01</td>
<td>1.261D-04</td>
<td>-2.960D+00</td>
<td>3.110D+00</td>
<td>2.997D+00</td>
<td>1.096D+01</td>
<td>1.040D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-7.268D+00</td>
<td>-2.011D+00</td>
<td>-2.922D+00</td>
<td>2.114D+00</td>
<td>2.065D+04</td>
<td>-1.747D+00</td>
<td>-3.436D+00</td>
<td>-6.773D+01</td>
<td>1.097D+01</td>
<td>1.030D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-1.296D+00</td>
<td>-2.832D+00</td>
<td>-1.118D+00</td>
<td>2.347D+00</td>
<td>2.464D+04</td>
<td>-1.335D+00</td>
<td>-3.843D+00</td>
<td>7.386D+00</td>
<td>1.097D+01</td>
<td>1.038D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-7.693D+00</td>
<td>-1.231D+00</td>
<td>-3.847D+00</td>
<td>4.135D+00</td>
<td>7.389D+00</td>
<td>7.389D+00</td>
<td>3.707D+00</td>
<td>1.302D+01</td>
<td>7.198D+00</td>
<td>1.026D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>6.442D+01</td>
<td>6.695D+00</td>
<td>1.139D+00</td>
<td>1.736D+00</td>
<td>2.155D+04</td>
<td>1.555D+00</td>
<td>3.922D+00</td>
<td>6.074D+00</td>
<td>1.097D+01</td>
<td>1.038D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-2.235D+00</td>
<td>3.905D+00</td>
<td>6.074D+00</td>
<td>4.222D+00</td>
<td>2.304D+04</td>
<td>1.498D+00</td>
<td>-3.784D+00</td>
<td>1.529D+00</td>
<td>1.097D+01</td>
<td>1.036D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-2.137D+00</td>
<td>2.047D+00</td>
<td>3.763D+00</td>
<td>4.093D+00</td>
<td>2.306D+04</td>
<td>1.498D+00</td>
<td>-3.784D+00</td>
<td>1.529D+00</td>
<td>1.097D+01</td>
<td>1.036D+04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4-8q Monte Carlo Mode Check Problem Output
<table>
<thead>
<tr>
<th>CASE</th>
<th>INFO VAR</th>
<th>DEL VAR</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
</table>

CASE 97
DEL VAR | -1.51670-01 | -6.15730-01 | 2.19720-01 | 4.42410-01 | 1.97670-01 | 0.006000-01 | 1.09710-01 | 1.03500-01 |

CASE 98
INFO VAR | -2.86210-01 | -2.84170-01 | -1.08000-01 | -4.66850-01 | 2.23260-01 | 1.27500-02 | -3.705500-02 | -1.58380-01 | 1.09750-01 | 1.03500-01 |
DEL VAR | -3.31490-00 | -3.70550-01 | -1.35280-01 | 4.32750-01 | 2.23260-01 | 1.27500-00 | -3.705500-02 | -1.58380-01 | 1.09750-01 | 1.03500-01 |

CASE 99
INFO VAR | 5.99330-01 | 1.23440-01 | 0.00320-01 | -2.35500-02 | 1.00310-00 | -5.37500-00 | 3.03120-00 | -7.97500-00 | 1.10530-01 | 1.02450-01 |
DEL VAR | -1.09270-01 | 3.33210-00 | -7.97500-00 | 1.00310-00 | -5.37500-00 | 3.03120-00 | -7.97500-00 | 1.10530-01 | 1.02450-01 |

CASE 100
DEL VAR | -4.94330-02 | 1.8640-01 | -3.6430-00 | 3.85880-00 | 4.74150-06 | 3.84390-00 | 1.48410-01 | -3.04040-01 | 1.09540-01 | 1.04260-01 |

Figure 4-8r Monte Carlo Mode Check Problem Output
## Monte Carlo Results

**Sample = 100**  
**Internal Rejects = 12**  
**User Rejects = 0**

### Dependent Variables with Internal Ranges

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Mean</th>
<th>Var</th>
<th>Interval</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>306</td>
<td>-0.239430 D 1</td>
<td>0.487650 D 01</td>
<td>0.113720 D 01</td>
<td></td>
</tr>
<tr>
<td>307</td>
<td>-0.141250 D 01</td>
<td>0.487650 D 01</td>
<td>0.113720 D 01</td>
<td></td>
</tr>
<tr>
<td>308</td>
<td>-0.141250 D 01</td>
<td>0.487650 D 01</td>
<td>0.113720 D 01</td>
<td></td>
</tr>
</tbody>
</table>

### Figure 4-8s Monte Carlo Mode Check Problem Output
Figure 4.8u Monte Carlo Mode Check Problem Output
### Figure 4-8x  Monte Carlo Mode Check Problem Output

![Image of the figure with Table 4-8x]
### Table 4-1 Monte Carlo Mode Output Interpretation

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
</table>
| CODE Number   | Independent/dependent variable identification number. | - See Table 3-3 for definitions.  
|               |             | - CODE NUMBER list is identical to CODE# list defined by Card Group 2. |
| MEAN          | Expected or mean value of the variable as determined by histogram. | - See Eq. (B.1-22) for algorithm. |
| VAr           | Variance of the variable as determined by the histogram. | - See Eq. (B.1-24) for algorithm. |
| INTERVAL      | Histogram cell size: | - NCELL is the number of cells as defined by Card Group 1.  
|               | \[
|               | \text{INTERVAL} = \frac{2 \times TOLERANCE}{\text{NCELL}} \]
|               |             | - TOLERANCE is defined later in this table. |
| NO. REJECTS   | Number of Monte Carlo experiments rejected from all histograms due to range limitations on this variable. | - Rejects occur only on dependent variables (i.e., TYPE = 7 or TYPE = 8).  
| (MEAN)        | Defines midpoint of histogram. The value is generally related to the mean or expected value of the variable. However, it is subject to varying definitions. | - See Appendix B.2.3, paragraph \textit{Monte Carlo}, for a discussion of rejects.  
|               | \text{TYPE} = 2, 3 Mean or expected value of variable as determined by VALUE on group 2 card. | - For TYPE = 2 or TYPE = 3, the input mean value is repeated for reference purposes.  
|               | \text{TYPE} = 4 Midpoint of abscissa table used to define the probability density function. | - For TYPE = 4, the actual mean value may be determined from the LIMITING CASES printed output. VALUE (as given there) is the mean value determined by the probability density function. |
### Table 4-1 Monte Carlo Mode Output Interpretation (cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
</table>
| (MEAN) (cont'd) | **TYPE = 7** The expected or mean value as determined by the Statistical Processor. | • See Eq. (B.1-5) for algorithm.  
• Details of the calculation depend on the IOPRN variable defined by Group 1 Card. |
| | **TYPE = 8** Definition depends on histogram range:  
• INTERNAL RANGES - **TYPE = 7** definition applies.  
• USER RANGES - Equal to VALUE as defined by Group 2 Card. | |
| (VAR) | Value sometimes related to variance ($\sigma^2$) of the variable. However, it is subject to varying definitions.  
**TYPE = 2, 3** Variance of variable as determined by STDEV on Group 2 Card:  
(VAR) = STDEV**2 | • For **TYPE = 2 or TYPE = 3**, the input variance is repeated for reference purposes.  
• For **TYPE = 4**, the actual variance ($\sigma^2$) may be determined from the LIMITING CASES printed output. TOLERANCE (as defined there) is related to the standard deviation ($\sigma$) determined by the input probability density function according to:  
\[
\sigma = 3 \times \text{TOLERANCE}
\]
so that the variance is  
\[
\sigma^2 = 9 \times \text{TOLERANCE}^2
\]
• Maximum value determined by search of probability density function table and used in generating random samples. See Appendix A.5.3 for full discussion. |
| | **TYPE = 4** Maximum tabulated value of probability density function. | |
Table 4-1 Monte Carlo Mode Output Interpretation (cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME (cont'd)</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE = 7</td>
<td>The variance ($\sigma^2$) as determined by the Statistical Processor.</td>
<td>- See Eq. (B.1-7) for algorithm.</td>
</tr>
<tr>
<td>TYPE = 6</td>
<td>Definition depends on histogram range:</td>
<td></td>
</tr>
<tr>
<td>- INTERNAL RANGES - TYPE = 7</td>
<td>definition applies.</td>
<td></td>
</tr>
<tr>
<td>- USER RANGES - VALUE determined by STDEV value on Group 2 Card according to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(VAR) = STDEV**2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOLERANCE</td>
<td>Defines the histogram half range. Subject to varying definitions.</td>
<td></td>
</tr>
<tr>
<td>TYPE = 2</td>
<td>The larger of 3*STDEV or TOL, both as defined on the Group 2 Card.</td>
<td>- Normally TOL is selected to be small for the purposes of evaluating the derivatives. Thus, the half range of the histogram will usually be TOLERANCE = 3*STDEV</td>
</tr>
<tr>
<td>TYPE = 3</td>
<td>The value of TOL as defined by the Group 2 Card.</td>
<td></td>
</tr>
<tr>
<td>TYPE = 4</td>
<td>Half the interval covered by the probability density function abscissa table.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4-1 Monte Carlo Mode Output Interpretation (concl'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
</table>
| TOLERANCE (cont'd) | TYPE = 7  The _σ_ value where _σ_ is defined by the Statistical Processor, i.e., 
  
  \[ \sigma = \sqrt{\text{VAR}} \]  
  TYPE = 8  Definition depends on histogram range:
  - INTERNAL RANGES - TYPE = 7  
    definition applies.
  - USER RANGES - TOL as defined by the Group 2 Card. |
| CLASS MARK | Midpoint of histogram cells. | |
| LOWER BOUND | Left hand cell boundary. | |
| UPPER BOUND | Right hand cell boundary. | |
| FREQ. | Number of occurrences of cell during Monte Carlo experiments. | • Total of FREQ Column is equal to the number of trials (NTRIAL on Group 1 Card) less the number of rejects. |
| DIST.FUNCT. | Estimated probability density function value for this interval. | • See Eq. (8.1-18) for computational algorithm. |
This report is a User's Manual for the Hypervelocity Inflight Trajectory Scatter (HITS) computer code. The primary purpose of the code is to evaluate "projectile dispersion" which is defined here as the miss distance associated with inflight behavior not anticipated by Fire Control. Projectile dispersion poses a fundamental limit on overall weapon system effectiveness. The code is central to a scientific, systematic approach to evaluating projectile dispersion. The method consists of (1) constructing a comprehensive model for each source of projectile dispersion (i.e., the error source model) and (2) processing it through the HITS code to obtain a detailed projectile dispersion error budget, replete with sensitivity coefficients. The error budget identifies the larger dispersion sources and those with significant potential, as well as the total dispersion. Both crossrange and downrange dispersion are computed. The error budget defines dispersion on a level conducive to interpretation and a comprehensive understanding.

The HITS code evaluates projectile dispersion statistics by manipulating the inputs to a set of trajectory equations representing the true trajectory and comparing the computed flight path to the trajectory predicted by a second set of equations representing Fire Control. The true trajectory is varied in accord with the error source model; the Fire Control Trajectory is based on nominal projectile characteristics. At the selection of the user, variations are governed by either analytical or Monte Carlo statistical methods.

The two sets of trajectory equations are closed form approximations to the full six degree of freedom (6 DOF) equations of motion. They consist of (1) a particle trajectory with a velocity dependent drag coefficient and (2) two perturbation equations which correct for angle of attack effects. Extensive 6 DOF simulations have been conducted to verify the approximations. These equations also have potential application to operational Fire Control computers, the computation of firing tables, and any other situation requiring rapid, accurate, low-cost trajectory determination.

"Hypervelocity" is used throughout this report to indicate the limitation of the present HITS code to projectiles whose velocity does not become transonic at any point along the trajectory. No other connotation is intended.
The HITS code consists of two portions. One portion performs all input/output functions and the statistical calculations. This portion is an adaptation of existing AVCO software (i.e., YP-58). It brings to the HITS code additional capability as a projectile design tool; a capability the code would not otherwise have had. The closed form trajectory equations constitute the second portion of the code. These equations were specifically developed for dispersion assessment under an earlier contract. The code was assembled under the earlier contract and used to evaluate the projectile dispersion of a reference design. The code, however, was developed only to the level of a research program and documented only to the extent necessary to support the analyses of the study. Detailed input/output information was not presented.

Under the present effort, the HITS code was advanced to the level of a production code. The code is maintained in the AVCO engineering computer code library as Production Code 5127. This report is the associated User's Manual. It places maximum emphasis on input/output information. The text contains all the day-to-day necessities required to operate the code and interpret the results of computation. Input/output information is summarized by tables featuring a quick-reference step-by-step format. Example problems are presented and discussed. They verify proper installation and illustrate the output format. The appendices discuss theoretical and programming aspects of the code. They contain the theoretical development of the closed form trajectory equations and a complete listing of the code.

APPENDIX A

ELEMENTS OF PROBABILITY THEORY AND STATISTICS

This appendix presents an overview of the pertinent aspects of probability theory. Sections A.1 through A.5 review basic concepts and theoretical relationships. Sections A.5 and A.6 relate the theory to the practical problems associated with Monte Carlo methods. More detailed treatments may be found in textbooks.\textsuperscript{1,2}

A.1 Random Phenomenon and Probabilities

Random phenomenon are those phenomena whose fundamental processes are either not completely understood or are too complex to define in an entirely satisfactory manner. In either case, the scientific interest in the phenomenon centers around the "average" rather than the "precise" outcome. Probability theory is a mathematical discipline for quantifying random phenomenon.

A.1.1 Fundamental Concepts

Probability theory has three building blocks: (1) the sample description space, (2) a collection of events, and (3) a probability function. This section discusses these concepts.

Sample Description Space

The sample description space is the collection of all possible outcomes of the random phenomena. Each occurrence is thought of as the result of an experiment. For instance, the sample description space for a coin flipping experiment would consist of the two possible outcomes (H, T). The sample description space for a temperature measurement taken at randomly selected time and location would include all values from $-\infty$ to $+\infty$, $(-\infty, +\infty)$.

Events

Random events (or simply "Events") are the groupings of the fundamental outcomes. They are subsets of the sample description space. Events are defined by the analyst. For instance, a meteorologist interested in the likelihood of sub-freezing temperatures would


select only two events to subdivide the sample description space of all possible temperatures: \((-\infty, 32)\) and \((32, +\infty)\).

**Probability Function**

The probability function, \(P\), assigns to each event a number between zero and one corresponding to the relative likelihood of occurrence of the event. The probability of an event, \(A\), is the theoretical ratio of the number of times it would occur, \(N\), to the total number of experiments, \(N_T\).

\[
P[A] = \frac{N}{N_T}
\]

(Monte Carlo procedures estimate probabilities by counting the results of experiments.) An event which (almost) never occurs has a probability of zero. An event which (almost) always occurs has a probability of one. For instance, if freezing temperatures occurred 40\% of the time the probability of that event would be

\[
P[(-\infty, 32)] = 0.40
\]

**Venn Diagrams**

A Venn diagram is a graphical method of depicting the relationship between the sample description space and events. (It also gives a convenient method for illustrating the rules of combination for probabilities.) Figure A-1 shows a Venn diagram. There are three events \(A\), \(B\), and \(C\). Events \(A\) and \(B\) overlap so \(A\) and \(B\) simultaneously occur for some experimental outcomes. Event \(C\) is exclusive of \(A\) and \(B\), since neither \(A\) nor \(B\) can occur when \(C\) occurs.

**A.1.2 Rules of Combination**

Let the probabilities of the individual events in Figure A-1 be \(P(A)\), \(P(B)\), and \(P(C)\). There are rules of combination. Let \(A + B\) denote the occurrence of either event, and let \(AB\) denote the simultaneous occurrence of both events. The probability of \(A + B\) is

\[
P(A + B) = P(A) + P(B) - P(AB)
\]
FIGURE A-1    THE VENN DIAGRAM
Referring to Figure A-1, $A + B$ is the area encompassed by either events $A$ or $B$, and $AB$ is the area common to both $A$ and $B$. With these visualizations, Eq. (A.1-3) simply states that $P(A+B)$ is equal to $P(A) + P(B)$ less a correction, $P(AB)$, for the common area which is counted twice in the sum $P(A) + P(B)$.

Since probabilities are positive numbers, Eq. (A.1-3) implies

$$P(A+B) \leq P(A) + P(B) \tag{A.1-4}$$

Considering a collection of events: $A$, $B$, $C$, ..., with the probabilities $P(A)$, $P(B)$, $P(C)$..., the probability that at least one of the events will occur is $P(A + B + C + ...)$ which is related to the individual probabilities through the inequality:

$$P(A + B + C + ...) \leq P(A) + P(B) + P(C) + ... \tag{A.1-5}$$

It is an inequality since it may be possible for at least two of the events to occur concurrently. If only one of the events can occur the events are said to be mutually exclusive and equality holds. If at least one of the events must occur, the ensemble of events is termed exhaustive in which case the left side of the above inequality is identically one.

A.1.3 Conditional Probabilities

Considering two events, $A$ and $B$, the probability of their simultaneous or joint occurrence is $P(AB)$. The probability $P(A/B)$ is the conditional probability of $A$ given $B$ and is defined by:

$$P(A/B) = \frac{P(AB)}{P(B)} \tag{A.1-6}$$

provided $P(B) > 0$. The conditional probability $P(A/B)$, is the probability $A$ will occur given that $B$ has occurred. Two events, $A$ and $B$, are said to be statistically independent if knowledge that $B$ has occurred is of no value in inferring that $A$ will occur so $P(A/B) = P(A)$. This means

$$P(ABC...) = P(A) P(B) P(C) ... \tag{A.1-7}$$
if A, B, C, ... are mutually independent events. That is, for statistically independent events, the probability of simultaneous occurrence is equal to the product of the individual probabilities.

A.2 Random Variables

The discussion of the previous section dealt with random phenomena in general. The discussion from this point on is restricted to random variables, i.e., those whose outcomes are numerically valued. The outside temperature, the instantaneous voltage across a resistor, and the displacement of the impact point from the target point are just three examples of random variables.

A.2.1 Distribution Functions

A numerically valued random phenomenon is one which is completely summarized by the value of a descriptive variable (i.e., temperature, voltage, or misdistance in the foregoing examples). The variable is referred to as a random or stochastic variable. The sample description space is all real numbers, and the events are all intervals and combinations of intervals. The probability function is defined by the (probability) distribution function, \( F \),

\[
F(X) = P(x \leq X)
\]

The distribution function evaluated at \( X \) is the probability that the random variable \( x \) has a value less than or at the most equal to \( X \). By definition, the cumulative distribution function has the properties \( F(-\infty) = 0, F(+\infty) = 1 \) and \( F(x) \) is a positive, non-decreasing function of \( x \). (Note: It is customary to notationally distinguish between a random variable, \( x \), and a value it could assume, \( X \), only when demanded by clarity.) A distribution function need not be a continuous function. It may have step-type discontinuities at discrete points corresponding to values with a positive probability of occurrence. Random variables may be "discrete," "continuous," or "mixed." The remainder of this section discusses these types.

A discrete random variable can take on specific discrete values \( X_1, X_2, \ldots X_n \). The numerical value shown on a pair of dice is an
example of a discrete random variable. Each experiment must result in an integer value between two and twelve. If \( f_i \) is the probability, \( X_i \) occurs, the distribution function is given as

\[
F(X) = \sum_{X_i \leq X} f_i = P(x \leq X)
\]  

(A.2-2)

and

\[
F(X_N) = \sum_{i=1}^{N} f_i = 1
\]  

(A.2-3)

If the distribution function derivative exists, the random variable is said to be continuous and the (probability) density function is defined according to

\[
f(x) = \frac{d}{dx} F(x)
\]  

(A.2-4)

The density function at \( X, f(X) \), is the probability of the event that the random variable \( x \) falls in the range

\[
X \leq x \leq X + dX
\]  

(A.2-5)

The probability of occurrence of any specific value of a continuous random variable is zero, since

\[
\lim_{\epsilon \to 0} P[X \leq x \leq X + \epsilon] = \lim_{\epsilon \to 0} \int_{X}^{X + \epsilon} f(x) \, dx
\]  

(A.2-6)

Air temperature is an example of a continuous random variable since the likelihood of any specific temperature is zero. As a result of the definitions

\[
\int_{-\infty}^{\infty} f(x) \, dx = F(\infty) - F(-\infty) = 1
\]  

(A.2-7)

The distribution of a mixed random variable contains both discrete values, with positive probability of occurrence, and a continuous distribution.
Most random variables treated by Monte Carlo methods are continuous. The density function is estimated by dividing up the range of the random variable into small intervals called cells. Histograms are then constructed which count the number of times the value of the random variable falls in each cell, $N_i$. The exact value is discarded and only the histogram is maintained during execution of the code. The density function is estimated as a piece-wise constant function by the relationship

$$f(x) = \frac{N_i}{N_T} \frac{1}{x_{b_{i+1}} - x_{b_i}}$$

where $N_T$ is the total number of trials and $x_{b_{i+1}}$ and $x_b$ are the upper and lower boundaries of the $i$th cell. It might be argued that Monte Carlo methods, in effect, approximate all random variables as discrete since only the histogram data is saved. Nevertheless, it is most convenient to treat only continuous random variables during the theoretical development. Equation (A.2-8) is all that need be recalled to make the translation back to the real world of Monte Carlo analysis.

A.2.2 Joint Distribution Functions

It is often necessary to describe two or more random variables in relation to each other. These situations are handled by treating each of the random variables $x_i$ as an element of a random vector $\mathbf{x}$. The joint (probability) distribution function is defined as

$$F(\mathbf{x}) = P(x_1 \leq X_1, x_2 \leq X_2, \ldots, x_n \leq X_N)$$

From the definition, the following limits hold

$$\lim_{x_1 \to -\infty} F(x) = 1$$

$$\vdots$$

$$\lim_{x_n \to -\infty} F(x) = 1$$

(A.2-10)
where $F_i(x_i)$ is the distribution function of the $i$th random variable. The joint (probability) density function is defined as

\[
\frac{\partial^n}{\partial x_1 \partial x_2 \ldots \partial x_n} F(x) = f(x) \tag{A.2-12}
\]

\[\lim_{x_1 \to \infty} F(x) - F_i(x_i) \]
\[
\vdots
\]
\[
x_{i-1} \to \infty
\]
\[
x_{i+1} \to \infty
\]
\[
\vdots
\]
\[
x_n \to \infty
\]  \hspace{1cm} (A.2-11)

A.2.3 Statistical Independence

Random variables are said to be statistically independent whenever the joint distribution functions factors such that

\[
F(x) = \prod_{i=1}^{n} F_i(x_i) \tag{A.2-13}
\]

or equivalently

\[
f(x) = \prod_{i=1}^{n} f_i(x_i) \tag{A.2-14}
\]

where $F_i$ is the distribution function and $f_i$ is the density function of the random variable $x_i$. Statistically independent random variables are independent in the sense that the value of one does not affect the value of another, and the definition is equivalent to the one given earlier for statistically independent random events.

A.3 Averages

Averages are of great practical importance. These statistics capture a random variable's pertinent characteristics: its average
value and quantitative assessments of randomness. Section A.3.1 discusses the mathematical operation of expectation, which is the means for computing averages. Specific averages (i.e., the moments) are discussed in Section A.3.2. Section A.3.3 discusses an important average used to assess the correlation between two random variables. Confidence limits are discussed in Section A.3.4.

### A.3.1 Expectation

If \( g(x) \) is any function of the random variable \( x \), the expectation (average value) of \( g(x) \) is denoted by \( E[g(x)] \). For a discrete random variable

\[
E[g(x)] = \sum_{i=1}^{n} g(x_i) f_i
\]

(A.3-1)

where \( f_i \) is the probability of occurrence of \( X_i \). In the case of a continuous random variable

\[
E[g(x)] = \int_{-\infty}^{\infty} g(x) f(x) \, dx
\]

(A.3-2)

where \( f \) is the density function. The development presented here considers only continuous random variables. However, it should be kept in mind that completely analogous formulas exist for all distribution types as illustrated by Eq. (A.3-1).

The expectation of a function of a random variable, as given by Eq. (A.3-2), is a weighted average of the function values with the weights determined by the distribution of the random variable. Expectation is a linear operator, which obeys

\[
E \left[ \sum_{i} a_i g_i(x_i) \right] = \sum_{i} a_i E[g_i(x_i)]
\]

(A.3-3)
where $a_i$ is any non-random number. The distributive law of multiplication,

$$\mathbb{E}\left[ \prod_i g_i(x_i) \right] = \prod_i \mathbb{E}[g_i(x_i)]$$  \hspace{1cm} (A.3-4)

holds only if the variables are statistically independent. The relation

$$\mathbb{E}[g(x)] = g(\mathbb{E}[x])$$  \hspace{1cm} (A.3-5)

is generally not true either.

The expectation of a vector valued function $g(x)$ is defined as

$$\mathbb{E}[g(x)] = \int_{-\infty}^{\infty} g(x) f(x) \, dx$$  \hspace{1cm} (A.3-6)

where $f(x)$ is the joint probability density function and integration is with respect to all variables. The expectation operator is a linear operator with respect to vector random variables.

**A.3.2 Moments**

The quantity $\mathbb{E}[x^k]$, is called the $k$th moment of $x$

$$\mathbb{E}[x^k] = \int_{-\infty}^{\infty} x^k f(x) \, dx$$  \hspace{1cm} (A.3-7)

In particular with $k = 1$, $\mathbb{E}[x] = \bar{x}$ is the mean value of the variable $x$. The quantity defined by $\mathbb{E}[(x-\bar{x})^k]$ is called the $k$th central moment of $x$. Moments and central moments are related. For example, the second central moment given as $\mathbb{E}[(x-\bar{x})^2]$ can be expressed in terms of the first two moments

$$\mathbb{E}[(x-\bar{x})^2] = \mathbb{E}[x^2] - \bar{x}^2$$  \hspace{1cm} (A.3-8)
by virtue of the fact that $E|x| = \bar{x}$ and the linear property of the expectation operator.

Several of the low order moments have practical interpretations. By definition, the first central moment is zero. The second central moment given as $E[(x-\bar{x})^2]$ is called the variance of $x$ and is denoted $\sigma^2$. It is a measure of dispersion of $x$ about the mean. The square root of the variance is called the standard deviation and denoted by $\sigma$.

Based on their frequency of general usage, the mean and variance are by far the most important moments. Occasionally, for non-zero mean random variables a coefficient of variation is defined

$$C_v = \frac{\sigma}{\bar{x}} \quad (A.3-9)$$

Of less frequent usage are the third and fourth central moments. The third central moment provides a measure of the asymmetry of the distribution about the mean value and is referred to as skewness. The fourth central moment is referred to as kurtosis and provides an additional measure of the clustering of the distribution about its mean value.

### A.3.3 Correlation

Let $x$ and $y$ be random variables with mean values $\bar{x}$ and $\bar{y}$. The expectation of the product

$$E[(x-\bar{x})(y-\bar{y})] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x-\bar{x})(y-\bar{y}) f(x,y) \, dx \, dy \quad (A.3-10)$$

is called the covariance of $x$ and $y$, $\text{Cov}(x, y)$. If $x$ and $y$ are statistically independent, in accordance with (A.2-4), then

$$\text{Cov}(x, y) = E[x-\bar{x}]E[y-\bar{y}] = 0 \quad (A.3-11)$$

The converse is not true. That is, in general, the vanishing of the covariance is not sufficient to insure statistical independence. Whenever the covariance is zero, the variables are said to be "uncorrelated."
Frequently a correlation coefficient is used rather than the covariance. The correlation coefficient is defined as

\[ \rho_{xy} = \frac{\text{Cov}(x, y)}{\sigma_x \sigma_y} \]  

(A.3-12)

where the \( \sigma_x \) and \( \sigma_y \) are the standard deviations of \( x \) and \( y \). It can be shown that the correlation coefficient always has values in the range \(-1\) to \(1\). Again, a non-zero value of the correlation coefficient implies statistical dependence but a vanishing correlation coefficient does not imply statical independence.

The interrelationship between two random phenomena, \( x \) and \( y \), is describable to first order by the correlation coefficient, \( \rho_{xy} \). The correlation coefficient is most clearly understood by an example. One method of assessing the presence of correlation between phenomenon \( y \) and phenomenon \( x \) would be to plot \( y \) versus \( x \). Any clustering of data points in the resulting "scatter diagram" would indicate correlation. The correlation can be quantified by fitting a straight line

\[ y = ax \]  

(A.3-13)

to the data (assuming zero means) using least squares techniques to determine \( a \). The result would be the best linear prediction formula for \( y \) given \( x \). This formula is referred to as the regression line. The regression line formula can be written in terms of the standard deviations and correlation coefficient

\[ \frac{\hat{y}}{\sigma_y} = \rho_{xy} \frac{x}{\sigma_x} \]  

(A.3-14)

Equation (A.3-14) shows that the correlation coefficient is the slope of the best-fit straight line when the data has been normalized with respect to the standard deviations. Since both \( x \) and \( y \) are random, the data will be scattered about the regression line. The scatter can be quantified by calculating the root mean square error

\[ \text{rms} = \sqrt{\sum_{i=1}^{n} [y_i - \hat{y}(x_i)]^2} \]  

(A.3-15)
where $i$ denotes the individual data points. The rms error can be shown to be

$$rms = \sqrt{1 - \rho_{xy}^2} \sigma_y$$  \hspace{1cm} (A.3-16)

Thus the correlation coefficient is a measure of the scatter about the regression line. If $\rho_{xy}$ is zero, the rms error is equal to the uncertainty in the data itself, $\sigma_y$. Thus uncorrelated random variables (i.e., $\rho_{xy} = 0$) are completely random relative to each other and no first order interrelationship exists. Since the rms error must be positive, Eq. (A.3-16) requires

$$-1 \leq \rho_{xy} \leq 1$$  \hspace{1cm} (A.3-17)

If $\rho_{xy} = \pm 1$ then the rms error is zero, all the data falls on the regression line, $y$ depends linearly on $x$, and $y$ is not random relative to $x$.

A.3.4 Confidence Limits

Confidence limits bound the uncertainty in the value of a random variable. The limits are selected to insure the random variable will fall inside, with a certain level of confidence. Since a random variable $x$ is expected to take on its average value $\bar{x}$, the $K$-confidence limit, $a_k$, is defined as the value for which

$$P(|x - \bar{x}| \leq a_k) = K$$  \hspace{1cm} (A.3-18)

This defines $a_k$ in terms of $K$ and the distribution. The random variable will fall inside the interval

$$\bar{x} - a_k \leq x \leq \bar{x} + a_k$$  \hspace{1cm} (A.3-19)

with a relative frequency of $K$. 

-153-
The Chebyshev inequality

\[
P[|x - \bar{x}| \leq h \sigma] \geq 1 - \frac{1}{h^2}
\]  

(A.3-20)

is plotted in Figure A-2 along with other representative distribution types. It can be seen to be a conservative lower bound. Substituting Eq. (A.3-20) into (A.3-18) and solving for the confidence limit yields

\[
a_k \leq \frac{\sigma}{\sqrt{1 - \alpha}}
\]  

(A.3-21)

Thus a random variable always has a value within $2\sigma$ of its mean 75% of the time and within $3\sigma$ of its mean 89% of the time. This illustrates that the standard deviation, $\sigma$, quantifies randomness regardless of the distribution.

A.4 Typical Distributions

It is generally difficult to precisely determine the distribution of a random variable. However, it is usually possible to assess characteristic properties which make it possible to select a distribution model. Presented here are four commonly used distribution types.

A.4.1 Uniformly Distributed Random Variable (TYPE = 3)

A uniformly distributed random variable is equally likely to take on any value in the interval $\bar{x} - T$ to $\bar{x} + T$. The density function is

\[
f(x) = \begin{cases} 
0 & |x - \bar{x}| > T \\
\frac{1}{2T} & |x - \bar{x}| \leq T
\end{cases}
\]  

(A.4-1)
Uniform Distribution

Gaussian Distribution

Chebyshev's lower bound

FIGURE A-2  CHEBYSHEV'S LOWER BOUND
The mean value is \( \bar{x} \); \( T \) is referred to as the tolerance, and

\[ \sigma = \frac{T}{\sqrt{3}} \]  
(A.4-2)

A uniform distribution is characteristic of error sources whose magnitude is limited by quality control procedures. Projectile length is a good example. Uncertainties in projectile length can be easily controlled by measuring each projectile after final machining. Those exceeding quality control limits would be discarded. Thus, off-nominal lengths beyond these limits do not occur. Within these limits, the length is equally likely to be any value.

A.4.2 Gaussian Distributed Random Variable (TYPE = 2)

A Gaussian distributed random variable is one whose distribution about the mean has the familiar bell shape. The density function is

\[ f(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{1}{2} \left( \frac{x - \bar{x}}{\sigma} \right)^2} \]  
(A.4-3)

where \( \bar{x} \) is the mean value and \( \sigma \) is the standard deviation. Figure A-3 illustrates the Gaussian curve. Since the interval \( \bar{x} \pm 3\sigma \) contains better than 99 percent of the outcomes, the tolerance \( T \) is defined as

\[ T = 3\sigma \]  
(A.4-4)

The effect of a change in value of \( \sigma \) is indicated in Figure A-4. Increasing \( \sigma \) effects a simultaneous lowering of the peak value and a spreading of the tails of the curve. Changes in the mean value \( \bar{x} \) results in a translation of the curve.

The Gaussian form is particularly interesting when sums of random variables are considered. For example, suppose there is a linear functional relation

\[ z = ax + by \]  
(A.4-5)

where \( a, b \) are constants and \( x, y \) are random variables. The
FIGURE A-3  GAUSSIAN DENSITY FUNCTION
FIGURE A-4  A COMPARISON OF TWO GAUSSIAN DENSITY FUNCTIONS WITH DIFFERENT STANDARD DEVIATIONS
variable $z$, by virtue of being a function of random variables, is also a random variable. If $x$ and $y$ are independent then one can conclude that

$$z = ax + by$$

$$\sigma_z^2 = a^2 \sigma_x^2 + b^2 \sigma_y^2$$  \hspace{1cm} (A.4-6)

by application of the principles previously presented. Now in general, the knowledge of the mean value of $z$ and its variance are insufficient to infer the distribution of $z$. For example, Figure A-5 illustrates two different density functions which have the same mean and variance. However, it can be shown that if $x$ and $y$ are Gaussian distributed then $z$ will also be Gaussian. This result can be extended to linear combinations of any number of Gaussian random variables.

The Central Limit Theorem states that under suitable conditions the sum of arbitrarily distributed independent random variables will become Gaussian as the number of variables becomes large. The necessary condition is that no single term of the sum can dominate. That is, the variance of any one term must not be of the same order as the sum of the variance of the other terms. For instance, the sum of twenty random variables, which are uniformly distributed between 0 and 1,

$$z = \frac{1}{20} \sum_{i=1}^{20} x_i$$  \hspace{1cm} (A.4-7)

is very nearly Gaussian.

In summary, a Gaussian uncertainty is one whose distribution about the mean has the familiar bell shape. It is appropriate for effects formed from a large number of independent random occurrences. For instance, variations in muzzle velocity are the result of a large number of independent effects occurring while the projectile traverses the barrel. Thus, the distribution of muzzle velocity would be expected to be Gaussian.

A.4.3 Jointly Gaussian Random Variables

Let $x$ and $y$ be jointly distributed Gaussian random variables. Their joint density function is
FIGURE A-5  TWO PROBABILITY DENSITY FUNCTIONS WITH IDENTICAL MEANS AND VARIANCES
\[ f(x,y) = \frac{1}{2\pi \sigma_x \sigma_y (1-\rho_{xy}^2)^{1/2}} \exp \left[ -\frac{(x-\bar{x})^2 - 2 \rho_{xy} \bar{x} \bar{y} + \bar{y}^2}{2(1-\rho_{xy}^2)} \right] \] (A.4-8)

where

\[ \bar{x} = \frac{x - \bar{x}}{\sigma_x} \] (A.4-9)

\[ \bar{y} = \frac{y - \bar{y}}{\sigma_y} \] (A.4-10)

and \( \bar{x} \) and \( \bar{y} \) are the mean values of \( x \) and \( y \), \( \sigma_x \) and \( \sigma_y \) are the standard deviations of \( x \) and \( y \), and \( \rho_{xy} \) is the correlation coefficient of \( x \) and \( y \). Considered separately, \( x \) and \( y \) are Gaussian distributed. If the correlation coefficient is zero, the density function factors and \( x \) and \( y \) are statistically independent. Thus, uncorrelated Gaussian random variables are statistically independent.

Six indices are commonly used to summarize the randomness of two jointly distributed random variables. The standard deviation, \( \sigma \), has already been described. There is a 50% probability the point \((x, y)\) will lie between two parallel lines which are equi-distant from the origin and are separated by twice the Linear Error Probable (LEP). The Circular Error Probable (CEP) is the radius of the circle with a 50% probability of occurrence,

\[ P\left[ \sqrt{x^2 + y^2} \leq \text{CEP} \right] = 0.5 \] (A.4-11)

The radius of the 80% circle is denoted \( R_{80} \). The mean radius \( \bar{J} \) is the average displacement of the point \((x, y)\) from the origin

\[ J = E \left[ \sqrt{x^2 + y^2} \right] \] (A.4-12)
(If \(x\) and \(y\) are the cross range deflections expressed as a fraction of the range, \(J\) is referred to as the "average jump angle"). The Radial Standard Deviation (RSD) is the rms value of \(x\) and \(y\)

\[
\text{RSD} = \sqrt{\text{E}(x^2 + y^2)}
\]

If \(x\) and \(y\) are jointly distributed Gaussian random variables, a readily available text book\(^1\) gives a detailed development of the analytic expressions relating the CEP to \(\sigma_x\), \(\sigma_y\) and \(\rho_{xy}\). The CEP is closely approximated by

\[
\text{CEP} = 0.589 (\sigma_x + \sigma_y)
\]

when \(\rho_{xy}\) is zero and \(\sigma_x\) and \(\sigma_y\) differ by no more than 80% of the larger. Provided \(\sigma_x\) und \(\sigma_y = \sigma\) and \(\rho_{xy} = 0\), all four indices are proportional to each other as indicated in Figure A-6. Each of these statistics may be used to define a circle which have the probabilities of occurrence as shown in Table A-1.

The above discussion has dealt exclusively with two jointly distributed Gaussian random variables. It is possible to have an arbitrarily large number of jointly distributed Gaussian random variables: \(x_1, x_2, \ldots, x_n\). The joint density function is defined elsewhere.\(^2\) The distribution function is completely determined by the \(N\times N\) covariance matrix

\[
\text{Cov}(x) = \begin{bmatrix}
\sigma_{x_1}^2 & \text{Cov}(x_1, x_2) & \ldots \\
\text{Cov}(x_2, x_1) & \sigma_{x_2}^2 & \ldots \\
\vdots & \vdots & \ddots
\end{bmatrix}
\]


FIGURE A-6  GAUSSIAN SCALE FACTORS
Table A-1  Probabilities of Occurrence

<table>
<thead>
<tr>
<th>Circle Radius</th>
<th>Probability of Occurrence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEP</td>
<td>20.3</td>
</tr>
<tr>
<td>CEP</td>
<td>50.0</td>
</tr>
<tr>
<td>J</td>
<td>54.4</td>
</tr>
<tr>
<td>RSD</td>
<td>63.2</td>
</tr>
<tr>
<td>R80</td>
<td>80.0</td>
</tr>
</tbody>
</table>
The covariance matrix is symmetric, positive definite and completely determined by $N$ the standard deviations $\sigma_{X_i}$ and the correlation coefficients $\rho_{X_iX_j}$.

A.4.4 Rayleigh Distributed Random Variable

Let $J$ be a Rayleigh distributed random variable. The probability density function is

$$f(J) = \frac{n}{2} \frac{J}{\bar{J}^2} \exp\left[-\frac{n}{4} \left(\frac{J}{\bar{J}}\right)^2\right] \quad (A.4-16)$$

where $\bar{J}$ is the mean value of the random variable $J$. The distribution is completely specified by $\bar{J}$ and

$$\sigma_J = \sqrt{\frac{4}{n} - 1} \quad \bar{J} = 0.52272 \bar{J} \quad (A.4-17)$$

The Rayleigh distribution is interesting because of its relationship to two jointly distributed Gaussian random variables. If $x$ and $y$ are identically distributed (i.e., $\sigma_x = \sigma_y = \sigma$) independent (i.e., $\rho_{xy} = 0$) Gaussian random variables, the radius

$$J = \sqrt{x^2 + y^2} \quad (A.4-18)$$

is Rayleigh distributed with mean $\bar{J}$ (i.e., as determined by Figure A-5). The orientation angle

$$\theta = \tan^{-1} \frac{y}{x} \quad (A.4-19)$$

is uniformly distributed between $\pm \pi$ radians. Thus the Rayleigh/uniform distribution describes the Gaussian distribution in polar coordinates.
A.5 Random Variable Generation

In Monte Carlo analyses it is necessary to generate random variables with known distributions. It is desired to make this procedure as simple as possible to conserve running time, simplify conversion to other machines, and not overcomplicate the programming.

