INFORMATION TO USERS

This was produced from a copy of a document sent to us for microfilming. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help you understand markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure you of complete continuity.

2. When an image on the film is obliterated with a round black mark it is an indication that the film inspector noticed either blurred copy because of movement during exposure, or duplicate copy. Unless we meant to delete copyrighted materials that should not have been filmed, you will find a good image of the page in the adjacent frame. If copyrighted materials were deleted you will find a target note listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed the photographer has followed a definite method in "sectioning" the material. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For any illustrations that cannot be reproduced satisfactorily by xerography, photographic prints can be purchased at additional cost and tipped into your xerographic copy. Requests can be made to our Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases we have filmed the best available copy.
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark \( \checkmark \).

1. Glossy photographs or pages ___
2. Colored illustrations, paper or print ___
3. Photographs with dark background ___
4. Illustrations are poor copy ___
5. Pages with black marks, not original copy ___
6. Print shows through as there is text on both sides of page ___
7. Indistinct, broken or small print on several pages \( \checkmark \)
8. Print exceeds margin requirements ___
9. Tightly bound copy with print lost in spine ___
10. Computer printout pages with indistinct print \( \checkmark \)
11. Page(s) _________ lacking when material received, and not available from school or author.
12. Page(s) _________ seem to be missing in numbering only as text follows.
13. Two pages numbered ___________. Text follows.
14. Curling and wrinkled pages ___
15. Other _____________________________________________________________________________

University
Microfilms
International
EVALUATION OF PROBABILISTIC SIMULATION METHODS AND
DEVELOPMENT OF OPTIMIZATION TECHNIQUES
FOR CAPACITY EXPANSION PLANNING OF
ELECTRIC POWER GENERATION SYSTEMS

DISSERTATION

Presented in Partial Fulfillment of the Requirements
for the Degree Doctor of Philosophy
in the Graduate School of The Ohio State University

by


* * * * *

The Ohio State University
1981

Reading Committee

Dr. Shoichiro Nakamura
Dr. Don W. Miller
Dr. Douglas N. Jones
Dr. Kevin A. Kelly

Approved by:

Advisor
Nuclear Engineering Program
Mechanical Engineering Department
DEDICATION

This dissertation is dedicated to my family
-- The one who raised me, and the one I am helping to raise.
ACKNOWLEDGEMENTS

I would like to express my appreciation to my advisor, Dr. Schoichiro Nakamura, for his sustained help, support, encouragement and guidance during my graduate studies at The Ohio State University.

I would also like to express my appreciation to Dr. Don W. Miller, for his encouragement and help throughout my graduate work and for his useful suggestions as member of the reading committee.

I wish to thank Dr. Douglas N. Jones and Dr. Kevin A. Kelly for their support and many useful suggestions during this research.

I am grateful to Dr. George E. Juras for his help in many occasions.

The help of Ms. Connie Mann in typing and of Messrs. James Munstaine, Jeffrey Shih, and Mohammad Harunuzzaman in computer programming is much appreciated.

Much of the research presented here was supported by the United States Department of Energy through a contract with the National Regulatory Research Institute. The support of the National Regulatory Research Institute is fully acknowledged.

Finally, I want to thank my wife, Eriphyle for her continuous encouragement and support throughout my graduate studies, and the many long nights she endured with me preparing and typing the manuscript.
VITA

October 27, 1948. . . . . . . . . . . . . Born - Thessaloniki, Greece

1971. . . . . . . . . . . . . . . . . . . . . . . . B.A., Physics, Aristotelion University of Thessaloniki, Greece

1974. . . . . . . . . . . . . . . . . . . . . . . . M.A., Department of Physics, Kent State University, Kent, Ohio

1976. . . . . . . . . . . . . . . . . . . . . . . . M.Sc., Department of Nuclear Engineering, The Ohio State University, Columbus, Ohio

1974-1978 . . . . . . . . . . . . . . . . . . Research Associate, Department of Nuclear Engineering, The Ohio State University, Columbus, Ohio

1978-1980 . . . . . . . . . . . . . . . . . . Research Associate, The National Regulatory Research Institute, The Ohio State University, Columbus, Ohio

1980-1981 . . . . . . . . . . . . . . . . . . Research Scientist, Battelle Columbus Laboratories, Columbus, Ohio

Areas of Specialty

Nuclear Reactor Theory
Numerical Analysis
Electric Utility Operations and Planning

Publications and Reports

Publications and Reports (Continued)


# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDICATION</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iii</td>
</tr>
<tr>
<td>VITA</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Objectives of the Research</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Need for the Research</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Research Performed</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Research Contributions</td>
<td>7</td>
</tr>
<tr>
<td>1.5 Organization of the Dissertation</td>
<td>7</td>
</tr>
<tr>
<td>2. CERES OVERVIEW</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>9</td>
</tr>
<tr>
<td>2.2 CERES Structure and Objectives</td>
<td>10</td>
</tr>
<tr>
<td>2.3 Input Requirements</td>
<td>13</td>
</tr>
<tr>
<td>2.4 Expansion Plan Optimization</td>
<td>14</td>
</tr>
<tr>
<td>2.4.1 PREP Submodule</td>
<td>14</td>
</tr>
<tr>
<td>2.4.2 Objective Function Optimization and Constraints</td>
<td>18</td>
</tr>
<tr>
<td>2.5 Summary</td>
<td>23</td>
</tr>
</tbody>
</table>
### 3. EVALUATION OF ELECTRIC GENERATING SYSTEM SIMULATION METHODS TO ESTIMATE RELIABILITY AND ENERGY PRODUCTION

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Introduction</td>
<td>24</td>
</tr>
<tr>
<td>3.2 Probabilistic Simulation of Electric Generating Systems</td>
<td>26</td>
</tr>
<tr>
<td>3.2.1 Load Demand and Outage Capacity as Random Variables</td>
<td>27</td>
</tr>
<tr>
<td>3.2.2 Equivalent Load Demand</td>
<td>31</td>
</tr>
<tr>
<td>3.2.3 Energy and Reliability Calculation</td>
<td>40</td>
</tr>
<tr>
<td>3.3 Approximations in Probabilistic Simulation</td>
<td>44</td>
</tr>
<tr>
<td>3.3.1 Piecewise Linear Approximation</td>
<td>44</td>
</tr>
<tr>
<td>3.3.2 Derate Method</td>
<td>46</td>
</tr>
<tr>
<td>3.3.3 Cumulant Approximation</td>
<td>47</td>
</tr>
<tr>
<td>3.4 Sensitivity Analysis</td>
<td>51</td>
</tr>
<tr>
<td>3.4.1 Methodology</td>
<td>51</td>
</tr>
<tr>
<td>3.4.2 Load Characteristics and Generating Systems Considered</td>
<td>55</td>
</tr>
<tr>
<td>3.4.3 Sensitivity Analysis Results</td>
<td>59</td>
</tr>
<tr>
<td>3.4.4 Analysis of the Results</td>
<td>77</td>
</tr>
<tr>
<td>3.5 Hybrid Use of Cumulant and Piecewise Linear Approximations</td>
<td>81</td>
</tr>
<tr>
<td>3.6 Load Forecast Uncertainty</td>
<td>83</td>
</tr>
<tr>
<td>3.6.1 Introduction</td>
<td>83</td>
</tr>
<tr>
<td>3.6.2 Uncertainty Algorithm</td>
<td>85</td>
</tr>
<tr>
<td>3.6.3 Effects of Uncertainty on Energy and LOLP</td>
<td>90</td>
</tr>
<tr>
<td>3.7 Conclusions</td>
<td>98</td>
</tr>
</tbody>
</table>

### 4. ITERATIVE DYNAMIC PROGRAMMING

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>101</td>
</tr>
<tr>
<td>4.2 Application of Dynamic Programming to Capacity Expansion Planning</td>
<td>104</td>
</tr>
<tr>
<td>4.2.1 Problem Definition</td>
<td>104</td>
</tr>
<tr>
<td>4.2.2 Fathoming Technique</td>
<td>106</td>
</tr>
<tr>
<td>4.3 Iterative Dynamic Programming with Tunnel Iterations</td>
<td>109</td>
</tr>
<tr>
<td>4.4 Stage Iterative Dynamic Programming</td>
<td>112</td>
</tr>
<tr>
<td>4.4.1 Algorithm Description</td>
<td>112</td>
</tr>
<tr>
<td>4.4.2 Test Problem Definition</td>
<td>115</td>
</tr>
<tr>
<td>4.4.3 Results and Analysis</td>
<td>117</td>
</tr>
</tbody>
</table>
4.5 Suggested Application of Stage Iterative Dynamic Programming to Capacity Expansion Planning .................................................. 136
5. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS ............ 140
  5.1 Summary and Conclusions .................................................. 140
  5.2 Recommendations for Further Research ......................... 143

APPENDICES
A. OPTIM MODULE SOURCE PROGRAM LISTING ....................... 144
B. CUMULANT FORMULAS .................................................... 204
C. PROSES SOURCE LISTING ............................................... 207
D. SENSITIVITY ANALYSES FIGURES .................................... 242
E. STAGE ITERATIVE DYNAMIC PROGRAMMING TEST PROGRAM SOURCE LISTING ............................................. 293

REFERENCES ......................................................................................... 300
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Load and Generating System Characteristics ..........57</td>
</tr>
<tr>
<td>3.2</td>
<td>Generating Unit Characteristics for Cases 1A through 5B ........................................58</td>
</tr>
<tr>
<td>3.3</td>
<td>Generating Unit Characteristics for Cases 6A, 6B, 7A, and 7B ...........................................60</td>
</tr>
<tr>
<td>3.4</td>
<td>Generating Unit Characteristics for Cases 8A, 8B, 9A, and 9B ...........................................61</td>
</tr>
<tr>
<td>3.5</td>
<td>Generating Unit Characteristics for Cases 10A, 10B, 11A, and 11B .....................................62</td>
</tr>
<tr>
<td>3.6</td>
<td>LOLP as Calculated by the Piecewise Linear and Cumulant Methods ..................................71</td>
</tr>
<tr>
<td>3.7</td>
<td>Total Energy Generation by the Piecewise Linear, Cumulant, and Derate Method ..................72</td>
</tr>
<tr>
<td>3.8</td>
<td>Computing Time for LOLP and Energy Calculations (AMDAHL 470V6, Cpu Time in System Seconds) 73</td>
</tr>
<tr>
<td>3.9</td>
<td>Qualitative Comparison of Hybrid, Piecewise, Cumulant, and Derate Methods for Expansion Planning .........84</td>
</tr>
<tr>
<td>3.10</td>
<td>LOLP With and Without Forecast Uncertainty. (80 and 60 percent probability for ±10% error in total energy demand) ..........96</td>
</tr>
<tr>
<td>3.11</td>
<td>Total Energy Generation With and Without Forecast Uncertainty. (80 and 60 percent probability for ±10% forecast error in total energy demand) ..........97</td>
</tr>
<tr>
<td>4.1</td>
<td>Various Objective Functions for the Test Problem ..........116</td>
</tr>
<tr>
<td>4.2</td>
<td>Objective Function Value and Number of Iterations Required for Convergence of SIDP for the Test Problem ....118</td>
</tr>
<tr>
<td>4.3</td>
<td>Number of Iterations and Computing Time for the Test SIDP Problem. .................................133</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>16</td>
</tr>
<tr>
<td>2.3</td>
<td>19</td>
</tr>
<tr>
<td>.1</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>30</td>
</tr>
<tr>
<td>3.3</td>
<td>32</td>
</tr>
<tr>
<td>3.4</td>
<td>33</td>
</tr>
<tr>
<td>3.5</td>
<td>36</td>
</tr>
<tr>
<td>3.6</td>
<td>42</td>
</tr>
<tr>
<td>3.7</td>
<td>43</td>
</tr>
<tr>
<td>3.8</td>
<td>43</td>
</tr>
<tr>
<td>3.9</td>
<td>63</td>
</tr>
<tr>
<td>3.10</td>
<td>64</td>
</tr>
</tbody>
</table>
LIST OF FIGURES (continued)

4.1 Initial Guess, Optimal, and Suboptimal Trajectories for Case 1. The Stage Step Size is Denoted by m ................................................................. 119