The HITS code contains a random number generator which provides uniformly distributed random numbers in the interval from zero to one. This section defines transformations which convert the output of the random number generator into one of three random variable types: uniformly distributed, Gaussian, or arbitrary (tabularly defined) distributed. It is convenient to concurrently describe the method if incrementing the histogram cell counters to record the distribution of the generated random sequence. For this discussion $N_c$ is defined to be the number of histogram cells. $R$ is the uniformly distributed number on the interval zero to one obtained from the random number generator.

A.5.1 Uniformly Distributed Variables (TYPE = 3)

For uniformly distributed variables it is assumed $\bar{x}$ is the mean value and $T$ is the tolerance. The upper and lower bounds defining the extent of the histogram are given by

$$x_l = \bar{x} - T$$

$$x_u = \bar{x} + T$$

The random variable

$$x = \bar{x} + (2R - 1)T$$

is uniformly distributed on between $x_l$ and $x_u$. The histogram cell number whose counter should be increased is

$$I = \text{Int}[N_c R] + 1$$

where $\text{Int}$ yields the largest integer which does not exceed the argument.
A.5.2 Gaussian Distributed Variables (TYPE = 2)

For the Gaussian distributed variables, it is assumed \( \bar{x} \) is the mean value and the tolerance \( T \) is \( 3\sigma \) where \( \sigma \) is the standard deviation. The upper and lower bounds for the histogram are

\[ x_l = \bar{x} - 3\sigma \]
\[ x_u = \bar{x} + 3\sigma \]  \hspace{1cm} (A.5-4)

so that the total range of the histograms is \( 6\sigma \).

A uniformly distributed random number \( R \) is converted into a Gaussian random number by the following procedure which uses the conversion table:

<table>
<thead>
<tr>
<th>( r_1 )</th>
<th>( r_2 )</th>
<th>( r_3 )</th>
<th>( r_4 )</th>
<th>( r_5 )</th>
<th>( r_6 )</th>
<th>( r_7 )</th>
<th>( r_8 )</th>
<th>( r_9 )</th>
<th>( r_{10} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.28</td>
<td>0.36</td>
<td>0.41</td>
<td>0.46</td>
<td>0.50</td>
<td>0.54</td>
<td>0.58</td>
<td>0.64</td>
<td>0.78</td>
</tr>
</tbody>
</table>

An index \( J \) is calculated according to

\[ J = \text{Int}[10 R] + 1 \]  \hspace{1cm} (A.5-5)

where the integer function \( \text{Int} \) has previously been defined. The Gaussian distributed number is then calculated from

\[ R_G = (10 R - J + 1)(r_j + 1 - r_j) + r_j \]  \hspace{1cm} (A.5-6)

\( R_G \) is Gaussian with a mean of \( \frac{1}{2} \) and a standard deviation of \( 1/\sqrt{6} \). The sample value, \( R_G \), is then scaled to form the sample of the desired Gaussian random variable

\[ x = \bar{x} + (2R_G - 1)T \]  \hspace{1cm} (A.5-7)

and the appropriate histogram cell number is given by

\[ I = \text{Int}[N_c R_G] + 1 \]  \hspace{1cm} (A.5-8)
A.5.3 Arbitrarily Distributed Variable (TYPE = 4)

For the arbitrarily distributed random variables, the probability density function is defined by a sequence of points \( x_i, f_i \). It is assumed that these values are tabulated with equal divisions of the \( x \) coordinate. Furthermore, the average value of the \( x \) coordinates is assumed equal to the mean value and occurs midway in the table, the probability of exceeding the \( x \) table is zero, and the maximum value for the distribution density function is known. That is

\[
\overline{x} = \frac{1}{2} (x_1 + x_k) \quad (A.5-9)
\]

\[
T = \frac{1}{2} (x_k - x_1) \quad (A.5-10)
\]

\[(f_i)_{\text{max}} \text{ is known} \quad (A.5-11)\]

Two uniformly distributed random numbers are generated and are designated \( R_1 \) and \( R_2 \). Using \( R_1 \), a tentative sample value is generated from the expression

\[
x = \overline{x} + (2R_1 - 1) T \quad (A.5-12)
\]

with an associated histogram interval number being given as

\[
I = \text{Int}[NcR_1] + 1 \quad (A.5-13)
\]

An index \( J \) is also determined from the first random number by the parallel expression

\[
J = \text{Int}[KR_1] + 1 \quad (A.5-14)
\]

where \( K \) is the number of tabulated points. The index \( J \) is an interpolative index which determines between which two values, \( x_J \) and \( x_{J+1} \), \( R_1 \) actually lies. Using this index the value of the probability density function is approximated by linear interpolation

\[
f(x) = f_j + \frac{(x-x_j)}{x_{j+1} - x_j} (f_{j+1} - f_j) \quad (A.5-15)
\]
The ratio of this value to the maximal value is calculated and compared with the second random number. If the relation,

\[
\frac{f}{(f_i)_{\text{max}}} \geq R_2
\]  

(A.5-16)

is satisfied then the sample value of the variable is accepted; if it is not, the sample value is rejected, two more random numbers are generated and the process repeated. This is performed until an acceptable pair of random numbers is found or a predetermined number of trials (20) have been performed in which case a system level error is generated and execution is suspended.

This method is called Von Newmann rejection sampling. The method will generate samples which emulate the given distribution function. However, it can be inefficient. Referring to Figure A-7, the area \(A_1\) represents the area under the distribution function in the transformed plane of the two random numbers. The method consists of accepting pairs of numbers which lie under the transformed distribution density curve.

![Figure A-7 Von Newmann Rejection Sampling Illustration](image)

The area \(A_2\) represents the region where rejection occurs. A sampling efficiency, \(\eta\), can be defined as the ratio of sample points accepted to the total number of trials. This efficiency is

\[
\eta = \frac{A_1}{A_1 + A_2} = A_1 = \frac{1}{(f_i)_{\text{max}} 2 T}
\]  

(A.5-17)
Typically, it can be expected that on the order of 50% of the samples will be rejected, which is an acceptable rate. For very peaked distributions this method could prove to be inefficient. A large number of arbitrarily distributed variables would compound the inefficiency. If this inefficiency proves to be a hinderance, other methods would be more appropriate and should be substituted.

A.6 Monte Carlo Inaccuracies

This section discusses two sources of error in Monte Carlo techniques: the grouping error and the sampling error. Guidelines are developed for minimizing these errors. In summary, these errors are small whenever the histogram cells are small and the sample size is large.

A.6.1 Grouping Errors

In a Monte Carlo analysis, the values resulting from the experiment are sorted into histogram cells and counted rather than recording the precise values. The purpose is to conserve computer storage. Subsequently, the histogram data is used to calculate moments. Since the exact values are lost in the process, an error, termed the grouping error, is introduced in the moments.

To illustrate the problem, consider the random variable x with density function f(x). During Monte Carlo experiments the observed values of x are grouped into cells which is tantamount to assuming that all observed values fell at the midpoint of the cell. This is an error. Thus, in reality the histogram does not correspond to the true distribution of x but rather to a discrete distribution where x can take only values associated with the midpoints of the cells. That is

\[ x = x_{Mi}, \quad i = 1, \ldots, N_c \]  

(A.6-1)

are the only possibilities where \( x_{Mi} \) are the midpoints. The probability associated with each of these values is

\[ P \left[ x_{bi} \leq x \leq x_{bi+1} \right] = \int_{x_{bi}}^{x_{bi+1}} f(x) \, dx \]  

(A.6-2)

where \( x_{bi} \) and \( x_{bi+1} \) are the lower and upper cell boundaries. The ensuing moment calculation based on the histogram is
As a prelude to evaluating the grouping error, consider the true \( n^{th} \) moment of \( x \) given by the integral

\[
\bar{x}^n = \int_{-\infty}^{\infty} x^n f(x) \, dx \quad (A.6-4)
\]

This integral can be subdivided into the contributions from each histogram cell to yield

\[
\bar{x}^n = \sum_{i=1}^{N_c} \int_{x_{b_i}}^{x_{b_{i+1}}} f(x) x^n \, dx \quad (A.6-5)
\]

where \( x_{b_i}, x_{b_{i+1}} \) are the boundary values of the cells and it is assumed

\[
f(x) = 0 \quad \text{if} \quad x < x_{b_0} \quad \text{or} \quad x > x_{b_{N_c}} \quad (A.6-6)
\]

The grouping error is the difference between the calculated moment, Eq. (A.6-3), and the true moment as given by Eq. (A.6-5). The exact magnitude of the grouping error in the \( n^{th} \) moment depends on the specific distribution. It cannot be evaluated in general unless the density function is approximated. A good first order assumption is all that is required: it is assumed that the density function, \( f(x) \), is constant over each histogram cell, i.e.,

\[
f(x) = f_i \quad x_{b_i} < x \leq x_{b_{i+1}} \quad (A.6-7)
\]

and each of the cells are assumed to be equal in size, \( \Delta x = x_{b_{i+1}} - x_{b_i} \). Thus, the midpoints of the cells are given by
The grouping error, \( \epsilon^N \), is the difference between the true and calculated moments of order \( N \)

\[
\epsilon^N = \bar{x}^N - x^N = \sum_{i=1}^{N_c} f_i \left[ \int_{x_{b_i}}^{x_{b_{i+1}}} x^n \, dx - x_{m_i}^n \, \Delta x \right] 
\]

With the aid of the identity

\[
\sum_{i=1}^{n} f_i \Delta x = \sum_{i=1}^{n} \int_{x_{b_i}}^{x_{b_{i+1}}} f(x) \, dx = 1 
\]

the grouping error can be evaluated for low order moments. In particular

\[
\epsilon^1 = 0 \\
\epsilon^2 = \frac{\Delta x^2}{12} 
\]

Thus, to first order, there is no grouping error in the calculation of the mean value; but, there is an error in the calculated variance equal to \( \Delta x^2/12 \), where \( \Delta x \) is the cell size.

The expressions for the grouping error given by Eq. (A.6-11) is quite accurate. A more general derivation with a more accurate approximation to the distribution function yields the same result. Corrections for grouping errors are called Sheppards corrections and are treated in greater detail in appropriate texts on statistics.¹ These corrections are not applied in the present Monte Carlo code because they can be made arbitrarily small. Since cell sizes are selected in proportion to the standard deviation of the variables, the percentage error due to grouping is of the

A random number generator seeks to generate random numbers which are uniformly distributed over the interval zero to one. Let a sequence of $N_T$ such numbers be sorted into $N_C$ histogram cells of equal size covering the range zero to one. It would be expected that each cell would occur $N_T/N_C$ times, or a relative frequency of $1/N_C$. Thus, for $N_C = 10$ and $N_T = 100$, 10 values should fall in each cell and the indicated probability of occurrence of any given cell would be $1/10$. However, when an actual sequence of random numbers is sorted it is generally found that neither of these theoretical expectations is true. This section discusses this problem. Only uniformly distributed random variables are discussed, although the results are generally true regardless of distribution type.

The above hypothetical situation can be explained with the aid of the binomial distribution. The binomial distribution states that if an event has a probability $p$ of occurring and a probability $q = 1 - p$ of not occurring, then the probability of exactly $X$ occurrences of the event in $N$ trials is

$$P(X) = \frac{N!}{X! (N - X)!} p^X q^{N-X} \quad (A.6-12)$$

For any particular cell, the probability that the generated random number will fall in that cell is $p = 1/N_C$, with the nonoccurrence probability being $q = 1 - 1/N_C$. According to Eq. (A.6-12), the probability any cell will occur exactly $N_T/N_C$ times in $N_T$
where it is assumed that \( NT/N_C \) is an integer quantity. With the assumption that the number of samples is large relative to the number of cells, \( NT \gg N_C \), Stirlings approximation

\[
N! = \sqrt{2\pi N} \ N^N e^{-N}
\]  
(A.6-14)

can be used to simplify Eq. (A.6-13):

\[
P\left( \frac{NT}{N_C} \right) = \frac{NT!}{\left( \frac{NT}{N_C} \right)! \left[ NT \left( 1 - \frac{NT}{N_C} \right) \right] \left( \frac{1}{N_C} \right)^{NT} \left( 1 - \frac{1}{N_C} \right)^{N_T} \left( 1 - \frac{1}{N_C} \right) \}
\]
(A.6-13)

\[
P\left( \frac{NT}{N_C} \right) = \sqrt{\frac{N_C}{2\pi NT\left( 1 - 1/N_C \right)}}
\]
(A.6-15)

Using this expression it is found that for a sequence of 100 random numbers sorted into 10 cells the probability any cell occurs exactly 10 times is about 1 chance in 8. Increasing the size of the sample to 1000 doesn't help, the probability of exactly 100 samples in a cell is about 1 chance in 24. Alternatively, sorting the sequence of 100 into 5 cells doesn't help either; the probability of any cell occurring exactly 20 times is about 1 chance in 10. This illustrates that as the sample size is increased, or the number of cells is decreased, the probability of observing the theoretically expected number of occurrences per cell diminishes.

It would appear there is no means for obtaining a more exact definition of the probability density function. This is not true, since the density function is more precisely defined in an average sense by larger sample sizes, as illustrated by the following example.

A sequence of 100 random numbers sorted into five cells has an expected number of occurrences of 20 per cell. A 5% deviation
in the actual occurrences per cell would be either 19, 20 or 21 in a cell. The associated probability is

\[ P(19) + P(20) + P(21) = 3P(20) = 0.3 \quad (A.6-16) \]

Increasing the sequence of random numbers tenfold, the expected number of occurrences is 200 per cell and a 5% deviation would span the twenty values 190, 191, ..., 210. The associated probability is

\[ \sum_{n=190}^{210} P(n) = 20P(200) = 0.6 \quad (A.6-17) \]

Thus, the chances of being within 5% of the theoretical are about twice as good with 1000 samples as they are for 100 samples. This example illustrates that the number of occurrences tends to the theoretical as the number of trials tends to infinity in an average sort of way. Thus, larger sample sizes improve the estimate of the density function. A general rule of thumb is contained in the old saw, "a few hundred is too few and a thousand is plenty."

**Estimated Moments**

Monte Carlo methods estimate the moments using Eq. (A.6-3) by approximating the density function with the observed relative frequency

\[ f(x) = \frac{N_i}{N_T} \frac{1}{x_{b_{i+1}} - x_{b_i}} \quad x_{b_i} \leq x \leq x_{b_{i+1}} \quad (A.6-18) \]

so that

\[ \int_{x_{b_i}}^{x_{b_{i+1}}} f(x) \, dx = \frac{N_i}{N_T} \quad (A.6-19) \]

where \( N_i \) is the number of occurrences of the \( i \)th cell, \( N_T \) is the total number samples, and \( x_{b_i} \) and \( x_{b_{i+1}} \) are the lower and upper bounds of the cell. Thus, errors in the estimated moments are the compound effect of grouping errors and density function uncertainties.
Section A.6.1 showed that grouping errors can be made arbitrarily small by using enough cells. This section treats the moment errors associated with uncertainties in the estimated density function. In order to evaluate the errors in the moments determined by Monte Carlo methods, let

\[ y = y(x) \]  \hspace{1cm} (A.6-20)

where \( x \) is the independent random variable and \( y \) is dependent. Since \( x \) is a random variable and \( y \) is a function of \( x \), \( y \) is also a random variable. Therefore, \( y \) has a density function \( f_y \) with moments \( \bar{y} \), the mean value, and \( \sigma_y^2 \), the variance, defined by

\[ \bar{y} = \int_{-\infty}^{\infty} y f(y) \, dy \]  \hspace{1cm} (A.6-21)

\[ \sigma_y^2 = \int_{-\infty}^{\infty} (y - \bar{y})^2 f(y) \, dy \]

The problem is to assess the errors in the mean value and variance computed from a finite Monte Carlo sequence.

In applying the Monte Carlo method a sequence of random values: \( x_1, x_2, \ldots, x_{NT} \) is generated and the resulting sample sequence \( y_1, y_2, \ldots, y_{NT} \) is computed. This latter sequence is a sequence of random numbers which in turn has a mean

\[ \mu_1 = \frac{1}{NT} \sum_{i=1}^{NT} y_i \]  \hspace{1cm} (A.6-22)

If a second sample is generated using a different sequence of random numbers, the second sample will have a mean given as

\[ \mu_2 = \frac{1}{NT} \sum_{i=1}^{NT} y_i \]  \hspace{1cm} (A.6-23)

which in general will be different from the first. If the process is repeated many times then a set of values, \( \mu_1, \mu_2, \ldots, \mu_J \)
will result. This set of values has a distribution function and a mean value and variance associated with it. The question arises then as whether the important properties of the sample mean distribution can be determined without actually determining the sampling distribution itself. This would provide the necessary guidelines for minimizing the sample errors in the moments. This can be done. The sample mean

\[ \mu = \frac{1}{N_T} \sum_{i=1}^{N_T} y_i \]  \hspace{1cm} (A.6-24)

is a function of \( N_T \) statistically independent variables; \( y_i, i = 1, \ldots, N \). Forming the expectation of Eq. (A.6-24) gives the mean value of the sample mean

\[ E[\mu] = \frac{1}{N_T} \sum_{i=1}^{N_T} E[y_i] \]  \hspace{1cm} (A.6-25)

which, according to Eq. (A.6-21), is

\[ E[\mu] = \bar{y} \]  \hspace{1cm} (A.6-26)

Therefore, the mean value of the sample mean is equal to the mean. The variance of the sample mean is found similarly. Thus noting

\[ \mu - \bar{y} = \frac{1}{N_T} \sum_{i=1}^{N_T} (y_i - \bar{y}) \]  \hspace{1cm} (A.6-27)

squaring both sides and forming the expectation

\[ \text{Var}(\mu) = E[(\mu - \bar{y})^2] = \frac{1}{N_T} E \left[ \sum_{i=1}^{N_T} (y_i - \bar{y})^2 \right] \]  \hspace{1cm} (A.6-28)

\[ \text{Var}(\mu) = \frac{1}{N_T^2} \sum_{i=1}^{N_T} E[(y_i - \bar{y})^2] \]  \hspace{1cm} (A.6-29)
Since $E (y_i - y)^2 = \sigma_y^2$ the variance of the sample mean is

$$\text{Var}(\mu) = \frac{\sigma_y^2}{N_T} \quad \text{(A.6-30)}$$

Thus for a sample size of $N_T$ of a random variable $y$, the standard deviation of the sample mean is $\sigma_y/\sqrt{N_T}$ where $\sigma_y$ is the standard deviation of $y$ and $N_T$ is the sample size. On the basis of a single Monte Carlo sample of size $N_T$, one would report the estimate of the mean value $\bar{y}$ as

$$\bar{y} = \mu \pm \frac{\sigma_y}{\sqrt{N_T}} \quad \text{(A.6-31)}$$

The $\sigma_y/\sqrt{N_T}$ is termed a sampling error for the mean. To reduce the sampling error by half requires a quadruple increase in the sample size. By the central limit theorem, for a large number of trials the distribution of the sample mean will tend to be Gaussian. It is therefore possible to associate confidence limits with it. For example, the true value of $\bar{y}$ should lie within $\mu \pm 3 \frac{\sigma_y}{\sqrt{N_T}}$ with 95% confidence.

Just as there is a distribution of the sample mean, there is also a distribution associated with the sample variance

$$S^2 = \frac{1}{N_T - 1} \sum_{i=1}^{N_T} (y_i - \mu)^2 \quad \text{(A.6-32)}$$

The expected value of the sample variance is

$$E[S^2] = \sigma_y^2 \quad \text{(A.6-33)}$$

The $(N_T - 1)$ factor in Eq. (A.6-32) is included so the sample variance is unbiased; that is, without an expected error. The standard deviation of the sample variance is

$$\sigma_{S^2} = \sigma_y \sqrt{\frac{2}{N_T - 1}} \quad \text{(A.6-34)}$$
To summarize, the moments computed from a Monte Carlo sample of size $N_T$ have the following characteristics

\[
\bar{y} = \mu = \frac{\sigma_y}{\sqrt{N_T}} \tag{A.6-35}
\]

\[
\sigma^2 = s^2 = \frac{\sigma_y^2}{N_T - 1} \tag{A.6-36}
\]

where $\mu$ and $s^2$ are the sample mean and sample variance, respectively. The HITS code calculates $\mu$ and $s^2$ for each of the variables of interest. The program does not determine the sampling errors involved. The sampling errors are plotted in Figures A-8 and A-9 for various confidence levels. Clearly, samples less than several hundred will produce moments with substantial errors. Samples exceeding a thousand provide adequate accuracy.

While the discussion of the sampling errors in the moments has been based on a single variable, the basic conclusions are applicable to the multiple variable case as long as the variables are statistically independent.
Figure A-8: Expected Error in Sample Mean in Percentage of Standard Deviation

Confidence Interval
B.1.1 Analytical Statistical Calculations

The theoretical dispersion assessment problem is to statistically determine the effect of independent variables on a dependent variable. For instance, the effect of the projectile error source model (independent variables) on the down-range dispersion (dependent variable). In general, an exact theoretical solution is not possible. However, the equations can be accurately approximated by a low order Taylor series for reasonable variations in the independent variables, and the mean value and variance of the dependent variable can be estimated. The Analytical Statistical mode mechanizes this concept.
Figure B-1  HITS Flow Diagram
Taylor Series Approximation

The relationship between the dependent variable, \( y \), and the independent variables, \( x_1, x_2, \ldots, x_n \), is representable as an algebraic equation

\[
y(x) = y(x_1, x_2, \ldots, x_n)
\]  

where \( x_1, \ldots, x_n \) are treated as the elements of a column vector \( \mathbf{x} \). In general, the equation is non-linear. With the assumption that the function \( y \) is differentiable and reasonably well-behaved, it can be accurately approximated by a second order Taylor series about the mean value of the independent variables, \( \overline{\mathbf{x}} = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n})^T \).

\[
y(x) = y(\overline{x}) + \frac{\partial y}{\partial \mathbf{x}} (\overline{x})(\mathbf{x} - \overline{x}) + \\
+ \frac{1}{2} (\mathbf{x} - \overline{x})^T \frac{\partial}{\partial \mathbf{x}} \left( \frac{\partial y}{\partial \mathbf{x}} \right)^T (\overline{x})(\mathbf{x} - \overline{x})
\]  

where \( T \) denotes the matrix transpose and

\[
\frac{\partial y}{\partial \mathbf{x}} (\overline{x}) = \left[ \frac{\partial y}{\partial x_1} (\overline{x}), \frac{\partial y}{\partial x_2} (\overline{x}), \ldots, \frac{\partial y}{\partial x_n} (\overline{x}) \right]
\]  

is the gradient row vector of first partial derivatives

---

and

\[
\frac{\partial}{\partial \bar{x}} \left( \frac{\partial y}{\partial \bar{x}} \right)^T \mathbf{z} =
\begin{bmatrix}
\frac{\partial^2 y}{\partial x_1^2} (\bar{z}) & \frac{\partial^2 y}{\partial x_1 \partial x_2} (\bar{z}) & \cdots & \frac{\partial^2 y}{\partial x_1 \partial x_n} (\bar{z}) \\
\frac{\partial^2 y}{\partial x_2 \partial x_1} (\bar{z}) & \frac{\partial^2 y}{\partial x_2^2} (\bar{z}) & \cdots & \frac{\partial^2 y}{\partial x_2 \partial x_n} (\bar{z}) \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 y}{\partial x_n \partial x_1} (\bar{z}) & \frac{\partial^2 y}{\partial x_n \partial x_2} (\bar{z}) & \cdots & \frac{\partial^2 y}{\partial x_n^2} (\bar{z})
\end{bmatrix}
\quad (B.1-4)
\]

is the Jacobian Matrix of second partial derivatives, both evaluated at the mean value of the independent variables, \( \bar{x} \).

**Dependent Variable Mean Value**

The mean value is found by taking the expectation of Eq. (B.1-2). After some manipulation, the mean value of the dependent variable can be shown to be

\[
\bar{y} = y(\bar{x}) + \frac{1}{2} \sum_{i=1}^{n} \frac{\partial^2 y}{\partial x_i^2} (\bar{z}) \sigma_{x_i}^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{\partial^2 y}{\partial x_i \partial x_j} (\bar{z}) \text{Cov}(x_i, x_j)
\quad (B.1-5)
\]

This equation states that \( \bar{y} \) is composed of the value of \( y \) at the mean of \( x \) plus second order corrections based on the variances and covariances of the independent variables. (Note: HITS includes the second term whenever the control variable IOPRNT \( \leq 3 \) and the third whenever IOPRNT \( \geq 5 \).)
Dependent Variable Variance

The variance of the dependent variable is computed according to

$$\sigma_y^2 = \mathbb{E}[y^2] - \gamma^2$$

Equations (B.1-2) and (B.1-5) are substituted and manipulated to obtain

$$\sigma_y^2 = \sum_{i=1}^{n} \left( \frac{\partial y}{\partial x_i} (\bar{x}) \right)^2 \sigma_{x_i}^2 +$$

$$+ 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{\partial y}{\partial x_i} (\bar{x}) \frac{\partial y}{\partial x_j} (\bar{x}) \text{Cov}(x_i, x_j) +$$

$$+ \frac{1}{2} \sum_{i=1}^{n} \left[ \frac{\partial^2 y}{\partial x_i^2} (\bar{x}) \right]^2 \sigma_{x_i}^4 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left[ \frac{\partial^2 y}{\partial x_i \partial x_j} (\bar{x}) \right]^2 \sigma_{x_i}^2 \sigma_{x_j}^2 +$$

Equation (B.1-7) states that \( \sigma_y^2 \) is composed of linear terms based on the variance and covariances of the independent variables and a profuse number of second order corrections. (Note: HITS includes the first term whenever IOPRNT \( \geq 3 \) and the second whenever IOPRNT \( \geq 5 \).) The second order corrections are exactly correct only when the independent variables are Gaussian, since the development of Eq. (B.1-7) assumed
Calculation of Derivatives

In order to estimate the mean and variance via Eqs. (B.1-5) and (B.1-7), it is necessary to evaluate the dependent variable and its derivatives at the mean values of the independent variables. The evaluation of \( y(\bar{x}) \) requires one reference to the projectile trajectory module. HITS determines the derivatives by manipulating the inputs to the projectile trajectory module in a systematic fashion. A central difference scheme is used to numerically approximate the derivatives. The independent variables are incremented one at a time both positively and negatively about their mean values. Letting \( \bar{x}_i \) represent the independent variable mean values and \( \Delta x_i \) a positive increment, two function evaluations, (i.e., calls to the projectile trajectory module), are performed to give

\[
y^+ = y(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_i + \Delta x_i, \ldots, x_n) \tag{B.1-9}
\]

\[
y^- = y(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_i - \Delta x_i, \ldots, x_n) \tag{B.1-9}
\]

(Note: HITS sets \( \Delta x_i \) equal to the input tolerance value TOL for each independent variable.) The derivatives are then calculated according to

\[
\frac{dy}{dx_i} = \frac{y^+ - y^-}{2 \Delta x_i} \tag{B.1-10}
\]
\[
\frac{\partial^2 y}{\partial x_i^2} (\overline{x}) = \frac{y'' - 2y(\overline{x}) + y'}{\Delta x_i^2} \quad \text{(B.1-11)}
\]

For the second order mixed derivatives, it is necessary to evaluate the function \( y \) four times for each pair of independent variables. The four function evaluations for the pair \( x_i \) and \( x_j \) are

\[
y^{++} = y \left( \overline{x}_1, \ldots, \overline{x}_i + \frac{\Delta x_i}{2}, \ldots, \overline{x}_j + \frac{\Delta x_j}{2}, \ldots, \overline{x}_n \right)
\]
\[
y^{+-} = y \left( \overline{x}_1, \ldots, \overline{x}_i + \frac{\Delta x_i}{2}, \ldots, \overline{x}_j - \frac{\Delta x_j}{2}, \ldots, \overline{x}_n \right)
\]
\[
y^{-+} = y \left( \overline{x}_1, \ldots, \overline{x}_i - \frac{\Delta x_i}{2}, \ldots, \overline{x}_j + \frac{\Delta x_j}{2}, \ldots, \overline{x}_n \right)
\]
\[
y^{--} = y \left( \overline{x}_1, \ldots, \overline{x}_i - \frac{\Delta x_i}{2}, \ldots, \overline{x}_j - \frac{\Delta x_j}{2}, \ldots, \overline{x}_n \right) \quad \text{(B.1-12)}
\]

The second mixed partial derivative is

\[
\frac{\partial^2 y}{\partial x_i \partial x_j} (\overline{x}) = \frac{y^{++} + y^{--} - (y^{+-} + y^{-+})}{\Delta x_i \Delta x_j} \quad \text{(B.1-13)}
\]

Since the second order mixed partial derivatives are symmetric with respect to the independent variables, there are \( n(n-1) \) distinct derivatives for any dependent variable, where \( n \) is the number of independent variables.

-189-
The increments used in the second order mixed partial derivative calculations are half the values used for the first order derivatives. The reason is illustrated in Figure B-2 which diagrams points in the \( x_i - x_j \) plane. An ellipse can be drawn through the four points used for the first derivative calculation. With regards to the assumed Taylor series approximation, the ellipse represents the boundary of the region over which the linearization is valid, by virtue of the selection of the \( \Delta x \)'s. For consistency, the off-axis coordinates are halved to insure they will always be interior to the ellipse.

**Summary**

Equations (B.1-5) and (B.1-7) are the basic equations for the Analytical Statistical mode of the Statistical Processor. If there is more than one dependent variable the equations are applied to each in turn. The objective is to determine the mean and variance of all dependent variables from the input statistics of the independent variables. This forces the calculation of first and second order partial derivatives, which is a major undertaking. Since the Analytical Statistical mode is exercised prior to the Monte Carlo mode (to set up the histograms), these calculations are pertinent to the Monte Carlo simulations.

**B.1.2 Monte Carlo Calculations**

The Monte Carlo mode consists of three parts: (1) generation of representative independent variable values, (2) processing of these values to determine the corresponding dependent variable values, and (3) condensing the ensemble of solutions to a usable form. Representative sequences of the independent variables are obtained from random number generators, as discussed in Appendix A. The trajectory equations of Appendix C relate the dependent variables to the independent. This section discusses the third point: data condensation using histograms.

**Cell Definition**

In order to achieve substantial condensation, the histograms must be constructed while performing the Monte Carlo experiments. In order to do this, the boundaries of the cells must be determined prior to the experiments. They are determined by an equal division of the range of the variable as defined by the upper bound, \( x_u \), and the lower bound, \( x_l \). The \( i \)th cell is \( (x_{bi}, x_{bi+1}) \), where the boundaries are given by
Points for evaluation of $\frac{\partial^2}{\partial \chi_i \partial \chi_j}$

Figure B-2 Coordinates for Derivative Evaluations
\[ x_{b_i} = x_\ell + (i - 1) \frac{x_u - x_\ell}{N_c} \]  

(B.1-14)

where \( N_c \) is the number of cells. For independent variables, \( x_1 \) and \( x_u \) are defined in Appendix A.5 for the various different types. For dependent variables, (TYPE = 7), \( x_1 \) and \( x_u \) are determined by the preliminary Analytical Statistical mode calculations. They are

\[ x_\ell = \bar{x} - 3\sigma \]
\[ x_u = \bar{x} + 3\sigma \]  

(B.1-15)

where \( \bar{x} \) and \( \sigma \) are the calculated values of the mean value and standard deviation. An alternate set of values may be input (TYPE = 8). The cell number for a given value, \( x \), is determined by evaluating

\[ i = \text{Int} \left[ \frac{x - x_\ell}{x_u - x_\ell} N_c \right] + 1 \]  

(B.1-16)

where \( \text{Int} \) is the largest integer less than the argument. The midpoint in each cell is termed the "class mark" and is denoted \( x_{m_i} \):

\[ x_{m_i} = \frac{1}{2} (x_{b_i} + x_{b_{i+1}}) = x_\ell + \left( i \frac{1}{2} - \frac{1}{2} \right) \frac{x_u - x_\ell}{N_c} \]  

(B.1-17)

Density Function Estimate

The Monte Carlo mode counts the number of times each cell occurs during the experimental sequence. The probability density function is estimated according to

\[ f(x) = \frac{N_i}{N_T} \frac{N_c}{x_u - x_\ell} \quad \text{for} \quad x_{b_i} \leq x \leq x_{b_{i+1}} \]  

(B.1-18)
where \( N_i \) is the number of times cell "i" occurred and
\[
N_T^* = \sum_{i=1}^{N_c} N_i
\]
(B.1-19)
is the total of all cell counts for this variable. Since it is possible for dependent variables to fall outside the range covered by the histogram cells, \((x_1, x_u)\); \( N_T^* \) is potentially less than the total number of experiments, \( N_T \).

**Moment Estimates**

The sample mean and standard deviation are estimated from the histograms. The mean value is
\[
\bar{x} = \int_{-\infty}^{\infty} x f(x) \, dx = \sum_{i=1}^{N_c} \int_{x_{b_i}}^{x_{i+1}} x f(x) \, dx
\]
(B.1-20)
which becomes the sample mean upon substitution of Eq. (B.1-18),
\[
\mu = \sum_{i=1}^{N_c} \frac{N_i}{N_T^*} x_{m_i}
\]
(B.1-21)

Using Eq. (B.1-17) the sample mean is
\[
\mu = x_\ell + \frac{x_u - x_\ell}{N_T^* N_c} \sum_{i=1}^{N_c} \left( i - \frac{1}{2} \right) N_i
\]
(B.1-22)
The variance is defined by
\[
\sigma^2 = \int_{-\infty}^{\infty} (x - \bar{x})^2 f(x) \, dx
\]
(B.1-23)
Upon substitution of Eq. (B.1-18), and using the sample mean instead of the mean value, Eq. (B.1-23) becomes an expression for the sample variance:

\[ s^2 = \left( \frac{x_n - x_p}{N_c} \right)^2 \left[ \sum_{i=1}^{N_c} \left( i - \frac{1}{2} \right)^2 \frac{N_i}{N_T} - \left( \sum_{i=1}^{N_c} \left( i - \frac{1}{2} \right) \frac{N_i}{N_T} \right)^2 \right] \]  

(B.1-24)

B.2 Code Architecture

This section discusses the architecture of the HITS code. The architecture was designed to meet the requirements of the Statistical Processor computational procedures just described. The four storage arrays discussed in Section B.2.1 form the basic structure. The input processor of Figure B-1 plays the intimate role described in B.2.2. Statistical Processor functions are presented in B.2.3. Subroutine definitions and linking are detailed in B.2.4. The facility for computer generated histogram plots is discussed in B.2.5.

B.2.1 Data Storage

The entire code is designed around four arrays in common storage. These arrays are

1) "OE" array - This one-dimensional array is a common input-output storage area to be shared by the projectile trajectory module and the Statistical Processor. That is, all subroutines are written with all their variables equivalenced to the OE array. The variable "code numbers" are addresses in the OE array.

2) "IA" array - This bicolumned array is used to store address links between the "OE" array and the additional storage areas of the B and C arrays described next. The first 10 rows of the IA array are reserved for dependent variables.

3) "B" array - This one-dimensional array is used to store most input information and some additional values.

4) "C" array - This one-dimensional array is used to store some input information and intermediate output information. It is a general scratch pad storage for the Statistical Processor.
The data transfers between the arrays and the computational modules that effect these transfers are indicated in Figure B-3.

B.2.2 Input Processor Functions

The Input Processor is a collection of subroutines. It determines the kind of calculations to be performed, the subroutines to be used, and the data required. The Input Processor reads the data, checks it for completeness and takes appropriate action if errors are detected or data is found to be missing. If a complete set of data has been input, control is relinquished to the Statistical Processor. This section discusses the functions of the Input Processor in detail.

The Input Processor determines which trajectory module is going to be exercised (at present, there is only the one described in Appendix C, which is specified by $I_Y$ = ). It clears the IA array, and stores a list of code numbers into the first column of the IA array starting at row 11. This is a complete list of all the addresses in the GE array for which data is required. The first 10 rows of IA are reserved for variables in the OE array which are defined by the input to be dependent variables. The OE array is initialized at preset (i.e., default) values.

The input cards are read and processed one at a time. Generally, the card contains a code number and five values as described in the text. The five values are placed in the B array with the aid of a counter. The first column of the IA array is searched to find the row containing the code number. When it has been located, the current value of the counter is stored in the second column, and the five values are placed in the corresponding location in the B array. Subsequently, the counter is advanced by six and the next data card is read and processed in a like fashion. The input proceeds until a card having a negative code number is encountered which signifies that all data has been read in. After all the data cards have been read, each row of the IA array has two addresses, the first is a location in the OE array, (i.e., the code number), and the second is a location in the B array where data is stored. Input data is extracted by searching the first column of the IA array, and then using the second column to vector into the B array. Thus, the order of the data cards is immaterial. Note: if any rows of the IA array are interchanged the information is not disturbed.
Figure B-3 Internal Data Transfers
If necessary data is missing, zeros will appear in the second column of the IA array. The second column of IA is checked for zeros with appropriate action taken whenever a zero is encountered. At present, a preset value is substituted. However, HITS has the facilities to access a "Time Phased Data Base." This option could be activated if real-time interactive terminals are available.

This description of the Input Processor is valid though over simplified. Several additions and/or modifications are necessary:

1) At the time each data card is read and processed, the variable value on the card is stored in both the OE array and B array. This saves a separate pass through the data to initialize the OE array.

2) For missing data, a "Time Phased Data Base" would be treated like a second source of data. That is, when the card reader is exhausted and data is found missing, the time phase data base is assumed to be implemented as a subroutine which can provide the missing data. This logic is present in the program, but for the present all missing data is filled in with preset values.

3) Dependent variable (TYPE = 7 or 8) code numbers are stored sequentially in the first 10 rows of the IA array. For a TYPE = 7, nothing is stored in the B array, and the second column of the IA array is not disturbed. For a TYPE = 8, the data is stored as previously described.

4) For a constant input value (TYPE = 5), a "$-1$" is inserted in the second column of the proper row of the IA array to signify that a value was read in. The value is then placed in the OE array.

5) A TYPE = 1 variable has a list of values associated with it. The list is stored in the C array. The TYPE, number of values, and the starting address for the list in the C array are stored sequentially in the B array. The appropriate index of the B array is stored in the IA array, in the appropriate row of the second column.
Since no nominal value is defined for this kind of variable, the first comment does not apply.

6) **TYPE = 4** variables have a two-dimensional list associated with them. They are handled like a **TYPE = 1** variable (see comment five). However, after read in, a nominal value, tolerance, variance, etc. are calculated and these values are stored in the B array as though they had been read in.

7) Inputs to the projectile trajectory module can be either floating point or fixed point. All input values are read in floating point format. To signify a value has to be stored in the OE array in an integer format, the negative of its code number is stored in the first column of the IA array. The code number is supplied by the Input Processor at initialization and therefore, there is no apparent difference to the user. Also note that integers and floating point numbers are stored in the B and OE arrays with mixed formats. The method is important to understanding the details of the code. Two equivalenced names are used, for instance, say U(5000) and IU(5000,2). The first is double precision floating and the second is single precision integer. When storing an integer value IU(*,1) is referenced, and when storing a floating number U(*) is referenced.

**B.2.3 Statistical Processor Functions**

This section describes the control functions performed by the Statistical Processor. There are four modes of operation: Single Trajectory, Range Check, Analytical Statistical, and Monte Carlo.

**Single Trajectory**

After all data has been read by the Input Processor, the nominal case is executed and the results are stored in the C array. If the input stream contained only **TYPE = 5** (constant) variables, the nominal case values are printed and a normal exit results.

**Range Check**

After all data has been read and all missing values filled in, the rows in the IA array are sorted so that **TYPE = 1** (Range Check) variables occupy rows 11, 12 and 13. If there are **TYPE = 1** variables, they are processed in a nested DO loop and the values
of the dependent variables for each combination are stored in the C array. After the DO loops have been satisfied, the C array information is printed and a normal exit results.

**Analytical Statistical**

If no TYPE = 1 variables are input, the IA array rows are sorted such that all TYPE = 2, 3 and 4 variables are situated at the top. After this sort, the nominal case is executed. The nominal case results are stored in the C array.