4.2 Initial Guess, Optimal, and Suboptimal Trajectories for Case 2. The Stage Step Size is Denoted by m ......................................................... 120

4.3 Initial Guess, Optimal, and Suboptimal Trajectories for Case 3. The Stage Step Size is Denoted by m ......................................................... 121

4.4 Initial Guess and Optimal Trajectories for Case 4. The Stage Step Size is Denoted by m ................................................................. 122

4.5 Initial Guess and Optimal Trajectories for Case 5. The Stage Step Size is Denoted by m ................................................................. 123

4.6 Initial Guess, Optimal, and Suboptimal Trajectories for Case 6. The Stage Step Size is Denoted by m ................................................................. 124

4.7 Initial Guess and Optimal Trajectories for Case 7. The Stage Step Size is Denoted by m ................................................................. 125

4.8 Initial Guess, Optimal, and Suboptimal Trajectories for Case 8. The Stage Step Size is Denoted by m ................................................................. 126

4.9 Initial Guess, Optimal, and Suboptimal Trajectories for Case 9. The Stage Step Size is Denoted by m ................................................................. 127

4.10 Initial Guess and Optimal Trajectories for Case 10. The Stage Step Size is Denoted by m ................................................................. 128

4.11 Initial Guess and Optimal Trajectories for Case 11. The Stage Step Size is Denoted by m ................................................................. 129

4.12 Initial Guess, Optimal, and Suboptimal Trajectories for Case 12. The Stage Step Size is Denoted by m ................................................................. 130

4.13 Initial Guesses for Case 1 ................................................................. 132

D.1 Percentage Differences of Original Load Probability Curves for Cases 2A and 2B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ......................................................... 243
LIST OF FIGURES (continued)

D.2 Percentage Differences of Final Equivalent Load Probability Curves in Case 2A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ............................................. 244

D.3 Percentage Differences of Final Equivalent Load Probability Curves in Case 2B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ............................................. 245

D.4 Percentage Differences of Original Load Probability Curves for Cases 3A and 3B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ............................................. 246

D.5 Percentage Differences of Final Equivalent Load Probability Curves in Case 3A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ............................................. 247

D.6 Percentage Differences of Final Equivalent Load Probability Curves in Case 3B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ............................................. 248

D.7 Percentage Differences of Original Load Probability Curves for Cases 4A and 4B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ............................................. 249

D.8 Percentage Differences of Final Equivalent Load Probability Curves in Case 4A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ............................................. 250

D.9 Percentage Differences of Final Equivalent Load Probability Curves in Case 4B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ............................................. 251

D.10 Percentage Differences of Original Load Probability Curves for Cases 5A and 5B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ............................................. 252
D.11 Percentage Differences of Final Equivalent 
Load Probability Curves in Case 5A, Between 
Each of the 4-, 5-, and 8-Cumulant Series 
Approximations and the Reference Piecewise 
Linear Method ................................................................. 253

D.12 Percentage Differences of Final Equivalent 
Load Probability Curves in Case 5B, Between 
Each of the 4-, 5-, and 8-Cumulant Series 
Approximations and the Reference Piecewise 
Linear Method ................................................................. 254

D.13 Percentage Differences of Original Load 
Probability Curves for Cases 6A and 6B, 
Between Each of the 4-, 5-, and 8-Cumulant 
Series Approximations and the Reference 
Piecewise Linear Method .................................................. 255

D.14 Percentage Differences of Final Equivalent 
Load Probability Curves in Case 6A, Between 
Each of the 4-, 5-, and 8-Cumulant Series 
Approximations and the Reference Piecewise 
Linear Method ................................................................. 256

D.15 Percentage Differences of Final Equivalent 
Load Probability Curves in Case 6B, Between 
Each of the 4-, 5-, and 8-Cumulant Series 
Approximations and the Reference Piecewise 
Linear Method ................................................................. 257

D.16 Percentage Differences of Original Load 
Probability Curves for Cases 7A and 7B, 
Between Each of the 4-, 5-, and 8-Cumulant 
Series Approximations and the Reference 
Piecewise Linear Method .................................................. 258

D.17 Percentage Differences of Final Equivalent 
Load Probability Curves in Case 7A, Between 
Each of the 4-, 5-, and 8-Cumulant Series 
Approximations and the Reference Piecewise 
Linear Method ................................................................. 259

D.18 Percentage Differences of Final Equivalent 
Load Probability Curves in Case 7B, Between 
Each of the 4-, 5-, and 8-Cumulant Series 
Approximations and the Reference Piecewise 
Linear Method ................................................................. 260

D.19 Percentage Differences of Original Load 
Probability Curves for Cases 8A and 8B, 
Between Each of the 4-, 5-, and 8-Cumulant 
Series Approximations and the Reference 
Piecewise Linear Method .................................................. 261

xlv
| D.20 | Percentage Differences of Final Equivalent Load Probability Curves in Case 8A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method | 262 |
| D.21 | Percentage Differences of Final Equivalent Load Probability Curves in Case 8B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method | 263 |
| D.22 | Percentage Differences of Original Load Probability Curves for Cases 9A and 9B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method | 264 |
| D.23 | Percentage Differences of Final Equivalent Load Probability Curves in Case 9A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method | 265 |
| D.24 | Percentage Differences of Final Equivalent Load Probability Curves in Case 9B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method | 266 |
| D.25 | Percentage Differences of Original Load Probability Curves for Cases 10A and 10B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method | 267 |
| D.26 | Percentage Differences of Final Equivalent Load Probability Curves in Case 10A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method | 268 |
| D.27 | Percentage Differences of Final Equivalent Load Probability Curves in Case 10B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method | 269 |
| D.28 | Percentage Differences of Original Load Probability Curves for Cases 11A and 11B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method | 270 |
D.29  Percentage Differences of Final Equivalent Load Probability Curves in Case 11A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ........................................ 271

D.30  Percentage Differences of Final Equivalent Load Probability Curves in Case 11B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method ........................................ 272

D.31  Percentage Differences in Plant Energy Production for Cases 2A and 2B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................ 273

D.32  Percentage Differences in Plant Energy Production for Cases 3A and 3B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................ 274

D.33  Percentage Differences in Plant Energy Production for Cases 4A and 4B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................ 275