Following the nominal case, limit cases are run with the independent variables sequentially permuted. The resulting dependent variable values are stored in the C array behind the nominal case results. The derivatives are calculated sequentially. The resulting derivative values are stored in the C array by overwriting the previous function evaluations. Using the derivatives, the function mean values and variances are calculated and stored in the C array immediately after the derivatives. The derivatives, means and variances are then printed.

These processes can be time consuming and require large amounts of computer storage. For purposes of estimating the storage and time requirements, let $N_D$ be the number of dependent variables (TYPE = 7 or 8), and $N_I$ be the number of independent variables (TYPE = 2, 3 or 4). As a general rule, storage requirements vary with the product $N_D N_I$, whereas the time consumption varies with $N_I$. Table B-1 states time (passes through the projectile trajectory module) and storage requirements. The C array allocates a total of 5,000 locations to data storage of the outputs as well as tabular storage for TYPE = 4 variables. This should be more than adequate for most purposes. Should more storage be required, it is a straightforward process to increase the size of the C array in each subroutine in which it appears.

**Table B-1  Time and Storage Requirements**

<table>
<thead>
<tr>
<th>Control Variable</th>
<th>Function Evaluations</th>
<th>Required Storage Locations (C Array)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I$\theta$PRNT $\geq$ 1</td>
<td>Nominal Case</td>
<td>1</td>
</tr>
<tr>
<td>I$\theta$PRNT $\geq$ 2</td>
<td>Limit Cases</td>
<td>$2N_I$</td>
</tr>
<tr>
<td>I$\theta$PRNT $\geq$ 4</td>
<td>Derivatives</td>
<td>0</td>
</tr>
<tr>
<td>I$\theta$PRNT $\geq$ 5</td>
<td>Mixed Derivatives</td>
<td>$4N_I(N_I + 1)$</td>
</tr>
<tr>
<td>I$\theta$PRNT $\geq$ 5</td>
<td>Covariances</td>
<td>0</td>
</tr>
</tbody>
</table>
Monte Carlo

The Monte Carlo mode performs four functions: It reads additional input, verifies the input, conducts Monte Carlo experiments and constructs histograms, and outputs the results. The input step of the Monte Carlo mode is preceded by execution of the Analytical Statistical mode. The Monte Carlo mode acquires additional control parameters from the input which define the sample size, number of cells, calculational option, and allowable number of rejected trial solutions.

The second function, input verification, consists of rearranging some of the values generated by the Analytical Statistical mode to be compatible with the Monte Carlo calculations. To efficiently describe the verification process, the range of all variables is assumed to be known in terms of

\[ \bar{x} - T \leq x \leq \bar{x} + T \]  

(B.2-1)

where \( \bar{x} \) is the mean value and \( T \) is the tolerance. The relationship of these variables to inputted and computed quantities has been stated elsewhere. These are summarized here along with some additions and/or modifications. In general for TYPE = 2, 3 or 8 variables the user specifies \( \bar{x}, T, \) and the variance \( \sigma^2 \), by input. The Analytical Statistical mode calculates these same quantities for TYPE = 4 and 7 variables. To insure proper sampling and to detect possible user errors the input is verified according to the following rules.

1) For TYPE = 2 or 7 variables the tolerance \( T \) is taken to be the larger of the input specified value or \( 3\sigma \) and the input value of \( \bar{x} \) is accepted.

2) For TYPE = 3 or 8 variables the given tolerance value and input value of \( \bar{x} \) are accepted and the variance value is ignored.

3) For TYPE = 4 variables the midrange value and tolerance are calculated from the extremal values of the input list with the input values ignored. For example, if ten values are entered

\[ \bar{x} = \frac{1}{2} (x_{10} + x_1) \]  

(B.2-2)

\[ T = \frac{1}{2} (x_{10} - x_1) \]  

(B.2-3)
For TYPE = 4 variables the maximum value of the distribution function is also found and stored in the location reserved for the calculated variance value. Any quantities changed in the verification process are stored and are subsequently printed on the output. The input verification step also includes the initialization of the histogram counters as well as the counting of the number of TYPE = 8 variables to be considered.

The third function in the computational process is the conduct of the Monte Carlo experiment and the construction of histograms. This is the heart of the whole computation. The independent variable values are generated, processed through the projectile trajectory module, and analyzed to increment appropriate histogram cell counters. The process is then repeated over and over until one of the following terminating conditions occurs:

1) Sampling difficulties for a TYPE = 4 variable occur.

2) The number of trial solutions which are outside of the internally generated histogram ranges exceeds a user defined limit.

3) The number of trial solutions which are outside of the user supplied histogram ranges exceeds a user defined limit.

4) The specified number of experiments have been performed.

Histograms are constructed for both the dependent and independent variables. Reasonable estimates of the ranges of all variables are available from input or are internally computed and cell boundaries are established. During the execution of the Monte Carlo experiments it is possible for one or more dependent variables to fall outside the range covered by its histogram. The results from such an experiment should be rejected from all histograms. This requires the code to take the special precautions described in the next two paragraphs when incrementing the histogram counters.

To accommodate these problems the following scheme is employed. The basic histograms are stored in five integer arrays. Two of these arrays, ID1 and III, are used to store the count for each cell belonging to the dependent and independent variables, respectively. For example, if each variable range
contains five cells, the counter for the second cell of the
dependent variable is the twelfth location of IDI. The
arrays IDI and IIl allow for a maximum of 20 cells with a maxi-
mum of 10 dependent variables and 20 independent variables.
Two additional integer arrays, IDT(10) and IIT(20), are used
for temporary storage of appropriate counter addresses in the
IDI and IIl arrays. As the independent variable values are
generated for each trial solution, the appropriate counter ad-
dresses are stored in the IIT array. After the dependent vari-
ables have been evaluated, their counter addresses are stored
in the IDT array. If all of dependent variable values are
within the range of their respective histograms, the addresses
in the IDT and IIT arrays are used to increment the appropriate
counters in the IDI and IIl arrays. If one or more of the de-
pendent variable values is not within range, then this trial
solution is discarded and none of the interval counters are
incremented. In this fashion the histograms count only accept-
able experiments. A fifth integer array, IRI, is used to count
the frequency with which each of the dependent variables causes
experiments to be discarded.

The user may desire histogram ranges for the dependent
variables which differ from those internally computed. To ac-
commodate this possibility three additional integer arrays,
ID2, II2 and IR2, are included. These arrays are replicas of
the basic arrays, IDI, IIl and IRI, and are manipulated in ex-
actly the same manner described above with one exception. The
exception is that user specifications in the form of TYPE = 8
variables are used instead of the internally generated ranges.
Thus, if the user specifies a TYPE = 8 dependent variable, the
routine provides two sets of histograms, one set using internal
ranges and one set using the TYPE = 8 specifications.

The fourth and final function performed on the Monte Carlo
mode consists of printing the Monte Carlo results. The histo-
gram values are calculated and printed. The sample mean and
variance are calculated and printed along with the corresponding
values from the Analytical Statistical mode. The output is on
a variable by variable basis. Occasionally the program hangs
up in the output section with an indicated error of a divide
fault. This usually occurs because none of the generated so-
lutions were within the histogram ranges. This is usually due
to either a gross input error, or an over restricted range for
a TYPE = 8 variable.
B.2.4 HITS Subroutines

The HITS computer code consists of a MAIN program and sixty-two subroutines. This section briefly describes the subroutines and their functional interconnections.

Table B-2 lists and briefly describes each of the subroutines. Further details can be gained from the program listings of Appendix D.

In Figure B-4 is a diagram depicting the subroutine linking arrangement of the HITS code. Referring to Figure B-4, there are four decision blocks labeled D1 through D4, each of which has several branch paths. The paths from the decision blocks are mutually exclusive and whenever the code is exercised only one path is followed. Decision block D1 is used to select the projectile trajectory module. At present, there is only one option (IY = 1); the others are dummy facilities. The second decision block, D2, determines which of the four Statistical Processor modes are to be exercised: Single Trajectory, Range Check, Analytical Statistical, or Monte Carlo. Decision block D2 is controlled by the TYPE input data. The decision block D3 is under direct control of the user and affects the decision to use a Taylor series approximation of the Projectile Trajectory module for the Monte Carlo experiments. Decision block D4 determines whether separate histograms are to be generated for user specified histogram ranges. This decision is made on the basis of whether or not any TYPE = 8 variables were input.

Figure B-5 diagrams the subroutine linking arrangement for the Projectile Trajectory module. Note the two distinct trajectory calculations based on the real world and fire control projectile characteristics.

B.2.5 Histogram Computer Plots

The HITS computer code is designed to facilitate creation of computer drawn histogram plots. Since computer plot software is hardware specific, the present code doesn't do the plotting. Rather it gathers the data together into one subroutine (HYSPLT) so that automatic plotting can be easily implemented by users who desire it. This section defines the format of the data.

The two-dimensional array TITLE (I,J) contains the titles for the histograms. The second subscript, J, denotes the variable
<table>
<thead>
<tr>
<th>SUBROUTINE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>Print input data and sequence other subroutine calls.</td>
</tr>
<tr>
<td>ARTLU</td>
<td>One-dimensional table look up routine.</td>
</tr>
<tr>
<td>A2</td>
<td>Computes $\int_0^x a^2 , dx$ using Real World parameters.</td>
</tr>
<tr>
<td>A2F</td>
<td>Computes $\int_0^x a^2 , dx$ using Fire Control parameters.</td>
</tr>
<tr>
<td>NAMES</td>
<td>Block data routine to initialize output variable names.</td>
</tr>
<tr>
<td>CHKIA</td>
<td>Perform check to see if any data is missing.</td>
</tr>
<tr>
<td>CHKIN</td>
<td>Check the input variables:</td>
</tr>
<tr>
<td></td>
<td>a) establish number of TYPE = 8 variables</td>
</tr>
<tr>
<td></td>
<td>b) force tolerance = 3$\sigma$ for TYPE = 2 and 7 variables</td>
</tr>
<tr>
<td></td>
<td>c) establish tolerance for TYPE = 4 variables</td>
</tr>
<tr>
<td>CNVRT</td>
<td>Convert uniform random numbers to Gaussian random numbers with the interval 0 to 1 as $3\sigma$ limits.</td>
</tr>
<tr>
<td>CRROSS</td>
<td>Computes the Real World projectile's oscillatory motion and crossrange (horizontal and vertical) velocity and position due to oscillatory motion.</td>
</tr>
<tr>
<td>CRROSSF</td>
<td>Computes the Fire Control projectile's oscillatory motion and crossrange (horizontal and vertical) velocity and position due to oscillatory motion.</td>
</tr>
<tr>
<td>CRVFT</td>
<td>Generate output variable values using 1st or 2nd order Taylor series expansions of projectile trajectory module.</td>
</tr>
<tr>
<td>DOABC</td>
<td>Provide automatic Range Check calculations.</td>
</tr>
<tr>
<td>DO234</td>
<td>Provide Analytical Statistical mode calculations.</td>
</tr>
<tr>
<td>EXTRA</td>
<td>Output routine for Range Check variable input data.</td>
</tr>
<tr>
<td>FC2987</td>
<td>Fire Control trajectory module interface.</td>
</tr>
<tr>
<td>SUBROUTINE</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>FEXP</td>
<td>Computes double precision exponents with error checking.</td>
</tr>
<tr>
<td>FILLA</td>
<td>Calculate mean values and variances from histograms.</td>
</tr>
<tr>
<td>FILLIN</td>
<td>Dummy routine to provide interface with time phased data base for missing information.</td>
</tr>
<tr>
<td>GQC9R</td>
<td>Computes fifth and sixth terms in Eq. (B.1-7).</td>
</tr>
<tr>
<td>GQUC</td>
<td>Computes third and fourth terms in Eq. (B.1-7).</td>
</tr>
<tr>
<td>G1795</td>
<td>Dummy alternate trajectory module interface.</td>
</tr>
<tr>
<td>G2440</td>
<td>Dummy alternate trajectory module interface.</td>
</tr>
<tr>
<td>G2987</td>
<td>Trajectory module interface.</td>
</tr>
<tr>
<td>HSTG1</td>
<td>Increment histogram counters using internally estimated variable range values.</td>
</tr>
<tr>
<td>HSTG2</td>
<td>Increment histogram counters using user defined (TYPE = 8) variable range values.</td>
</tr>
<tr>
<td>HYSPLT</td>
<td>Interface for automatic histogram bar chart plots.</td>
</tr>
<tr>
<td>INCARD</td>
<td>Reads data cards and decodes the TYPE numbers</td>
</tr>
<tr>
<td>INCON</td>
<td>Computes algebraic constants using Real World initial conditions.</td>
</tr>
<tr>
<td>INCNF</td>
<td>Computes algebraic constants using Fire Control initial conditions.</td>
</tr>
<tr>
<td>INITIL</td>
<td>Selects projectile trajectory module, presets data values and initialize INCARD.</td>
</tr>
<tr>
<td>INLHST</td>
<td>Initialize all histogram and error counters to zero.</td>
</tr>
<tr>
<td>LOADER</td>
<td>Collects and stores data for automatic histogram bar chart plotter.</td>
</tr>
<tr>
<td>SUBROUTINE</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>MCRL</td>
<td>Sequence Monte Carlo operations and monitor number of trials and number of rejects.</td>
</tr>
<tr>
<td>MOVEUP</td>
<td>Internally sorts input data to move variables to top of storage arrays.</td>
</tr>
<tr>
<td>NCØV</td>
<td>Locates covariances of independent variables in the C array.</td>
</tr>
<tr>
<td>NMEAN</td>
<td>Locates mean values of dependent variables in the C array.</td>
</tr>
<tr>
<td>NVAR</td>
<td>Locates variances of dependent variables in the C array.</td>
</tr>
<tr>
<td>NVARX</td>
<td>Locates variances of independent variables in B array.</td>
</tr>
<tr>
<td>NLD</td>
<td>Locates first order derivatives in C array.</td>
</tr>
<tr>
<td>N2D</td>
<td>Locates second order derivatives in C array.</td>
</tr>
<tr>
<td>PICK1</td>
<td>Branch routine to select projectile trajectory module.</td>
</tr>
<tr>
<td>PRNTMC</td>
<td>Provides printed output of the Monte Carlo results.</td>
</tr>
<tr>
<td>QCØR</td>
<td>Computes second term of Eq. (B.1-5) and second term of Eq. (B.1-7).</td>
</tr>
<tr>
<td>RANDØM</td>
<td>Formats random numbers.</td>
</tr>
<tr>
<td>RANDU</td>
<td>Random number generator for supplying uniform random numbers on interval 0 to 1.</td>
</tr>
<tr>
<td>RW2987</td>
<td>Real World trajectory module interface.</td>
</tr>
<tr>
<td>SAMPLE</td>
<td>Generates random sample values for all TYPE = 2, 3 or 4 variables.</td>
</tr>
<tr>
<td>SPRNT</td>
<td>Formats and prints contents of OE array.</td>
</tr>
<tr>
<td>STDDEV</td>
<td>Prints standard deviations with greater precision.</td>
</tr>
<tr>
<td>STØREC</td>
<td>Data storage routine for efficient packing of the C array.</td>
</tr>
</tbody>
</table>
Table B-2  HITS Subroutine List (Concl'd)

<table>
<thead>
<tr>
<th>SUBROUTINE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TG1795</td>
<td>Dummy alternate trajectory module preset routine.</td>
</tr>
<tr>
<td>TG2440</td>
<td>Dummy alternate trajectory module preset routine.</td>
</tr>
<tr>
<td>TG2987</td>
<td>Initializes trajectory parameters at preset (default) values.</td>
</tr>
<tr>
<td>TRAJT</td>
<td>Computes Real World projectile velocity and range as a function of time, including the influence of oscillatory motion on drag.</td>
</tr>
<tr>
<td>TRAJX</td>
<td>Computes Real World projectile velocity and time as a function of range, including the influence of oscillatory motion on drag.</td>
</tr>
<tr>
<td>TRAJXF</td>
<td>Computes Fire Control projectile velocity and time as a function of range, including the influence of oscillatory motion on drag.</td>
</tr>
<tr>
<td>TVX</td>
<td>Computes projectile velocity and range as a function of time for a particle (no oscillatory drag) trajectory.</td>
</tr>
<tr>
<td>T4NTV</td>
<td>Computes mean value, variance and tolerance for arbitrarily distributed variables.</td>
</tr>
<tr>
<td>VXT</td>
<td>Computes projectile velocity and time as a function of range for a particle (no oscillatory drag) trajectory.</td>
</tr>
<tr>
<td>WIND</td>
<td>Computes projectile horizontal velocity and position brought about by crosswinds.</td>
</tr>
<tr>
<td>XXC</td>
<td>Computes the range at which the projectile's oscillatory motion has converged.</td>
</tr>
<tr>
<td>ZATAN2</td>
<td>Computes the arctangent of a function, placing the resulting angle in the proper quadrant. Includes provisions to avoid computational errors for the special case of non-oscillatory motion.</td>
</tr>
<tr>
<td>ZZ</td>
<td>Computes appropriate value of h as defined by Eq. (C.3-44)</td>
</tr>
</tbody>
</table>
Figure B-4 HITS Subroutine Linking Diagram
Figure B-5  Projectile Trajectory Module
Subroutine Linking Diagram
associated with the histogram. These are, at most, $KD + K234$
variables, where $KD$ is the number of dependent variables (appearing first) and $K234$ is the number of independent variable (appearing last). The first subscript defines specific items relating to the variable as defined by Figure B-6. The actual histograms are stored in the three-dimensional array $ZDATAD(I,J,K)$. The index $K$ denotes the $KD + K234$ variables with the dependent variables appearing first and the independent last. Figure B-7 illustrates the format of a two-dimensional slice of the array, $ZDATAD(I,J,\ast)$. The upper portion applies to the dependent variables and the lower defines the array for independent variables.
<table>
<thead>
<tr>
<th>ROW</th>
<th>INTERNAL NAME</th>
<th>DESCRIPTION</th>
<th>REMARKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IPASS</td>
<td>Histogram Indicator</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>IV</td>
<td>Code Number</td>
<td>X-Axis Label</td>
</tr>
<tr>
<td>3</td>
<td>ITYPE</td>
<td>Variable Type</td>
<td>2, 3, 4, 7 or 8</td>
</tr>
<tr>
<td>4</td>
<td>NRR</td>
<td>Number of Rejects</td>
<td>Rejects</td>
</tr>
<tr>
<td>5</td>
<td>CM</td>
<td>Histogram Mean</td>
<td>MEAN</td>
</tr>
<tr>
<td>6</td>
<td>CV</td>
<td>Histogram Variance</td>
<td>VAR</td>
</tr>
<tr>
<td>7</td>
<td>PM</td>
<td>Estimated Mean</td>
<td>(MEAN)</td>
</tr>
<tr>
<td>8</td>
<td>PV</td>
<td>Estimated Variance</td>
<td>(VAR)</td>
</tr>
<tr>
<td>9</td>
<td>FCTR</td>
<td>Histogram Cell Size</td>
<td>Interval</td>
</tr>
<tr>
<td>10</td>
<td>PT</td>
<td>Tolerance</td>
<td>TOL</td>
</tr>
<tr>
<td>11</td>
<td>NRR</td>
<td>Number of Rejects</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>CM</td>
<td>Histogram Mean</td>
<td>MEAN</td>
</tr>
<tr>
<td>13</td>
<td>CV</td>
<td>Histogram Variance</td>
<td>VAR</td>
</tr>
<tr>
<td>14</td>
<td>PM</td>
<td>Estimated Mean</td>
<td>(MEAN)</td>
</tr>
<tr>
<td>15</td>
<td>PV</td>
<td>Estimated Variance</td>
<td>(VAR)</td>
</tr>
<tr>
<td>16</td>
<td>FCTR</td>
<td>Histogram Range</td>
<td>Interval</td>
</tr>
<tr>
<td>17</td>
<td>PT</td>
<td>Tolerance</td>
<td>TOL</td>
</tr>
<tr>
<td>18</td>
<td>NCELL</td>
<td>Number of Cells</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>SAVE</td>
<td>Maximum Relative Frequency</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>ISMØØD</td>
<td>Indicator for Spline</td>
<td>1 for TYPE = 2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Smoothing Option</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>zero</td>
</tr>
</tbody>
</table>

Figure B-6 Histogram Title Array
### (a) Dependent Variables

<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>X-Axis</th>
<th>Y-Axis</th>
<th>User Ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$X_{b_0}$</td>
<td>$100 \frac{N_0}{N_T^*}$</td>
<td>$X_{b_0}$</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>$X_{b_1}$</td>
<td>$100 \frac{N_1}{N_T^*}$</td>
<td>$X_{b_1}$</td>
</tr>
</tbody>
</table>

### (b) Independent Variables

<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>X-Axis</th>
<th>Y-Axis</th>
<th>Y-Axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Figure B-7  Histogram Storage Array ZDATAD (I,J,*)
APPENDIX C

CLOSED FORM TRAJECTORY EQUATIONS

This appendix derives closed form expressions for the trajectory of a hypervelocity projectile. The equations define the trajectory in time as well as space. The closed form expressions result from realistic simplifying approximations to the full six degree of freedom (6 DOF) equations of motion. All assumptions were verified by comparison to exact solutions as determined by numerical integration of the equations of motion. In all cases, the closed form equations agreed to within engineering tolerances. The equations are of general interest. Avco has employed them in: (1) the aerodynamic design of a hypervelocity projectile, (2) the interpretation of ballistic range test data, and (3) the evaluation of projectile dispersion. Potential applications include incorporation in operational fire control computers, computation of firing tables, and any other situation requiring rapid, accurate, low-cost trajectory determination. Section C.1 presents background material and an overview of the analytical development. Particle trajectory equations are presented in Section C.2. Sections C.3 and C.4 present cross-range and downrange perturbation equations, respectively, used to correct the basic particle trajectory.

1.1 Introduction

The complete coupled, nonlinear equations of motion of a projectile in free flight cannot be solved in a general closed form. To obtain such solutions, it is necessary to make simplifying approximations. Many such solutions have been developed in the past under various assumptions for application to a wide variety of problems. These include the gross evaluation of range-velocity-time histories, ballistic range data reduction, and the development of jump angle expressions for specific launch disturbances. One of the unique aspects of the trajectory equations presented here is that both downrange as well as crossrange dynamics are taken into account. The key has been to build upon the existing work and to piece together various solutions in such a manner as to accurately describe the downrange and crossrange dynamics of hypervelocity projectiles.

The trajectory model has three parts. The first part describes the basic particle trajectory including the effects of constant velocity winds in both the crossrange and downrange
directions. The second part deals with angle of attack oscillatory motion and subsequent crossrange deflection from the particle trajectory. The final part of the model makes use of the angle of attack history from the oscillatory motion solution to compute range, velocity, and flight time perturbations for correction of the basic particle trajectory. These values account for the increase in drag brought about by angle of attack oscillations. The three parts are then superimposed to describe the complete trajectory history in six degrees of freedom.

The coordinate system used in the trajectory model is shown in Figure C-1. A consistent set of symbols is used throughout this appendix and is presented in Table C-1 along with code numbers which relate the variables to the HITS Code.

C.2 Particle Trajectory

This section presents the basic particle trajectory closed form equations. The development of the equations is discussed in Section C.2.1. Section C.2.2 presents results which verify the equations.

C.2.1 Analytical Development

The particle trajectory solution is fundamentally the same as that developed and reported in an earlier study, with the formulation extended to include the effects of winds in the crossrange and downrange directions. The primary assumptions involved in the solution are the following:

- The drag coefficient varies as $1/V^2$.
- The wind has constant speed and direction.
- The projectile weathercocks into the wind and flies a static zero angle of attack trajectory.
- A uniform density atmosphere.
- The absence of gravity.

The drag coefficient is allowed to vary with velocity according to

$$C_D = C_{D_\infty} + \frac{K_D}{V^2} \quad \text{(C.2-1)}$$

FIGURE C-1 TRAJECTORY COORDINATE SYSTEM
<table>
<thead>
<tr>
<th>HITS CODE NUMBERS</th>
<th>SYMBOL</th>
<th>DEFINITION AND COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R.W.</td>
<td>F.C.</td>
<td>C.O.</td>
</tr>
<tr>
<td>122</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>404</td>
<td></td>
<td></td>
</tr>
<tr>
<td>412</td>
<td></td>
<td></td>
</tr>
<tr>
<td>419</td>
<td></td>
<td></td>
</tr>
<tr>
<td>418</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{l_a} )</td>
<td>( C_{N_a} = \left( C_{D_a} + \frac{K_D}{v_0^2} \right) )</td>
<td></td>
</tr>
<tr>
<td>( C_M )</td>
<td>pitching moment coefficient about center of gravity, ( \frac{1}{M} )</td>
<td></td>
</tr>
<tr>
<td>413</td>
<td></td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

1 Denotes Real World projectile parameter
2 Denotes Fire Control projectile parameter
3 Denotes Computed quantity
<table>
<thead>
<tr>
<th>HITS CODE NUMBERS</th>
<th>SYMBOL</th>
<th>DEFINITION AND COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R.W.</td>
<td>F.C.</td>
<td>C.O.</td>
</tr>
<tr>
<td>110</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>434</td>
<td></td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>206</td>
<td>ICNCL</td>
</tr>
<tr>
<td>126</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>435</td>
<td></td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>206</td>
<td>ICNCL</td>
</tr>
<tr>
<td>127</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>430</td>
<td></td>
<td></td>
</tr>
<tr>
<td>473</td>
<td></td>
<td>JA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Table C-1

**List of Symbols** (Cont'd)

<table>
<thead>
<tr>
<th>Hits Code Numbers</th>
<th>Symbol(s)</th>
<th>Definition and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>R.W. F.C. C.O.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>130 30</td>
<td>$J_{Ay}$</td>
<td>vertical component of jump angle, radians</td>
</tr>
<tr>
<td></td>
<td>$J_{Az}$</td>
<td>horizontal component of jump angle, radians</td>
</tr>
<tr>
<td>410</td>
<td>$K_{Cx}$</td>
<td>constant describing variation of $C_x$ with angle of attack, rad$^{-2}$</td>
</tr>
<tr>
<td>411</td>
<td>$K_{D}$</td>
<td>constant describing variation of $C_x$ with velocity, sec$^2$/ft$^2$</td>
</tr>
<tr>
<td>216,217</td>
<td>$K_1, K_2$</td>
<td>parameters defined by Eqs. (C.3-14) and (C.3-15)</td>
</tr>
<tr>
<td>123 23</td>
<td>$L$</td>
<td>projectile length, ft</td>
</tr>
<tr>
<td>431</td>
<td>$\alpha^*$</td>
<td>$\frac{\pi}{\rho A_{e} V_0}$, sec</td>
</tr>
<tr>
<td>128 28</td>
<td>$P$</td>
<td>projectile spin rate, rad/sec</td>
</tr>
<tr>
<td></td>
<td>$P_{cr}$</td>
<td>projectile resonant spin rate, rad/sec</td>
</tr>
<tr>
<td>441,442,443,444</td>
<td>$R_1, R_2, R_3, R_4$</td>
<td>parameters defined by Eqs. (C.3-16) through (C.3-19)</td>
</tr>
<tr>
<td>445</td>
<td>$R_{trim}$</td>
<td>static trim angle amplification factor due to spin</td>
</tr>
<tr>
<td>111 11</td>
<td>$SM$</td>
<td>projectile static margin, $\frac{X_{ef} - X_{es}}{L}$</td>
</tr>
<tr>
<td>450</td>
<td>$t$</td>
<td>flight time, sec</td>
</tr>
<tr>
<td></td>
<td>$\frac{t}{V_0}$</td>
<td>sec</td>
</tr>
<tr>
<td>500</td>
<td>$t_n$</td>
<td>nominal flight time, sec</td>
</tr>
</tbody>
</table>
# TABLE C-1

**LIST OF SYMBOLS** (Cont'd)

<table>
<thead>
<tr>
<th>HITS CODE NUMBERS</th>
<th>SYMBOL</th>
<th>DEFINITION AND COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R.W.</td>
<td>P.C.</td>
<td>C.Q.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HITS CODE NUMBERS</td>
<td>SYMBOL</td>
<td>DEFINITION AND COMMENTS</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>R.W</td>
<td>E.C.</td>
<td>C.Q.</td>
</tr>
<tr>
<td>426</td>
<td></td>
<td></td>
</tr>
<tr>
<td>427</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>402</td>
<td></td>
<td></td>
</tr>
<tr>
<td>476</td>
<td></td>
<td></td>
</tr>
<tr>
<td>403</td>
<td></td>
<td></td>
</tr>
<tr>
<td>401</td>
<td></td>
<td></td>
</tr>
<tr>
<td>409</td>
<td></td>
<td></td>
</tr>
<tr>
<td>408</td>
<td></td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>405</td>
<td></td>
<td></td>
</tr>
<tr>
<td>406</td>
<td></td>
<td></td>
</tr>
<tr>
<td>407</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE C-1

LIST OF SYMBOLS (Concl'd)

<table>
<thead>
<tr>
<th>HITS CODE NUMBERS</th>
<th>SYMBOL</th>
<th>DEFINITION AND COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R.W.</td>
<td>F.C.</td>
<td>C.Q.</td>
</tr>
<tr>
<td>116</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>464, 465, 466, 469</td>
<td>λ₀, λ₁, λ₂, Δλ</td>
<td>components of damping rates, sec⁻¹</td>
</tr>
<tr>
<td>467, 469</td>
<td>ν₁, ν₂</td>
<td>phase angles defined by Eqs. (C.3-24) and (C.3-25), respectively, radians</td>
</tr>
<tr>
<td>104</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>437, 438, 439</td>
<td>φ₀, φ₁, φ₂</td>
<td>phase angles defined by Eqs. (C.4-15), (C.3-45), and (C.3-46), respectively, radians</td>
</tr>
<tr>
<td>A₀</td>
<td></td>
<td>meridional orientation of static trim, radians</td>
</tr>
<tr>
<td>Δφ</td>
<td>meridional shift of static trim orientation due to spin, radians</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>460, 461, 462, 468</td>
<td>ω₀, ω₁, ω₂, Δω</td>
<td>components of oscillation frequency defined by Eqs. (C.3-7), (C.3-3), (C.3-4), and (C.3-8), respectively; rad/sec</td>
</tr>
</tbody>
</table>
where \( C_{D\infty} \) and \( K_D \) are constants evaluated for the specific projectile and velocity regime. Knowledge of the projectile drag coefficient at two velocities within the range of interest allows determination of the two constants \( C_{D\infty} \) and \( K_D \). The drag coefficient approximation may also be expressed in terms of the ballistic coefficient, \( \beta \) (i.e., \( \beta = \frac{W}{C_D A} \)), as indicated in Figure C-2 which shows the accuracy of a typical fit.

With the inclusion of constant velocity winds, the previous particle trajectory solution takes the form

\[
V_x = \left\{ (V_0 - W_x)^2 e^{2f \sqrt{B} C_{D\infty} (W_x - x)} + \frac{K_D}{B C_{D\infty}} \left( e^{2f \sqrt{B} C_{D\infty} (W_x - x)} - 1 \right)^{1/2} \right\}^{1/2} + W_x
\]  \hspace{1cm} (C.2-2)

\[
t = \frac{1}{f \sqrt{C_{D\infty} K_D}} \left\{ \tan^{-1} \left[ \sqrt{\frac{B C_{D\infty}}{K_D}} (V_0 - W_x) \right] - \tan^{-1} \left[ \sqrt{\frac{B C_{D\infty}}{K_D}} (V_x - W_x) \right] \right\}
\]  \hspace{1cm} (C.2-3)

\[
V_{z_w} = W_x \left[ 1 - \frac{V_x - W_x}{V_0 - W_x} \right]
\]  \hspace{1cm} (C.2-4)

\[
Z_w = \frac{W_x}{V_0 - W_x} (V_0 - x)
\]  \hspace{1cm} (C.2-5)

It is most important to notice that range is the independent variable in the particle trajectory equations. Subsequently, velocity and time-of-flight corrections are computed and applied as a function of range and not time. Equation (C.2-3) is indeterminate for the case of constant drag coefficient (\( C_D = C_{D\infty} \) and \( K_D = 0 \)). Numerical evaluation of this case can be made using any sufficiently small value for \( K_D \), such that \( K_D / \sqrt{B} << C_{D\infty} \). In the presence of a wind component in the downrange direction, \( W_x \), an iterative procedure must be used to solve the equations, since the product of the unknown flight time, \( t \), and \( W_x \) appears in the expression for velocity, i.e., Eq. (C.2-2).

---

Figure C-2 Typical Ballistic Coefficient Variation with Mach Number

Sea Level

Analytic Model: \( \frac{1}{\beta} = \frac{1}{\beta_\infty} + \frac{K}{V^2} \)

Asymptotic Limit, \( \beta_\infty \)

Detailed Calculation

Flight Ballistic Coefficient

Mach Number = \( \frac{V}{V_s} \)

\( V_s \) = Speed of Sound
C.2.2 Verification

The particle trajectory equations, including the drag coefficient approximation, were verified by comparing them to exact numerical solutions of the 6 DOF equations of motion. Figure C-3 summarizes one comparison. It shows the range-velocity-time histories from the analytic model and the 6 DOF simulation agree.

A separate comparison was made to evaluate the effects of winds on crossrange error. This comparison (as well as all subsequent check cases in this appendix) was made using a smaller projectile whose aerodynamic and mass properties are listed in Table C-2. With a muzzle velocity of 11,000 ft/sec and a constant crossrange wind velocity, $W_z$, of 15 ft/sec, the 6 DOF calculations showed an 11.2 ft crossrange drift after 2 seconds of flight. The analytic model agreed to within 2 percent.

C.3 Crossrange Perturbations

This section presents the crossrange perturbation equations that are used to correct the basic particle trajectory. The development of the equations is discussed in Section C.3.1. Section C.3.2 presents results which verify the equations.

C.3.1 Analytical Development

The crossrange perturbation model addresses the oscillatory motion and corresponding crossrange effects. A solution for the angle of attack history was obtained from a NACA report.\(^1\) This model employs the following typical ballistic range assumptions:

- Constant velocity
- Constant spin rate
- Linear aerodynamics
- Small trim angles due to shape or mass asymmetries
- Small angles ($\cos \theta = 1, \tan \theta = \sin \theta = \theta$)
- Uniform density atmosphere
- The absence of gravity

---

\[ \theta = 4.5^\circ \]
\[ \frac{R_N}{R_B} = 0.053 \]
\[ R_B = 0.56'' \]
\[ L = 6.76'' \]
\[ W = 0.25 \text{ LB} \]
\[ \beta_F = 1078 \text{ LB/FT}^2 \]

**Figure C-3 Trajectory Comparison**

- Analytic Model
- Digital Code

---

-225-
### TABLE C-2

**TEST CASE PROJECTILE PROPERTIES**

<table>
<thead>
<tr>
<th>CHARACTERISTIC</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass Properties:</td>
<td></td>
</tr>
<tr>
<td>Cone Angle, $\theta_C$</td>
<td>$5.5^\circ$</td>
</tr>
<tr>
<td>Length, $L$</td>
<td>3.7&quot;</td>
</tr>
<tr>
<td>Nose Radius, $R_N$</td>
<td>.019&quot;</td>
</tr>
<tr>
<td>Base Radius, $R_B$</td>
<td>.375&quot;</td>
</tr>
<tr>
<td>Weight, $W$</td>
<td>.11 lbs.</td>
</tr>
<tr>
<td>Roll Moment of Inertia, $I_x$</td>
<td>$0.538 \times 10^{-6}$ slug-ft$^2$</td>
</tr>
<tr>
<td>Pitch Moment of Inertia, $I_y$</td>
<td>$0.787 \times 10^{-5}$ slug-ft$^2$</td>
</tr>
<tr>
<td>Aerodynamic Characteristics</td>
<td></td>
</tr>
<tr>
<td>Flight Ballistic Coefficient ($M = 15$), $\delta_F$</td>
<td>1000 PSF</td>
</tr>
<tr>
<td>Static Margin, $SM$</td>
<td>6.2% L</td>
</tr>
</tbody>
</table>
At first appearance, the constant velocity assumption may seem objectionable since significant velocity variations occur along the actual trajectory. The constant velocity assumption is required to achieve a closed form solution with a nonzero spin rate. In reality, the method of applying the constant velocity crossrange corrections to the particle trajectory minimizes the errors incurred in calculating the perturbation assuming a constant velocity. Velocity affects crossrange displacement through its influence on the aerodynamic force in the crossrange direction. This force is composed of two components: the drag force and the normal force. For low drag, slender projectiles, the normal force dominates and the drag force can be safely ignored. The normal force is proportional to the angle of attack. Thus, assuming small trim angles of attack, the normal force is only important while the projectile is oscillating in angle of attack. Furthermore, the first few oscillations are the most important because subsequent oscillations tend to average out. Since these occur near the muzzle where velocity is essentially constant, the constant velocity assumption is realistic. The crossrange perturbation is applied to the basic particle trajectory as a function of range. This scales the "constant velocity" crossrange perturbation in accord with the downrange velocity variation of the particle trajectory. Thus, the crossrange perturbation equations simulate the true velocity variation.

The solution given in the NACA report\(^1\) is in terms of the body fixed coordinate system shown in Figure C-4. The form of the solution is

\[
\alpha = K_1 e^{\lambda t'} \sin (\omega_1 t' + \nu_1) - K_2 e^{\lambda t'} \sin (\omega_2 t' - \nu_2) + \alpha_{\text{trim}} \tag{C.3-1}
\]

\[
\beta = K_1 e^{\lambda t'} \cos (\omega_1 t' + \nu_1) + K_2 e^{\lambda t'} \cos (\omega_2 t' - \nu_2) + \beta_{\text{trim}} \tag{C.3-2}
\]

where \(\alpha\) and \(\beta\) are, respectively, the pitch and yaw body angles of attack as illustrated in Figure C-4. These equations are written in terms of the range expressed as a pseudo flight time, \(t' = X/V_0\). This effects the appropriate scaling between

the crossrange perturbation solution assuming constant velocity and the realistic velocity variations contained in the particle trajectory. The frequency and damping constants are defined as follows:

\[ \omega_1 = \omega_o - \Delta \omega \]  
(C.3-3)

\[ \omega_2 = \omega_o + \Delta \omega \]  
(C.3-4)

\[ \lambda_1 = \lambda_o + \Delta \lambda \]  
(C.3-5)

\[ \lambda_2 = \lambda_o - \Delta \lambda \]  
(C.3-6)

where

\[
\omega_o = \frac{\sqrt{2}}{4} \left\{ \frac{4C_{M\alpha}}{1'} - \left( \frac{\rho}{m'} \right)^2 \left( \frac{C_{N\alpha}}{m'} + \frac{C_{M\theta}}{1'} \right)^2 + \left[ - \frac{4C_{M\alpha}}{1'} + \frac{\rho}{l_y} \frac{l_x}{I_y} \right]^2 \right\}^{1/2}
\]

(C.3-7)

\[
\Delta \omega = P \left( 1 - \frac{l_x}{2l_y} \right)
\]

(C.3-8)

\[
\lambda_o = \frac{1}{2} \left( \frac{C_{M\theta}}{1'} - \frac{C_{N\alpha}}{m'} \right)
\]

(C.3-9)

\[
\Delta \lambda = \frac{P l_x}{4 \omega_o l_y} \left( \frac{C_{N\alpha}}{m'} + \frac{C_{M\theta}}{1'} \right) + \frac{P C_{M\phi}}{2 \omega_o l_y}
\]

(C.3-10)

where

\[
C_{M\alpha} = -C_{N\alpha} \frac{L}{D}
\]

(C.3-11)

\[
I' = \frac{2l_y}{\rho AD V_o^2}
\]

(C.3-12)
\[
\omega' = \frac{2V}{\rho A g V_0}
\]  \hspace{1cm} (C.3-13)

The terms \(K_1\) and \(K_2\) in Eqs. (C.3-1) and (C.3-2) are constants involving the initial conditions

\[
K_1 = \sqrt{R_1^2 + R_2^2} \hspace{1cm} (C.3-14)
\]

\[
K_2 = \sqrt{R_3^2 + R_4^2} \hspace{1cm} (C.3-15)
\]

where

\[
R_1 = \frac{\omega_o [\dot{\beta}_o - \omega_2 (\beta_o - \beta_{\text{trim}}) - \lambda_2 (a_o - a_{\text{trim}})] + \Delta \lambda [\dot{\beta}_o - \omega_2 (\beta_o - \beta_{\text{trim}}) - \lambda_2 (a_o - a_{\text{trim}})]}{2(\omega_o^2 + \Delta \lambda^2)}
\]  \hspace{1cm} (C.3-16)

\[
R_2 = \frac{-\omega_o [\dot{\beta}_o - \omega_2 (\beta_o - \beta_{\text{trim}}) - \lambda_2 (a_o - a_{\text{trim}})] + \Delta \lambda [a_o - \omega_2 (\beta_o - \beta_{\text{trim}}) - \lambda_2 (a_o - a_{\text{trim}})]}{2(\omega_o^2 + \Delta \lambda^2)}
\]  \hspace{1cm} (C.3-17)

\[
R_3 = \beta_o - \beta_{\text{trim}} - R_1 \hspace{1cm} (C.3-18)
\]

\[
R_4 = a_o - a_{\text{trim}} - R_2 \hspace{1cm} (C.3-19)
\]

The initial conditions \(a_o, \beta_o, \dot{a}_o, \) and \(\dot{\beta}_o\) in Eqs. (C.3-16) and (C.3-17) are related to the projectile inertial orientations and rates by

\[
a_o = \theta_o - \frac{V_o}{V_0}
\]  \hspace{1cm} (C.3-20)
The phase angles \( \nu_1 \) and \( \nu_2 \) in Eqs. (C.3-1) and (C.3-2) are computed according to

\[
\nu_1 = \sin^{-1} \left( \frac{R_2}{K_1} \right) = \cos^{-1} \left( \frac{R_1}{K_1} \right)
\]

\[
\nu_2 = \sin^{-1} \left( \frac{R_4}{K_2} \right) = \cos^{-1} \left( \frac{R_3}{K_2} \right)
\]

Both the sine and cosine definitions are given here to establish the proper quadrants in which the phase angles \( \nu_1 \) and \( \nu_2 \) lie.