D.34  Percentage Differences in Plant Energy Production for Cases 5A and 5B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................ 276

D.35  Percentage Differences in Plant Energy Production for Cases 6A and 6B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................ 277

D.36  Percentage Differences in Plant Energy Production for Cases 7A and 7B, Between the
Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................278

D.37 Percentage Differences in Plant Energy Production for Cases 8A and 8B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................279

D.38 Percentage Differences in Plant Energy Production for Cases 9A and 9B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................280

D.39 Percentage Differences in Plant Energy Production for Cases 10A and 10B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................281

D.40 Percentage Differences in Plant Energy Production for Cases 11A and 11B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted. ........................................282

D.41 Percentage Difference in Plant Energy Production for Case 2B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty .................................................................283

D.42 Percentage Difference in Plant Energy Production for Case 3B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty .................................................................284

D.43 Percentage Difference in Plant Energy Production for Case 4B, Between the Reference
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.44</td>
<td>Percentage Difference in Plant Energy Production for Case 5B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty</td>
</tr>
<tr>
<td>D.45</td>
<td>Percentage Difference in Plant Energy Production for Case 6B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty</td>
</tr>
<tr>
<td>D.46</td>
<td>Percentage Difference in Plant Energy Production for Case 7B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty</td>
</tr>
<tr>
<td>D.47</td>
<td>Percentage Difference in Plant Energy Production for Case 8B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty</td>
</tr>
<tr>
<td>D.48</td>
<td>Percentage Difference in Plant Energy Production for Case 9B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty</td>
</tr>
<tr>
<td>D.49</td>
<td>Percentage Difference in Plant Energy Production for Case 10B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty</td>
</tr>
<tr>
<td>D.50</td>
<td>Percentage Difference in Plant Energy Production for Case 11B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty</td>
</tr>
</tbody>
</table>
1.1 Objectives of the Research

The objectives of this dissertation are (a) to evaluate the use of the cumulant approximation in probabilistic simulation of electric generating systems for the purpose of capacity expansion planning, and (b) to develop fast and efficient algorithms for capacity expansion planning programs.

1.2 Need for the Research

Technological advances in the sixties have made possible the construction of large nuclear plants for electricity generation. The long construction times and the large capital investments required to build these plants demand careful assessment of the need for nuclear power. In order to determine the need for nuclear power, if any, all the available means of electricity generation should be evaluated, for each specific electric power system (electric utility), and over the long planning horizon necessary for the long construction times and operational lifetimes of the generating units. The objective of electric capacity expansion planning is to determine when and what generating units should be added to an existing generating system so that the "objective function", i.e., the total cost to construct the new units...
and operate the generating system, is minimized under prescribed reliability and operating constraints. The long range capacity planning is a multidimensional, non-linear optimization problem that becomes even more complicated by the uncertainty and random nature of future electricity demand and the limited reliability of the electric power plants.

Many electric utilities, universities, research organizations, and vendors of electric power plants have developed (or updated) several capacity expansion models. Some of the better known capacity expansion planning models have been reviewed by S. Lee in the excellent comparative analysis of Reference [1]. The MIT Energy Laboratory is now developing a large scale Electric Generation Expansion Analysis System (EGEAS)* and has evaluated some of the most widely used capacity planning models [2]. G. W. Evans has reviewed several more models in Reference [6], in order to analyze the optimization techniques used. Extensive bibliographies on the models mentioned above are provided in the references cited.

All of the above models share two major disadvantages: (a) their use is cumbersome and requires highly trained personnel, and (b) the production costing and optimization methods used are either oversimplified and inaccurate or require excessive computing times.

* EGEAS is being developed by the MIT Energy Laboratory for the Electric Power Research Institute and is expected to be finished by the end of 1981 [3]. EGEAS is expected to include, among others, the Generation Expansion Model (GEM) developed by MIT [4], and the Stone & Webster's Optimal Generation (OPTGEN) model [5].
There are two explanations for this. First, on-line program execution has only recently been made possible with the increase of computer memory and speed. Second, recent breakthroughs in production costing and optimization methods, that reduced the memory and computing requirements of capacity expansion planning models, have not yet been incorporated in most of the existing models.

Three of the most significant recent developments in capacity expansion planning are (a) the cumulant approximation of probabilistic simulation of the operations of electric generating systems [7], (b) the fathoming technique [8] used in optimization by the dynamic programming method, and (c) the Generalized Bender's Decomposition principle for capacity planning optimization [2, 9]. Currently, (a) and (c) of the above methods are being incorporated in EGEAS by the MIT Energy Laboratory [2, 10], and (a) and (b) are implemented by the Tennessee Valley Authority in TARANTULA [11, 12]. However, neither EGEAS nor TARANTULA are expected to be simpler and easier to use than

* The term on-line in this dissertation refers to the immediate execution of a program from an interactive, time-sharing terminal. It is different from the "off-line" (batch) execution by the fact that in the latter case the computer executes the program when enough core memory becomes available. The results of the on-line execution can be printed immediately on the time-sharing terminal or stored on disk. The off-line execution results are stored on disk, or printed on a high speed printer.
any of the other already mentioned capacity planning models. Rather, both EGEAS and TARANTULA are utilizing the recently developed methodologies, in order to increase the detail with which the operations of the electric generating systems are simulated.

Until now the WASP group of programs (WASP-I, WASP-II, WASP-II(OSU), WASP-III) [13-18] were the only sophisticated capacity expansion planning programs, available in the public domain, that could be obtained from the original developers with relative ease. WASP has been used extensively both in the United States and by the international community [19-22]. However, the highly trained personnel required to use WASP are often not available at most of the state regulatory agencies and small electric utilities. Therefore, many regulatory agencies and electric utilities rely on the services of consulting firms [6,23-28] for their needs in capacity expansion planning. This practice is not only expensive, but the expansion planning models used by most consulting firms are proprietary and difficult to verify.

It should be evident from the above discussion that a user oriented, accurate and computationally efficient capacity expansion model is needed. This need gave the impetus for the development of the Capacity Expansion and Reliability Evaluation System (CERES) [29]. However, during the development of CERES, it became apparent that more research was needed for the evaluation of the cumulant and fathoming methods. This evaluation, which is part of the research presented in this dissertation, provided the motivation for further research which led to the development of four new algorithms for use in capacity expansion planning.
1.3 Research Performed

The cumulant* method approximates the plant and load data of electric utility systems in order to facilitate the probabilistic simulation of the systems' operations. The literature review of the cumulant method [30-37] led to the conclusion that more research was needed for the evaluation of the method's accuracy and computational efficiency. A comprehensive sensitivity analysis showed that the cumulant method alone cannot be used in a satisfactory manner for capacity expansion planning. Therefore, a new hybrid method for probabilistic simulation of electric generating systems was developed, in which cumulants are used for the calculation of plant energy production and piecewise linear polynomials for the evaluation of the generating systems reliability. Furthermore, another new method was developed that provides the possibility of efficiently incorporating energy demand forecast uncertainty into capacity expansion planning.

In the optimization of capacity expansion planning, the generalized Bender's decomposition method and the fathoming technique of dynamic programming were reviewed. The Bender's decomposition method [2] is still under development [3] and therefore no decision could be made on its accuracy and computational efficiency. The fathoming technique [8, 38, 39] has already been used in capacity expansion planning [6] and in other optimization problems [40]. However, fathoming alone is not sufficient for the reduction of the excessive computer memory and time required for capacity expansion planning, contrary to what is claimed by

* Details in Section 3.3.2
Evans* in Reference [6]. Therefore, two new iterative optimization methods were developed.

In the first iterative optimization method the "manual" tunnel iterations (see Section 4.3), used in WASP, were automated and fathoming was introduced. The second method, called "Stage Iterative Dynamic Programming," is based on successive application of traditional dynamic programming on subperiods of the optimization period (study horizon) (details in Section 4.4).

CERES is intended for use at a time-sharing terminal through remote access of a central computer. Moreover, the program is made interactive through "program-user" dialogues that are written in "plain English" and are embedded in various parts of the code. Thus the execution of the program is extremely simplified. The input data are entered and/or modified through the interactive dialogue. Furthermore, sensitivity analyses are very easy and inexpensive, since the user is given ample chance to modify key parameters during the program execution. CERES is faster and more accurate than WASP, since the hybrid approximation of probabilistic simulation and the iterative dynamic programming with the automated tunnel iterations are incorporated into the CERES program.

* Evans applied fathoming with traditional dynamic programming in capacity planning of unrealistically small electric generating systems and with a simplistic production costing algorithm.
1.4 **Research Contributions**

The most significant contributions of this research are:

(a) the comprehensive evaluation of the cumulant approximation of the probabilistic simulation of the operation of electric generating systems

(b) the development of the hybrid approximation in probabilistic simulation, which uses cumulants for energy and piecewise linear polynomials for Loss-Of-Load Probability (LOLP)* calculations

(c) the development and evaluation of a new algorithm for incorporation of energy forecast uncertainty in capacity expansion planning

(d) the automated use of tunnel iterations and fathoming in dynamic programming optimization in capacity expansion planning

(e) the formulation of a new stage iterative dynamic programming algorithm and suggested methods for its application to capacity expansion planning problems

(f) the incorporation of some of the above techniques in CERES

1.5 **Organization of the Dissertation**

Chapter 2 of this dissertation describes the CERES program and outlines the CERES objectives, the program structure and major

*LOLP is a measure of the generating system reliability. For any time period, LOLP is defined as the ratio of the total time the generating system cannot meet the load demand to the length of the time period.*
algorithms used. Chapter 3 includes the evaluation of the cumulant method through a comprehensive sensitivity analysis, the description of the hybrid probabilistic simulation method, and the forecast uncertainty algorithm. Chapter 4 presents the tunnel iterations and fathoming techniques and describes the stage iterative dynamic programming algorithm and its numerical application to a test optimization problem. Finally, Chapter 5 contains a summary of the research and recommendations for further work.
CHAPTER 2
CERES OVERVIEW

2.1 Introduction

The Capacity Expansion and Reliability Evaluation System (CERES) has provided the motivation and the framework for the research described in this dissertation. CERES is briefly overviewed here so that algorithms described in chapters 3 and 4 can be more easily understood and appreciated. Writing the CERES program has not been the focus of this research. Rather, the purpose has been the evaluation of methodologies and the development of new algorithms for use in capacity expansion planning. Therefore, only the parts of the CERES program that are relevant to the algorithms described in the subsequent chapters are outlined here in some detail. Detailed information on CERES is provided in Reference [29].

CERES is a modular computer program that is designed to find the economically optimal generation expansion plan for electric power generating systems. CERES is designed to be both easy to use and computationally efficient. These advantages are achieved by making the program "interactive" and using new algorithms for the most computationally intensive parts of the program. The interactive aspect of CERES provides the capability of on-line use of the program by remote access of a computer through a
time-sharing terminal. The computational efficiency was improved with respect to WASP by adapting a hybrid production costing algorithm and an iterative dynamic programming optimization technique. These techniques are described in detail in chapters 3 and 4 of this thesis.

CERES is divided into two major modules; the INPUT module and the OPTIM module. The author was responsible for the development of the OPTIM module, and this is his principal contribution in CERES.

Section 2.2 of this chapter, describes the objective, the structure, and the capabilities of CERES. Sections 2.3 and 2.4 give an outline of the INPUT and OPTIM modules, respectively.

2.2 CERES Structure and Objectives

The CERES objective is to find the most economical expansion policy for an electric generating system under specified reliability, plant type availability and generating system operation constraints. This objective is accomplished by analyzing alternative expansion plans in study horizons of up to 20 years.

The objective function, i.e., the function that is optimized, is the sum of the discounted generating system production cost and the construction expenditures of added units. What type of generating units, if any, and when they will be added are determined so as to satisfy the system reliability requirements and minimize the objective function.

The production cost and the system Loss-Of-Load Probability (LOLP) are calculated through probabilistic simulation of the
generating system for each combination of existing and added units. Although generating system expansion is assumed to occur only once a year, four simulation periods (seasons) per year are used so that the effects of scheduled plant maintenance and seasonal electric load variation are simulated more accurately. The probabilistic simulation algorithm is described and evaluated in Chapter 3.