The terms \( a_{\text{trim}} \) and \( \beta_{\text{trim}} \) in Eqs. (C.3-1), (C.3-2), (C.3-16), and (C.3-17) are body fixed trim values for a spinning body (i.e., the "rolling trim") and are related to the body fixed static trim values, \( a_{\text{ST}} \) and \( \beta_{\text{ST}} \), by

\[
a_{\text{trim}} = R_{\text{trim}} \sqrt{a_{\text{ST}}^2 + \beta_{\text{ST}}^2} \cos (\phi_{\text{ST}} + \Delta \phi)
\]

\[
\beta_{\text{trim}} = R_{\text{trim}} \sqrt{a_{\text{ST}}^2 + \beta_{\text{ST}}^2} \sin (\phi_{\text{ST}} + \Delta \phi)
\]

where

\[
\phi_{\text{ST}} = \sin^{-1} \frac{\beta_{\text{ST}}}{\sqrt{a_{\text{ST}}^2 + \beta_{\text{ST}}^2}} = \cos^{-1} \frac{a_{\text{ST}}}{\sqrt{a_{\text{ST}}^2 + \beta_{\text{ST}}^2}}
\]

\[
\frac{a_{\text{ST}}}{\sqrt{a_{\text{ST}}^2 + \beta_{\text{ST}}^2}} = \frac{\beta_{\text{ST}}}{\sqrt{a_{\text{ST}}^2 + \beta_{\text{ST}}^2}}
\]
From the work presented in an AIAA journal,\(^1\) \(R_{\text{trim}}\) and \(\Delta \phi\) are given by

\[
P_{\text{trim}} = \left[ \left( 1 - \left( \frac{P}{P_{\text{CR}}} \right)^2 \right) + \left( 2\mu \left( \frac{P}{P_{\text{CR}}} \right)^2 \right) \right]^{-1/2}
\]

(C.3-29)

\[
\Delta \phi = \begin{cases} 
\tan^{-1} \left[ \frac{2\mu \left( \frac{P}{P_{\text{CR}}} \right)}{1 - \left( \frac{P}{P_{\text{CR}}} \right)^2} \right] & \text{for } 1 - \left( \frac{P}{P_{\text{CR}}} \right)^2 > 0 \\
\pi + \tan^{-1} \left[ \frac{2\mu \left( \frac{P}{P_{\text{CR}}} \right)}{1 - \left( \frac{P}{P_{\text{CR}}} \right)^2} \right] & \text{for } 1 - \left( \frac{P}{P_{\text{CR}}} \right)^2 < 0
\end{cases}
\]

(C.3-30)

where

\[
P_{\text{CR}} = \left[ \frac{-C_{\text{Na}} \rho V_o^2 AD}{3(l_y - l_z)} \right]^{1/2}
\]

(C.3-31)

\[
\mu = \left[ \frac{-C_{\text{Mq}} \rho AD}{2(l_y - l_z)} \right]^{1/2}
\]

(C.3-32)

The term $R_{trim}$ represents the amplification factor for a spin rate, $P$, close to the critical (or resonant) frequency, $P_{CR}$. The term $\Delta \phi$ represents the meridional shift of the plane of the rolling trim depending on the value of the spin rate with respect to $P_{CR}$. At resonance conditions ($P = P_{CR}$), the amplification is maximum and the rolling trim plane is shifted by $90^\circ$ with respect to the plane of the static trim.

With the angle of attack history established, the lift force is known. The crossrange displacement is found by integration of the equations of motion normal to the flight path. The remainder of this section is devoted to this integration.

Because of the spinning motion of the projectile, the body fixed coordinate system in which $\alpha$ and $\beta$ are described, see Figure C-4, rotates with respect to the inertial frame in which the trajectory is described, see Figure C-1. At exit from the muzzle ($t = 0$) the two coordinate systems are aligned. At this instant, positive $\alpha$ is nose up and creates a lift force in the vertical direction which is the positive $y$ direction in the trajectory frame. Owing to the nature of the body fixed coordinate system, positive $\beta$ at $t = 0$ is nose left and produces a horizontal yaw force in the negative $z$ direction. With positive spin being "right wing down," the body fixed lift and yaw forces rotate with respect to the trajectory coordinate frame as illustrated in Figure C-5. The force components in the inertial trajectory coordinate system are written as follows.\(^1\)

\[
L_{\alpha} = \frac{C_{N\alpha}}{2} \rho A V_o^2 \alpha
\]

\[
L_{\beta} = \frac{C_{N\beta}}{2} \rho A V_o^2 \beta
\]

\[
\phi = P t'
\]

\[
F_y = L_{\alpha} \cos \phi + L_{\beta} \sin \phi
\]

\[
F_z = L_{\alpha} \sin \phi - L_{\beta} \cos \phi
\]

\(^1\) In most applications, it is more appropriate to use $C_{L\alpha}$ instead of $C_{N\alpha}$ in the ensuing development. This affects only the definition of $h$, as noted in Eq. (C.3-44).
FIGURE C-5 LIFT FORCE COMPONENTS IN THE TRAJECTORY FRAME
Thus the equations of motion are

\[ \frac{w}{g} \frac{d^2 Y}{d t'^2} = \frac{C_{N_d} \rho A V_0^2}{2} \left[ a \cos P t' + \beta \sin P t' \right] \quad \text{(C.3-38)} \]

\[ \frac{w}{g} \frac{d^2 Z}{d t'^2} = \frac{C_{N_d} \rho A V_0^2}{2} \left[ a \sin P t' - \beta \cos P t' \right] \quad \text{(C.3-39)} \]

where \( \alpha \) and \( \beta \) are given by Eqs. (C.3-1) and (C.3-2). The notation \( t' \) is maintained in Eqs. (C.3-38) and (C.3-39) since it is not the true flight time.

Integrating Eqs. (C.3-38) and (C.3-39) gives the solutions for crossrange velocity and displacement brought about by oscillatory motion

\[ V_y = \frac{V}{V_o} \left\{ \frac{K_1}{\sqrt{\lambda_1^2 + (\omega_1 + P)^2}} \left[ e^{\lambda_1 t'} \sin[(\omega_1 + P)t' + \nu_1 + \phi_1] - \sin(\nu_1 + \phi_1) \right] \right. \]

\[ \left. + \frac{K_2}{\sqrt{\lambda_2^2 + (\omega_2 - P)^2}} \left[ e^{\lambda_2 t'} \sin[(\omega_2 - P)t' - \nu_2 + \phi_2] - \sin(\phi_2 - \nu_2) \right] \right\} \]

\[ + \frac{a_{\text{trim}}}{p} \frac{\sin P t'}{p} - \frac{\beta_{\text{trim}}}{p} \left( \cos P t' - 1 \right) + \frac{1}{h} \frac{d y}{d t} \]  

\quad \text{(C.3-40)}
\[ V_z = \frac{h}{V_0} \left\{ \frac{-K_1}{\sqrt{\lambda_1^2 + (\omega_1 + P)^2}} \left[ e^{\lambda_1 t'} \cos [(\omega_1 + P) t' + \nu_1 + \phi_1] - \cos (\nu_1 + \phi_1) \right] \right. \]

\[ - \frac{K_2}{\sqrt{\lambda_2^2 + (\omega_2 - P)^2}} \left[ e^{\lambda_2 t'} \cos [(\omega_2 - P) t' - \nu_2 + \phi_2] - \cos (\phi_2 - \nu_2) \right] \]

\[ - \frac{\beta_{\text{trim}}}{p} \sin P t' - \frac{a_{\text{trim}}}{p} \left( \cos P t' - 1 \right) + \frac{1}{h} \frac{dz}{dt} \bigg\} \]

\[ Y = h \left\{ \frac{K_1}{\lambda_1^2 + (\omega_1 + P)^2} e^{\lambda_1 t'} \sin [(\omega_1 + P) t' + \nu_1 + 2\phi_1] \right. \]

\[ - \frac{K_2}{\lambda_2^2 + (\omega_2 - P)^2} e^{\lambda_2 t'} \sin [(\omega_2 - P) t' - \nu_2 + 2\phi_2] \]

\[ - \left[ \frac{K_1 \sin (\nu_1 + \phi_1)}{\sqrt{\lambda_1^2 + (\omega_1 + P)^2}} - \frac{K_2 \sin (\phi_2 - \nu_2)}{\sqrt{\lambda_2^2 + (\omega_2 - P)^2}} - \frac{1}{h} \frac{dy}{dt} \bigg\} \right] \]

\[ - \frac{K_1 \sin (\nu_1 + 2\phi_1)}{\lambda_1^2 + (\omega_1 + P)^2} + \frac{K_2 \sin (2\phi_2 - \nu_2)}{\lambda_2^2 + (\omega_2 - P)^2} \]

\[ + a_{\text{trim}} \left[ \frac{1 - \cos P t'}{p^2} \right] + \beta_{\text{trim}} \left[ \frac{P t' - \sin P t'}{p^2} \right] \left\} \right. \]

(C.3-41)
\[
Z = h \left\{ \frac{-k_1}{\lambda_1^2 + (\omega_1 + P)^2} e^{\lambda_1 t'} \cos [(\omega_1 + P)t' + \nu_1 + 2\phi_1] \right. \\
- \frac{k_2}{\lambda_2^2 + (\omega_2 - P)^2} e^{\lambda_2 t'} \cos [(\omega_2 - P)t' - \nu_2 + 2\phi_2] \\
+ \left[ \frac{k_1 \cos (\nu_1 + \phi_1)}{\sqrt{\lambda_1^2 + (\omega_1 + P)^2}} + \frac{k_2 \cos (\phi_2 - \nu_2)}{\sqrt{\lambda_2^2 + (\omega_2 - P)^2}} + \frac{1}{h} \frac{dz}{dt} \right] t' \\
+ \left[ \frac{k_1 \cos (\nu_1 + 2\phi_1)}{\lambda_1^2 + (\omega_1 + P)^2} + \frac{k_2 \cos (2\phi_2 - \nu_2)}{\lambda_2^2 + (\omega_2 - P)^2} \right] t' \\
- \beta_{\text{trim}} \left[ \frac{1 - \cos Pt'}{p^2} \right] + \alpha_{\text{trim}} \left[ \frac{Pt' - \sin Pt'}{p^2} \right] \right\} \\
\text{(C.3-43)}
\]

where
\[
h = \begin{cases} 
\frac{C_{N_a} \rho A_g V_o^2}{2W} & \text{ICNCL} = 0 \\
\frac{C_{L_a} \rho A_g V_o^2}{2W} & \text{ICNCL} = 1
\end{cases} \\
\phi_1 = \sin^{-1} \frac{-(\omega_1 + P)}{\sqrt{\lambda_1^2 + (\omega_1 + P)^2}} = \cos^{-1} \frac{\lambda_1}{\sqrt{\lambda_1^2 + (\omega_1 + P)^2}} \text{ (C.3-45)} \\
\phi_2 = \sin^{-1} \frac{-(\omega_2 - P)}{\sqrt{\lambda_2^2 + (\omega_2 - P)^2}} = \cos^{-1} \frac{\lambda_2}{\sqrt{\lambda_2^2 + (\omega_2 - P)^2}} \text{ (C.3-46)}
\]

I. All numerical verification results presented in this appendix use \( C_{N_a} \). This is a worst case assumption for the reference projectile.
again take note of the respective quadrants for $\phi_1$ and $\phi_2$.

The "jump angle" is defined as the root sum of the squares of the coefficients of the linear term in Eqs. (C.3-42) and (C.3-43) divided by the velocity. It is

$$J_A = \sqrt{J_A^2 + J_A^2}$$  \hspace{1cm} (C.3-47)

where

$$J_A^2 = \frac{h}{V_0} \left[ \frac{K_2 \sin (\phi_2 - \nu_2)}{\sqrt{\lambda_2^2 + (\omega_2 - P)^2}} - \frac{K_1 \sin (\nu_1 + \phi_1)}{\sqrt{\lambda_1^2 + (\omega_1 + P)^2}} \right]$$  \hspace{1cm} (C.3-48)

$$J_A^2 = \frac{h}{V_0} \left[ \frac{K_1 \cos (\nu_1 + \phi_1)}{\sqrt{\lambda_1^2 + (\omega_1 + P)^2}} + \frac{K_2 \cos (\phi_2 - \nu_2)}{\sqrt{\lambda_2^2 + (\omega_2 - P)^2}} \right]$$  \hspace{1cm} (C.3-49)

C.3.2 Verification

Verification of the crossrange perturbation equations was established by comparison to four 6 DOF simulations. The results are summarized in Table C-3. In all cases, the analytic solution agrees to within 6%. An additional comparison is shown in Figure C-6, in which crossrange error due to trim angle of attack is shown as a function of spin rate. The analytic model is seen to be in excellent agreement with the 6 DOF calculations.

C.4 Downrange Perturbations

This section presents the downrange perturbation equations that are used to correct the basic particle trajectory. The development of the equations is discussed in Section C.4.1. Section C.4.2 discusses results which verify the equations.

C.4.1 Analytical Development

The downrange perturbation equations model the downrange effects of angle of attack oscillations. In addition to the assumptions stated in Section C.3.1, the following assumptions are required to achieve an accurate closed form solution:
TABLE C-3

TRAJECTORY MODEL VERIFICATION

\( V_0 = 11000 \text{ ft/sec} \)
Flight Time = 0.2 Sec

<table>
<thead>
<tr>
<th>CASE</th>
<th>( \theta_0 )</th>
<th>( \alpha )</th>
<th>( \omega_0 ) (RAD/SEC)</th>
<th>( \omega ) (RAD/SEC)</th>
<th>( t )</th>
<th>( t ) (RAD/SEC)</th>
<th>( P ) (RAD/SEC)</th>
<th>( X ) (FT)</th>
<th>( Y ) (FT)</th>
<th>( Z ) (FT)</th>
<th>( V ) (FT/SEC)</th>
<th>JUMP ANGLE (KILES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6 DOF</td>
<td>2091.4</td>
<td>0.0</td>
<td>0.0</td>
<td>9947.0</td>
</tr>
<tr>
<td>2</td>
<td>10°</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>6 DOF</td>
<td>2084.4</td>
<td>0.37</td>
<td>-0.13</td>
<td>9901.5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>6 DOF</td>
<td>2090.1</td>
<td>2.75</td>
<td>0</td>
<td>9931.8</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>-50</td>
<td>70</td>
<td>0.2</td>
<td>0</td>
<td>200</td>
<td>6 DOF</td>
<td>2089.7</td>
<td>-1.29</td>
<td>2.59</td>
<td>9931.8</td>
</tr>
</tbody>
</table>
Trim angles of attack cause negligible drag.
Downrange effects can be captured by mean drag coefficients.
Pitch oscillations are lightly damped.

In addition to the crossrange perturbation, oscillatory motion has a downrange effect. Angle of attack oscillations cause the projectile to experience a net increase in drag. The exact mechanism is illustrated in Figure C-7. As shown in Figure C-7(a), drag is defined as the force acting in line with the velocity vector. There are two contributions when the body is at angle of attack, a component of the normal force, \( C_N \), acts in the drag direction, increasing the drag force. The second component is the axial force coefficient, \( C_x \), which in general is a function of angle of attack, as illustrated in Figure C-7(b).

The drag coefficient is

\[
C_D = C_x \cos \alpha + C_N \sin \alpha \tag{C.4-1}
\]

which, for small angles, becomes

\[
C_D \approx C_x + C_N \alpha \tag{C.4-2}
\]

Modeling the axial force as parabolic in angle of attack,

\[
C_x = C_{x_0} + K C_x \alpha^2,
\]

and the normal force as linear in angle of attack (i.e., \( C_N = C_{N_0} \alpha \)) the drag coefficient becomes

\[
C_D = C_{x_0} + (C_{N_0} + K C_x) \alpha^2 \tag{C.4-3}
\]

The term \( C_{x_0} \) is the value of the drag coefficient at zero angle of attack. It is given by the drag model of Section C.2.1, Eq. (C.2-1). For the general case in which oscillations occur in both the pitch and yaw planes, the angle of attack in these expressions becomes the total angle of attack (i.e., the angle between the body axis and the velocity vector), denoted hereafter as \( \alpha \). For small angles

\[
\alpha^2 = a^2 + \beta^2 \tag{C.4-4}
\]
FIGURE C-7  DOWNRANGE FORCES DUE TO ANGLE OF ATTACK
where $\alpha$ and $\beta$ are, respectively, the pitch and yaw angles of attack in the body fixed coordinate system. Upon substitution of Eqs. (C.3-1) and (C.3-2) into Eq. (C.4-4), the total attack (with no trim) becomes

$$\bar{a}^2 = K_1^2 \frac{2\lambda_1 x}{V_o} + K_2^2 \frac{2\lambda_2 x}{V_o} + 2K_1K_2 e \frac{2\lambda_0 x}{V_o} \cos \left( \frac{2\omega_0 x}{V_o} + \nu_1 - \nu_2 \right)$$

(C.4-5)

where $t'$ has been replaced by $X/V_o$. The trim angle of attack is neglected so $\bar{a}^2$ may be expressed in a concise, workable form. This is a reasonable approximation since in reality the trim angles are small and do not have an appreciable effect on the drag.

Substitution of Eq. (C.4-5) into Eq. (C.4-3), yields the drag coefficient

$$C_D = C_{Dm} + \frac{K_D}{V^2} + (C_{Na} + K_{Cz}) \left[ \frac{2\lambda_1 x}{V_o} + \frac{2\lambda_2 x}{V_o} \right]$$

$$+ 2K_1K_2 e \frac{2\lambda_0 x}{V_o} \cos \left( \frac{2\omega_0 x}{V_o} + \nu_1 - \nu_2 \right)$$

(C.4-6)

The downrange equation of motion is

$$\frac{w}{\rho} \frac{d^2 x}{dt^2} = -\frac{\rho A V^2}{2} \left[ C_{Dm} + \frac{K_D}{V^2} + (C_{Na} + K_{Cz}) \left[ \frac{2\lambda_1 x}{V_o} + \frac{2\lambda_2 x}{V_o} \right]$$

$$+ 2K_1K_2 e \frac{2\lambda_0 x}{V_o} \cos \left( \frac{2\omega_0 x}{V_o} + \nu_1 - \nu_2 \right) \right]$$

(C.4-7)

which cannot be solved in closed form. As a result, perturbations are calculated for use as corrections. The perturbations are based on mean values of the drag coefficient with and without oscillations, where the mean drag coefficient is defined as
\[ C_D = \frac{\int_0^x C_D \, dx}{\int_0^x dx} \quad (C.4-8) \]

For the case without oscillations (zero angle of attack)

\[ C_D = \frac{\int_0^x \left( C_{D_0} + \frac{K_D}{v^2} \right) \, dx}{\int_0^x dx} \quad (C.4-9) \]

The equation of motion becomes

\[ \frac{w}{g} \frac{dV}{dt} = \frac{w}{g} \frac{dV}{dx} - \frac{pA}{2} \left( C_{D_0} + \frac{K_D}{v^2} \right) v^2 \quad (C.4-10) \]

or equivalently

\[ \left( C_{D_0} + \frac{K_D}{v^2} \right) dx = - \frac{2w}{\rho A g} \frac{dV}{V} \quad (C.4-11) \]

Substituting Eq. (C.4-11) into the Eq. (C.4-9) and integrating yields

\[ C_{D_0} = 0 = \frac{1}{x} \frac{2w}{\rho A g} \ln \frac{V_0}{V} \quad (C.4-12) \]

For the case with oscillatory motion, the mean drag coefficient is

\[ C_{D_a} = C_{D_a} + \frac{C_{N_a} + K_C x}{x} \int_0^x \bar{a}^2 \, dx \quad (C.4-13) \]

Upon substituting the expression for \( \bar{a}^2 \), Eq. (C.4-7), and integrating
The mean drag coefficients defined by Eqs. (C.4-12) and (C.4-14) are then used in constant drag coefficient equations for the motion:

$$V = V_0 e^{-\frac{\rho A_g}{2w} C_{Dx}}$$

(C.4-16)

$$t = \frac{2w}{\rho A_g} \frac{1}{V_0} \frac{1}{C_D} \left( e^{-\frac{\rho A_g}{2w} C_{Dx}} - 1 \right)$$

(C.4-17)

The two results are subtracted to compute the perturbation quantities $\Delta V$ and $\Delta t$:

$$\Delta V = V_a - V_{a=0} = V_0 \left[ e^{-\frac{\rho A_g}{2w} C_{Da}} - e^{-\frac{\rho A_g}{2w} C_{Da=0}} \right]$$

(C.4-18)

$$\Delta t = t_a - t_{a=0} = \frac{2w}{\rho A_g} \frac{1}{V_0} \left[ \frac{1}{C_{Da}} - 1 \right] - e^{-\frac{\rho A_g}{2w} C_{Da=0}}$$

(C.4-19)
For values of range, \( X \), prior to angle of attack convergence, these perturbations are applied directly to the velocity and flight time computed from the particle trajectory solution given in Section C.2, Eqs. (C.2-2) and (C.2-3). At values of range beyond the point of convergence, the particle trajectory solution from Eqs. (C.2-2) and (C.2-3) is restarted at the range of convergence, \( X_c \). The initial conditions corresponding to the velocity and time at the convergence range, \( X_c \), are the values for the particle trajectory equations (without angle of attack oscillations) perturbed by \( \Delta V \) and \( \Delta t \).

The range at convergence, \( X_c \), is found by determining the point at which the angle of attack oscillations decay to a point where the drag is no longer appreciably affected by angle of attack. From Eq. (C.4-5), the upper envelope of the oscillations is given approximately by

\[
\tilde{s}_{\text{upper}} = \frac{\lambda_0 x}{v_o} \left[ \frac{\Delta \lambda}{v_o} + \frac{\Delta \lambda}{v_o} \right]
\]

The damping is assumed to be small compared to the natural frequency, so that many oscillations are required to reach half amplitude. For the purposes of numerical calculations, experience has shown pitch oscillations beyond the point where \( \tilde{s}_{\text{upper}} = 0.5\text{o} \) have little effect on the trajectory. Thus, the range at convergence is defined as the point at which this occurs. (This is a computer code input parameter and can be easily changed to any other value.)

The downrange error, \( \Delta X \), at nominal time, \( t_n \), is computed by iterating the solution in range until the computed flight time matches \( t_n \). If desired, the iteration can be avoided by an additional approximation. The downrange error is closely approximated by

\[
\Delta x \approx (t_n - t) v
\]

C.4.2 Verification

Verification of the downrange perturbation equations was established by comparison to four numerical solutions of the
full equations of motion given in Table C-3. The downrange position agrees to within 0.05%. The downrange perturbation, \( \Delta X \) (the difference between the downrange position with oscillatory motion and the downrange position without oscillatory motion at the same flight time) agrees to approximately 10%.
This appendix presents the HITS\textsuperscript{1} program listing. The code was designed to facilitate conversion to real-time operation from remote key board terminals. The program is written in FORTRAN IV and requires less than 175 K bytes of computer memory on an IBM 360.

The code generates seven (7) warning level (i.e., Level 4) diagnostics when compiled on an IBM-360/75: four (4) in Subroutine DOABC and three (3) in Subroutine DO234. These are to be expected and ignored.

\footnote{HITS is maintained in the Avco engineering computer code library as Production Code 5127.}
**HYPERVELOCITY INFIGHT TRAJECTORY SCATTER CODE**

**PREPARED BY**
AVCO SYSTEMS DIVISION
201 LOWELL STREET
WILMINGTON, MASSACHUSETTS 01887

**PREPARED FOR**
RESEARCH DIRECTORATE
GENERAL THOMAS J. RODMAN LABORATORY
ROCK ISLAND ARSENAL
ROCK ISLAND, ILLINOIS 61201

**EFFECTIVE DATE - JANUARY 1976**
D.1 MAIN Program (continued)

IMPLICIT REAL*8 (A-H,O-Y)
LOGICAL *1 CARD(80)
COMMON /CARDCH/ B(200), C(5000), IA(200,2), IDI(10,2), INC.
COMMON /CSPRT/ IOPRNT, IOUT, IY, KARS1(3), KA, KB, KC, KD, KTL, K234, LABC1(3)
COMMON /CINM/ INOM
COMMON /CISPRNT/ ISPRNT
COMMON /IMCH/IDI(200), ID2(200), II(400), I2(400), IDT(10), IIT(20),
I11(10), I22(10), MTRIAL, NCELL, NCALC, KCOPT, IRJ1, IRJ2, ING, NCC, NR1, NR2
COMMON /HISPL/ IOPLOT, ISPL, ZTITLE(20,30), ZDATAD(21,4,30)
IOUT = 6
INC = 5
INC = 10
C REWIND INC
C WRITE (IOUT,9001)
C9001 FORMAT(* THE INPUT CARDS ... */)
C 1 READ (5,9002,END=2) CARD
C9002 FORMAT(80A1)
C WRITE (INC,9002) CARD
C WRITE (IOUT,9003) CARD
C9003 FORMAT(IX,80A1)
C GO TO 1
C 2 REWIND INC
8 CONTINUE
WRITE(IOUT,9005)
9005 FORMAT(IN1,/*)
A 3IT9, 4(*H*), T30, 4(*H*), T40, 25(*I1), T71, 25(*I1), T102, 25(*S1),/>
B10(T9, 4(*H*), T30, 4(*H*), T50, 4(*I1), T81, 4(*T*), T102, 4(*S1),/>
C 3IT9, 25(*H*), T30, 4(*H*), T50, 4(*I1), T81, 4(*T*), T102, 25(*S1),/>
D11(T9, 4(*H*), T30, 4(*H*), T50, 4(*I1), T81, 4(*T*), T102, 25(*S1),/>
E 3IT9, 4(*H*), T30, 4(*H*), T40, 25(*I1), T81, 4(*T*), T102, 25(*S1),/>
F1H0 *T15 *HYPER VELOCITY *T48, *IN AILERT *T78, *TRAJECTORY, *T110, *SCATHIT
25705082
G1* */*
H1H0, T55, *CCCC 00000 DDDD EEEEEE
I1H * T55, *C 0 0 D D E E
J1H * T55, *C 0 0 0 D E E
K1H * T55, *C 0 0 0 0 D E
L1H * T55, *CCCC 00000 DDDD EEEEEE
M/* */*
N1T55, *PREPARED BY, T85, *PREPARED FOR, */*
NT21 *AVCO SYSTEMS DIVISION, T91 *RESEARCH DIRECTORATE, */*
OT21, 201 LOWELL STREET T10, GENERAL THOMAS J. RODMAN LABORATORY, */*
P*/21, WILMINGTON, MASSACHUSETTS 01887, */*
Q* T91, ROCK ISLAND, ILLINOIS 61201, */*
RT50, *EFFECTIVE DATE - JANUARY 1976*
 DO 6 I=1,10
6 ID(1,2) = 0
C INOM USED IN S2987.
INOM = 1
C READ THE FIRST CARD AND PLACE THE PROPER SUBSCRIPTS INTO THE IA
C ARRAY.
CALL INITIL
D.1 MAIN Program (continued)

IF (WCOPT.EQ.0) GO TO 9
IF (IOPRINT.LT.3) IOPRINT=3
IF (WCALC.EQ.3) IOPRINT = 4
9 CONTINUE
C READ THE QE NAME, TYPE, NOMINAL VALUE, TOLERANCE, VARIANCE AND TABLE
C LENGTH AND PLACE INTO PROPER ARRAYS.
CALL INCARD
KSAVE = KC + 1
C MOVE EITHER THE TYPE 1 OR TYPES 2, 3, AND 4 UP TO THE START OF THE
C IA ARRAY.
CALL MOVEUP
C
C PRINT NAMES OF VARIABLES
C
520 FORMAT(IX)
501 FORMAT(IHI,39X,*** STOCHASTIC INDEPENDENT VARIABLES ***,//)
DO 500 IK = 1,K234
   IOES = IA(IK,1)
   WRITE(IOUT,520)
500 CONTINUE
C CALL FILLIN(IOES,IOUT)
505 CONTINUE
IF(KTI.EQ.0) GO TO 506
WRITE(IOUT,507)
507 FORMAT(IHI,39X,*** RANGE-CHECK INDEPENDENT VARIABLES ***,//)
DO 508 IK = 1,KT1
   IOES = IA(IK,1)
   WRITE(IOUT,520)
508 CONTINUE
C CALL FILLIN(IOES,IOUT)
506 CONTINUE
WRITE(IOUT,502)
502 FORMAT(IHI,47X,*** DEPENDENT VARIABLES ***,//)
DO 503 IK = 1,KD
   IOES = ID(IK,1)
   WRITE(IOUT,520)
503 CALL FILLIN(IOES,IOUT)
      WRITE(IOUT,509)
509 FORMAT(IHI,39X,*** VARIABLES RESET BY INPUT SEQUENCE ***,//)
DO 510 IK = 1,200
   IF(I(AIK,2).NE.-1) GO TO 510
   IOES = IA(IK,1)
   WRITE(IOUT,520)
510 CALL FILLIN(IOES,IOUT)
C CHECK COLUMN 2 OF IA FOR MISSING DATA.
C CALL CHKIA
C ARE WE RUNNING THE 3-DIM. MATRIX OR NON-TOL-VAR OR JUST NOMINAL VALUES
C IF((KTI.NE.0).OR.(K234.NE.0)) CALL EXTRA
IF (KTI.EQ.0) GO TO 10

HITS0101
HITS0102
HITS0103
HITS0104
HITS0105
HITS0106
HITS0107
HITS0108
HITS0109
HITS0110
HITS0111
HITS0112
HITS0113
HITS0114
HITS0115
HITS0116
HITS0117
HITS0118
HITS0119
HITS0120
HITS0121
HITS0122
HITS0123
HITS0124
HITS0125
HITS0126
HITS0127
HITS0128
HITS0129
HITS0130
HITS0131
HITS0132
HITS0133
HITS0134
HITS0135
HITS0136
HITS0137
HITS0138
HITS0139
HITS0140
HITS0141
HITS0142
HITS0143
HITS0144
HITS0145
HITS0146
HITS0147
HITS0148
HITS0149
HITS0150
D.1 MAIN Program (concluded)

INOM = 0
CALL DOABC
GO TO 30
10 IF ( K234 .EQ. 0 ) GO TO 20
ISAVIS = ISPRNT
ISPRNT = 0
CALL DO234
ISPRNT = ISAVIS
GO TO 30
20 CALL PICK1
30 CONTINUE
IF ( MCOPT .EQ. 0 ) GO TO 8
KC = KCSAVE
CALL MCR1
IF ( IOPLOT .EQ. 0 ) GO TO 8
CALL HYSPLT
GO TO 8
END
D.2 Subroutine ARTLU

SUBROUTINE ARTLU(X, Y, XT, YAT, YB, YBT, YC, YCT, YD, YT, YE, YFT, YG, YGT, YH, YHT, YI, YIT)

IMPLICIT REAL*8 (A-H, O-Z)

DIMENSION XT(2), YAT(2), YBT(2), YCT(2), YD(2), YT(2), YFT(2), YGT(2)

YHT(2), YIT(2)

N=J

DO 10 I=1, N

IF (X - XT(I+1)) 200, 200, 100

200 P = (X - XT(I)) / (XT(I+1) - XT(I))

GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), N

1 CONTINUE

9 YI=YIT(I)+P*(YIT(I+1)-YIT(I))

8 YH=YHT(I)+P*(YHT(I+1)-YHT(I))

7 YG=YGT(I)+P*(YGT(I+1)-YGT(I))

6 YF=YFT(I)+P*(YFT(I+1)-YFT(I))

5 YE=YTE(I)+P*(YTE(I+1)-YTE(I))

4 YD=YDT(I)+P*(YDT(I+1)-YDT(I))

3 YC=YCT(I)+P*(YCT(I+1)-YCT(I))

2 YB=YBT(I)+P*(YBT(I+1)-YBT(I))

1 YA=YAT(I)+P*(YAT(I+1)-YAT(I))

GO TO 300

10 CONTINUE

300 RETURN

END
D.4 Function A2F

```
FUNCTION A2F (X)
IMPLICIT REAL * 8 (A-H, O-Z)
COMMON/OCC/OA(O600) UTS0243
EQUVALENCE (V1,OE(1)),(V2,OE(2)),UITS0244
A(PHINW,OE(3)),(CD1,OE(5)) HITS0245
C(XN,OE(6)),(CO2,OE(7)) HITS0246
C(IN,OE(9)),(CNA,OE(10)) HITS0247
D(CM1,OE(12)),(CMA,OE(13)) HITS0248
E(PHITRI,OE(15)),(PHIANG,OE(17)) HITS0249
F(RATE,OE(18)),(PHIAT,OE(19)) HITS0250
G(PHIGNA,OE(21)),(A,OE(22)) HITS0251
H(D,OE(24)),(W,OE(25)) HITS0252
I(AIY,OE(27)),(P,OE(28)) HITS0253
J(CAA,OE(30)) HITS0254
EQUVALENCE (ALPHA,OE(400)),(ATOT,OE(401)) HITS0255
A(ALP,OE(402)),(ALP400,OE(403)) HITS0256
B(BET,OE(405)),(BETA,OE(406)) HITS0257
C(COT1,OE(407)),(COT2,OE(408)) HITS0258
D(DIAM,OE(411)),(DIAM,OE(412)) HITS0259
E(EHMTHA,OE(414)),(EHMTHA,OE(415)) HITS0260
F(ETRAN,OE(417)),(ETRAN,OE(418)) HITS0261
G(DEL,OE(420)),(DELV,OE(421)) HITS0262
H(DLX,OE(423)),(DLX,OE(424)) HITS0263
I(DOLX,OE(426)),(DOLX,OE(427)) HITS0264
J(DOL,OE(429)),(DOL,OE(430)) HITS0265
K(DK,OE(432)),(DK,OE(433)) HITS0266
L(DLD,OE(435)),(PSID,OE(436)) HITS0267
M(MPH,OE(438)),(MPH2,OE(439)) HITS0268
N(NCL,OE(441)),(N2,OE(442)) HITS0269
Q(QPA,OE(444)),(QTRIA,OE(445)) HITS0270
P(POL,OE(447)),(TS,OE(448)) HITS0271
Q(QT,OE(450)),(QT,OE(451)) HITS0272
R(ROL,OE(453)),(VC,OE(454)) HITS0273
S(SCL,OE(456)),(X2,OE(458)) HITS0274
EQUIVALENCE (WZ,OE(459)),(W,OE(460)) HITS0275
A(A1,OE(461)),(W2,OE(462)) HITS0276
B(BLAM,OE(464)),(BLAM,OE(465)) HITS0277
C(CN1,OE(467)),(CN2,OE(468)) HITS0278
D(DXU,OE(470)),(XJAY,OE(471)) HITS0279
E(EJXY,OE(473)),(EJZ,OE(474)) HITS0280
F(FALPH,OE(476)) HITS0281
```

```
XNMP = XNUL - XNU2 + PHI0
A2F = CAY1 * V0 * (CAY1 / 2.000 / XLM1 * (FEXP( 2.000 * XLM1 / XLM2 + 2.000 * X / V0 ) * DEXP( 2.000 * XLM2 + 2.000 * X / V0 + XNMP ) * DEXPS( 2.000 * XLM2 + 2.000 * X / V0 ) * X ) ) + CAY2 * CAY2 * V0 / XLM2 * (FEXP( 2.000 * XLM2 + 2.000 * X / V0 + XNMP ) * DEXP( 2.000 * XLM2 + 2.000 * X / V0 ) * X ) )
END
```

D.6 Subroutine CHCKIA

```
SUBROUTINE CHCKIA
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CARDCM/ R(20), C(500), IA(200*2), ID(I), K, INC,
1 IOUT, IY, KARS1, KA, KB, KC, KD, KT1, K274, LABC(3)
WRITE(IOUT,8)C2
B800 FORMAT(1H1,4X,4CH*** VARIABLES ASSIGNED PRESET VALUES ***/)
C IF COLUMN 2 OF IA HAS A ZERO, WE MUST FILL IN SOME DATA.
DO 10 I=1,KA
   IAT = IA(I,1)
   IF ( IA(I,2) .EQ. 0 ) CALL FILLIN(IAT,JKOUT)
10 CONTINUE
WRITE(IOUT,8)C1
B801 FORMAT(1H1)
RETURN
END
```

WIT50327
WIT50328
WIT50329
WIT50330
WIT50331
WIT50332
WIT50333
WIT50334
WIT50335
WIT50336
WIT50337
WIT50338
WIT50339
WIT50340
WIT50341
WIT50342
D.7 Subroutine CHKIN

SUBROUTINE CHKIN
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CARDCH/ 8(200),C(5000),IA(200),ID(10),INC,IPRINT,NDHIT
IIY,KARS1(3),KB,KC,KD,KT,1234,LAB(3)
COMMON/IMCH/1D(200),ID(200),III(400),II2(400),IOT(10),IIIT(20),IIR(1),IIR(2)
C(I),NTRIAL,NCELL,MCALC,MCOPT,IRJ1,IRJ2,IN8,NCC,NR1,NR2
DIMENSION IB(2,200)
EQUIVALENCE (IB(1,1),B(1))
C THIS SUBROUTINE CHECKS THE INPUTS AND
...TYPE 9 COUNTS THE VARIABLES
...TYPE 2 FORCES THE TOLERANCE TO BE A 3 SIGMA MINIMUM
...TYPE 4 DEFINES TOLERANCE=5*(XU-XL), MEAN=5*(XU+XL) AND PUTS
...THE MAXIMUM VALUE OF THE TAB. DIST. FUNCT. IN THE VAR
IN8=0
DO 2 I=1,KD
IF(ID(I,2))2,2,1
1 IN8=IN8+1
2 CONTINUE
DO 9 I=1,K234
IADD=IA(I,2)
ITYP=IB(I,IADD)
IF(ITYP=3,3,9,5)
3 FOR ITYP=2 IT IS GAUSSIAN IT, IV=8 ADDRESSES OF TOL AND VAR
IT=IADD+2
IV=IT+1
TOL=3.5*SORTB(IV)
9 IF(TOL-B(1))9,9,4
4 B(1)=TOL
GO TO 9
C TYPE 4 DISTRIBUTED VARIABLE ENTRY
VAR=1,
5 IT=IADD+4
ILT=IT+1,IT
8=IB(1,IT)
IV=ILT+IS-1
C IT IS THE ADDRESS IN B OF THE 1ST C ADDRESS FOR THE TAB. DIST. FUNCTION
ILT IS THE NUMBER OF ENTRIES IN C
IV IS THE LAST ADDRESS IN C OF THE DIST. FUNCTION.
DO 6 J=1,IV
IF(VAR=C(J))7,6,6
6 VAR=C(J)
7 CONTINUE
C VAR IS THE MAX. VALUE STORE IN VARIANCE POSITION OF B ENTRIES
C CALCULATE 1ST AND LAST ADDRESSES OF VARIANCE TABLE IN C
IT=IV+1
IV=IT+ILT-1
IADD=IADD+1
B(IADD)=.5*(C(IT)+C(IV))
IADD=IADD+1
D.7 Subroutine CHKIN (concluded)

```
B(IADD)=.S*(C(IV)-C(IT))
IADD=IADD+1
B(IADD)=VAR
9    CONTINUE
RETURN
END
```
D.8 Subroutine CNVRT

```
SUBROUTINE CNVRT(RN)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION R(11)
DATA R/0.0, 0.28, 0.36, 0.41, 0.46, 0.50, 0.54, 0.59, 0.64, 0.69, 1.0/
C THIS PROGRAM CONVERTS RECT. DIST. NOS. TO GAUSSIAN DIST. NOS.
AJ=10.*RN
J=AJ
BJ=J
RN=(AJ-BJ)*(R(J+2)-R(J+1))+R(J+1)
RETURN
END
```

D.9 Subroutine CROSS

SUBROUTINE CROSS(X, V )

[Explicit REAL Variables] [HITS0410]

COMMON /FVE/OE(500)

EQUVALENCE (VC, OE(101)), (WIND, OE(102))

ALPHAT, OE(173), (RH0, OE(104)), (CD1, OE(105))

AV1, OE(176), (C02, OE(107)), (V2, OE(108))

CXN, OE(159), (CNA, OE(110)), (SN, OE(111))

DCNO, OE(112), (CMPA, OE(113)), (TRIM, OE(114))

EFPMET0, OE(115), (ANGC, OE(116)), (PHIANG, OE(117))

FCSF0, OE(114), (CMAT, OE(119)), (GAM0, OE(120))

GTSUM, OE(121), (A, OE(122)), (FLL, OE(123))

HE, OE(124), (#, OE(125)), (AIK, OE(126))

IAB, OE(127), (P, OE(128)), (ALPCON, OE(129))

JICAA, OE(130)

EQUVALENCE (IDUMP, OE(204)) [HITS0423]

ALPHA, OE(400), (ATOM, OE(401)) [HITS0424]

BETA, OE(402), (ALPHA, OE(403)) [HITS0425]

BETT0, OE(405), (BETA, OE(406)) [HITS0426]

COTA, OE(408), (ALPHA, OE(409)) [HITS0427]

CNET, OE(411), (CD0, OE(412)) [HITS0428]

CMHT0, OE(414), (CMHTA, OE(415)) [HITS0429]

DACA, OE(417), (CDAOB, OE(418)) [HITS0430]

DELT, OE(420), (DELV, OE(421)) [HITS0431]

MODLX, OE(423), (DYOX, OE(424)) [HITS0432]

DNO, OE(427), (DZTX, OE(428)) [HITS0433]

EYEP, OE(429), (EYEP, OE(430)) [HITS0434]

EMF, OE(432), (EDLT, OE(433)) [HITS0435]

FGR, OE(435), (FGR0, OE(436)) [HITS0436]

GPHI1, OE(438), (PHI12, OE(439)) [HITS0437]

GRI, OE(441), (G42, OE(442)) [HITS0438]

HRT, OE(444), (HRIV, OE(445)) [HITS0439]

IR, OE(447), (IR, OE(448)) [HITS0440]

LTV, OE(450), (LTV0, OE(451)) [HITS0441]

MLT, OE(453), (MLT, OE(454)) [HITS0442]

MUVX, OE(456), (MUVX, OE(457)) [HITS0443]

EQUVALENCE (WZ, OE(459)), (W0, OE(460)) [HITS0444]

PW, OE(461), (#2, OE(462)) [HITS0445]

XLAM0, OE(464), (XLAM0, OE(465)) [HITS0446]

XNU0, OE(467), (XNU02, OE(468)) [HITS0447]

XNUM, OE(470), (XJAY, OE(471)) [HITS0448]

XJX, OE(473), (XJX, OE(474)) [HITS0449]

FA, OE(476), (FA, OE(477)) [HITS0450]

TP = X/V0

TL1 = XLAM0 * TP

TL2 = XLAM0 * TP

ELT1 = FEXP(TL1)

ELT2 = FEXP(TL2)

W1T = W1 * TP

W1T = W1 * TP + XNU0

W2T = W2 * TP
D.9 Subroutine CROSS (continued)

```fortran
1000 FORMAT(1X * TP ALPHA BETA ATOT VCONZ VCONY VCONX VCONH VCONW)
2/IX, IP10E12+4 // )
4 ALPHA = ELT1*CAY1*DSIN(W1TV) -$ ELT2*CAY2
1 BETA = ELT1*CAY1*DCOS(W1TV) + ELT2*CAY2
1 TP2 = TP * TP
A2 = A*00
B2 = B*00
IF( DABS(ALPHA) < GT. 1.0-38) A2= ALPHABETA
IF( DABS(BETA) < GT. 1.0-38) B2=BETA*BETA
ATOT= DSORT( A2*B2)
P2 = P * P
P*P = P*P
IF( P < EQ. * ) GO TO 1
SINPT = DSIN( PTP )
COSPT = DCOS( PTP )
VCONY = (ALPHAP2 + SINPT - BETTRM. *(COSPT - 1.000)) / P
VCONZ = -(BETTRM. + SINPT - ALTRM. *(COSPT - 1.000)) / P
YCONA = ALTRM. *(1.000 - COSPT) + BETTRM. *(PTP-SINPT) / P
ZCONA = -(BETTRM. *(1.000 - COSPT) + ALTRM. *(PTP-SINPT)) / P
GO TO 2
1 CONTINUE
VCONY = TP * ALTRM
VCONZ = -BETTRM * TP
YCONA = TP2 * ALTRM / 2.000
ZCONA = -TP2 * BETTRM / 2.000
2 CONTINUE
HV = H / V
HV = HV * V
ELWP1 = XLAM1 ** 2 + (W1 + P) ** 2
ELWP2 = XLAM2 ** 2 + (W2 - P) ** 2
CAY1LW = CAY1 / DSORT(ILWP1)
CAY2LW = CAY2 / DSORT(ILWP2)
XNUPH1 = XNU1 + PHI1
XNUPH2 = XNU2 + PHI2
WPTVP1 = (W1 + P) * TP + XNUPH1
WPTVP2 = (W2 - P) * TP + XNUPH2
COSNP1 = DCOS(XNUPH1)
COSNP2 = DCOS(XNUPH2)
SINNP1 = DSIN(XNUPH1)
SINNP2 = DSIN(XNUPH2)
VYA = HVV *( CAY1LW *(ELT1 * DSIN( WPTVP1) - SINNP1 ) )
1 -CAY2LW *(ELT2 * DSIN( WPTVP2) - SINNP2 )
2 + VCONY + DYDT0 / H )
VZA = HVV *( -CAY1LW *(ELT1 * DCOS( WPTVP1) - COSNP1 ) )
1 -CAY2LW *(ELT2 * DCOS( WPTVP2) - COSNP2 )
```
D.9 Subroutine CROSS (concluded)

2 + YCONZ + DZDTH/H 
WCON1 = WPTVPI + PHI1 
WCON2 = WPTVPI + PHI2 
IF( IDUMP = 1 ) WRITE(*,1) TP, ALPHA, BETA, ATOT
1. VCCNY, YCONZ, YCONA, ZCONA, VYA, VZA
YA = H * ( CAY1/EWLP1 * ELT1 * DSIN(WCON1) - CAY2/EWLP2 * ELT2)
1 * DSIN(WCON1) - (CAY1LW * DSIN(XNUPHI) - CAY2LW * DSIN(XNUPHI)) - DYDT 
1 * DSIN(WCON2) - (CAY1LW * DSIN(XNUPHI) - CAY2LW * DSIN(XNUPHI)) - DYDT
2) /H ) * TP - CAY1/EWLP1 * DSIN(XNUPHI + PHI1) + CAY2/EWLP2 * DSIN(WCON1)
3 (XNUPH2 + PHI2) + YCONA )
ZA = 4 * (-CAY1/EWLP1 ) 
ELT1 * DCOS(1.5/4N1 ) - CAY2/EWLP2 * ELT2
1 * DCOS(WCON1) + (CAY1LW * DCOS(XNUPHI) + CAY2LW * DCOS(XNUPHI) + DZDTH1 )
2 /H ) * TP + CAY1/EWLP1 * DCOS(XNUPHI + PHI1) + CAY2/EWLP2 * DCOS(WCON2)
3 (XNUPH2 + PHI2) + ZCONA )
EOLT = FEXP( ALMAO * TP) 
EOLT = DEXP( DELLA * TP) 
ALPMA = EOLT * ( CAY1 * EOLT + CAY2 / EOLT ) 
ALPMIN = EOLT * ( CAY1 * EOLT - CAY2 / EOLT ) 
XJAY = -4 * ( CAY1LW * SINNP1 - CAY2LW * SINNP2 ) * 1.* D3 
XJAZ = -4 * ( CAY1LW * COSNP1 + CAY2LW * COSNP2 ) * 1.* D3 
XJA = DSORT( XJAY ** 2 + XJAZ ** 2 )
IF( IDUMP = 1 ) WRITE(*,1) YA, ZA, EOLT, EDLT
1. ALPMA, ALPMIN, XJAY, XJAZ, XJA
1001 FORMAT( 8X,10H,10H )
1002 FORMAT( 8X,10H,10H )
24 RETURN 
END
D.10 Subroutine CROSSF

SUBROUTINE CROSSF( X, V )

IMPLICIT REAL *8 (A-H, O-Z)

COMMON/DE/DE(650)

EQUIVALENCE (VO, OE( 1)), (VWIND, OE( 2)), (RHOD, OE( 4)), (CD1, OE( 5)), (RHO, OE( 4)), (CDII, OE( 5)), (CD2, OE( 7)), (V2, OE( 8)), (CNA, OE( 13)), (SM, OE( 11)), (CMPA, OE( 13)), (PRIM, OE( 14)), (ANQO, OE( 15)), (PHIANG, OE( 17)), (PHIT, OE( 18)), (PHIRT, OE( 19)), (GANO, OE( 20)), (A, OE( 22)), (ELL, OE( 23)), (W, OE( 25)), (AIX, OE( 26)), (P, OE( 28)), (ALPCON, OE( 29)), (IDUMP, OE( 204)), (ATOT, OE( 401))

EQUIVALENCE (ALPHA, OE( 400)), (ATOT, OE( 401))

A(432), (ALPHD0, OE( 403)), (B, OE( 404))

B(235), (SETA, OE( 405)), (BETADO, OE( 407))

C(408), (ALPMAX, OE( 409)), (ALPMIN, OE( 410))

C(411), (CD8, OE( 412)), (CNA, OE( 413))

E(144), (CMHTA, OE( 415)), (CAYI, OE( 416))

F(417), (CDAOB, OE( 418)), (CDAB, OE( 419))

G(420), (DELV, OE( 421)), (DELT, OE( 422))

H(423), (DYDX, OE( 424)), (DZDX, OE( 425))

I(426), (CDT, OE( 427)), (DELW, OE( 428))

J(429), (EYE, OE( 430)), (EMP, OE( 431))

K(432), (FDLT, OE( 433)), (F, OE( 434))

L(435), (PSID, OE( 436)), (PHI0, OE( 437))

M(438), (PHI2, OE( 439)), (PS10, OE( 440))

N(441), (P2, OE( 442)), (R3, OE( 443))

O(444), (RTRM, OE( 445)), (TC, OE( 446))

P(447), (TS, OE( 448)), (TGO, OE( 449))

Q(450), (THETAO, OE( 451)), (THETD, OE( 452))

R(453), (VC, OE( 454)), (VC0, OE( 455))

S(456), (VZ, OE( 457)), (WX, OE( 458))

EQUIVALENCE (WZ, OE( 459)), (W0, OE( 460))

A(461), (W2, OE( 462)), (WL2, OE( 463))

B(464), (XLM1, OE( 465)), (XLM2, OE( 466))

C(467), (XNU1, OE( 468)), (XG, OE( 469))

D(470), (XJAY, OE( 471)), (XJAZ, OE( 472))

E(473), (YA, OE( 474)), (ZA, OE( 475))

F(476)

TP = X/VO

TL1 = XLM1 * TP

TL2 = XLM2 * TP

FLEX1 = FEXP(TL1)

FLEX2 = FEXP(TL2)

W1T = W1 * TP

W1T = W1T * XNU1
D10 Subroutine CROSS (continued)
D.10 Subroutine CROSSF (concluded)

2 + VCONZ + DZDT0/ H )
WCON1 = WPTVP1 + PHI1
WCON2 = WPTVP2 + PHI2
IF ( IDUMP YEQ Y1 ) WRITE(6,1001) TP,ALPHA,BETA,ATOT
     YA = H *( CAY1/ ELW1* ELT1 * DSIN(WCON1) )-CAY2/ELWP2* ELT2
1 * DSIN(WCON2)- (CAY1LW * DSIN( XNUPHI) -CAY2LW * DSIN(XNUPH2)-DYDT
20 /H ) * TP -CAY1/ ELW1 * DSIN(XNUPHI+PHI1) +CAY2/ELWP2 * DSIN
3(XNUPH2 +PHI2) + YCONA )
ZA = H*(CAY1/ELW1* ELT1 * DCOS(WCON1) )-CAY2/ELWP2* ELT2
1 * DCOS(WCON2)- (CAY1LW * DCOS(XNUPHI) +CAY2LW * DCOS(XNUPH2)+DZDT
20 /H ) * TP + CAY1/ELWP1 * DCOS(XNUPHI+PHI1) +CAY2/ELWP2 * DCOS
3(XNUPH2 +PHI2) + ZCONA )
EOLT = FEXP( XLAM0 * TP)
EDLT = DEXP( DELLAW* TP)
ALPmax = EOLT *( CAY1 * EDLT + CAY2 / EDLT )
ALPmin = EOLT *( CAY1 * EDLT - CAY2 / EDLT )
XJAY = -HV * ( CAYILW * S1NPHI -CAY2LW * S1NP2 )1.03
XJAZ = HV * ( CAYILW * COSPHI1 +CAY2LW * COSPH2 )1.03
XJA = DSQRT( XJAY ** 2 + XJAZ ** 2 )
IF (IDUMP YEQ 1 ) WRITE(6,1001) "A", ZA, EOLT, EDLT
1 * ALPmax, ALPmin, XJAY, XJAZ, XJA
1001 FORMAT( * YA ZA EDLT EDLT ALPHI)
I MAX ALPmin XJAY XJAZ XJA */IX,1P11E12.
24 /*
RETURN
END

HITS0637
HITS0638
HITS0639
HITS0640
HITS0641
HITS0642
HITS0643
HITS0644
HITS0645
HITS0646
HITS0647
HITS0648
HITS0649
HITS0650
HITS0651
HITS0652
HITS0653
HITS0654
HITS0655
HITS0656
HITS0657
HITS0658
HITS0659
HITS0660
HITS0661
HITS0662
HITS0663
D.11 Subroutine CRVFT

SUBROUTINE CRVFT(KOPT)
IMPLICIT REAL*8(A-H,O-Z)
COMMON/CARDCM/8(230),C(500),IA(200),ID(10,2),INC,IOPRNT,IOUT
K...KAVSN(3),KAK,KB,KC,KD,KTI.K234,LABC(3)
COMMON/DEC/660
DIMENSION DELX(20),FUN(10)
C THIS ROUTINE USES THE T.S. EXPANSIONS OF YP-58,YP3
C USES 1ST,2ND UNMIXED OR FULL 2ND ORDER FOR KOPT =1,2,3 RESP.
C CREATE DELX SAMPLE TABLE
DO 1 I=1,K234
JB=IA(I,2)+1
IOE=IA(I,1)
1 DELX(I)=OE(IOE)-8(JB)
C INITIALIZE EXPANSIONS SUMS TO NOMINAL VALUES
DO 2 I=1,KD
FUN(I)=C(KC+I-1)
2 START LOOP ON INDEPENDENT UNMIXED DERIVATIVE SUMS
DO 4 I=1,K234
I1=KC+KD*(2*I-1)-1
I2=I1+KD
GO TO(12,11,11),KOPT
11 DO 13 J=1,KD
13 F...UN(J)=UN(J)+DELX(I)*C(I1+J)
GO TO 4
11 CONTINUE
10 END OF LOOP --CHECK FOR INCLUSION OF MIXED DERIVATIVES
IF(KOPT=2)9,9,5
19 I1=I2+3*KD
2 I2=K234-1
DO 8 I=1,12
J3=I+1
DO 7 J=J3,K234
3 L=I+KD
6 FUN(L)=UN(L)+DELX(I)*DELX(J)*C(I1+L)
7 I1=I+KD
8 CONTINUE
9 STORE RESULTS IN DE ARRAY
DO 10 I=1,KD
DE(IOE)=FUN(I)
RETURN
END
D.12 Subroutine DOABC

SUBROUTINE DOABC
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL*1 BL
LOGICAL*1 OTT(10) /10H1234567890/
LOGICAL*1 FMT1(46) /46H ' IND. VAR.', (A1,OE('4',4,'14','1'),1PE12.4)/
A/
LOGICAL*1 FMT2(29) /29H('3X', (A1,OE('4','4',)) ')/
LOGICAL*1 FMT3(18) /18H('5X','1P E12.4',)/
COMMON /CARDOM/, B(202), C(5000), IA(200,2), ID(17,2), INC.
COMMON/IDC/IDC(1), KAS(3), KA, KB, KC, KD, KTL, K234, LABC(3)
COMMON/IDC/IDC(1)
COMMON /CISPROM/ ISPRNT
DIMENSION IOES(3), JC(3), IOE(2, 600), IB(2,200),
ABCOE(3), OUTOE(10)
EQUIVALENCE (IOES(1), IA(1), IA(1), IBOE), (IOES(2), IBOE), (IOES(3), ICOE),
(JC(1), JCA), (JC(2), JCB), (JC(3), JCC), (OE(1), IOE(1,1)),
IF(ISPRNT .EQ. 0) WRITE(IOUT,500)
500 FORMAT(1H1,39X,*** RANGE CHECK INDEPENDENT VARIABLE COMBINATIONS
****,///) 501 FORMAT(1H1,39X, '*** RANGE CHECK INDEPENDENT VARIABLE COMBINATION
****,///)
C SET UP THE FORMATS.
FMT1(15) = OTT(KTL)
FMT2(5) = OTT(KD)
FMT3(3) = OTT(KD)
IF ( KD LT 10 ) GO TO 2
FMT2(5) = OTT(11)
FMT3(3) = OTT(11)
2 CONTINUE
C IOES(I) IS THE OE SUBSCRIPT FOR THE I,TH TYPE 1 ARRAY.
C JC(I) IS ONE LESS THAN THE ADDRESS IN C FOR THE I,TH TYPE 1 ARRAY.
C THE OE SUBSCRIPTS FOR THE TYPE 1 ARRAYS HAVE BEEN MOVED UP TO THE
C START OF THE IA ARRAY.
C GE, THE OE SUBSCRIPTS AND THEIR C ADDRESS.
DO 5 I=1,KT1
IOES(I) = IA(I,1)
JB = IA(I,2)
5 JC(I) = IB(I,JB+2) - 1
IAR = 0
10 IAR = IAR + 1
IF ( IA(QE .GT. 0 ) ) GO TO 12
L = -IA(E
IOE(I,L) = C(JCA+IAR)
GO TO 15
12 QE(I(IAE)) = C(JCA+IAR)
15 ABCOE(I) = C(JCA+IAR)
C IF THERE IS ONLY ONE TYPE 1 ARRAY, IBR=1 TO GET PAST 3 IF'S AT END.
IAR = 1
IF ( KT1 LT 2 ) GO TO 40
HITS0726
HITS0727
HITS0728
HITS0729
HITS0730
HITS0731
HITS0732
HITS0733
HITS0734
HITS0735
HITS0736
HITS0737
HITS0738
HITS0739
HITS0740
HITS0741
HITS0742
HITS0743
HITS0744
HITS0745
HITS0746
HITS0747
HITS0748
HITS0749
HITS0750
HITS0751
HITS0752
HITS0753
HITS0754
HITS0755
HITS0756
HITS0757
D.12 Subroutine DOABC (concluded)

```
GO TO 21
27  IRR = IBR + 1
21  IF ( I9OE .GT. 0 ) GO TO 22
    L = -I9OE
    ICE(1,L) = C(JCB+IBR)
    GO TO 25
22  OEO(I9OE) = C(JCR+IBR)
25  ABCOE(I) = C(JCB+IBR)
C IN THE EVENT THERE IS NO THIRD TYPE 1 ARRAY, ICR=1 WILL GET US PAST
C THE 3 IF'S AT THE END.
    ICR = 1
    IF ( KT1 .LT. 3 ) GO TO 40
    GO TO 31
31  ICR = ICR + 1
30  IF ( I9OE .GT. 0 ) GO TO 32
    L = -I9OE
    OEO(1,L) = C(JCC+ICR)
    GO TO 35
32  OEO(I9OE) = C(JCC+ICR)
35  ABCOE(I) = C(JCC+ICR)
40  CALL PICK!
    IF(ISPRINT.EQ.1) WRITE(IOUT,501)
    WRITE (IOUT,FMT1) (BL, IA(I,1), ABCOE(I), J=1,KT1)
    DO 50 I=1,KD
    JOE = 10(I,1)
50  WRITE (IOUT,FMT2) (BL, JOE(I,1), J=1,KD)
      WRITE (IOUT,FMT3) (OUTOE(I), J=1,KD)
      IF ( ICR .LT. LABC3 ) GO TO 30
      IF ( I3R .LT. LABC2 ) GO TO 20
      IF ( IAR .LT. LABC1 ) GO TO 10
RETURN
END
```
D.13 Subroutine DO234

SUBROUTINE DO234
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL*1 OTT(10),/10H1234567890/  
LOGICAL*1 FMT2(29),/29H2(5X, (A1,OE(','),I4,') ')./)  
LOGICAL*1 FMT3(14),/14H5X,1P  E12,4/)  
DIMENSION IBB(2,200)  
COMMON /CARDCM/ B(200), C(500), IA(200), ID(10,2), INC.
1 IOPRT, IOUT, IY, KARS1(3), KA, KB, KC, KD, KT1, K234, LABC(3)
COMMON/DEC/DE(600)
EQUIVALENCE (IBB(1,1),B(1))
DIMENSION IOES(2), LN(2)
EQUIVALENCE (IOES(1), IOE1), (IOES(2), IOE2), (LN(1), L1),
(I(LN(2), L2))
COMM/INDEX/JO,1234,ICYBAR,ICSUMS,ICDGUB,KCOV
IJB1 = 0
IJB2 = 0
IJB3 = 0
IJB4 = 0
DO 20 JBD = 1,KD
IJB = ID(JBD,1)  
IF(IJB.EQ.0) IJB1 = JBD
IF(IJB.EQ.30) IJB2 = JBD
IF(IJB.EQ.31) IJB3 = JBD
IF(IJB.EQ.32) IJB4 = JBD
20 CONTINUE
IJB = IJB1 + IJB2 + IJB3 + IJB4
C SET UP THE FORMATS.
FMT2(6) = OTT(KD)  
FMT3(8) = OTT(KD)  
IF (KD .LT. 1 ) GO TO 2
FMT2(5) = OTT(1)  
FMT3(7) = OTT(1)  
2 CONTINUE
C KYBAR IS ONE LESS THAN THE START OF THE YBAR OUTPUT IN THE C ARRAY.
KYBAR = KC
IF((IJB.GT.0).AND.(IDUMP.EQ.0)) GO TO 201
WRITE (IOUT,90033) (BL, IAI(1), OEIAI(1,1)), I=1,K234)
9003 FORMAT('INOMINAL CASE, INDEPENDENT VARIABLES ***\// ((5X, 5(A1, '
1 * 'DEF', 'I4,') = ', 1PE12,4))')
201 CONTINUE
C THE NOMINAL VALUES.
CALL PICK1
CALL STOREC
K = KYBAR - KD
IF((IJB.GT.0).AND.(IDUMP.EQ.0)) GO TO 801
WRITE (IOUT,9005)  
9005 FORMAT('INOMINAL CASE, DEPENDENT VARIABLES ****\//
WRITE (IOUT,FMT2) (BL, IAI(I,1), I=1,KD)
HITS0791
HITS0792
HITS0793
HITS0794
HITS0795
HITS0796
HITS0797
HITS0798
HITS0799
HITS0800
HITS0801
HITS0802
HITS0803
HITS0804
HITS0805
HITS0806
HITS0807
HITS0808
HITS0809
HITS0810
HITS0811
HITS0812
HITS0813
HITS0814
HITS0815
HITS0816
HITS0817
HITS0818
HITS0819
HITS0820
HITS0821
HITS0822
HITS0823
HITS0824
HITS0825
HITS0826
HITS0827
HITS0828
HITS0829
HITS0830
HITS0831
HITS0832
HITS0833
HITS0834
HITS0835
HITS0836
HITS0837
HITS0838
HITS0839
HITS0840
D.13 Subroutine DO234 (continued)

WRITE (IOUT,FMT3) (C(K+1), I=1,KD)

C INDEPENDENT VARIABLES.
C CHANGE THEM ONE AT A TIME.
C
WRITE (IOUT,9006) (BL, I(I), I=1,KD)
9006 FORMAT('ILIMITING CASES', /5X, 'INDEPENDENT VARIABLE', 48X,
1 'DEPENDENT VARIABLE'// /10X, 'NAME', 'VALUE', 'TOLERANCE', 5X,
2 8(A1, 'OE('', I4, '))')//35X,2(A1, 'OE('', I4, '))')

R-1 CONTINUE
DO 10 I=1,K234
10 DE = IA(I,1)
JBC = IA(I,2)
DE(IEO) = B(JB+1) + B(JB+2)
CALL PICK1
CALL STOREC
K = KC - KD
IF((IJG.GT.0).AND.((IDUMP.EQ.0))) GO TO 202
WRITE (IOUT,9007) IOE, B(JB+1), B(JB+2), (C(K+L), L=1,KD)
9007 FORMAT('K', '15.15', '1P2E12.4', '4X', '8E12.4//' /34X,2E12.4//' )

R-2 CONTINUE
DE(IEO) = B(JB+1) - B(JB+2)
CALL PICK1
CALL STOREC
K = KC - KD
IF((IJG.GT.0).AND.((IDUMP.EQ.0))) GO TO 203
WRITE (IOUT,9008) (C(K+L), L=1,KD)
9008 FORMAT('34X, 1P2E12.4//' /34X, 2E12.4//' )

R-1 CONTINUE
C RESTORE OF(IOE) TO ITS NOMINAL VALUE.
DE(IEO) = B(JB+1)
1: CONTINUE
C IF IOPRNT=1, WE STOP HERE.
KMBB = KCYBAR
IF (IOPRNT.EQ.1) GO TO 900

C GET THE FIRST AND SECOND UNMIXED PARTIALS AND STORE THEM IN C OVER THE
C KPAR IS ONE LESS THAN THE STARTING SUBSCRIPT IN THE C ARRAY OF THE
C PARTIAL DERIVATIVE (Y(J)PLUS - Y(J)MINUS) / (2*DELX(1))
C
KCYBAR = KCYBAR + KD
KD2 = 2 * KD
KYP = KCYBAR - KD
KRM = KCYBAR
IF((IJG.GT.0).AND.((IDUMP.EQ.0))) GO TO 204
WRITE (IOUT,9009) (D(I+1), I=1,KD)
9009 FORMAT('IOERIVATIVES', 5X, 'IN', 5X, 'VAR', 5X, 'DEPENDENT VARIABLE'
1 /8X, '1:112'// )

204 CONTINUE
C IF (IDUMP.EQ.0) GO TO 2000
DIMENSION IRA(2,2)
*o00@0C0"~aa~~a*.a-"

vo" 0

NNNNN NNNMMPAMP MMVMM

00

0t

-y

0

X (I

N 40

No0t
0

CL M~ N

0.*o
V

00

Vo

*

D

6)4

MVa

to

4
"a

00~

9-sMm

4

0o4

'-0

1ff00.N

~
*rl

0.~u11*"
- ( I-

19
OV

4J

e"aN).VX. w X

0

4Ca".a-"00

'

Fý1 . 1,i f- a I.-0
v

ON-

> W4W<WW4WW-

n
p-

r-

oil

x(ill

0'

NO

+

-W

0
00

)0

0

o-

W.3U~~3

1.1

*

le-r.pop'
ONOaa0 ++ +

)f1,-*

N.

90.
I Pa
0.00-."a

N

ftoft~t(...Z

M W

0

I

"a

&o%

~ ~IA~N-~aU

o

1L

Ulu

0

i

)

X.O4 Ir-

110

V0V

J0

0.UJ

: C- :)c

le

0

LV4pUU C)

U-I

44"

0

0-

moo
0.(fM

ri0..-.

0 +

N.82
+0..C
+ +

4DVt i,

19W11110

)P-UuwvIL3ILUuv'
>u
4
N
'~

LI

99

4.

a9C~0

00.

V

X'

++
,W

N

2

0%IIV
O

Valevowl
0

u

HIt It

v"IbvVv


D.13 Subroutine DO234 (continued)

SYJ = SYJ + C(KRMJ) * VAR
SYJS = SYJS + C(KRP+J)**2 * VAR

4 CONTINUE
C(KCMSJ) = C(KCYB) + 0.500 * SYJ
C(KCMSJ+KD) = SYJS
5 CONTINUE
 IF((IJ) GT 0) AND (IDUMP.EQ.0) GO TO 256
C PRINT E(Y(J)) AND E(Y(J)**2).
WRITE(10UT,6969)
WRITE (1OUT,9012) (I)(I,1), I=1,KD)
FORMAT (112 2ND ORDER MEANS AND 1ST ORDER VARIANCES // 62X,
A*DEPENDENT VARIABLES // 8X, 10I12)
WRITE (1OUT,9013) (C(KCMSUMS+I), I=1,KD)
FORMAT(11H: 4X,5HMEANS,IX,IP10E12,4)
WRITE (1OUT,9014) (C(KCMSUMS+KD+I), I=1,KD)
FORMAT(11H: 9HVARIANCES,IX,IP10E12,4)
DO 75,75 = 1,KD
75 CALL STODEVIC(KCMSUMS+KD+I),I,KD,1OUT)
256 CONTINUE
JOB = KCMSUMS
KMBO = KCMYBAR
IF (IOPRINT .EQ. 3) GO TO 970
C UPDATE K C DUE TO E(Y(J)) AND E(Y(J)**2).
KC = KC + 2 * KD
KDOUT = KC
C VARY THE INDEPENDENT VARIABLES TWO AT A TIME.
IF((IJ3 GT 0) AND (IDUMP.EQ.0)) GO TO 2C7
WRITE (1OUT,9014) (I)(I,1), I=1,KD)
FORMAT(11H: 2ND ORDER MIXED PARTIALS; // * IND. VAR.** 50X,
*DEPENDENT VARIABLES // (8X, 10I12)
2C7 CONTINUE
C KCDR WILL STORE THE SECOND DERIVATIVES.
KCDR = KCDOUT - KD
KM1 = K234 - 1
KD3 = 3 * KD
DO 90 I=1,KM1
IDOE = IA(I,1)
IR = IA(I,2)
IP1 = I + 1
DO 90 J=IP1,K234
JOE = IA(J,1)
JR = IA(J,2)
BB = 3(I+2) * B(J+2)
C THE NEXT DOUBLE DO LOOP WILL CREATE THE FOUR COMBINATIONS OF NOMINAL
C VALUE PLUS AND MINUS TOLERANCE.
DO 60 I2=1,2
OE10F = B(I+1) + (-1.000)**(3-12) * B(I+2) / 2.000
DO 60 J2=1,2
OE10F = 3(J+1) + (-1.000)**(3-J2) * B(J+2) / 2.000
60 CALL DPK1
D.13 Subroutine DO234 (continued)

CALL STOREC
60 CONTINUE
KDOR = KCDR + KD
C THE COMBINATIONS OF PLUS AND MINUS ARE KD APART.
DO 70 L = 1, KD
KDOR = KDOR + L
C(KDOR) = ( C(KDOR) + C(KDOR+KD3) - C(KDOR+1) ) / KB
70 CONTINUE
IF((IJB.GT.0).AND.((IDUMP.EQ.0))) GO TO 208
WRITE (IOUT,9015) IOE, JDE, (C(KDOR+L), L = 1, KD)
9015 FORMAT(300, 215, 2X, 1P10E12.4)
208 CONTINUE
C RESTORE THE SECOND OE OF THE PAIR TO ITS NOMINAL VALUE.
OE(OE) = B(IJB+1)
C BRING KC UP TO DATE.
KD = KCDR + KD
80 CONTINUE
C RESTORE THE FIRST OE OF THE PAIR TO ITS NOMINAL VALUE.
OE(OE) = B(IJB+1)
90 CONTINUE
J = 800 = KCSUMS
KMB = KCYBAR
IF((IOPRNT.EQ.4)) GO TO 900
C GAUSSIAN CORRECTION FOR VARIANCES
C
I234 = K234
ICYBAR = KICYBAR
ICSUMS = KCSUMS
KCDUB = KCDUB
KD = KD
KC = KC
IF((IOPRNT.EQ.5)) GO TO 803
IF((IJB.GT.0).AND.((IDUMP.EQ.0))) GO TO 805
WRITE (IOUT,6969)
WRITE (IOUT,9050) (ID(I,1), I = 1, KD)
9050 FORMAT(2D ORDER MEANS AND VARIANCES ASSUMING GAUSSIAN INDEPENDENT VARIABLES // 62X, 'DEPENDENT VARIABLES' // 8X, 10112)
803 CONTINUE
C GOUC
CALL GOUC
IF((IJB.GT.0).AND.((IDUMP.EQ.0))) GO TO 806
WRITE (IOUT,9013) (C(KCSUMS+I), I = 1, KD)
WRITE (IOUT,9023) (C(KCSUMS+KD+I), I = 1, KD)
DO 75 I = 1, KD
75 CONTINUE
CALL STDDEV(C(KCSUMS+KD+I), I = 1, KD)
806 CONTINUE
IF((IOPRNT.EQ.6)) GO TO 900
803 CONTINUE
KDAVE = (K234 * (K234 + 1)) / 2
D.13 Subroutine D0234 (continued)

DO 85:0 IDAV = 1,KDAVE
IDA = KCOV + IDAV
IF(IDA.GT.9999) WRITE(6,8590) IDA
86: FORMAT(9X,9IDA = ' ',K15,1X,'IN D0234 WHICH EXCEEDS THE C ARRAY')
85: C(IDA) = 0.D0
C THE CARD FOR THIS READ IS DIRECTLY BEHIND THE NAME-TYPE-NOM-TCL-VAR-
C TABLE LENGTH CARDS.
WRITE(IOUT,9002)
8002: FORMAT(1H1,41X,9*** INDEPENDENT VARIABLE CORRELATIONS ***,//.
A4.X,6X,'CODE'6X,'6X,'CODE'6X,2X,'CORRELATION'6X/'
B4.X,5X,'NUMBER'5X,5X,'NUMBER'5X,2X,'COEFFICIENT'6X,///)
100: READ (INC,9001) IOE1, IOE2, RHO
9001: FORMAT(15,15,1F13.1)
IF(IOE1.LT.0) GO TO 802
WRITE(IOUT,9001) IOE1, IOE2, RHO
9001: FORMAT(40X,5X,15,6X,5X,15,6X,2X,1PE11.0,///)
C PUN THRU IA AND GET SUBSCRIPTS (OF THE IA ARRAY) WHERE IOE1 AND
C IOE2 ARE LOCATED.
DO 13 J=1,2
DO 11 I=1,K234
ISAV = 1
IF (IA(I,1).EQ. IOE(J)) GO TO 120
11: CONTINUE
WRITE (IOUT,9002)
9002: FORMAT(39H ILLEGAL OE SUBSRIPT READ IN WITH RHO.)
CALL EXIT
12: LN(J) = ISAV
13: CONTINUE
C GET THE SMALLEST IA SUBSCRIPT INTO LN(1).
IF ( LN(1) .LT. LN(2) ) GO TO 140
LS = LN(1)
LN(1) = LN(2)
LN(2) = LS
C GET THE SUBSCRIPTS TO PICK UP THE VARIANCES IN THE B ARRAY.
140: IB1 = IA(L1,2)
IB2 = IAIL2,2)
COV = RHO * DSRT( B(IB1+3) * B(IB2+3) )
C STORE COVARIANCES IN THE C ARRAY
C MCOV = 'MCOV(L1,L2)
C(MCOV) = COV
GO TO 130
802 CONTINUE
C COMUUTE 2ND ORDER MEANS AND 1ST ORDER VARIANCES WITH CORRELATIONS
CALL QCOR
IF(IOPRNT.EQ.7) GO TO 804
IF((IJ3.GT.0).AND.(IDUMP.EQ.)) GO TO 807
WRITE(IOUT,6980)
D.13 Subroutine DO234 (continued)

WRITE(IOUT,9051) (ID(I,1),I = 1,ND)
9051  FORMAT(2 * 2ND ORDER MEANS AND 1ST ORDER VARIANCES WITH CORRELATION FACTORS, 82X, 'DEPENDENT VARIABLES' //
     & ATTD INDEPENDENT VARIABLES' //)
     & CONTINUE
     & GO TO 900
     & WRITE(IOUT,69.69)
     & WRITE(IOUT,9052) (ID(I,1),I = 1,ND)
807  CONTINUE
     & IF((IJ8*GT.1).AND.(IDUMP.EQ.0)) GO TO 900
     & WRITE(IOUT,9013)(C(KCSUMS+I),I = 1,ND)
     & WRITE(IOUT,9023)(C(KCSUMS+KD+I),I = 1,ND)
     & DO 753 I = 1,ND
     & CALL STDDEV(C(KCSUMS+KD+I),I,KD,IOUT)
807  CONTINUE
     & IF((IJ9*EQ.0)) RETURN
     & IF((IJ9*EQ.0)) GO TO 225
     & M000 = 0
     & IF(J00 = 1)J01
     & WRITE(IOUT,8000)
     & B000 FORMAT(IH1,*50X,14H*** NOTICE *** //,
     & A30X,'THE SPECIFICATION OF EITHER RHO OR RHOX (IE
     & B30X,'VARIABLES 320 OR 321) OR BOTH AS OUTPUT VARIABLES
     & C30X,'NECESSITATES UP-DATING THE NOMINAL VALUES, DERIVATIVES,
     & D30X,'MEANS, AND VARIANCES, THE LIMITING CASES FOR AND PARTIAL
     & E30X,'DERIVATIVES OF RHO AND RHOX ARE MEANINGLESS. THUS THE
     & F30X,'FULL PROJECTILE DISPERSION CODE SHOULD BE UTILIZED IN
     & G30X,'MONTE CARLO SIMULATIONS (MCALC = 0) TO OBTAIN
     & H30X,'MEANINGFULL VALUES OF RHO AND/OR RHOX, IF A TAYLOR
     & I30X,'SERIES APPROXIMATION OF THE PROJECTILE DISPERSION CODE
     & J30X,'(MCALC = 1, 2, OR 3) IS USED IN MONTE CARLO
     & K3X,'SIMULATIONS THE RHO AND RHOX RESULTS ARE MEANINGLESS. ** 
D.13 Subroutine DO234 (continued)

```
00 503  IB08 = 1,KD
IF(ID(IB08,1) .EQ. 306) GO TO 504
IF(ID(IB08,1) .EQ. 307) GO TO 505
IF(ID(IB08,1) .EQ. 308) GO TO 506
GO TO 503
504 OXTO = C(KMB08 + IB08)
MB08 = MB08 + 1
IF (IOPRNT.LT.3) GO TO 503
OE(301) = C(JB08 + IB08)
BVAR1 = C(JBG0 + KD + IB08)
GO TO 503
505 OYTO = C(KMB08 + IB08)
MB08 = MB08 + 1
IF (IOPRNT.LT.3) GO TO 503
OE(302) = C(JB08 + IB08)
BVAR2 = C(JBG0 + KD + IB08)
GO TO 503
506 OZTO = C(KMB08 + IB08)
MB08 = MB08 + 1
IF (IOPRNT.LT.3) GO TO 503
OE(303) = C(JB08 + IB08)
BVAR3 = C(JBG0 + KD + IB08)
CONTINUE
IF(MBA0 .EQ. 3) GO TO 507
WRITE(IOUT,601)
VARIABLES OXTO, OYTO, OZTO (IE 306, 307, 308) MUST
BE DEFINED AS EITHER TYPE 7 OR 8,/%,1H)
CALL EXIT
507 C(KMBO8+1JB08)-DSQRT((DXT-0E(301))**2+(DYT-0E(302))**2+
A(DXT-0E(303))**2)
DO 221 IB08 = 1,K234
C(KMBO8+1JB08+K0+2*(IB08-1)+KD) = 0.000
221 C(KMBO8+1JB08+K0+2*IB08+KD) = 0.000
IF (IOPRNT.LT.3) GO TO 220
AVAR = DSQRT(BVAR1) + DSQRT(BVAR2) + DSQRT(BVAR3))/3.*
C(JB08) = 1.59576900 * AVAR
C(JB08 + 1JB08 + KD) = 0.453520900 * AVAR * AVAR
IF (IOPRNT.LT.4) GO TO 220
KAR = (K234-1) * K234/2
DO 222 IB08 = 1,KAR
222 C(KCDBOB+1JB08+(308-1)+KD) = 0.000
CONTINUE
IJ8 = IJ8 - 1
IF(IJ8 .EQ. 1) GO TO 299
CONTINUE
IF (IJ8 .EQ. 0) WRITE(IOUT,3000)
3000 IB08 = 1,KD
```