The optimization algorithm is based on forward dynamic programming. The system configurations in each year that do not meet the dynamic programming optimization path or do not meet the system reliability requirements are rejected, and only those expansion histories (sequences of plant additions in each year of the study horizon) that satisfy the dynamic programming optimality and reliability requirements are recorded. Costs of unique histories are compared if there are multiple acceptable expansion plans. The optimum expansion plan is the one with the minimum cost among all the plans. The optimization algorithm developed for CERES is described in Chapter 4.

CERES consists of two major modules, each of which has two submodules. The INPUT module consists of the PLANT and LOAD submodules. The OPTIM module consists of the PREP and DYNO submodules. The information created by each of the CERES submodules is passed to the other submodules through disk files. The flow of information among the program submodules is schematically shown in Figure 2.1. The term "FILES" shown in this Figure refers to files stored on disk. It should be noted that the sequence of executing the PLANT and LOAD submodules is
FIGURE 2.1 Flow of Calculations Among CERES Submodules
immaterial. However, both of these submodules should be executed before PREP. Finally, DYNO execution should follow the execution of PREP.

The PLANT and LOAD submodules contain interactive dialogues through which the input data are entered and/or modified by the user. The data are subsequently supplied to the PREP submodule. The latter calculates, and transfers to DYNO, some additional parameters that are required by the optimization algorithm in DYNO.

Since some of the algorithms described in chapters 3 and 4 were incorporated in the OPTIM module, the Fortran source of this module is listed in Appendix A.

2.3 Input Requirements

CERES input data are entered through the interactive dialogues of the PLANT and LOAD submodules. PLANT and LOAD are independent programs and the order in which they are executed is immaterial.

The plant data are entered and/or modified through the interactive dialogue of the PLANT submodule. These data are:

1. Starting year of expansion study
2. Last year of expansion study
3. Maximum number of expansion plant types considered, and
4. The engineering and economic parameters needed to describe the existing generating units and the plant types considered as expansion candidates (e.g., base and maximum capacity, scheduled and forced outage rates, operating and construction costs, etc.).

All input data except the first and last year of the study horizon (expansion study period) may be revised at execution time.
(on-line) through the PLANT submodule interactive dialogue. After all the revisions, the data are stored on disk in the PLANT file.

The load data are entered through the LOAD submodule. This is the only CERES submodule that requires some of its input data not to be entered interactively. These data are the electric generating system hourly load demand data for at least one year, and it is too detailed to be entered interactively. In addition to the above, the user must enter interactively at his or her time sharing terminal the following load forecast data:

1. four seasonal energy multipliers for each year
2. four seasonal load factors for each year

The LOAD submodule interactive dialogue provides ample chance for the user to check and correct the input data. After the user input procedure is completed, load duration curves and cumulants (see Chapter 3) are calculated and written on disk for each season and each year.

Details on the plant and load data and entering procedures are given in Reference [29].

2.4 Expansion Plan Optimization

2.4.1 PREP submodule

Preparatory calculations are needed before DYNO is applied. These calculations are performed in the PREP submodule which reads and restructures the input data transferred from the INPUT module; reads the minimum acceptable (critical) LOLP value; calculates the
plant maintenance schedule for the existing units; calculates the loading order of all units considered; estimates the minimum reserve margins; calculates the plant cumulants and the lower bounds of annual operating costs. When these calculations are completed, all above information is stored on the PREP files for use by the DYNO submodule. It is essential that PREP is executed before DYNO. The flowchart in Figure 2.2 shows the sequence of calculations in the PREP submodule.

The maintenance requirement for each plant type is specified by input in number of days of maintenance shutdown per year per unit. The PREP submodule determines the season in which the maintenance of each generating unit of the existing system will take place, so that the reserve margin in each season (i.e., the difference between the system capacity and peak load demand) is the same for all seasons. The maintenance schedule of the expansion units is considered in the DYNO submodule. For more details in the maintenance scheduling see Reference [41].

The minimum reserve margin requirement ($RMIN_n$) for each year is applied in the DYNO submodule to examine the generation configurations. $RMIN_n$ is calculated, based on the Critical Loss-Of-Load Probability, denoted here by CLOLP, and represents the lower bound for the reverse margin of the generating system in year $n$ that satisfies the CLOLP. $RMIN_n$ is used to exclude the generation configurations that do not satisfy the CLOLP criterion without calculating CLOLP, which is more costly.
FIGURE 2.2 PREP Submodule Flowchart

START

READ CRITICAL LOLP

YEAR
N = 1

RESTRUCTURE PLANT DATA

READ PLANT AND LOAD DATA

CALCULATE EXISTING PLANT MAINTENANCE SCHEDULE

CALCULATE MINIMUM RESERVE MARGIN

STORE ON DISK

CALCULATE ECONOMIC LOADING ORDER

CALCULATE PLANT CUMULANTS

CALCULATE LOWER BOUND OF OPERATING COSTS

FINAL YEAR?

No

N = N + 1

PREP FILES

END

n = n + ri ~

FIGURE 2.2 PREP Submodule Flowchart
RMIN\textsubscript{n} is found by the following procedure.* For each year \( n \), the seasonal LOLP for the system, consisting only of scheduled plant types, is calculated for each season. If the seasonal LOLP is greater than CLOLP, the expansion candidate units allowed for the year are hypothetically added one by one to the system until LOLP becomes less than CLOLP. The forced outage rate of the added units is assumed to be equal to that of the lowest forced outage rate among all the allowed expansion units for the year. In adding the units, the smallest unit available is added first, and then the next smallest unit, and so on. This is repeated for each season of the year \( n \). The maximum total capacity of expansion units thus added is found and denoted by \( CA_n \). Then RMIN\textsubscript{n} is calculated by

\[
RMIN_n = CS_n + CA_n - P_n
\]

where RMIN\textsubscript{n} is the minimum reserve margin for year \( n \)

\( CS_n \) is the total capacity of the scheduled units for year \( n \)

\( CA_n \) is the total capacity of hypothetically added expansion units as mentioned above

\( P_n \) is the annual peak of the load demand in year \( n \)

The plant cumulants are calculated from the plant moments. Details of this are in Appendix B.