D.13 Subroutine D0234 (continued)

    IF(ID(I808,1) .EQ. 309) GO TO 704
    IF(ID(I808,1) .EQ. 310) GO TO 735
    GO TO 703

704   DXYO = C(KMB06+I808)
    MB06 = MB08 + 1
    IF(IPRINT.LT.3) GO TO 703
    OE(304) = C(IJOB8 + I808)
    BVAR1 = C(IJOB8 + I808 + K0)
    GO TO 703

705   DZX0 = C(KMB06+I808)
    MB06 = MB08 + 1
    IF(IPRINT.LT.3) GO TO 703
    OE(305) = C(IJOB8 + I808)
    BVAR2 = C(IJOB8 + I808 + K0)

706   CONTINUE
    IF(MB08.EQ.2) GO TO 707
    WRITE(IOUT,8001)

617   FORMAT(IHI,**//.7 *H VARIABLES DXY AND DZX (IE 309, 310) MUST BE DEFINED AS EITHER TYPES 7 OR 8,**//.1HI)
    CALL EXIT

707   C(KMB09+IJOB8) = DSORT((DXY - OE(304))*2 + (DZX0 - OE(305))*2)
    GO TO 1800 = I, K234
    IF(IJOB8+K0+2*(I808-1)*K0) = 0.000

233   C(KMB09+IJOB8+2*I808*K0) = 0.000
    IF(IPRINT.LT.3) GO TO 240

708   AVAR = DSORT(BVAR1) + DSORT(BVAR2))/2.
    C(IJOB8 + I808) = 1.25331*D0 * AVAR
    C(IJOB8 + I808 + K0) = 2.420293367*D0 * AVAR * AVAR
    EXECUTE A VARIANCE ON A VARIANCE
    IF(IPRINT.LT.3) GO TO 240
    KAR = (K234-1) * K234/2
    GO 232 I808 = I,KAR

232   C(KCDOUB+IJOB8+(I808-1)*K0) = 0.000

24   CONTINUE
    IJB = IJB - 1J02
    IF(IJB .EQ. 0) GO TO 799

226   CONTINUE
    WRITE(IOUT,8001)

8001  FORMAT(IHI,53X,14H*** NOTICE ***//** A30X* THE SPECIFICATION OF EITHER RADIUS OR RADX IS GUI **/ B30X* VARIABLES 318 OR 319) OR BOTH AS OUTPUT VARIABLES **/ C30X* NECESSITATES UP-DATING THE DERIVATIVES, MEANS, **/ D30X* AND VARIANCES, PARTIAL DERIVATIVES OF RADIUS AND RADX ARE **/ E30X* MEANINGLESS, THUS THE FULL PROJECTILE DISPERSION CODE **/ F30X* SHOULD BE UTILIZED IN MONTE CARLO SIMULATIONS (MCALC = 0)**/ G30X* TO OBTAIN MEANINGLESS, PARTIAL DERIVATIVES OF RADIUS AND RADX ARE **/ H30X* A TAYLOR SERIES APPROXIMATION OF THE PROJECTILE **/ I30X* DISPERSION CODE (MCALC = 1, 2, OR 3) IS USED IN MONTE **/ J30X* CARLO SIMULATIONS THE RADIUS AND RADX RESULTS ARE **/ K30X* MEANINGLESS, RADX AND/OR RADX MEANS AND VARIANCES ARE **/ L30X* TAKEN FROM THE TYPE 8 SPECIFICATION, **/ **/
D.13 Subroutine DO234 (continued)

IF (IJB3.EQ.0) GO TO 300
IJOB = IJB3
IF ( IOPRNT.EQ.1 ) GO TO 350
DO 351 IJOB = 1, K234
C(KMB08+IJOB*KD+2*(IJOB-1)*KD) = 0.0D0
351 C(KMB08+IJOB*KD) = 0.0D0
IF ( IOPRNT.EQ.2 ) GO TO 350
IF ( IJOB .GE. ID(IJOB2) ) GO TO 301
WRITE(1OUT,8003)
8003 FORMAT(1H1,"30H VARIABLE RADT (IE 318) MUST BE TYPE 8,"1H1)
CALL EXIT
350 CONTINUE
IJOB = IJB4
IF ( IOPRNT.EQ.1 ) GO TO 299
DO 361 IJOB = 1, K234
C(KMB08+IJOB*KD+2*(IJOB-1)*KD) = 0.0D0
361 C(KMB08+IJOB*KD) = 0.0D0
IF ( IOPRNT.EQ.2 ) GO TO 299
IF ( IJOB .GE. ID(IJOB2) ) GO TO 362
WRITE(1OUT,8003)
8003 FORMAT(1H1,"30H VARIABLE RXDX (IE 319) MUST BE TYPE 8,"1H1)
CALL EXIT
362 C(JBO8+IJOB8) = B(ID(IJOB8,2)+1)
C(JBO8+IJOB8*KD) = d(ID(IJOB8,2)+3)
IF ( IOPRNT.EQ.3 ) GO TO 299
KAR = (K234-1)*K234/2
DO 369 IJOB = 1, KAR
C(KDC08U8+IJOB8*(IJOB8-1)*KD) = 0.0D0
369 CONTINUE
WRITE(1OUT,9003) (BL,B(I1,1)+B(I1,2)+1), I = 1, K234
WRITE(1OUT,9005)
WRITE(1OUT,FMT2) (BL,B(I1,1), I = 1, KD)
WRITE(1OUT,FMT3) (C(KMB08I), I = 1, KD)
IF ( IOPRNT.GT.1 ) GO TO 380
WRITE(1OUT,9006) (BL,B(I1,1), I = 1, KD)
DO 381 I = 1, K234
WRITE(1OUT,9007) IAI(I1-1)+B(I1,2)+1), B(I1,2)+2),
ACC(KMB08+2*KD+2*(I-1)*KD)+L = 1+KD)
381 WRITE(1OUT,9008) C(KMB08+2*KD*I+L), L = 1, KD)
WRITE(1OUT,9009) (ID(I1,1), I = 1, KD)
DO 382 I = 1, K234
RETURN
D.13 Subroutine DG234 (continued)

```fortran
WRITE(IOUT,9010) (A(I,J),J=1,L),1, I=1,KD)  
382 WRITE(IOUT,9011) (C(K4808+KD*L+2*KD*I-I)), L = I,KD)  
IF (IUPRN.LT.3) RETURN  
IF (IUPRN.EQ.3) GO TO 385  
WRITE(IOUT,901K) (ID(I,I),I = 1,KD)  
KMM = K234 - 1  
LMNO = KD008  
DO 383 I = 1, KMM  
KMM = I + 1  
DO 384 J = KMM,K234  
WRITE(IOUT,9015) (A(I,J),I=1,KD)  
384 LNO = LMNO + KD  
385 WRITE(IOUT,6969)  
6969 FORMAT(1I1)  
KPRNT = IUPRN - 2  
GO TO (875,875,875,877,878,KPRNT)  
875 WRITE(IOUT,9012) (ID(I,I),I = 1,KD)  
GO TO 880  
876 WRITE(IOUT,9051)(ID(I,I),I = 1,KD)  
GO TO 880  
877 WRITE(IOUT,9052)(ID(I,I),I = 1,KD)  
GO TO 880  
878 CONTINUE  
WRITE(IOUT,9013)(C(J808+I),I = 1,KD)  
WRITE(IOUT,9023) (C(J808+KD*I), I = 1,KD)  
DO 754 I = 1,KD  
754 CALL STDDEVIC(J808*KD*I),I=1,KD,IOUT)  
D.14 Subroutine EXTRA

SUBROUTINE EXTRA
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CARDG/ 3(200), C(500), I(A(200)+2), ID(10+2), INC,
1 DOPRINT, IOUT, IY, KARS(3), KAK, KB, KC, KI, ILE(234), LAB(3)
DIMENSION I(B(2,200))
EQUIVALENCE (I9(I1,1),B(I1))
IF(KI.EQ.0) GO TO 100

C PRINT RANGE CHECK VALUES
WRITE(IOUT,5000)
DO I = 1, KI
KOE = IA(I-1)
KB = IA(I,1) + 1
NL = IB(I,KB)
KC = IB(I,KB+1)
WRITE(IOUT,5000) IOE, (C(KC-I+J), J=1, ITL)
5000 FORMAT(1H1, 49X, '*** RANGE CHECK VALUES '///,
$22X, 5X, 'NUMBER', 5X, 'VALUE', 6X)
5001 FORMAT(22X, 4X, 15, 6X, 5(3X, '1PE11.4', 2X), 
$5(26X, '5(3X, '1PE11.4', 2X)'))
A/

100 CONTINUE
IF(K234.EQ.0) RETURN
DO 2 I = 1, K234
IOE = IA(I-1)
KB = IA(I,1)
IF(IB(I,KB),NE,4) GO TO 2
WRITE(IOUT,5003) IOE
5003 FORMAT(1H1, 38X, '*** CODE NUMBER', 5, '1X',
$*PROBABILITY DENSITY FUNCTION' HITS1357
8, 41X, 5X, 'NUMBER', 5X, 'VALUE', 6X, '4X, 'DENSITY', '////
8

ITL = IB(I, KB+4)
KC = IB(I, KB+5)
DO 3 J = 1, ITL
WRITE(IOUT, 5004) J, C(KC-1+ITL+J), C(KC-1+J)
3 CONTINUE
WRITE(IOUT, 5004) J, C(KC-1+ITL+J), C(KC-1+J)
2 CONTINUE
5004 FORMAT(40X, 6X, 14, 6X, 2(3X, '1PE11.4', 2X))
RETURN
END
D.15 Subroutine FC2987

SUBROUTINE FC2987
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/DEC/OE(600)

EQUIVALENCE (VO,OE(1)), (VWIND,OE(2)),
        (RH0,OE(4)), (CD1,OE(5)),
        (DI0,OE(7)), (V2,OE(8)),
        (CNAD,OE(10)), (SM,OE(11)),
        (CMA,OE(13)), (TRIM,OE(14)),
        (ANGO,OE(16)), (PHIANG,OE(17)),
        (PHIRAT,OE(19)), (GAM0,OE(20)),
        (A,OE(22)), (ELL,OE(23)),
        (W,OE(25)), (AIX,OE(26)),
        (P,OE(28)), (ALPCON,OE(29)),
        (ATOT,OE(401)),
        (ALPHAD0,OE(403)), (B,OE(404)),
        (ALPHM,OE(409)), (ALPMIN,OE(410)),
        (CD8,OE(412)), (CMA,OE(413)),
        (CMHTA,OE(415)), (CAY1,OE(416)),
        (CDAB,OE(418)), (CDAB,OE(419)),
        (DELV,OE(421)), (DELT,OE(422)),
        (DYDXO,OE(424)), (DZDXO,OE(425)),
        (DZDTO,OE(427)), (DELLW,OE(428)),
        (EYEPC,OE(430)), (EMP,OE(431)),
        (EDLT,OE(433)), (F,OE(434)),
        (PSID0,OE(436)), (PHI10,OE(437)),
        (PHI2,OE(439)), (PSIO,OE(440)),
        (PHI2,OE(442)), (R3,OE(443)),
        (PSID1,OE(444)), (RTRIM,OE(445)),
        (T,OE(446)), (TG0,OE(449)),
        (TS,OE(448)), (THETD0,OE(452)),
        (THETA0,OE(451)), (T,OE(446)),
        (VC,OE(454)), (VC0,OE(455)),
        (VO,OE(457)), (WX,OE(458)),
        (VZA,OE(459)), (W,OE(460)),
        (WZ,OE(459)), (W1,OE(462)),
        (W2,OE(463)),

EQUIVALENCE (WZ,OE(459)), (W1,OE(462)),
        (W2,OE(463)), (VL2,OE(463)),
        (XLAM1,OE(465)), (XLAM2,OE(466)),
        (XNAD,OE(466)), (XNAD,OE(466)),
        (XNU1,OE(468)), (XG,OE(469)),
        (XNU2,OE(468)), (XU1,OE(470)),
        (XJAY,OE(471)), (XJAZ,OE(472)),
        (XJAY,OE(473)), (XJAZ,OE(473)),
        (XKA,OE(474)), (XK,OE(475)),

EQUIVALENCE (TN,OE(500)), (XTFC,OE(501)),
        (YTFC,OE(502)), (YTFC,OE(502)),
        (VXTFC,OE(504)),

CALL INCONF
A57 = ALPCON/57, 2957795100
CALL XXC(XLAM0,XC,VO,DELLAM,CAY1,CAY2,TOL,A57)
CALL TRAJXF(XN,VC,TN)
CALL CROSSF(XN,VC)
CALL WIND(VX,TH,XN,WZ,WX,VO,VZW,ZW)
D.15 Subroutine FC2987 (concluded)

\[ \begin{align*}
XTFC &= XN \\
YTF &= YA \\
ZTFC &=ZA + ZW \\
VXTFC &= VX \\
VYTF &= VYA \\
VZTFC &= VZA + VZW \\
\text{return} \\
\text{end}
\end{align*} \]
D.16 Function FEXP

FUNCTION FEXP(X)
  IMPLICIT REAL*8(A-H,O-Z)
  IF( X .LT. -90.00) GO TO 1
  FEXP = DEXP( X)
  RETURN
  FEXP = 0.00
  RETURN
END
D.17 Subroutine FILLA

SUBROUTINE FILLA (I,FCTR,PM,PT,CM,CV,NCELL)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(25),S
C THIS ROUTINE FILLS OUT THE OUTPUT ARRAY
CELL=2.00*PT/FCTR
ENN=0.00
SUM1='.'DO
SUM2='.'DO
C COUNT TOTAL FREQUENCY
DO 1 I=1,NCELL
ENN=ENN+A(I,4)
1 DO 2 I=1,NCELL
A(I,5)=A(I,4)/ENN
FI=I
FIM=F1-.500
FI=FIM*A(I,5)
SUM1=SUM1+FI
SUM2=SUM2+FI*FIM
A(I,5)=A(I,5) / FCTR
A(I,1)=PM+PT*(2.*FIM/CELL-1.)
A(I,2)=A(I,1)-S00*FCTR
A(I,3)=A(I,2)*FCTR
CV=(SUM2-SUM1**2)*FCTR**2
CM=PM+FCTR*(SUM1-.5D0*CELL)
RETURN
END
D.18 Subroutine FILLIN

SUBROUTINE FILLIN(IA, IOUT)
IMPLICIT REAL*8 (A-H, O-Z)
COMMON/NAMES/HEAD(180)
IAT = IA
IF (IA.LT.0) IAT = - IAT
IF(IAT.GT.100) GO TO 1
WRITE(IOUT,8000) HEAD(IAT),IAT
RETURN
1 CONTINUE
IF(IAT.GT.200) GO TO 2
WRITE(IOUT,8001) HEAD(IAT-100),IAT
RETURN
2 CONTINUE
IF(IAT.GT.300) GO TO 3
WRITE(IOUT,8002) HEAD(IAT-165),IAT
RETURN
3 CONTINUE
IF(IAT.GE.400) GO TO 4
WRITE(IOUT,8003)ヘッド(IAT-259),IAT
RETURN
4 CONTINUE
IF(IAT.GE.500) GO TO 5
WRITE(IOUT,8004) HEAD(IAT-315),IAT
RETURN
5 CONTINUE
IF(IAT.GT.600) GO TO 6
WRITE(IOUT,8004) HEAD(IAT-338),IAT
RETURN
6 WRITE(IOUT,8005)
RETURN
8000 FORMAT(1H ,36X, A6, 1X, 'FIRE CONTROL ', 9X, ' CODE NUMBER ', 13)
8001 FORMAT(1H ,36X, A6, 1X, 'REAL WORLD ', 9X, ' CODE NUMBER ', 13)
8002 FORMAT(1H ,36X, A6, 1X, 'SYSTEM CONTROL', 9X, ' CODE NUMBER ', 13)
8003 FORMAT(1H ,36X, A6, 1X, 'STATISTICAL VARIABLE', 3X, ' CODE NUMBER ', 13)
8004 FORMAT(1H ,36X, A6, 1X, 'TRAJECTORY VARIABLE', 3X, ' CODE NUMBER ', 13)
8005 FORMAT(1H,50('/ 1X, 'DE ADDRESS EXCEEDS 600')
END
D.20 Subroutine GQUC

SUBROUTINE GQUC
IMPLICIT REAL*8(A-H, O-Z)
COMMON /CARDCM/ B(200), C(500), IA(200,2)
COMMON INDEX/ K23, K234, KCYBAR, KCSUMS, KCO0UB, KCOV
    ) = K234 - 1
DO 1 = 1, KD
    SUM = 0.0
    DO 2 J = 1, K234
        N2 = N2D(I, J, K)
        NX = NVARX(J)
        SUM = SUM + ( C(N2) * B(NX) )
        IF( K1 = E0, 1 ) GO TO 6
        DO 3 J = 1, K1
            NXJ = NVARX(J)
            BNXJ = B(NXJ)
            KB = J + 1
            DO 3 K = KB, K234
                N2 = N2D(I, J, K)
                NXK = NVARX(K)
                SUM = SUM + C(N2) * B(NXJ) * B(NXK)
            3 CONTINUE
        2 CONTINUE
        IF( K1 = E0, 1 ) GO TO 6
        1 CONTINUE
        RETURN
    END
D.23 Subroutine G2987

SUBROUTINE G2987
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/DEC/DE(600)
COMMON/CISPRNT/ISPRNT
COMMON/CINUM/INJM
COMMON/EQUIVALENCE
A(ZBART,OE(133)) , (YBART,OE(134)) , (ZBART,OE(135))
B(DYX,OE(324)) , (DXT,OE(325)) , (DYE,OE(326))
C(DYXI,OE(327)) , (DYXI,OE(328)) , (DYXI,OE(329))
D(DYXI,OE(330)) , (DYXI,OE(331)) , (DYXI,OE(332))
E(DYXI,OE(333)) , (DYXI,OE(334)) , (DYXI,OE(335))
F(DYXI,OE(336)) , (DYXI,OE(337)) , (DYXI,OE(338))
G(DYXI,OE(339)) , (DYXI,OE(340)) , (DYXI,OE(341))
H(DYXI,OE(342)) , (DYXI,OE(343)) , (DYXI,OE(344))

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

FIRE CONTROL LOGIC

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

IF((.NOT.((IFC*NE.1) .AND. (IFC*EQ.1))) .AND. (IFC*NE.1)) GO TO 1
DO 2 I = 1,100
2 QF(I) = QF(I+100)
1 CONTINUE
IF((IFC*NE.1) .AND. (IFC*NE.1)) GO TO 3
TOL = 10.**ICNV
CALL FC2987
IF(INUM*EQ.1) GO TO 500
IF(ISPRNT*EQ.1) GO TO 600
500 CALL SPRNT
D.23 Subroutine G2987 (continued)

600 CONTINUE
3  IFC = 2
CALL RW2987

C INCREMENTAL OUTPUTS

DXT  = XT  - XTFC
DYT  = YT  - YTFC
DZT  = ZT  - ZTFC
DXY  = YR  - YTCF
DZX  = ZR  - ZTCF
DVXT = VXI - VXTCF
DVYT = VYT - VYTCF
DVZT = VZI - VZTCF
DVXX = VXI - VXTCF
DVYX = VYR - VYTCF
DVZX = VXR - VZTCF
DT  = TX  - TN
DXTI = DXT - XBAR
DYTI = DYT - YBAR
DZTI = DZT - ZBAR
DXTE = DXT - XBAR
DYTE = DYT - YBAR
DZTE = DZT - ZBAR
DXXI = DXI - XBAR
DZXI = DZI - ZBAR
DXYE = DXY - YBAR
DZXE = DZI - ZBAR

C SEP AND CEP CALCULATIONS USING INTERNAL AND EXTERNAL CENTERS

RADT = DSQRT(DXTE**2 + DYTE**2 + DZTE**2)
RADX = DSQRT(DXIE**2 + DZXE**2)
RHOT = DSQRT(DXTI**2 + DYTI**2 + DZTI**2)
RHOX = DSQRT(DXII**2 + DZII**2)

C CORRELATIONS

DXDYT = DXT * DYT
DXDZT = DXT * DZT
D.23 Subroutine G2987 (concluded)

DYDZT = OYT * DZT
DYDZX = OYX * DZX
DXYT1 = OXT1 * DYTI
DXDZX = OXZ1 * DZTI
DYDZT1 = OYT1 * DZTI
DYOZXI = OYXI * OZXI
DXYTE = OXTE * DYTE
DXDZTE = OXZTE * DZTE
DYDZTE = OYTE * DZTE
DYDZXE = OYXE * DZXE

C INOM = 1 IN MAIN*
IF ( INOM .EQ. 1 ) GO TO 260
IF ( ISPANT .EQ. 0 ) RETURN

260 CALL SPRT
INOM = 0
RETURN
END

HITS1666
HITS1667
HITS1668
HITS1669
HITS1670
HITS1671
HITS1672
HITS1673
HITS1674
HITS1675
HITS1676
HITS1677
HITS1678
HITS1679
HITS1680
HITS1681
HITS1682
D.24 Subroutine HSTGL

SUBROUTINE HSTGL(IER)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CARDCM/ B(200),C(5000),IA(200,2),ID(10,2),INC,OPRNT,OUT,
IIV,KAPS1(3),KA,KB,KC,KD,KT1,K234,LABC(3)
COMMON/DEC/DE(600)
COMMON/INCH/ID2(200),IID2(200),IID1(400),IID2(400),IDT(10),IIT(20),I,
JR1(JR1,JR2,JR8,NCC,NRI,2)
EQUIVALENCE (KADD,KC)
C THIS SUBROUTINE CHECKS THE SAMPLE OUTPUT AGAINST THE INTERNALLY
generated histogram ranges. If a reject occurs it sets IER=-1.
C THIS SUBROUTINE requires KADD the nominal block address in C = K
ENN=NCCELL
IBN=1
LADD=2*KD+K234+KD+(KADD-1)
DO 1 I=1,KD
1 ADD=ADD+I
C IADD is address of output variable value-calculate addresses
C of variable mean and tolerance values in C array
IMN=LADD+1
IVAR=IMN+KD
TOL=3.0*DSORT(C(IOVAR))
FJ=(OE(IADD)-C(IMN)+TOL)*.5*ENN/TOL
IF ( (FJ)4.77,77
77 J=FJ
J=J+1
C J is histogram interval to be incremented-check if valid
IF((J)4.9,2
2 IF(J-NCELL)3,4
C IF J LT ZERO OR GT NCELL reject sample
4 IF (I-1 N I1(I)+1
1 IDT(I)=0
IER=-1
RETURN
C Store histogram cell address temporarily
1 IBN=IBN+NCCELL
C IF this point reached the sample is OK so increment histograms
DO 5 I=1,KD
5 IDT(I)=IDT(I)+1
DO 6 I=1,K234
6 IDT(I)=IDT(I)+1
RETURN
END

MITS1683
MITS1684
MITS1685
MITS1686
MITS1687
MITS1688
MITS1689
MITS1690
MITS1691
MITS1692
MITS1693
MITS1694
MITS1695
MITS1696
MITS1697
MITS1698
MITS1699
MITS1700
MITS1701
MITS1702
MITS1703
MITS1704
MITS1705
MITS1706
MITS1707
MITS1708
MITS1709
MITS1710
MITS1711
MITS1712
MITS1713
MITS1714
MITS1715
MITS1716
MITS1717
MITS1718
MITS1719
MITS1720
MITS1721
MITS1722
MITS1723
MITS1724
MITS1725
MITS1726
MITS1727
D.25 Subroutine HSTG2

SUBROUTINE HSTG2(IER)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CAPDCM/ B(200),C(500),IA(200),2,1D(10,2),1NC,1OPRNT,1OUT
IIY,KARS(3),KAK,KB,KD,KT1,K234,LABC(3)
COMMON/DEC/DE(600)
COMMON/IMCH/ID1(200),ID2(200),II1(400),II2(400),IDT(10),IIT(20),
IIR(11),IOP(10),NTRIAL,NCELL,MCALC,MCOPT,IRJ1,IRJ2,INB,NCC,NR1,NR2
EQUIVALENCE (KADD,KC)
C THIS ROUTINE CHECKS THE SAMPLE OUTPUT AGAINST THE USER SPEC
EN= NCELL
IBN= 1
DO 1 I= 1, KO
L.DD(1,2)
C CHECK IF IT IS A TYPE 8 VARIABLE
IF(L)8,8,2
C IF IT IS NOT A TYPE 8 CHECK IF IT PASSED THE PREVIOUS TIME
9 IF (IDT(1))4,4,1
C IF NOT A TYPE 8 AND REJECTED BEFORE REJECT IT AGAIN IF PASSED
2 N=ID1(I,1)
FJ=(.5*(N)-B(L+1)+B(L+2))*.5*ENN/B(L+2)
IF (FJ )4,77,77
77 J=FJ
J=J+1
C J IS THE HISTOGRAM INTERVAL NUMBER CHECK IF VALID
3 IF(J-NCELL)5,5,4
4 IR2(1)=IR2(1)+1
IER=-1
RETURN
5 IDT(1)=IBN+J-1
1 IBN=IBN+NCELL
C IF GETS HERE IT IS SUCCESSFUL SO INCREMENT HISTOGRAMS
DO 6 I= 1, KO
L=IDT(I)
6 ID2(L)=ID2(L)+1
DO 7 I=1,K234
L=IIT(I)
7 II2(L)=II2(L)+1
RETURN
END
D.26 Subroutine HYSPLT

```
SUBROUTINE HYSPLT
  IMPLICIT REAL * 8(A-H, O-Y)
  COMMON/ HISPLT/ IOPLT, ISPLT
  COMMON /CARDCM/ B(200), C(5000), IA(200,2), ID(10,2), INC
  COMMON /LABC(3)
  NCELL1 = ZTITLE(18,1) + 1
  WRITE(IOUT,1,500)
  1 FORMAT(1H1)
     WRITE(IOUT,10,0)
  10 FORMAT(+J
      K = KD + K234
      DO 1 J = 1, K
      1 FORMAT(1001) J, (ZTITLE(I,J), I=1,20)
  1001 FORMAT(106X, 1P10E12.4, /, 12X, 1P10E12.4)
      DO 2 J = 1, K
      WRITE(IOUT,500)
      WRITE(IOUT,1002) J, ((ZDATAD(I*, L, J) + L = 1,4), I=1, NCELL
      1002 FORMAT(106X, '
      RETURN
END
```
D.27 Subroutine INCARD

SUBROUTINE INCARD
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CARDOM/ B(200), C(500), IA(200,2), IO(10,2), INC.
1 IOPRT, IGUT, IY, KARS1(3), KA, KB, KC, KD, KTI: K234, LABC(3)
COMMON/OEC/OE(600)
DIMENSION IOE(2,600)
D. MENSION IB(2,200)
EQUIVALENCE (18(1,1), B(1)), (IOE(1,1), OE(1))
DO 15 I=1,3
15 LABC(1)=1
C KTI WILL COUNT TYPE 1, K234 WILL COUNT TYPES 2, 3 AND 4.
C KD FOR THE DEPENDENT VARIABLES.
C KB FOR THE CURRENT ENTRY IN B.
C KC FOR THE CURRENT ENTRY IN C.
KT1 = 0
K234 = 0
KD = 0
KB = 0
KC = 0
EXIT = 0
20 READ (INC,9002) IS,JE, IY, TYPE, FNOM, TOL, STDEV, ITL
9002 FORMAT(IS,12,3F13.1,14)
IF ( ISOE (LT,0) ) GO TO 100
WRITE(ICOUT,5500)ISOE,ITY,FNOM,TOL,STDEV,ITL
5500 FORMAT(10X,5X,IS,5X,3X,7X,12X,7X,3(3X,1PE11.4,2X),6X,99)
V&R = STDEV * STDV
GO TO (100, 203, 203, 400, 500, 999, 708, 708), IY
100 KTI = KTI + 1
IF ( KTI LE, 3 ) GO TO 110
WRITE(ICOUT,9004)ISOE
9004 FORMAT(26H MORE THAN 3 TYPE 1, NAME=, I4)
EXIT = 1
GO TO 108
110 ITLADD = 1
GO TO 101
C 120 SAVE THE KAR SUBSCRIPT FOR LATER MOVING THE TYPE 1 INFO UP TO
C THE START OF IA.
120 KARS1(KTI) = KAR
LABC(KTI) = ITL
GO TO 20
203 K234 = K234 + 1
GO TO 110
C 210 SAVE 2 SLOTS IN IB (FOR TABLE LENGTH AND ADDRESS IN C).
210 KB = KB + 2
GO TO 710
400 K234 = K234 + 1
C SKI03 SLOTS FOR NOM, TOL AND VAR.
C ITLADD = 4
14 KEEP3 TRACK OF READING THE W AND Y TABLES.
C I4 = 1
D.27 Subroutine INCARD (continued)

GO TO 1010  
410 IF ( I4.EQ. 2) GO TO 420
    IA = 2
    KC = KC + 1
    GO TO 107:
420 IAW = IB(1,KB) - 1
    IAY = IAW + ITL
    KBR = KB - 5
    CALL TANTV ( IAW, IAY, ITL, KBR )
C PUT NOMINAL VALUE INTO OE.
    LOE = IA(KAR,1)
    OE(LOE) = B(KBR+1)
    GO TO 20
500 GO TO 1010
C 510 PUT A -1 IN COLUMN 2 FOR A LATER CHECK.
    IA(KAR,2) = -1
C IS IT AN INTEGER.
    LOE = IA(KAR,1)
    IF ( LOE .LE. GT, 0) GO TO 710
    LOE = - LOE
    IOE(1,LOE) = FNOM
    GO TO 20
710 KD = KD + 1
    IO(KD,1) = ISOE
    IF ( ITYPE .EQ. 7) GO TO 20
    KB = KB + 1
    IO(KD,2) = KB
    ITLADD = 1
    GO TO 1046
710 OE(ISOE) = FNOM
    GO TO 20
999 WRITE (IOUT,1000)
1000 FORMAT(25H INCARD ERROR, CALL EXIT.)
    CALL EXIT
1010 KAR = KAR + 1
    IF ( IABS(IA(KAR,1)) .EQ. ISOE ) GO TO 1030
    IF ( KAR .LE. KA ) GO TO 1020
    EXIT = 1
    WRITE (IOUT,9003) ISOE, ITYPE
9003 FORMAT(24H INCARD, COULD NOT FIND * I4, 16H IN IA FOR TYPE = I2)
    IF ( ITYPE .EQ. 1 OR ITYPE .EQ. 4 ) GO TO 1080
    GO TO 20
1030 GO TO (1049,1040,1040,1040,510,999,999,999,999), ITYPE
1040 KB = KB + 1
    IA(KAR,2) = KB
1045 IB(1,KB) = ITYPE
    KB = KB + ITLADD
    GO TO (1060,1050,1050,1060,999,999,999,1050), ITYPE
1050 X(KB) = FNOM
D.27 Subroutine INCARD (concluded)

```
KB = KB + 1
B(KB) = TOL
KB = KB + 1
B(KB) = VAR
GO TO ( 999, 210, 210, 999, 999, 999, 999, 710), ITYPE
1060 IB(I,KB) = ITL
KC = KC + 1
KB = KB + 1
IB(I,KB) = KC
1070 KCM1 = KC - 1
READ (INC,9005) (C(I+KCM1), I=1,ITL)
9005 FORMAT(5F13.1)
C THE LAST ADDRESS USED IS KCM1 + ITL*
KC = KCM1 + ITL
GO TO ( 122, 999, 999, 41), 999, 999, 999, 999), ITYPE
1080 READ (INC,9005) (C(I), I=1,ITL)
IF ( ITYPE = EQ, 1 ) GO TO 20
C FOR ITYPE=4 WE HAVE TO BYPASS BOTH W AND Y TABLES.
ITYPE = 1
GO TO 1080
1100 CONTINUE
C FINISHED READING ALL THE INPUT CARDS, WERE THERE ANY ERRORS.
IF ( IEXIT = EQ, 1 ) CALL EXIT
RETURN
END
```
D.28 Subroutine INCON

SUBROUTINE INCON

IMPLICIT REAL * 8 (A-H, O-Z)

COMMON/DEC/OE(600)

EQUIVALENCE

ALPHWIN, OE(103), (RHO, OE(104)), (CD1, OE(105)),

BVI, OE(106), (CD2, OE(107)), (V2, OE(108)),

CIV, OE(109), (CNA, OE(110)), (SM, OE(111)),

DCNO, OE(112), (CMPA, OE(113)), (TR1M, OE(114)),

EPNITRI, OE(115), (ANGQ, OE(116)), (PHIANG, OE(117)),

FIRATE0, OE(113), (PHIRAT, OE(119)), (GAM0, OE(120)),

GRPHIGAM, OE(121), (A, OE(122)), (ELL, OE(123)),

MD, OE(124), (W, OE(125)), (AIL, OE(126)),

M(IAIY, OE(127)), (P, OE(128)), (ALPCON, OE(129)),

JICAA, OE(130)

EQUVALENCE

V0, OE(101), (VWIND, OE(102)), (V2/V2 - 1.00 / VI2)

CAYD = (CD2 - CD1) / (1.00 / VI2)

CD1 = CD1 - CAYD / VI2

B = 1.00 + (WZ / (V0 - WX)) ** 2

F = RHD * A * G

/ 57.2957795100 * 32.17400 /
D.28 Subroutine INCON (continued)

IF(IDUMP .EQ. 1) WRITE(6,1001) WX, WZ, CAYD, CD8, B, F
1000 FORMAT(/ WX, WZ, CAYD, CD8, B, F)
               F = 1/IX, 1P6E12.4 /1

THETA0 = ANGO/RFAC
PS10 = PHIANG/RFAC
THETD0 = RATE0
PS10 = PHIRAT
DYDX0 = GAM0/RFAC
DZDX0 = PHIGAM/RFAC

DYTO = V0 * DYDX0
DZTO = V0 * DZDX0

ALPHD0 = THETD0 - DYDX0
BETA0 = DZDX0 - PS10

IF(IDUMP .EQ. 1) WRITE(6,1001)
               THETA0, PS10, THETD0,

1 PS10, DYDX0, DZDX0, DYTO, DZTO

FCV = F * CNA * V0

ALPHD0 = THETD0 - FCV * ALPH0 - P * BETA0

IF(INCL .EQ. 1) ALPHD0=ALPHD0+F*V0*THETA0*(CD8+CAYD/V0/V0)
               P = 1/IX, 1P6E12.4 /1

BETA0 = ALPHD0 * FCV * BETA0 + P * ALPH0

IF(INCL .EQ. 1) BETA0=BETA0-F*V0*PS10*(CD8+CAYD/V0/V0)
               P = 1/IX, 1P6E12.4 /1

CMA = - CNA * ELL * SM / D

EYEP = 2*D0 * A1Y/RHO / A /D *V0/V0

EMP = 2*D0 * A/RHO / A / G / V0

D2V = D2V/D2V/V0

CMTHTD = CMQA * D2V

CMHTA = CMPA * D2V

M = ZZ(CNA, CD8, CAYD, V0, EMP, ICNCL)

CCON = CNA / EMP + CMTHTD / EYEP

PII = P * A1Y

WCON = - 4*D0 * CMA / EYEP + PII * PII - CCON * CCON

IF(IDUMP .EQ. 1) WRITE(6,1002) ALPH0, BETA0, ALPHD0, BETA0, CNA, EYPH
               P = 1/IX, 1P6E12.4 /1

1, EMP, CMTHTD, CMHTA, M

W0 = DSORT(-1.0) *( WCON + DSORT(WCON * WCON + 4*D0*PII)

1 * CCON + 2*D0 * P * CMHTA / EYEP ) ** 2 )

IF( CNA + LT. 0.* ) GO TO 3

PCR=0.00

XR=0.00

RTX=1.00

PHIROL= 0.00

GO TO 4

3

CONTINUE

PCRV= DSORT(-CMA*RHO * A /2*D0*D/(A1Y-A1X))

PCR= PCRV + DABS(V0)

XMU= ( CNA*RHO/4*D0 * A/WG- CMQA*D0/RH0*A/(A1Y-A1X))/PCRV

PPCR= P/PCR

PPCR1=1.00 - PPCR**2

XMU2=2.00 * XMU*PPCR

RTX=1.00 + DSORT(PPCR1 + XMU2**2)

PHIROL = DATAN(XMU2/PPCR)

IF( PPCR1 *LT. 0.* ) PHIROL = PHIROL+3.1415+2653589880

HITS1968
HITS1969
HITS1970
HITS1971
HITS1972
HITS1973
HITS1974
HITS1975
HITS1976
HITS1977
HITS1978
HITS1979
HITS1980
HITS1981
HITS1982
HITS1983
HITS1984
HITS1985
HITS1986
HITS1987
HITS1988
HITS1989
HITS1990
HITS1991
HITS1992
HITS1993
HITS1994
HITS1995
HITS1996
HITS1997
HITS2000
HITS2001
HITS2002
HITS2003
HITS2004
HITS2005
HITS2006
HITS2007
HITS2008
HITS2009
HITS2010
HITS2011
HITS2012
HITS2013
HITS2014
HITS2015
HITS2016
HITS2017
D.28 Subroutine INCON (continued)

ABAR = DSORT(TRIM**2 + PNITRI**2)
PNITRR = ZATAN2(PNITRI, TRIM)

1
CONTINUE
2
ALPTRM = RTRIM * ABAR / RFAC
PTPR = PNITRR + PHIROL
BETTRM = ALPTRM * DSINC PTPR
ALPTRM = ALTRM * DCOS(PTPR)

IF(DUMP .EQ. 1) WRITE(6,1006) XMU, RTRIM, PHIROL, ALPTRM

1006 FORMAT(/' 'XMU RTRIM PHIROL ALPTRM BETHI

1TRM PCR*)
2
1X, IP10E12.4 //)
DEW = P - PII / 2.D0
XLAM0 = 500* (CMHTDA / EYEP - CNA/EMP)
DELM0 = PII / 4.D0 / W0 * CCON + P * CMHTA / 2.D0 / W0 / EYEP
W1 = W0 - DEW
W2 = W0 + DEW
AA = ALPHO - ALPTRM
XLAM1 = XLAM0 + DELAM
XLAM2 = XLAM0 - DELAM
BB = BETAO - BETTRM
RCON1 = ALPHO0 + W2 * BB
RCON2 = BETAO0 - W2 * AA
W2 = 2.D0 * (W0 * W0 + DELAM ** 2)
R1 = (W0 * RCON1 + DELAM * RCON2) / W2
R2 = -(W0 * RCON2 + DELAM * RCON1) / W2

IF(DUMP .EQ. 1) WRITE(6,1004) W0, DELW, XLAM0, DELAM * W1, W2

1004 FORMAT(/' 'W0 DELW XLAM0 DELAM */WHITS2046

2
R2 = ' /1X, IP11E12.4 // )
R4 = AA - R2
R3 = BB - R1
CAY1 = DSORT( R1 * R1 + R3 * R3 )
CAY2 = DSORT( R3 * R3 + R4 * R4 )
XNU1 = ZATAN2( R2, R1 )
XNU2 = ZATAN2( R4, R3 )
PHI1 = DATAN2(-W1 - P, XLAM1)
PHI2 = DATAN2(-W2 + P, XLAM2)
PHI0 = DATAN2(-W0, XLAM0)

1001 FORMAT(/' 'YTHETA0 PS10 THEHITS2058

1D00 PS100 DDX0 DDX0 DDX0 DYDT0 DDX0 DDX0/WHITS2059
2 /1X, IP19E12.4 /IWHITS2060

1002 FORMAT(/' 'ALPH0 BETAO ALPHD0 BETADL */CHITS2061
1/1X, IP11E12.4 /) IWHITS2062
2 /1X, IP19E12.4 /IWHITS2063

IF(DUMP .EQ. 1) WRITE(6,1005) R3, R4, CAY1, CAY2

1005 FORMAT(/' 'R3 R4 CAY1 CAY2 */XHITS2066

1UI XNU1 XNU2 PHI1 PHI2 PHI0 */1X, IP10E12 */IWHITS2067
D.29 Subroutine INCONF

SUBROUTINE INCONF
IMPLICIT REAL * 8 (A-H, O-Z)
COMMON/OCC/OE(600)
EQUIVALENCE (VO, OE(  1)), (VWIND, OE(  2)),
            (A, PHWIN, OE(  3)), (RHO, OE(  4)), (CD1, OE(  5)),
            (V0, CD2, OE(  7)), (V2, OE(  8)),
            (C, CMA, OE( 10)), (SM, OE( 11)),
            (D, CMQ, OE( 12)), (CAMA, OE( 13)), (TRIM, OE( 14)),
            (E, PNI0, OE( 15)), (ANG0, OE( 16)), (PHIANG, OE( 17)),
            (F, RATE0, OE( 18)), (PHIR, OE( 19)), (GAM0, OE( 20)),
            (G, PHIGAM, OE( 21)), (A, OE( 22)), (ELL, OE( 23)),
            (H, OE( 24)), (W, OE( 25)), (AIX, OE( 26)),
            (I, A1Y, OE( 27)), (P, OE( 28)) (ALPCON, OE( 29)),
            (JC, OE( 301))
EQUIVALENCE (IDUMP, OE( 204)), (ICNCL, OE( 206)),
            (ALPHA, OE( 400)), (ATDT, OE( 401)),
            (ALP, ALPH0, OE( 403)), (B, OE( 404)),
            (BETR, BETRM, OE( 405)), (BETA0, OE( 406)), (BETAD0, OE( 407)),
            (C, ALP0, OE( 408)), (ALPMAX, OE( 409)), (ALPMIN, OE( 410)),
            (D, CD8, OE( 411)), (CMA, OE( 412)),
            (E, CMHT, OE( 414)), (CMHTA, OE( 415)), (CA1, OE( 416)),
            (F, CDAY2, OE( 417)), (CDAB, OE( 418)),
            (G, DEL, OE( 420)), (DELT, OE( 421)), (DELT0, OE( 422)),
            (H, DELX, OE( 423)), (D9Y2, OE( 424)), (D9X0, OE( 425)),
            (I, DYT, OE( 426)), (DZDT, OE( 427)), (DELW, OE( 428)),
            (J, DELAM, OE( 429)), (EYE, OE( 430)), (EM, OE( 431)),
            (K, OELT, OE( 432)), (ELT, OE( 433)),
            (L, HI, OE( 435)), (PSIDO, OE( 436)),
            (M, PHI, OE( 438)), (PHI2, OE( 439)), (PSIO, OE( 440)),
            (N, RI, OE( 441)), (R2, OE( 442)), (R3, OE( 443)),
            (Q, R4, OE( 444)), (RTRIM, OE( 445)), (TC, OE( 446)),
            (P, TC, OE( 447)), (TS, OE( 448)), (TG0, OE( 449)),
            (Q, TP, OE( 450)), (TMLAG0, OE( 451)), (TMAD0, OE( 452)),
            (R, TOL, OE( 453)), (VC, OE( 454)), (VCO, OE( 455)),
            (S, VYA, OE( 456)), (VZA, OE( 457)), (WX, OE( 458)),
            (T, W, OE( 459)), (W0, OE( 460)),
            (EQUIVALENCE (VZ1 = V1 * V1)
            (CAYD = (CD2 - CD1) / (1.00), V2/V2 - 1.00 / V1^2)
            (CD8 = CD1 - CAYD / V1^2)
            (B = 1.00 * (WZ / (V0 - W) ** 2)
            (F = RHO * A * G / 2.0 * Du / W

DATA RFA, G / 57.2957795100 * 32.17400 /
WX = VWIND
WZ = PHWIN
V1^2 = V1 * V1
CAYD = (CD2 - CD1) / (1.00), V2/V2 - 1.00 / V1^2
CD8 = CD1 - CAYD / V1^2
B = 1.00 * (WZ / (V0 - W) ** 2)
F = RHO * A * G / 2.0 * Du / W

HITIS2071
HITIS2072
HITIS2073
HITIS2074
HITIS2075
HITIS2076
HITIS2077
HITIS2078
HITIS2079
HITIS2080
HITIS2081
HITIS2082
HITIS2083
HITIS2084
HITIS2085
HITIS2086
HITIS2087
HITIS2088
HITIS2089
HITIS2090
HITIS2091
HITIS2092
HITIS2093
HITIS2094
HITIS2095
HITIS2096
HITIS2097
HITIS2098
HITIS2099
HITIS2100
HITIS2101
HITIS2102
HITIS2103
HITIS2104
HITIS2105
HITIS2106
HITIS2107
HITIS2108
HITIS2109
HITIS2110
HITIS2111
HITIS2112
HITIS2113
HITIS2114
HITIS2115
HITIS2116
HITIS2117
HITIS2118
HITIS2119
HITIS2120
D.29 Subroutine INCONF (continued)

```fortran
IF(IDUMP .EQ.1) WRITE(6,1000) WX,WZ,CAYD, CD8, B, F
1000 FORMAT(// WX WZ CAYD CD8 ) HITS 2121
1 F = /1X,1P6E12.4 /
THETA0 = ANGO/RFAC
PSIO = PHIANG/RFAC
THETO0 = RATE0
PSIDO = PHIRAT
DYDX0 = GAMO/RFAC
DZDX0 = PHIGAM/RFAC
DYTO = VO * DYDX0
DZTO = VO * DZDX0
THETAO = THETA0 - DYDX0
BETAO = DZDX0 - PSIO
THETO = THETO0
PSIDO, DYDX0, DZDX0, DYTO, DZTO
FCV = F * CNA * V0
ALPHD0 = THETO0 - FCV * ALPHO - P * BETAO
IF(ICNCL.EQ.1) ALPHDO=ALPHD0+F*V0*THETAO*(CD8+CAYD/V0/V0)
BETADO = -PSIDO - FCV * BETAO + P * ALPHO
IF(ICNCL.EQ.1) BETADO=BETADO-F*V0*PSIO*(CD8+CAYD/V0/V0)
CMA = - CNA * ELL * SM / D
EYEP= 2.00 * AIY / RHO / A / D / V0 / V0
EMP = 2.00 * W / RHO / A / G / V0
D2V = D / 2.00 / V0
CMTHDA = CMQ * D2V
CMTHTA = CMQA* D2V
H = ZZ(CNA, CD8, CAYD, VO, EMP, ICNCL)
CCON = CNA / EMP + CMTHD / EYEP
P11 = P * AIY / AIY
WCON = -4.00 * CMA / EYEP + PII * PII -CCON * CCON
IF(IDUMP .EQ.1) WRITE(6,1002)ALPHO, BETAO, ALPHDO, BETADO, CMA, EYEP
HITS 2151
1 * EMP, CMTHD, CMTHA, H
HITS 2152
W0 = DSGRT(.125 * ( WCON + DSGRT(WCON + WCON + 4.00*(PII)))
HITS 2153
1 * CCON + 2.00 * P * CMTHA / EYEP ) ** 2 ) ) )
HITS 2154
IF( CMA .LT. 0.) GO TO 3
HITS 2155
PCR=0.00
HITS 2156
XMU=0.00
HITS 2157
RTRIM=1.0
HITS 2158
PHIROL= 0.
HITS 2159
GO TO 4
HITS 2160
3 CONTINUE
HITS 2161
PCRV = DSGRT(-CMA*RHO * A / 2.00*D/(AIY-AIX))
HITS 2162
PCR= PCRV*DABS(V0)
HITS 2163
XMU= (CNA*RHO/4.00 *A/ W*G-CM0*D0/8.00*RHO*A/(AIY-AIX))/PCRV
HITS 2164
PPCR = P/PCR
HITS 2165
PPCRI=1.00- PPCR**2
HITS 2166
XMU2=2.00 * XMU1*PPCR
HITS 2167
RTRIM= 1.00 /DSGRT(PPCR1**2 + XMU2**2)
HITS 2168
PHIROL= DATA(XMU2/PPCR1)
HITS 2169
IF( PPCRI .LT. 0.) PHIROL = PHI0UL+3.14159265358980
HITS 2170
```

D.29 Subroutine INCONF (continued)

```
ABAR = OSORT(TRIM**2 + PNHITR**2)
PNHITR = ZATAN2(PNHITR,TRIM)
CONTINUE
ALPTRM = RTRIM * ABAR / RFAC
PTPR = PNHITR * PHIROL
BETTRM = ALPTRM * DSIN(PTPR)
ALPTRM = ALPTRM * DCOS(PTPR)
IF(IDUMP *EQ.1) WRITE(6,1006) XMU,RTRIM,PHIROL,ALPTRM
1006 FORMAT(*,
1     PCRA/ XMU  RTRIM  PHIROL  ALPTRM
1TRM
2  W1 = W0 - DELW
W2 = W0 + DELW
AA = ALPH0 - ALPTRM
XLAM1 = XLAM0 + DELLAM
XLAM2 = XLAM0 - DELTAM
BB = BET40 - BETTRM
RCON1 = ALPHD0 + W2 * BB
RCON2 = BETADO - W2 * AA - XLAM2 * BB
WL2 = 2.00 * (W0 * W0 + DELLAM ** 2)
R1 = (W0 * RCON1 + DELTAM * RCON2) / WL2
R2 = (-W0 * RCON2 + DELTAM * RCON1) / WL2
IF(IDUMP *EQ.1) WRITE(6,1004) W0, DELW, XLAM0, DELTAM, W1, W2
1004 FORMAT(*, W0, DELW, XLAM0, DELTAM, R1
1W2, XLAM1, XLAM2, WL2, R1, R2
2  XI, Y, 1P11E12.4, 1P11E12.4 )
1001 FORMAT(*, THETAO, PSI0, THEH
LTD0, PS1D0, DYD0, DZD0, DYD0, DZD0, THE
1002 FORMAT(*, ALPHD0, BETADO, CMNTHD, CMNTHA, H*
2/IX, 1P11E12.4 )
```

HITS2171
HITS2172
HITS2173
HITS2174
HITS2175
HITS2176
HITS2177
HITS2178
HITS2179
HITS2180
HITS2181
HITS2182
HITS2183
HITS2184
HITS2185
HITS2186
HITS2187
HITS2188
HITS2189
HITS2190
HITS2191
HITS2192
HITS2193
HITS2194
HITS2195
HITS2196
HITS2197
HITS2198
HITS2199
HITS2200
HITS2201
HITS2202
HITS2203
HITS2204
HITS2205
HITS2206
HITS2207
HITS2208
HITS2209
HITS2210
HITS2211
HITS2212
HITS2213
HITS2214
HITS2215
HITS2216
HITS2217
HITS2218
HITS2219
HITS2220
**D.30 Subroutine INITIL**

```fortran
SUBROUTINE INITIL
IMPLICIT REAL*8 (A-H,O-Y)
COMMON /CARDOM/ B(200), C(5000), IA(200,2), ID(10+2), INC.
1 IOPRNT, IOUT, IY, KARS1(3), KA, KB, KC, KD, KY, K234, LABC(3)
COMMON /CISPRN/ ISPRNT
COMMON /CMCPRT/ MCPRNT
COMMON /CRAN/ IRANNO
COMMON/IMCH/ID1(200), ID2(200), I1(400), I2(400), IDT(10), IIT(20),
1R1(10), IR2(10), NTRIAL, NCELL, MCALC, MCPT, IRJ1, IRJ2, IN8, NCC, NR1, NR2
COMMON/HTSPLIC/ IOPLT, ISPLT, ZTITLE(20,30), 2DATAD(21,4,30)
DIMENSION I8(2,200), IOE(160), IDE(150), IDE(460), I13(3)
1 KOE(3)

EQUIVALENCE (IOE(1), IOE(1)), (IOE(161), IOE(1)),
1 (IOE(311), IOE(11)), (IB1(1), B(1))

DATA IOEA / 1, 2, 3, 4, 5, 6, 7,
A 8, 9, 10, 11, 12, 13, 14, 15, 16,
AIRJ1 17, 18, 19, 20, 21, 22, 23, 24, 25,
C 26, 27, 28, 29, 30, 101, 102, 103, 104,
E 105, 106, 107, 108, 109, 110, 111, 112, 113,
F 114, 115, 116, 117, 118, 119, 120, 121, 122,
G 123, 124, 125, 126, 127, 128, 129, 130, 131,
DATA KOE / 70, 150, 150 /
DU 10 J=1,2
DO 10 I=1,1,200
10 IA(I,J) = 0
DO 20 J=1,1,200
20 B(I) = 0.000
C FIRST CARD IN TELLS WHICH OF THE 3 IOE DATA STATEMENTS (IY) WILL BE
C PLACED INTO THE FIRST COLUMN OF IA.
READ (INC*9001, END=70) IY, IOPRNT, ISPRNT, MCPT, MCALC, NCELL,
1 NTRIAL, IRJ1, IRJ2, MCPRNT, IRANNO
3, IOPLT, ISPLT
9001 FORMAT(1I1, 2I3, 1I5)
C
C INPUT LISTING
WRITE(IOUT, 5000)
5000 FORMAT(1Hi, 43X, '*** SIMULATION INPUT SUMMARY ***///)
WRITE(IOUT, 5001) IY, IOPRNT, ISPRNT, MCPT, MCALC, NCELL, NTRIAL,
1 IRJ1, IRJ2, MCPRNT, IRANNO, IOPLT, ISPLT
5001 FORMAT(30X,
1 'SINGLE-CASE, RANGE-CHECK, AND STATISTICAL PROCESSOR CON
4 ATROLS', ///
4 'IY = .4X, I1.5X, IOPRNT = .2X, I3.5X, ISPRNT = .4X, I5.5X
1 3X, .5X, IRJ1 = .6X, IRJ2 = .5X, MCPRNT = .5X, MCALC = .5X, NTRIAL = .5X, IN8 = .5X
1 NCELL = .6X, MCPT = .6X, MCPT = .6X
3 1 /, 25X, IOPLT = .5X, ISPLT = .5X
E///24X, 'IRJ2 = .5X, MCPRNT = .5X, IRANNO = .5X
1 E///24X, 'IOPLT = .5X, ISPLT = .5X
```
D.30 Subroutine INITIL (concluded)

WRITE(IOUT,5002)
5002 FORMAT(IHI,42X,'*** INPUT VARIABLE SPECIFICATIONS ***',//, 
A10X,6X,'CODE',6X,4X,'VARIABLE',4X,5X,'NOMINAL',4X,4X,'TOLERANCE', 
B3X,4X,'STANDARD',6X,3X,'SUBSEQUENT',2X,/
C10X,5X,'NUMBER',5X,6X,'TYPE',6X,6X,'VALUE',5X,1' ',4X,'DEVIATION', 
D5X,5X,'POINTS',5X,/) 
CALL RAND ( IRANDO ) 
DO 25 I=1,20
25 X = RANDOM ( Y )
C SINCE THE ENTRIES IN THE DATA STATEMENTS ARE PACKED TIGHT, IADD WILL 
C INDICATE HOW MANY TO ADD ON TO THE LOOP COUNTER (IN ORDER TO PICK 
C UP THE CORRECT I0E).
C
IADD = 0
IF ( IY .GT. 1 ) IADD = 160
IF ( IY .GT. 2 ) IADD = 310
C KA IS THE COUNT OF THE INDEPENDENT VARIABLES IN THE FIRST COLUMN OF IA
KA = KOE(IY)
DO 30 I=1,KA
30 IA(I+1) = I0E(I+IADD)
GO TO (40, 50, 60), IY
40 CALL TG2987
   RETURN
50 CALL TG2440
   RETURN
60 CALL TG1795
   RETURN
70 WRITE(IOUT,9002)
9002 FORMAT(IHI,59,/,1X,6('THATS ALL FOLKS',5X))
   CALL EXIT
   RETURN
   END
D.31 Subroutine INLHST

SUBROUTINE INLHST

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/IMCH/ID1(200),ID2(200),II1(400),II2(400),IDT(10),IIIT(20),
II1(1*),IT2(10),NTRIAL,NCALL,MCALC,MCALM,IRJ1,IRJ2,INB,NCC,NR1,NR2

C THIS SUBROUTINE ZEROS OUT THE HISTOGRAM AND MONTE CARLO COUNTERS
NCC=0
NR1=0
NR2=0
INB=0
DO 1 I=1,10
  IR1(I)=0
  IR2(I)=0
1  DO 2 I=1,200
  ID1(I)=0
  ID2(I)=0
2  DO 3 I=1,400
  II1(I)=0
  II2(I)=0
3  RETURN
END
D.32 Subroutine LOADER

SUBROUTINE LOADER (IV, CM, CV, FCTR, NRR, PM, PV, PT, NCELL)

1   * A * IPASS *)
   IMPLICIT REAL * 8 (A-H, O-Y)
   DIMENSION A(20, 5)
   1, IB(2,200)
   EQUIVALENCE (IB(1,1), B(1))
   COMMON/HISPLT/ IOPLLOT, ISPLLOT
   COMMON /CARDCM / B(200), C(5000), IA(200,2), ID(10,2), INC
   1, IOPRT, IOUT, IY, KARSI(3), KA, KB, KC, KD, KTI, K234
   2, LABC(3)
   DATA PI / 3.14159265400 /
   DATA J /0 /
   J= J+1
   IF( J .GE. KD+K234) J= l
   GO TO ( 1, 2, 3,4 ) . IPASS
   1 JSTRT = 1
   JD = 0
   J2 = 2
   JEND = KD
   JS = 4
   GO TO 5
   2 JEND = KD + K234
   JSTRT = 1 + KD
   JS=4
   J2 = 2
   JD = 0
   GO TO 5
   3 JSTRT = 1
   JEND = KD
   JS=11
   J2 = 4
   JD = 10
   GO TO 5
   4 JSTRT = 1 + KD
   JEND = KD + K234
   JS=11
   J2 = 3
   GO TO 5
   5 FPI = 1.00 / DSQRT( 2.00 * PI )
   NCELL = NCELL +1
   IF( J .LE. KD ) GO TO 10
   ITYPE = IB(1,IA(J-KD,2))
   ZTITLE(20, J) = 0.
   IF ( ITYPE .EQ. 2 ) ZTITLE(20, J) = 1.
   GO TO 11
   10 ITYPE = 7
   IF ( ID(J,2) .NE. 0 ) ITYPE = 8
   11 ZTITLE(1, J) = IPASS
   ZTITLE(2, J) = IV
D.32 Subroutine LOADER (continued)

```
ZTITLE(J) = [TYPE
ZTITLE(JS,J) = NRR
ZTITLE(JS+1,J) = CM
ZTITLE(JS+2,J) = CV
ZTITLE(JS+3,J) = PM
ZTITLE(JS+4,J) = PV
ZTITLE(JS+5,J) = FCTR
ZTITLE(JS+6,J) = PT
ZTITLE(18,J) = NCELL
    SUM = 0.0
    DO 7 IN = 1, NCELL
        SUM = SUM + A(IN,4)
    SUM = 100.0 / SUM
    SAVE = -1.020
    DO 8 IN = 1, NCELL
        ZDATAD(IN, J2,J) = A(IN,4) * SUM
        IF ( SAVE .LT. ZDATAD(IN, J2,J) ) SAVE = ZDATAD(IN, J2,J)
    IF ( IPASS .EQ. 4 ) GO TO 8
    IF ( IPASS .EQ. 2 ) GO TO 9
    ZDATAD(IN, IPASS,J) = A(IN,2)
    GO TO 8
9    ZDATAD(IN, J) = A(IN,2)
    IF ( ITYPE .NE. 2 ) GO TO 20
        F = DEXP( -(A(IN,1) - ZTITLE(7,J)) **2 /2. 
    1 /ZTITLE(8,J) ) * FPI / SQRT( ZTITLE(8,J) )
    ZTITLE(20,J) = 1
    GO TO 21
20 IF ( ITYPE .NE. 3 ) GO TO 22
    ZTITLE(20,J) = 0
    F = 1.00 / ( ZDATAD( NCELL +1,1,J) -ZDATAD(1,1,J) )
    GO TO 21
22 IF ( ITYPE .NE. 4 ) GO TO 23
    KB = IA(J-KD,2)
    ITL = IB(1, KB + 4)
    KC = IB(1, KB + 5)
    DATA( IN,1)
    CALL ARTLU(1, DATA, C(KC+ITL), F, C(KC) )
23 CONTINUE
6 CONTINUE
    IF ( IPASS .NE. 2 ) GO TO 25
    ZDATAD(NCELL+1,J) = A(NCELL,3)
    ZDATAD(NCELL+2,J) = 0.0
    CONTINUE
25 IF ( IPASS .EQ. 1 ) GO TO 27
    IF ( IPASS .EQ. 4 ) GO TO 28
C IPASS = 3
```

D.32 Subroutine LOADER (concluded)

ZOATAD(NCELL1, 3, J) = A(NCELL + 3)
ZOATAD(NCELL1, 4, J) = 0.0
GO TO 26
ZOATAD(NCELL1, 3, J) = 0.0
26 CONTINUE
IF ( IPASS .EQ. 1 .OR. IPASS .EQ. 2 ) GO TO 14
IF( ZTITLE(19, J) .LT. SAVE ) ZTITLE(19, J) = SAVE
GO TO 15
ZTITLE(19, J) = SAVE
14 CONTINUE
15 CONTINUE
C IF( IPASS .EQ. 4 )
C CALL HISPLT
RETURN
END
D.33 Subroutine MCRL

SUBROUTINE MCRL
IMPLICIT REAL*8 (A-H,O-Y)
COMMON/CARDCH,B(200),C(5000),IA(200),ID(10,2),INC,IOPTN,IOUT,HITS2429
IIY,KARS(3),KAT,KB,KC,KT1,K234,LABC(3)
COMMON/MCPRT/,MCPRNT
COMMON/INCH/ID(200),IO2(200),II1(400),II2(400),IOT(10),IIT(20),
IIT(110),ID2(10),NTRIAL,NC,MC2,MCF,IRJ1,IRJ2,IN8,NC,MR1,MR2
COMMON/HSPLIT/I0PL0T,ISPL0T,ZTITLE(3120)
COMMON/OCE/DE(600)
KCAS = 0
DO 400 IBB = 1,3120
400 ZTITLE(IBB) = 0.0
CALL INLMST
CALL CHKIN
IF(MCPRNT.EQ.1) WRITE(IOUT,5000)
5000 FORMAT(I1,39X,"*** SUMMARY OF MONTE CARLO RANDOM EXPERIMENTS***")
A./////1
C START MONTE CARLO LOOP WITH SAMPLE GENERATOR
1: L SAMPLE
C RUN CASE
KCASE = KCASE + 1
IF(MC2CALC) 20,20,25
20 CALL PICK1
GO TO 22
22 IF(MCPRNT) 24,24,23
23 CONTINUE
C USE THE LAST TEN ENTRIES IN THE C ARRAY FOR OUTPUT,
X = 0
231 JC = 4990
232 K = K + 1
IF(K.GT.K234) GO TO 233
JA = IA(K+1)
JC = JC + 1
C(JC) = DE(JA)
IF(JC.LT.5000) GO TO 232
233 WRITE(IOUT,9002)KCASE,(C(I)),I=4991,JC
9002 FORMAT(I1X,"CASE",15X,"IND. VAR.*",1P10E12.4)
IF(K.LT.K234) GO TO 231
K = 0
234 JC = 4990
235 K = K + 1
IF(K.GT.KD) GO TO 235
JD = ID(K+1)
JC = JC + 1
C(JC) = DE(JD)
GO TO 236
236 WRITE(IOUT,9003) (C(I)),I=4991,JC
9003 FORMAT(I1X,"DEP. VAR.*",1P10E12.4,///)
24 IER = 0
D.33 Subroutine MCRL (concluded)

C PERFORM INTERVAL CHECK
IF=0
C . HSTSl(IER)
IF(IER)2,3,3
2 NR1=NR1+1
IF(NR1-IRJ1)3,3,7
3 IF(IN8)6,6,A
C IF THERE ARE TYPE 8,5 PERFORM CHECK ON USER SPECS
4 IER=0
CALL HSTG2(IER);
IF(IER)5,6,5
5 NR2=NR2+1
IF(NR2-IRJ2)6,6,7
C INCREMENT SAMPLE COUNTER AND CHECK -- END OF MONTE CARLO LOOP
6 NCC=NCC+1
IF(NCC-NTRIAL)1,7,7
7 IF(IN8)6,6,9
C KEND=2
GO TO 10
9 KEND=4
C START PRINTING RESULTS
10 WRITE(IOUT,200) NTRIAL, NR1, NR2
200 FORMAT(1H4.12X,'MONTE CARLO RESULTS',10X,'SAMPLE = ',15.4X,'INTERNAL RANGES'
1 REJECTS = ',14.4X,'USER REJECTS = ',14)
JO 21 L=1,KEND
GO TO(12,13,14,15),L
12 WRITE(IOUT,201)
201 FORMAT(//20X,'DEPENDENT VARIABLES WITH INTERNAL RANGES')
GO TO 11
13 WRITE(IOUT,202)
202 FORMAT(//20X,'INDEPENDENT VARIABLES WITH INTERNAL RANGES')
GO TO 11
14 WRITE(IOUT,203)
203 FORMAT(//20X,'DEPENDENT VARIABLES WITH USERS RANGES')
GO TO 11
15 WRITE(IOUT,204)
204 FORMAT(//20X,'SORTED INDEPENDENT VARIABLES FOR USERS RANGES')
11 CALL PRNTMC(L)
21 CONTINUE
RETURN
END
.34 Subroutine MOVEUP

SUBROUTINE MOVEUP
IMPLICIT REAL*8 (A-H, O-Z)
COMMON /CARDCM/ B(200), C(5000), IA(200, 2), ID(10, 2), MC,
ICORNT, IOLT, IV, KARS1(3), KA, KB, KC, KD, KT1, K234, LABC(3)
DIMENSION IB(2, 200)
EQUIVALENCE (IB(1, 1), B(1))
C IF THERE WERE ANY TYPE 1, MOVE THEM UP TO THE START OF THE IA ARRAY.
IF (KT1 .EQ. 0) GO TO 20
KTIR = 0
17 KTIR = KTIR + 1
J1 = KARS1(KTIR)
JS1 = IA(KTIR, 1)
JS2 = IA(KTIR, 2)
IA(KTIR, 1) = IA(J1, 1)
IA(KTIR, 2) = IA(J1, 2)
IA(J1, 1) = JS1
IA(J1, 2) = JS2
IF (KTIR .LT. KT1) GO TO 10
GO TO 50
20 CONTINUE
C RUN THRU IA. GET SUBSCRIPT OF B. IF TYPE 2, 3, OR 4, MOVE UP IN IA.
K234 = 0
IAR = 0
30 IAR = IAR + 1
IF (IA(IAR, 1) .EQ. 0 .OR. IA(IAR, 2) .EQ. -1) GO TO 40
IF (IB(IAR, 1) .EQ. 1 .OR. IB(IAR, 2) .GT. 4) GO TO 40
K234 = K234 + 1
JS1 = IA(K234, 1)
JS2 = IA(K234, 2)
IA(K234, 1) = IA(IAR, 1)
IA(K234, 2) = IA(IAR, 2)
IA(IAR, 1) = JS1
IA(IAR, 2) = JS2
40 IF (K234 .EQ. K234) GO TO 50
IF (IAR .LT. KA) GO TO 30
WRITE (10, OUT, 9006)
9006 FORMAT(71H MOVEUP ERROR . . . REACHED END OF IA BEFORE FINDING ALL
TYPE 2, 3 AND 4,)
CALL EXITS
50 CONTINUE
RETURN
END
FUNCTION NCOV(I, J)
IMPLICIT REAL*8(A-H, O-Z)
COMMON /CARDCM/ 8(200), C(5000), IA(200,2), ID(10,2), INC.
1 IOPRNT, IOUT
COMMON /INDEX/ KD, K234, KCYBAR, KCSUMS, KCOUB, KCOV
IF(I .NE. J) GO TO 100
WRITE(IOUT, 99) II, JJ
99 FORMAT(///, ' DO234 TRIES TO LOOK UP VARIANCE IN COVARIANCE ARRAY I = ', I3, ' J = ', I3)
100 CONTINUE
IF (JJ .LT. II) GO TO 1
I = II
J = JJ
GO TO 2
1 I = JJ
J = II
2 CONTINUE
NSUM = 0
II = I-1
IF (II .EQ. 0) GO TO 3
DO 4 IK = 1, II
4 NSUM = NSUM + K234 - IK
3 CONTINUE
NCOV = NCOV + NSUM * J - I
IF(NCOV .LE. 2000) RETURN
WRITE(IOUT, 98) NCOV
98 FORMAT(///, ' COVARIANCE MATRIX EXCEEDS THE C ARRAY. NCOV = ', I4)
RETURN
END
D.36 Function NMEAN

```
FUNCTION NMEAN(I)
IMPLICIT REAL*8(A-H, O-Z)
COMMON /CARDCM/ B(200), C(5000), IA(200,2)
COMMON /INDEX/ KD, K234, KCYBAR, KCSUMS, KCOUB, KCOV
NMEAN= KCSUMS+I
RETURN
END
```
D.37 Function NVAR

```
FUNCTION NVAR(I)
IMPLICIT REAL*8(A-H, O-Z)
COMMON /CARDCM/ B(200), C(5000), IA(200,2)
COMMON /INDEX/ KD, K234, KCYBAR, KCSUMS, KCDQUB, KCOV
NVAR = KCSUMS *KD + I
RETURN
END
```
D.38 Function NVARX

```
FUNCTION NVARX(I)
IMPLICIT REAL*8(A-H, O-Z)
COMMON /CARDCM/ B(200), C(5000), IA(200,2)
COMMON /INDEX/ KD, K234, KCVBAR, KCSUMS, KCDRUB, KCOV
NVARX= IA(I,2) + 3
RETURN
END
```
D.39 Function N1D

FUNCTION N1D(I,J)
IMPLICIT REAL*8(A-H, O-Z)
COMMON /CARDCM/ B(200), C(5000), IA(2000)
COMMON /INDEX/ KD, K234, KCYBAR, KCUMS, KCDOUB, KCOV
N1D= KCYBAR + (2*J-1)*K + I
RETURN
END
D.40 Function N2D

FUNCTION N2D(I, JJ, KK)
IMPLICIT REAL*8(A-H, O-Z)
COMMON /CARDCM/ B(20), C(5000), IA(200,2)
COMMON /INDEX/ KO, K234, KCYBAR, KCSUMS, KCDOUB, KCOV
IF (KK*LT. JJ) GO TO 1
J = JJ
K = KK
GO TO 2
1
J = KK
K = JJ
GO TO 3
2
CONTINUE
IF (J.GE. K) GO TO 4
N2D = KCYBAR + 2*J*KD + I
RETURN
3
CONTINUE
J1 = J-1
NSUM=0
IF (J1*EQ. 0) GO TO 5
DO 4 JK = 1, J1
4 NSUM = NSUM + K234 - JK
CONTINUE
N2D = KCDOUB + KD *(NSUM + K - J - 1) + I
RETURN
END
D.41 Subroutine PICK1

SUBROUTINE PICK1
IMPLICIT REAL*8 (A-H,O-Z)
COMMON / CARDCM/ B(200), C(50, 0), IA(200,2), ID(10,2), INC,
1 IDPRNT, ICUT, IY, KARSI(3), KB, KC, KD, KTI, K234, LABC(3)
GO TO (1, 2.3), IY
1 CALL G2987
RETURN
2 CALL G2440
RETURN
3 CALL G1795
RETURN
END
D.42 Subroutine PRNTMC

```
SUBROUTINE PRNTMC(IPASS)
  IMPLICIT REAL*8 (A-H,O-Y)
COMMON/CARDCM, B(200), C(5000), IA(200,2), ID(10,2), INC, IOPRTN, IOUT,
       IIV, KARS(3), KA, KB, KC, KD, KT1, K234, LABC(3)
COMMON/IMCH/I0D(200), ID(400), IDT(10), IIT(20), INR1(10), IR2(10),
       NTRIAL, NCELL, MCALC, MCQPT, NR, JR1, JR2, IN8, NCC, NR, N2R
DIMENSION A(20,5)
COMMON/HI, SPLOT, ISPLOT, ZTITLE(20,30), ZDATAD(21,4,30)
  EQUIVALENCE (A(1,1),C(4900))
  EQUIVALENCE (KADD, KC)
  THIS ROUTINE NEEDS THE KADD ADDRESS OF THE NOMINAL CASE RESULTS
  ISN=0
  ENCL=NCELL
  GO TO (1*2,1,2), IPASS
  KE=KD
  LADD=2*KD*234 + KD + (KADD - 1)
  GO TO 3
  KE=K234
  DO 17 I=1,KE
  GO TO (4-5,6,5), IPASS
  FIRST PASS FOR THE DEPENDENT VARIABLES
  DEFINITIONS PM = PRIOR MEAN VALUE            PV = PRIOR VARIANCE
        PT = PRIOR TOLERANCE                        FCTR = CLASS INTERVAL
        IV = VARIABLE CODE NO                      NRR = NUMBER OF Rejects
  IV=ID(I,1)
  INN=LADD + I
  IVAR=INN+KD
  PV=C(IVAR)
  PT=3.#DOSORT(PV)
  PM=C(INN)
  FCTR = 2.000 * PT / ENCL
  NRR=IR1(1)
  GO TO 9
  FIRST AND THIRD PASS FOR INDEPENDENT VARIABLES
  IV=IA(I,1)
  L=IA(I,2)
  NRR=0
  PM=B(L+1)
  PT=B(L+2)
  PV=3(L+3)
  FCTR = 2.000 * PT / ENCL
  GO TO 9
  FOURTH PASS FOR USER SPEC DEP. VARIABLES
  L=ID(I,1)
  CHECK IF IT IS A TYPE 8
  IF(L)4,4,7
  IV=ID(I,1)
  NRR=IR2(1)
  GO TO 8
  CONVERT AND TRANSFER HISTOGRAMS TO OUTPUT ARRAY A
```

D.42 Subroutine PRNTMCH (concluded)

```
9  DO 14 J=1,NCELL
   10  A(J)=ID1(IBN+J)
   11  GO TO 14
   12  A(J)=ID2(IBN+J)
   13  CONTINUE
C  FILL IN THE OUTPUT ARRAY AND CALC MEAN AND VARIANCE
   14  CALL FILL(A,FCTR,PM,PT,CV,NCELL)
   15  IF(IOPLOT,NE,0)
   16  ICALL LOADER(IV,CN,CV,FCTR,NNR,PM,PT,NCELL,A,IPASS)
   17  WRITE(1DOUT,200) IV,CN,CV,FCTR,NNR,PM,PT
   200  FORMAT(92X,'CODE NUMBER '*'E14.6','4X,'MEAN '*'E14.6','4X,'VAR '='E14.6',6HITS2733
   1X,'INTERVAL '='E14.6','2X,'NO. REJECTS '='E15.2X,'(MEAN) '='E14.6',6HITS2734
   22X,'TOLERANCE '='E14.6/4X,'CLASS MARK '='E14.6/4X,'3X,'LOWER BOUND '='E14.6/4X,'3X,'UPPER BOUND '='E14.6/4X,'3X,'FREQ '='E14.6/4X,'3X,'DIST. FUNCT. ')
   18  DO 16 K=1,NCELL
   19  WRITE(1DOUT,201) A(K),L=1,5
   201  FORMAT(3E14.6,F5.0,E14.6)
   21  IBN=IBN+NCCELL
   22  RETURN
END
```

HITS2718 HITS2719 HITS2720 HITS2721 HITS2722 HITS2723 HITS2724 HITS2725 HITS2726 HITS2727 HITS2728 HITS2729 HITS2730 HITS2731 HITS2732 HITS2733 HITS2734 HITS2735 HITS2736 HITS2737 HITS2738 HITS2739 HITS2740 HITS2741 HITS2742
D.43 Subroutine QCOR

SUBROUTINE QCOR
IMPLICIT REAL*8(A-H, O-Z)
COMMON /CARDCM/ B(200), C(5000), IA(200,2)
COMMON /INDEX/ KD, K234, KCYBAR, KCSUMS, KCOV
K1 = K234 - 1
IF( K1 .EQ. 0) RETURN
DO 1 I = 1, KD
   DEL = 0.D0
   DELP = 0.D0
   DO 2 J = I, K1
      NXJ = NID(I, J)
      YXJ = C( NXJ)
      KB = J + 1
      DO 2 K = KB, K234
         N2 = N2D(I, J, K)
         NK = N1D(I, K)
         NC = NCOV(J, K)
         CV = C(NC)
         DEL = DEL + C(N2) * CV
         DELP = DELP + YXJ * C(NK) * CV
         DELP = DELP + 2.D0
         NY = NMEAN(I)
         C(NY) = C(NY) + DEL
         NV = NVAR(I)
         C(NV) = C(NV) + DELP
      END
1 CONTINUE
RETURN
END
D.44  Function RANDOM