* Similar RMIN\textsubscript{n} calculating procedure and usage in optimization are mentioned by G.W. Evans [6].
The loading order of base and peak blocks of the units is determined in the strict order of economic merit; i.e., the least expensive units (or capacity blocks for units represented with two capacity blocks) are added first.

The calculation procedure for the lower bound of operating cost, and its implementation in DYNO, are described in Chapter 4.

2.4.2 Objective Function Optimization and Constraints

All CERES input and calculated information is transferred to the DYNO submodule. This submodule performs the economic evaluation of the alternative expansion plans, and determines the best expansion policy for the system. An iterative forward dynamic programming method is used to find the expansion plan that gives the minimum discounted cash flow of capital and operating expenditures over the study period. The value of the objective function (total cost), which is to be minimized through dynamic programming, is calculated from the cost of each system configuration in each study year. DYNO provides the option of sensitivity analysis whereby the user can study the effects of allowing different expansion types, different discount rates, and different definitions of objective function. A report of the optimal or the few next best suboptimal solutions, is produced at the user's request. A flowchart of the DYNO functions is shown in Figure 2.3. For each case study N, the plant expansion types, the maximum reserve margin, the discount rate, and the form of the objective function should be specified. Following the objective function
READ INPUT DATA

CASE STUDY
N = 1

READ THE USER INPUT INTERACTIVELY

FIND THE OPTIMAL SOLUTION

RESIMULATE THE OPTIMAL AND THE SUBOPTIMAL SOLUTIONS REQUESTED

PREP FILES

A

B

FIGURE 2.3 DYNO Submodule Flowchart
FIGURE 2.3 (continued)
optimization, the optimal and some (up to 10, as specified by the user) next best solutions are resimulated and printed by the report writer. The resimulation is necessary for the recovery of details that are not stored in memory during the optimization.

CERES has two alternative definitions of objective functions: a traditional definition [13,14] and an alternative definition using the levelized fixed charge rate [42-44]. Either of the two options can be chosen for sensitivity analysis at execution time (on-line). The traditional objective function is defined as the sum of the operating costs and capital costs for construction minus the salvage value of the plants, both of which are discounted to a specified base year (see Section 4.2.1). The salvage value considered in the objective function represents the credit given for the unused portion of the unit life (see discussion at the end of Section 4.5 and References [13,14]). In the fixed charge rate (FCR) option, the objective function is the sum of (a) the present worth of the total operating cost for all generating units during the study horizon, and (b) the present worth of the fixed charges of all generating units that are added to the system during the study horizon.

The constraints imposed on the allowable generation configurations for each year are divided into two types as follows:

a. User-specified constraints
   i) LOLP < CLOLP
   ii) Maximum reserve margin, $R_{MAX_n}^n$
   iii) Upper and lower bound on the number of expansion units allowed in each plant type
b. Computational constraints

iv) Minimum reserve margin, RMIN

v) Tunnel constraints

The reserve margin constraints are applied first, since their calculation is much less expensive than that of LOLP. If the reserve margin of any configuration (state) in year n is less than RMIN or more than RMAX, that state is immediately disqualified because either constraint (i) or (ii) will not be satisfied. Tunnel constraints are used to limit temporarily the number of states to be considered in a run of dynamic programming optimization and thus reduce the storage and computing requirements. More details on the use of tunnels are given in Chapter 4.

The constraints are applied in the following hierarchy:

1. Upper and lower bounds on the number of expansion units allowed
2. RMIN
3. RMAX
4. LOLP
5. Tunnels

The fathoming technique, which is also used to restrict the number of admissible states, is discussed in Chapter 4.

Finally, DYNO provides the option of performing on-line sensitivity analysis on expansion plant types; discount rate; minimum and maximum reserve margins; and form of objective function. Several report options of varying detail are available.
2.5 Summary

CERES is a highly efficient and easy to use electric power capacity expansion optimization program. It is easy to use because most of the man-machine interactions are performed through "plain English" dialogues. It is efficient, because the new algorithms used (cumulant method, fathoming, automatic tunnel iterations) and the overall CERES design make CERES faster and more accurate than WASP. CERES should be of particular use to regulatory agencies and electric utilities for its simplicity and accuracy.
CHAPTER 3

EVALUATION OF ELECTRIC GENERATING SYSTEM SIMULATION
METHODS TO ESTIMATE RELIABILITY AND ENERGY PRODUCTION

3.1 Introduction

The fast and accurate calculation of system reliability and of plant energy generation is one of the most important requirements in capacity expansion planning of electric generating systems. A typical expansion planning study requires simulation of a large number of generating unit configurations. Although a very accurate simulation of the generating system may be expensive, accuracy should not be compromised, otherwise the calculated optimum expansion plan may not be the true optimum.

Considerable research has been devoted in an effort to solve this problem of high accuracy and low computing cost in the simulation of electric generating systems. Since both the electric load and the forced outages of the generating units are random variables, most production costing and capacity planning models use probabilistic simulation [13-17, 45-48]. However, accurate representation of electric generating system plant and load data has been a significant problem [16, 49-50] because this data must be represented in such a form that would facilitate the simulation of the operations of the generating system. For example, the system load demand may be represented with the
load demand for each hour of the simulation period. However, hourly simulation of the operation of the generating system is so time consuming, that it is used only for small simulation time periods and a few generating system configurations. Therefore hourly simulation is used only in detailed production costing models. The recent introduction of the cumulant method in probabilistic simulation of electric generating systems [7, 30, 35] has been a breakthrough that attempts to satisfy the requirements of both high accuracy and short computing time. Past sensitivity analyses to validate the cumulant method have been limited to either comparative evaluation of LOLP, calculated from capacity outage tables with no load demand considerations [31-33], or to idealized load probability distributions [36].

Numerical tests during the development of CERES showed that the cumulant method alone can not be used in a satisfactory manner for capacity expansion planning. The accuracy and computational efficiency of the cumulant method were evaluated in a comprehensive sensitivity analysis, for both energy and reliability calculations. The cumulant method was compared to two widely used probabilistic simulation methods, namely the piecewise linear and derate methods; the first is known to be very accurate [16, 49] and the second very fast [27, 28].