```
FUNCTION RANDOM(X)
IMPLICIT REAL*(A-H,O-Z)
CALL RANDU(IX, IY, X)
IX = IY
RANDOM = X
RETURN
ENTRY RAND(IX)
RETURN
END
```
D.45 Subroutine RANDU

SUBROUTINE RANDU(IX, IY, YFL)
IMPLICIT REAL*8(A-H,O-Z)

I = 1
YFL = 1.483647 + 1
RETURN
END

-329-
D.46 Subroutine RW2987 (concluded)

A57 = LPCON / 57.29577951D0
CALL XXC(XLAM0, XC, VO, DELLA, CAY1, CAY2, TOL, A57)
CALL TRAJT(TN, XC, VXT, XT)
CALL CROSS(XT, VXT)
CALL WND(VXT, TN, XT, WZ, WX, VO, VZW, ZW)
YT = YA
ZT = ZA + ZW
VYT = VYA
VZT = VZA + VZW
CALL TRAJX(XN, XC, VXN, TX)
CALL CROSS(XN, VXN)
CALL WND(VXN, TX, XN, WZ, WX, VO, VZW, ZW)
YR = YA
ZR = ZA + ZW
VYR = VYA
VZR = VZA + VZW
RETURN
END
D.47 Subroutine SAMPLE

SUBROUTINE SAMPLE
IMPLICIT REAL*(A-H,O-Z)
COMMON/CA,CMY/B(200),IA(200),ID(10),INC,IPRNT,ROUT.
COMMON/DEC/DF(600)
COMMON/INCH/D(200),ID(200),I1(I400),I2(400),IDT(10),IIT(20)
IIR(I1),I2(I1),NCX,MCALC,MCOTP,IRJ1,IRJ2,IN8,NCC,NRR
DIMENSION IB(2,200)
EQUIVALENCE (I9(I1,1),B(I))
C THIS ROUTINE GENERATES THE INPUT SAMPLES AND REQUIRES A UNIFORM
C RANDOM NUMBER GENERATOR CALLED RANDOM,
IAN=1
ENN=NCX
DO 12 I=1,K234
12 LP=0
IA=IA(I,2)
1 CALL RANDOM(RN)
IF(I1,IADD-3),1,2
C IF TYPE NO. IS 2 CONVERT RN TO A GAUSSIAN DISTRIBUTION
2 CALL CNVX(RN)
VAL=B(IADD+1)+I2*R4(RN-1)*B(IADD+2)
RNS=FN*N
INDEX=NS
I1=INDEX
VAL IS THE RANDOM VALUE AND IIT IS THE HISTOGRAM ADDRESS
IF(I1,IADD),5,4
C STORE THE RANDOM VALUE IN THE GE ARRAY
5 IADD=IA(I,1)
QE(IADD)=VAL
GO TO 12
C FOR TYPE 4 CALCULATE ACTUAL PROB. DENSITY FOR CHOSEN RANDOM VALUE
4 INDEX=IADD+4
JY=IA(I,INDEX)
FEN=JY
FEN=FN*RN
J=FEN
INDEX=INDEX+1
JF=IA(I,INDEX)+J
JF=JY+JF
C JF AND JY ARE ADDRESSES IN THE C ARRAY OF DIST, FUNC, AND VARIAB
C VALUE RESPECT--THAT IS VAL IS GT C (JY) AND LT C(JY+1)--INTERPOLATE
C TO DETERMINE THE ACTUAL VALUE OF DIST, FCN, FOR CHOSEN VAL
C FACT=C(JF)+VAL-C(JY))(C(JF+1)-C(JF))/C(JY+1)-C(JY))
C CALCULATE RANDOM PROBABILITY
CALL RANDOM(RN)
FRY=RN*B(IADD+3)
IF(FACT-TRY1,6,5.5
C IF THE RANDOM VALUE IS GREATER THAN THE ACTUAL REJECT THIS SAMPLE
6 IF(I1P-2010,1):11
IF(I1P-2010,11)
D.47 Subroutine SAMPLE (concluded)

C     IF A VALUE CANT BE FOUND IN 20 TRYS QUIT
11    WRITE(IOUT,200)
20    FORMAT(10X,'TYPE 4 RANDOM SAMPLE ERROR')
     CALL EXIT
12    I3N=IBN+NCCELL
     RETURN
     END
D.48 Subroutine SPRNT

SUBROUTINE SPRNT
IMPLICIT REAL*8 (A-H, O-Z)
COMMON/HEAD1(180), IUNIT1(132), IUNIT2(48), IUNIT3(180), IUNIT4(133), IUNIT5(600)
COMMON/IFOR(50)
DATA IFOR1/30/, IFOR2/5/, IFOR3/6/, IFOR4/43/, IFOR5/77/, IFOR6/19/
B*(NONE)/ FT/SEC, FT/SEC, FT/SEC
C*RA/D/ SEC, SEC
D*SL-F/ SEC, SEC
E*FT/ SEC, SEC
F*(NONE)/ SEC, SEC
G*FT/ SEC, SEC
H*FT/ SEC, SEC
I*FT/SEC, SEC
J*FT/ SEC, SEC
K*FT/ SEC, SEC
L*SQ-FT/ SEC, SEC
M*SQ-FT/ SEC, SEC
N*RA/D/ SEC, SEC
O*RA/D/ SEC, SEC
P*SEC/ SEC, SEC
Q*SEC/ SEC, SEC
R*SEC/ SEC, SEC
S*SEC/ SEC
A*(NONE)/ SEC, SEC
B*SEC/ SEC, SEC
C*SEC/ SEC, SEC
D*SEC/ SEC, SEC
E*SEC/ SEC, SEC
F*SEC/ SEC, SEC
G*SEC/ SEC, SEC
H*IFOR(50)
I*INPECT, IFOR(50)
WRITE(IOUT,500)
WRITE(IOUT,501)
WRITE(IOUT,505)/ (HEAD1(1).OE(1).UNIT1(1),I=1,IFOR1)
IFOR = IFOR + IFOR2
WRITE(IOUT,502)
WRITE(IOUT,505)/ (HEAD1(1).OE(1+100).UNIT1(1),I=1,IFOR1)
WRITE(IOUT,503)
WRITE(IOUT,506)/ (HEAD1+IFOR, OE(1+200).UNIT1(1+IFOR),I=1,IFOR3)
IFOR = IFOR + IFOR3
WRITE(IOUT,504)

HITS2913
HITS2914
HITS2915
HITS2916
HITS2917
HITS2918
HITS2919
HITS2920
HITS2921
HITS2922
HITS2923
HITS2924
HITS2925
HITS2926
HITS2927
HITS2928
HITS2929
HITS2930
HITS2931
HITS2932
HITS2933
HITS2934
HITS2935
HITS2936
HITS2937
HITS2938
HITS2939
HITS2940
HITS2941
HITS2942
HITS2943
HITS2944
HITS2945
HITS2946
HITS2947
HITS2948
HITS2949
HITS2950
HITS2951
HITS2952
HITS2953
HITS2954
HITS2955
HITS2956
HITS2957
HITS2958
HITS2959
HITS2960
HITS2961
HITS2962
D.48 Subroutine SPRNT (concluded)

```fortran
WRITE(OUT,505)(HEAD(I+IFOR),OE(I+300),UNITS(I+IFOR),I=1,IFOR4)
500  FORMAT(1H1,38X,42H### PROJECTILE DISPERSION CASE SUMMARY ###)
      IFOR = IFOR + IFOR4
      WRITE(OUT,505)(HEAD(I+IFOR),OE(I+399),UNITS(I+IFOR),I=1,IFOR5)
      IFOR = IFOR + IFOR5
      WRITE(OUT,505)(HEAD(I+IFOR),OE(I+499),UNITS(I+IFOR),I = 1,IFOR6)
504  FORMAT(1H,49X,19HCOMPUTED QUANTITIES)
505  FORMAT(4(2X,A6,3H = .IPD11,4,1X,A6,2X))
506  RETURN
END
```
D.49 Subroutine STDDEV

SUBROUTINE STDDEV(A,K,KD,IOUT)
IMPLICIT REAL*8 (A-K,O-Z)
DIMENSION B(10)
B(K) = DSORT(A)
IF(K.NE.KD) RETURN
WRITE(IOUT,8887)(A(I), I = 1,KD)
WRITE(IOUT,8888)(B(I), I = 1,KD)
8887 FORMAT(1H*,9HSTD. DEV.,1X,1P1?E12,4)
8888 FORMAT(1H*,//,3X,*STANDARD DEVIATIONS WITH HIGHER PRECISION*,/)
RETURN
END
D.50 Subroutine STOREC

SUBROUTINE STOREC
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CARDCM/ B(200), C(5000), IA(200), ID(10,2), INC.
I IPRNT, IOUT, IY, KARS1(3), KA, KB, KC, KD, KTI, K234, LABC(3)
COMMON/DEC/DE(600)
C PLACE THE KD DEPENDENT VARIABLES INTO THE C ARRAY.
DO 10 I=1,KD
   JOE = ID(I,1)
   KC = KC + 1
   IF(KC GT 5000) WRITE(IOUT,900) KC
   900 FORMAT(10X,'KC = ',I5,1X,'IN STOREC WHICH EXCEEDS THE C ARRAY')
   C(KC) = DF(JOE)
10 CONTINUE
RETURN
END
D.51 Subroutine TG1795

SUBROUTINE TG1795
RETURN
END
D.53 Subroutine TG2987

SUBROUTINE TG2987
IMPLICIT REAL *8 (A-H,O-Z)
COMMON /CARDOM/ B(200), C(5000), IA(200,2), ID(10,2), INC
COMMON /IOPRNT, IOUT, IY, KARS1(3), KA, KB, KC, KD, KTI, K234, LABC(3)
COMMON/OEC/OE(600)
EQUIVALENCE (VOF, OE( 1)), (WXF, OE( 2)), (CBF, OE( 3)),
(AYF, OE( 3)), (AYF, OE( 4)), (CDI, OE( 5)),
(US, OE( 6)), (UW, OE( 7)), (V2F, OE( 8)),
(DF, OE( 9)), (DF, OE(10)), (SMF, OE(11)),
(CRM, OE(12)), (CRM, OE(13)), (ATRMS, OE(14)),
(EARMS, OE(15)), (EARMS, OE(16)), (PS1O, OE(17)),
(CGDO, OE(18)), (CGDO, OE(19)), (GAO, OE(20)),
(CGDO, OE(21)), (CGDO, OE(22)), (ELF, OE(23)),
(HOF, OE(24)), (HOF, OE(25)), (AIXF, OE(26)),
(JCAAF, OE(27)), (JCAAF, OE(28)), (ALPCOF, OE(29)),
(EQV, OE(30)), (QV, OE(31)), (WX, OE(102)),
(AWY, OE(103)), (RHO, OE(104)), (CD1, OE(105)),
(BVI, OE(106)), (CD2, OE(107)), (V2, OE(108)),
(CXN, OE(109)), (CNA, OE(110)), (SM, OE(111)),
(DCM, OE(112)), (CMAPA, OE(113)), (ATRMS, OE(114)),
(EARMS, OE(115)), (EARMS, OE(116)), (PS1O, OE(117)),
(FD, OE(118)), (FD, OE(119)), (GAMO, OE(120)),
(CZO, OE(121)), (CZO, OE(122)), (ELL, OE(123)),
(HO, OE(124)), (HO, OE(125)), (AIX, OE(126)),
(IATY, OE(127)), (IATY, OE(128)), (ALPCON, OE(129)),
(JCA, OE(130)), (JCA, OE(131)), (JCA, OE(132)),
(AZBART, OE(133)), (YABART, OE(134)), (ZBARX, OE(135)),
(EQV, OE(136)), (EQV, OE(137)), (EQV, OE(138)),
(AZBART, OE(203)), (YABART, OE(204)), (ZBARX, OE(205)),
(EQV, OE(206)), (EQV, OE(207)), (EQV, OE(208)),
(AZBART, OE(303)), (YABART, OE(304)), (ZBARX, OE(305)),
(EQV, OE(306)), (EQV, OE(307)), (EQV, OE(308)),
(TOL, OE(453))}

FIRE CONTROL SYSTEM PARAMETERS

VGF = +1.1000000D4
WVF = +0.0000000D0
WYF = +0.0000000D0
RHOF = +0.002378D0
CDIF = +3.5853980D-2
V1F = +1.6740000D4
CD2F = +1.1951330D-1
V2F = +3.906000D3
D.53 Subroutine TG2987 (continued)

\[
\begin{align*}
\lambda & = +1.00000004 \\
CNAF & = +1.97670000 \\
SNF & = +6.20000000-2 \\
CMQF & = -7.50000000 \\
CMPAF & = +0.00000000 \\
ATRMSF & = +0.00000000 \\
BTRMSF & = +0.00000000 \\
THETOF & = +0.00000000 \\
PS1OF & = +0.00000000 \\
TDOTOF & = +0.00000000 \\
PDDOTOF & = +0.00000000 \\
GAMOF & = +0.00000000 \\
AZOF & = +0.00000000 \\
AF & = +3.06800000-3 \\
ELLF & = +3.10000000-1 \\
DF & = +6.25000000-2 \\
WF & = +1.10000000-1 \\
AIXF & = +1.23460000-6 \\
AIYF & = +2.11030000-5 \\
PFF & = +4.000000002 \\
ALPCOF & = +5.00000000-1 \\
CAAF & = +0.000000000 \\
\end{align*}
\]

C

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

C PROJECTILE PARAMETERS (REAL WORLD)

C

\[
\begin{align*}
V0 & = +1.10000004 \\
WX & = +0.00000000 \\
WY & = +0.00000000 \\
RHO & = +0.000000002 \\
CD1 & = +3.58539800-2 \\
V1 & = +1.67400004 \\
CD2 & = +1.19513300-1 \\
V2 & = +3.96000003 \\
XN & = +1.00000004 \\
CNA & = +1.97670000 \\
SN & = +6.20000000-2 \\
CMQ & = -7.50000000 \\
CMPA & = +0.00000000 \\
ATRMS & = +0.00000000 \\
BTRMS & = +0.00000000 \\
THET0 & = +0.00000000 \\
PS10 & = +0.00000000 \\
TDOT0 & = +0.00000000 \\
PDDOT0 & = +0.00000000 \\
GAMO & = +0.00000000 \\
AZO & = +0.00000000 \\
A & = +3.06800000-3
\end{align*}
\]
D.54 Subroutine TRAJT (continued)

```
EFC = DEXP( F * CDAB * XC )
DELV = V2 * (1.0D/EF - 1.0D/EFC )
DELT = ((EF - 1.0D)/CDAB - (EF0 -1.0D)/CDAB0 )/ F /V2
TC = TCC + DELT
TS = T - TC

IF( IDUMP *EQ. 1 ) WRITE(6,1004) DELV,DELT, TC, TS
1004 FORMAT( 5X,1PE12.4 )
     1   X, IP111E12.4
     IF( DABS( TS ) ) LT, TOL ) GO TO 1
     IF( T LT, TC ) GO TO 4
     VT = VCO + DELV
     CALL TVX( VC + TS, VX, X, TS, X )
     X = XG + X

     IF( IDUMP *EQ. 1 ) WRITE(6,1003) T, XC, VX, X, VC0, TC0

     1. A2X, CDAB, CDAB, EF, EFO
     CONTINUE
     RETURN

     C \backslash T \less \backslash T
     4 \backslash CONTINUE
     TCC = T
     CALL TVX( V0, TGC, VX0, XG )
     D0.2 I = 1, I00
     A2X = A2( XG )
     CDAB = CDAB + DLOG( (CDAB + CAYD/VX0/VX0 )/ (CDAB + CAYD/VX/VX0 ) )
     I = 2,00 /F /XG
     CDAB = CDAB + (CNA + CAA) / XG * A2X
     EF = DEXP( F * CDAB * XG)
     EFO = DEXP( F * CDAB0 * XG)

1003 FORMAT( * T, XC, VX, X )
     IC0  A2X CDAB CDAB EF
     2 \backslash EFO*I1*IP111E12.4/8/
     DELT = ((EF -1.0D)/CDAB - (EF0 -1.0D)/CDAB0)/ F /V2
     V0 = (1.0D/EF - 1.0D/EFC )
     TC = TCC + DELT
     VX = VX0 + DELV

     IF( DUMP *EQ. 1 ) WRITE(6,1005) TG0, XG, VX0, DELT, DELV

     1. VX, TG, XG

1005 FORMAT( * T, TG0, XG, VX0, DELT, DELV, DEHIT )
     ILV VX VG V0*I1*IP111E12.4/8/
     IF( DABS( TG - T ) LT, TOL ) GO TO 3
     VG = VX - VX0 \backslash T \backslash T
     CALL VX(VX0, TG0, VG0, VG, F, B, CDAB, XG, CAYD, TOL )
     CONTINUE
     WRITE(6,1001)
```
D.55 Subroutine TRAJX

SUBROUTINE TRAJX( X , XC , VX , T )
IMPLICIT REAL * 8 ( A-H , O-Z )
C, COMMON / OES / I X, Y, Z
EQUIVALENCE ( VX , YE ( 131 ) ) , ( WX , YE ( 132 ) )
A ( PHIWIN , YE ( 133 ) ) , ( RH0 , YE ( 134 ) ) , ( CD1 , YE ( 135 ) )
B ( V1 , YE ( 136 ) ) , ( CD2 , YE ( 137 ) ) , ( V2 , YE ( 138 ) )
C ( XN , YE ( 139 ) ) , ( CMA , YE ( 140 ) ) , ( SM , YE ( 141 ) )
D ( CMQ , YE ( 142 ) ) , ( CMPA , YE ( 143 ) ) , ( TRIM , YE ( 144 ) )
E ( PNITRI , YE ( 145 ) ) , ( ANG0 , YE ( 146 ) ) , ( PHIANG , YE ( 147 ) )
F ( RATE , YE ( 148 ) ) , ( PHIRAT , YE ( 149 ) ) , ( GMA , YE ( 150 ) )
G ( PHIGAM , YE ( 151 ) ) , ( A , YE ( 152 ) ) , ( ELL , YE ( 153 ) )
H ( OD , YE ( 154 ) ) , ( AI , YE ( 155 ) ) , ( AI , YE ( 156 ) )
J ( IALY , YE ( 157 ) ) , ( P , YE ( 158 ) ) , ( ALPCON , YE ( 159 ) )
J ( CAA , YE ( 160 ) )
EQUIVALENCE ( IDUMP , YE ( 204 ) ) , ( ATJT , YE ( 401 ) )
A ( ALPH , YE ( 402 ) ) , ( ALPHOC , YE ( 403 ) ) , ( a , YE ( 404 ) )
B ( BETAW , YE ( 405 ) ) , ( BETAC , YE ( 406 ) ) , ( BETAD0 , YE ( 407 ) )
C ( PHI , YE ( 408 ) ) , ( ALPMAX , YE ( 409 ) ) , ( ALPMIN , YE ( 410 ) )
D ( CAYD , YE ( 411 ) ) , ( CD8 , YE ( 412 ) ) , ( CMA , YE ( 413 ) )
E ( CMHTD , YE ( 414 ) ) , ( CMHTA , YE ( 415 ) ) , ( CAY1 , YE ( 416 ) )
F ( CAY2 , YE ( 417 ) ) , ( CDAB , YE ( 418 ) ) , ( CDAB , YE ( 419 ) )
G ( DELFU , YE ( 420 ) ) , ( DELV , YE ( 421 ) ) , ( DELT , YE ( 422 ) )
H ( ODELX , YE ( 423 ) ) , ( ODXD , YE ( 424 ) ) , ( ODXC , YE ( 425 ) )
I ( IDOT , YE ( 426 ) ) , ( IDOT , YE ( 427 ) ) , ( IDLEW , YE ( 428 ) )
J ( ODELL , YE ( 429 ) ) , ( EYPE , YE ( 430 ) ) , ( EMP , YE ( 431 ) )
K ( EOLT , YE ( 432 ) ) , ( EDLT , YE ( 433 ) ) , ( EPC , YE ( 434 ) )
L ( PHI , YE ( 435 ) ) , ( P110 , YE ( 436 ) ) , ( P110 , YE ( 437 ) )
M ( PHI1 , YE ( 438 ) ) , ( PHI2 , YE ( 439 ) ) , ( PSI , YE ( 440 ) )
N ( PHI2 , YE ( 441 ) ) , ( R2 , YE ( 442 ) ) , ( R3 , YE ( 443 ) )
O ( R4 , YE ( 444 ) ) , ( RTRIM , YE ( 445 ) ) , ( TC , YE ( 446 ) )
P ( TCO , YE ( 447 ) ) , ( TS , YE ( 448 ) ) , ( TG , YE ( 449 ) )
Q ( TQ , YE ( 450 ) ) , ( THETA0 , YE ( 451 ) ) , ( THETA0 , YE ( 452 ) )
R ( TOL , YE ( 453 ) ) , ( VC , YE ( 454 ) ) , ( WC , YE ( 455 ) )
S ( VYA , YE ( 456 ) ) , ( VZA , YE ( 457 ) ) , ( VX , YE ( 458 ) )
EQUIVALENCE ( WZ , YE ( 459 ) ) , ( W0 , YE ( 460 ) )
A ( W1 , YE ( 461 ) ) , ( W2 , YE ( 462 ) ) , ( W3 , YE ( 463 ) )
B ( XLM , YE ( 464 ) ) , ( XLM1 , YE ( 465 ) ) , ( XLM2 , YE ( 466 ) )
C ( XNU1 , YE ( 467 ) ) , ( XNU2 , YE ( 468 ) ) , ( XG , YE ( 469 ) )
D ( XM , YE ( 470 ) ) , ( XJAY , YE ( 471 ) ) , ( XJAZ , YE ( 472 ) )
E ( XJX , YE ( 473 ) ) , ( YA , YE ( 474 ) ) , ( ZA , YE ( 475 ) )
F ( ALPH , YE ( 476 ) )

IF( X . LE . XC ) GO TO 1
X1 = XC
GO TO 2
1
X1 = X
2
CONTINUE
IF( X1 . EQ . 0 . ) X1 = 1 . D -10
CALL VX ( VC0 , TCO , W , WX , F , B , CD8 , X1 , CAYD , TOL)
IF( DB ( X1 , 1 . D -10 ) . EQ . VC0 . )}
D.55 Subroutine TRAJX (concluded)

\[
\begin{align*}
A2X &= A2(X1), \\
F &= F \cdot X1 \\
CDA0B &= CD8 + 0.0001/(CD8 + CAYD/VC0/VC0)/(CD8 + CAYD/VC0/VC0) \\
1 &/ 2.0000 / F / X1 \\
CDAB &= CDA0B + (CNA + CAA)/X1 \cdot A2X \\
EFCX &= DEXP( F1 \cdot CDAB ) \\
EFCX0 &= DEXP( F1 \cdot CDA0B ) \\
DELV &= V0 \cdot (1.0000/EFCX - 1.0000/EFCX0) \\
DELT = (EFCX - 1.0000)/CDAB - (EFCX0 - 1.0000)/CDA0B)/F/VC0 \\
VC &= VC0 + DELV \\
TC &= TC0 + DELT \\
IF( X \cdot LE. X0 ) \ GO TO 3 \\
xS &= X - XC \\
CALL &= VXT(VX, TS, VC, VX, F, B, CD8, XS, CAYD, TOL) \\
T &= TS + TC \\
GO TO 4 \\
3 \ CONTINUE \\
VX &= VC \\
T &= TC \\
4 \ IF( I0000 \ EQ. 1 ) WRITE(6,1000) X, T, X1, VCO, TCO \\
1, A2X, CDA0B, CDAB, EFCX, EFCX0 \\
1000 FORMAT( \ 
1, X, T, X1, VCO, A2X, CDA0B, CDAB, EFCX, EFCX0 \\
1C) \\
2 \ IF( I0001 \ EQ. 1 ) WRITE(6,1001) DELV, DELT, VC, TC, XS, VX \\
1001 FORMAT( \ 
1, VX, DELV, DELT, VC, TC, XS, VX \\
1S) \\
IS \ RETURN \\
END \\
\end{align*}
\]
D.56 Subroutine TRAJXF

SUBROUTINE TRAJXF( X, XC, VX, T )

COMMON/DEC0/D(100)

EQUIVALENCE ( VD, J(E( 11)), (VWINO, O(E( 21)), HITS3331
A(PHIWIN, J(E( 31)), (RH0, O(E( 41)), (CDI, O(E( 51)), HITS3332
B(VI, J(E( 61)), (CD2, O(E( 71)), (V2, O(E( 81)), HITS3333
C(XN, J(E( 91)), (CNA, O(E( 10)), (CSM, O(E( 11)), HITS3334
D(CM), J(E( 12)), (CMPA, O(E( 13)), (TRIM, O(E( 14)), HITS3335
E(PNITRI, J(E( 15)), (ANG0, O(E( 16)), (PHIANG, O(E( 17)), HITS3336
F(YATE, J(E( 18)), (PRHAT, O(E( 19)), (GAM0, O(E( 20)), HITS3337
G(PHIGAM, J(E( 21)), (CM2, O(E( 22)), (ELL, O(E( 23)), HITS3338
H(), J(E( 24)), (P, O(E( 25)), (AIX, O(E( 26)), HITS3339
I(KAY, J(E( 27)), (P, O(E( 28)), (ALPCON, O(E( 29)), HITS3340
J(CAA, J(E( 30)), HITS3341

EQUIVALENCE (IOWN, J(E( 204)), (ATJT, O(E( 421)), HITS3342
A(ALPTX, J(E( 42)), (ALPHA, O(E( 43)), (ATJT, O(E( 421)), HITS3343
B(GET), J(E( 465)), (BETA, O(E( 465)), (BETAT, O(E( 465)), HITS3344
C(RETA, J(E( 448)), (ALPMA, O(E( 439)), (ALPMA, O(E( 439)), HITS3345
D(CAY), J(E( 411)), (CD2, O(E( 412)), (CMA, O(E( 413)), HITS3346
E(CMITHD, J(E( 414)), (CMTHTA, O(E( 415)), (CAY1, O(E( 416)), HITS3347
F(CAY2, J(E( 417)), (CMATHA, O(E( 418)), (CMTHA, O(E( 419)), HITS3348
G(DELT), J(E( 423)), (DELV, O(E( 421)), (DELV, O(E( 423)), HITS3349
H(DLX), J(E( 4231)), (NYDX, O(E( 424)), (NYDX, O(E( 425)), HITS3350
I(XDYT, J(E( 426)), (YZDT, O(E( 427)), (DELW, O(E( 428)), HITS3351
J(DYTL), J(E( 430)), HITS3352
K(E1LT), J(E( 432)), (EDLT, O(E( 433)), (F, O(E( 434)), HITS3353
L(EM), J(E( 435)), (PSID, O(E( 436)), (PSID, O(E( 437)), HITS3354
M(Phi1, J(E( 438)), (PHI2, O(E( 439)), (PSI1, O(E( 440)), HITS3355
N(Re1, J(E( 441)), (R1, O(E( 442)), (R1, O(E( 443)), HITS3356
O(CRA), J(E( 445)), (CTRIM, O(E( 446)), (TC, O(E( 447)), HITS3357
P(CT), J(E( 447)), (TS, O(E( 448)), (TS, O(E( 449)), HITS3358
Q(TP), J(E( 450)), (THETA, O(E( 451)), (THETA, O(E( 452)), HITS3359
R(TOL), J(E( 453)), (VC, O(E( 454)), (VC0, O(E( 455)), HITS3360
S(YYA, J(E( 456)), (VZA, O(E( 457)), (W, O(E( 458)), HITS3361
T(EVIALENE, J(E( 458)), (VZ, O(E( 459)), (WZ, O(E( 460)), HITS3362
U(LE), J(E( 461)), (W2, O(E( 462)), (W2, O(E( 463)), HITS3363
V(LAM), J(E( 464)), (XLAM, O(E( 465)), (XLAM, O(E( 466)), HITS3364
W(LX), J(E( 467)), (XL2, O(E( 468)), (XL2, O(E( 469)), HITS3365
X(LXM), J(E( 469)), (XL2, O(E( 470)), (XL2, O(E( 471)), HITS3366
Y(LX), J(E( 462)), (WZ, O(E( 463)), (WZ, O(E( 464)), HITS3367
Z(LX), J(E( 463)), (WZ, O(E( 464)), (WZ, O(E( 465)), HITS3368
IF( X .LE. XC ) GO TO 1
XI = XC
GO TO 2
1 XI = X
2 CONTINUE

IF( X1 .EQ. 0. ) X1 = 1.0-10
CALL VX(T(VCO), TCO, VX, W, F, B, CD8, X1, CAYD, TOL )
IF( DMA5( X1, .EQ. 1.0-10) = VCO )
D.56 Subroutine TRAJXF (concluded)

```
A2X = A2F(X1)
FX1 = F * X1

CDA08 = CD8 + DLOG((CD8 + CAYD/VC0)/VC0)/(CD8 + CAYD/VO/VO)

1 / 2.0D0 / F / X1

CDA8 = CDA08 + (CMA + CAA)/X1 * A2X

EFCX = DEXP( FX1 * CDAB )
EFCX0 = DEXP( FX1 * CDA08 )

DELV = VO + (1.0D0 / EFCX - 1.0D0 / EFCX0 )
DELTT = ((EFCX - 1.0D0 / CDAB - (EFCX0 - 1.0D0/CDA08)/F /VO

VC = VCO + DELV

TC = TCO + DELTT

IF( X LE. XC ) GO TO 3

XS = X - XC

CALL VXV( VX, TS, VC, WX, F, B, CD8, XS, CAYD, TOL )

T = TS + TC

GO TO 4

3 CONTINUE

VX = VC
T = TC

4 IF( IDUMP *EQ. 1 ) WRITE( 6, 1000 ) X, T, X1, VC0, TCO

1 1000 FORMAT( * , A2X, CDAB, EFCX, EFCX0
1001 FORMAT( * , A2X, CDAB, EFCX, EFCX0

2 IF( IDUMP *EQ. 1 ) WRITE( 6, 1001 ) DELV, DELTT, VC, TC, XS, VX

1001 FORMAT( * , DELV, DELTT, VC, TC

1S RETURN
END
```
D.57 Subroutine TVX

SUBROUTINE TVX(VI, T, VX, X)
C IMPLICIT REAL * 8 (A-H, O-Z )
COMMON/DEC/DEA(400)
EQUIVALENCE (TDUMP, DE(204))
EQUIVALENCE (ALPHA, DE(430)), (ATJT, DE(431))
A(ALPTE, DE(432)), (ALPHD0, DE(433)), (B, DE(434))
B(BETTPW, DE(455)), (BETAW0, DE(456)), (BETADO, DE(457))
C(3ETA, DE(458)), (ALPMAX, DE(459)), (ALPMIN, DE(460))
D(CAYD, DF(111)), (CAYD, DE(112)), (CMA, DE(113))
E(CMTHTD, DE(414)), (CMHTDA, DE(415)), (CAY1, DE(416))
F(CAY2, DE(417)), (CDAB, DE(418)), (CAB, DE(419))
G(ELT, DE(420)), (EDLT, DE(421)), (DEL, DE(422))
H(D2 ELX, DE(423)), (D2DXC, DE(424)), (D2DXD, DE(425))
I(DYDT, DE(426)), (DYDTA, DE(427)), (DELW, DE(428))
J(EDLLAM, DE(429)), (EYE P, DE(430)), (EMP, DE(431))
K(ELT, DE(432)), (EDLT, DE(433)), (F, DE(434))
L(HPHI1, DE(435)), (HPHI2, DE(436)), (PHI0, DE(437))
M(N2, DE(441)), (R2, DE(442)), (R3, DE(443))
O(ORA, DE(444)), (RTRIM, DE(445)), (TC, DE(446))
P(PTC, DE(447)), (TS, DE(448)), (TG, DE(449))
Q(TP, DE(450)), (THETA), DE(451)), (THETD0, DE(452))
R(TOL, DE(453)), (VC, DE(454)), (VCF, DE(455))
S(VVY, DE(456)), (VZA, DE(457)), (WX, DE(458))
EQUIVALENCE (WZ, De(459)), (D0, DE(460))
A(A, DE(461)), (W2, DE(462)), (W2, DE(463))
R(XLAM, DE(464)), (XLAML, DE(465)), (XALM2, DE(466))
C(XNU, DE(467)), (XNU2, DE(468)), (XG, DE(469))
D(XMU, DE(470)), (XMJ, DE(471)), (XMJ, DE(472))
F(JAXA, DE(473)), (YJ, DE(474)), (ZA, DE(475))
F(ALP1, DE(476))

VX = VI - WX
CAYBC = CAYD / 3 /CD8
BCDK = DSQRT( 1.0D0 / CAYBC )
TFCKT = TAN(F * DSQRT(CDBA + CAYD) ) / T
VX = (VI - WX - TFCKT / BCDK) / ( 1.0D0 + BCDK * VX * TFCKT) + VX
HITS3445
X = WX * T - DLOGI ( (VX-WX)**2 + CAYBC ) / ( (VI-WX)**2 + CAYBC )
HITS3446
1 )
2 IF(IDUMP.EQ.0) GO TO 69
WRITE(6,100) X, VX, CAYBC, VX, BCDK, TFCKT
CONTINUE
69 FORMAT(//, X=' ', IP8E15.8)
RETURN
END

HITS3454
D.58 Subroutine T4NTV

SUBROUTINE T4NTV ( IAW, IAY, LT, KBR )
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CARDCM/ B(200), C(5000), IA(200,2), ID(10,2), INC,
1 IQPRNT, IQUT, IY, KARS1(3), KA, KB, KC, KD, KT1, K234, LABC(3)
SWF = 0.000
SYWF = 0.000
DO 10 I = 1, LT
K = IAW + I
SWF = SWF + C(K)
J = IAY + I
SYWF = SYWF + C(J) * C(K)
10 SYSWF = SYSWF + C(J)**2 * C(K)
C NOMINAL VALUE
EY = SYWF / SWF
B(KBR+1) = EY
C VARIANCE
B(KBR+3) = SYSWF / SWF - EY**2
C TOLERANCE
B(KBR+2) = 3.000 * DSORT ( B(KBR+3) )
RETURN
END

HITS3455  HITS3456  HITS3457  HITS3458  HITS3459
HITS3460  HITS3461  HITS3462  HITS3463  HITS3464
HITS3465  HITS3466  HITS3467  HITS3468  HITS3469
HITS3470  HITS3471  HITS3472  HITS3473  HITS3474
HITS3475  HITS3476
D.59 Subroutine VXT

SUBROUTINE VXT(VX, T, VI, WX, F, B, CD8, X, CAYD, TOL) 
IMPLICIT REAL * 8(A-H, O-Z) 
COMMON/DEC/DE(600) 
EQUIVALENCE (IDUMP, DE(204)) 
F2 = 2.00 * F 
VIWX = VI - WX 
VIWX2 = VIWX * VIWX 
3CK = B * CD8 / CAYD 
BCD8K0 = DSORT(BCK) 
T1 = DATAN( BCD8K0 * VIWX ) 
T0 = 0.00 
T = 0.00 
F2BC = F2 * DSQR(B) * CD8 
VX0 = 0.00 
FSCDK0 = F * DSQR(CD8 * CAYD) 
DO 1 I = 1, 100 
E = FEXP(F2BC * (WX * T - X)) 
VX = DSQR(VIWX2 + E + (E-1.00)/BCK) + WX 
T = T1 + DATAN(BCD8K0*(VX-WX)) 
1 CONTINUE 
IF(IOW .EQ. 0) GO TO 69 
69 CONTINUE 
100 FORMAT(* ,I3, "T = ", 1PE18.8, ". T-OLD=", 1PE18.8, ". VX =", 1PE18.8 ) 
IF(DABS((VX-VX0)/VX) + DABS((T-T0)/T), LT, TOL) 
1 RETURN 
VX0 = VX 
T0 = T 
1 CONTINUE 
WRITE(6, '1001) 
1001 FORMAT(*, VX AND T ROUTINE EXCEEDED 100 ITERATIONS..." //) 
RETURN 
END
D.60 Subroutine WIND

SUBROUTINE WIND( VX, T, X, WZ, WX, VO, VZ, Z )
  IMPLICIT REAL * 8 (A-H, O-Z)
  VX = VX - WX
  WZ = WZ * (1.0D0 - (VX - WX)/VO WX)
  Z = WZ / VO WX * (VO * T - X)
  RETURN
END
D.61 Subroutine XXC

SUBROUTINE XXC (XL0, XC, V0, DELL, CAY1, CAY2, TOL, AC)
IMPLICIT REAL * 8(A-H, O-Z)
COMMON/OES/OE(600)
EQUIVALENCE (IDUMP, OE(204))
IF(CAY1+CAY2 NE. 0.) GO TO 2
XC=1.0-3
RETURN
2 CONTINUE
XLV = XL0 / V0
XCO= DLG( AC/(CAY1 + CAY2 ) ) / XLV
DLV0= DELL / V0
DO 1 I = 1, 100
   E = DEXP( DLV0 * XCO )
   E1 = CAY1 * E
   E2 = CAY2 / E
   A = E1 + E2
   A2 = E1 - E2
   B = FEXP(- XCO * XLV )
   C = ( A - AC * B) / ( A * XLV + DLV0 * A2 )
   XC = XCO - C
   IF(IDUMP, NE.1) GO TO 69
   WRITE(6,1000) I, XC, XCO, C, E, E1, E2, A, A2, B
69 CONTINUE
1000 FORMAT( 'NO.*, I4.*, XC=*, IP1E18.8, XC-OLD=*, IP1E18.8')
   IF( CHANGE = 1, IP1E18*6 / IP8E15.6 )
   IF( DBS( C / XC ) .GT. TOL ) GO TO 5
   IF( XC .LT. 1.0-3 ) XC = 1.0-3
   RETURN
5 CONTINUE
   XCO = XC
1 CONTINUE
   WRITE(6,1001)
1001 FORMAT('XXC ROUTINE EXCEEDED 100 ITERATIONS WITHOUT CONVERGING/*')
   IF(XC .LT. 1.0-3) XC = 1.0-3
   RETURN
END
D.62 Function ZATAN2

```
FUNCTION ZATAN2(R1, R2)
IMPLICIT REAL * 8(A-H, O-Z)
IF( R1 .NE. 0.) GO TO 1
IF( R2 .NE. 0.) GO TO 1
ZATAN2=0.00
RETURN
ZATAN2= DATAN2(R1, R2)
RETURN
END
```

FUNCTION ZATAN2(R1, R2)
IMPLICIT REAL * 8(A-H, O-Z)
IF( R1 .NE. 0.) GO TO 1
IF( R2 .NE. 0.) GO TO 1
ZATAN2=0.00
RETURN
ZATAN2= DATAN2(R1, R2)
RETURN
END
D.63 Subroutine ZZ

FUNCTION ZZ(CNA, CD8, CAYD, V0, EMP, ICNCL)
IMPLICIT REAL * 8 (A-H, O-Z)
IF (ICNCL.EQ.1) GO TO 1
ZZ = CNA * V0/EMP
RETURN
1
CLA = CNA - (CD8 + CAYD/V0/V0)
ZZ = CLA * V0/EMP
RETURN
END
DISTRIBUTION

A. Department of Defense

Defense Advanced Research Projects Agency
ATTN: Mr. Charles R. Lehner, Jr.
1400 Wilson Blvd.
Arlington, VA 22209
1

Defense Documentation Center
ATTN: TIPDR
Cameron Station
Alexandria, VA 22314
2

B. Department of the Army

HQ DA (DAMA-WSZ, Dr. V. Garber)
Washington, DC 20310
1

HQ DA (DAMA-WSW, LTC Gibson)
Washington, DC 20310
1

Commander
US Army Material Command
ATTN: AMCRD-T, Dr. H. M. El-Bisi
AMCRD-T, Mr. D. Stefanye
AMCRD-FW
1
5001 Eisenhower Avenue
Alexandria, VA 22333

Commander
US Army Missile Command
ATTN: AMSMI-RGT, Mr. Knoblach
AMSMI-RDF, Mr. Morrow
Huntsville, AL 35801
1

Commander
US Army Armament Command
ATTN: AMSAR-RD, Mr. J. Brinkman
AMSA-RDT, Mr. L. Johnson
Rock Island, IL 61201
1

Commander
US Army Training Doctrine Command
ATTN: ATCD-CM-A
Fort Monroe, VA 23651
1
DISTRIBUTION (Cont'd)

Commander
US Army Aviation Systems Command
ATTN: AMSAV-EVW
P. O. Box 209
St. Louis, MO 63166

Director
US Army Ballistic Research Laboratories
ATTN: AMXBR-CA, Dr. Eichelberger
Mr. C. M. Frank
Mr. A. S. Elder
Aberdeen Proving Ground, MD 21005

Director AMXSY-ADE
AMSAA
ATTN: Mr. John L. Meredith
Aberdeen Proving Ground, MD 21005

Commander
US Army Air Defense Agency
Fort Bliss, TX 79916

Commandant
US Army Armor School
Director, Combat Development
ATTN: ATSD-CD-MM, Mr. M. Falkovich
Fort Knox, KY 40121

Commandant
US Army Air Defense School
ATTN: ATSAD-DL-P
Fort Bliss, TX 79916

President
US Army Air Defense Board
ATTN: STEBD-TD-S
Fort Bliss, TX 79916

Commander
Edgewood Arsenal
ATTN: SAREA-CL-PA, Mr. A. Flatau
Edgewood, MD 21010

Copies
1
3
1
1
1
1
1
1
DISTRIBUTION (Cont'd)

Commander
Frankford Arsenal
ATTN: SARFA-J700, Mr. C. M. Dickey 3
      SARFA-MDS-BP, Mr. P. Ayyoub 1
      Mr. A. Korr, Chairman-Invention Evaluation Committee
      Building 64-4

      Dr. Sid Ross 1
      Philadelphia, PA 19137

Commander
Picatinny Arsenal
ATTN: SARFA-PR-E, Mr. Hirskowitz 1
      SARPA-AD-D-A-2, Mr. Michelson 1
      SARPA-AD, Mr. Painter 1

      Dover, NJ 07801

Commander
Rock Island Arsenal
ATTN: SARRI-LA, Mr. Wrenn 1
      SARRI-LE 1
      SARRI-LA, Mr. Wells 1
      SARRI-LW, Mr. Stickling 1
      SARRI-Lk-Y 15

      Rock Island, IL 61201

Commander
Watervliet Arsenal
ATTN: Mr. Austin 1
      Mr. Rummel 1

      Watervliet, NY 12189

1. Department of the Air Force

      AFAL/DLDG (Maj. Bilsbury)
      Eglin AFB, FL 32542 1

      AFATC/DLM (Mr. Walter Detrick)
      Eglin AFB, FL 32542 1
### DISTRIBUTION (Concl'd)

<table>
<thead>
<tr>
<th>Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

**AEDC/ARO (Mr. G. Norfleet)**
Arnold AFS, TN 37389

**D. Department of the Navy**

<table>
<thead>
<tr>
<th>Commander</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naval Surface Weapons Laboratory</td>
</tr>
<tr>
<td>ATTN: Dr. V. Dawson</td>
</tr>
<tr>
<td>Mr. N. Sheetz</td>
</tr>
<tr>
<td>White Oak, MD 25989</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Commander</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naval Research Laboratory</td>
</tr>
<tr>
<td>ATTN: Mr. W. Atkins</td>
</tr>
<tr>
<td>Washington, DC 20315</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Naval Weapons Center</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code 5115</td>
</tr>
<tr>
<td>ATTN: Mr. Ward</td>
</tr>
<tr>
<td>China Lake, CA 93555</td>
</tr>
</tbody>
</table>

**E. General**

<table>
<thead>
<tr>
<th>Avco Systems Division</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTN: Dr. W. G. Reinecke</td>
</tr>
<tr>
<td>201 Lowell Street</td>
</tr>
<tr>
<td>Wilmington, MA 01887</td>
</tr>
</tbody>
</table>