The advantages and disadvantages of the cumulant method led to the following two new developments. First, a new hybrid method for probabilistic simulation of electric generating systems was developed, in which cumulants are used for energy and piecewise linear polynomials for reliability calculations. Second, a new method for incorporating
load forecast uncertainty in expansion planning was formulated. In this method, the effects of the uncertainty in future energy demand are incorporated explicitly into the calculation of both the generating system reliability and the expected energy production of the generating units. The author's contributions in this chapter consist of the two new methods outlined above, and the determination of the relative merits of the piecewise, cumulant and derate methods, in a comprehensive sensitivity analysis.

This chapter is organized as follows. Section 3.2 contains an outline of the basic principles of probabilistic simulation of electric generating systems. Although these principles have been described elsewhere [50-57], the outline is necessary for the understanding of the remainder of this chapter. In Section 3.3 the cumulant, piecewise linear, and derate methods are described, and their comparative evaluation is presented in Section 3.4. The hybrid and forecast uncertainty algorithms are described in Sections 3.5 and 3.6 respectively. The most important results of this chapter are summarized in Section 3.7.

3.2 Probabilistic Simulation of Electric Generating Systems

This section outlines the basic principles of the probabilistic simulation of electric generating systems. The purpose of probabilistic simulation in capacity expansion planning is to estimate the generating system's reliability and the expected energy production of each generating unit. The former is used to reject unreliable expansion candidate configurations while the latter provides the means for calculating the generating system production costs.
3.2.1 Load Demand and Outage Capacity as Random Variables

The need for probabilistic simulation of electric generating systems rises from the inability to forecast the exact future load demand and generating unit forced outages. Only the probability distribution of future load demand and generating unit outages can be forecasted. Therefore, both the load demand and the generating unit outages are treated as random variables in the context of probabilistic simulation.

In capacity expansion planning, the simulation period for production costing are usually months or seasons [58] although some models use weeks [11, 47] and others half-years [59]. Shorter periods provide more accurate simulation of scheduled maintenance, pump-storage and hydro plants, but require more computing cost.

The future load demand is usually specified by the "load probability function", \( L(x) \). This is also known as "inverted load duration curve" and "complementary distribution function". \( L(x) \) is defined as the probability that a random load \( \tilde{x} \) will equal or exceed a demand level of \( x \) MW. That is:

\[
L(x) = P (\tilde{x} \geq x) \tag{3.1}
\]

The load frequency (density) function, \( l(x) \), is defined by

\[
l(x)dx = P (x < \tilde{x} \leq x+dx) \tag{3.2}
\]
where \( l(x)dx \) is the probability that a random load of \( x \) MW takes a value between \( x \) MW and \( x+dx \) MW. From Equation (3.1) and (3.2) it is evident that

\[
L(x) = \int_x^{\infty} l(x')dx' \tag{3.3}
\]

Typical load probability and load frequency curves are shown in Figures 3.1 and 3.2.

The forced outages of a generating unit are specified with the outage capacity frequency function, \( q(c) \), as

\[
q(c)dc = P(\tilde{c} \leq c < c + dc) \tag{3.4}
\]

where the \( q(c)dc \) is the probability that a random outage of \( \tilde{c} \) MW will be between the unit's generation level \( c \) and \( c + dc \) MW. If \( C \) is the unit's rated capacity then

\[
\int_0^{C} q(c)dc = 1 \tag{3.5}
\]

The unit's availability probability \( p \) is

\[
p = \int_0^{C} q(c)\delta(c-0)dc \tag{3.6}
\]

where \( \delta \) is the Dirac delta function. The probability, \( q \), that the unit is unable to provide any power is
Figure 3.1  Typical Load Probability Curve
Figure 3.2 Typical Load Frequency Curve
q = \int_{0}^{C} q(c) \delta(c-C) dc \quad (3.7)

In many instances [57, 58] generating units are represented with two states: a) available and capable of full power generation and b) on total forced outage. In this representation q and p are the unit's forced outage and availability probability, and Equation (3.5) reduces to

\[ p + q = 1 \quad (3.8) \]

A typical two state discrete outage capacity frequency function is shown in Figure 3.3, where p and q are represented with two spikes, at zero and full capacity respectively.

3.2.2 Equivalent Load Demand

Assume that the generating system consists of N generating units with loading order 1,2,3, ..., n, ..., N and with capacities \( C_1, C_2, \ldots, C_n, \ldots, C_N \) where \( C_n \) is the n-th unit's capacity in MW and \( P_1, P_2, \ldots, P_n, \ldots, P_N \) are the units' availability probabilities. Let the load probability function be, \( L(x) \), as depicted in Figure 3.4. If the simulation period is \( T \) hours, the expected energy generation, \( E_1 \), of the first unit is

\[ E_1 = T \ p_1 \int_{0}^{C} L(x) dx \quad (3.9) \]

as depicted by the shaded area in Figure 3.4. In order to calculate the expected energy generation of the second unit, we should
Figure 3.3 Two State Representation of a Generating Unit's Outage Capacity Frequency Function
Figure 3.4 Load Probability Curve. The Shaded Area Depicts the Expected Energy Generation by the First Generating Unit
realize that the second unit will shift in the loading order to replace the whole or part of the first unit, when the latter is totally or partially on forced outage. This means that the load demand for the second unit will be higher when the first unit is on outage than otherwise. Therefore, before calculating the energy of the second unit, we should increase the load demand by the amount of outage of the first unit. The new load demand is called equivalent load demand, and its load probability function, $EL_1(x)$, is defined as the probability that a random equivalent load of $\tilde{x}_1$ MW will equal or exceed a demand level of $x$ MW

$$EL_1(x) = P(\tilde{x}_1 > x)$$ (3.10)

where $\tilde{x}_1$ is the sum of random load demand $\tilde{x}$ and outage capacity $\tilde{c}_1$, of the first unit

$$\tilde{x}_1 = \tilde{x} + \tilde{c}_1$$ (3.11)

Let $l_1(x,c)$ be the combined frequency function of $x$ and $c_1$, i.e.

$$l_1(x,c) = P(x < \tilde{x} \leq x+dx, \ c < \tilde{c}_1 \leq c+dc)$$ (3.12)

Assume that $\tilde{x}$ and $\tilde{c}_1$ are independent random variables. Any pair of random variables $\tilde{x}$ and $\tilde{c}_1$ that satisfy the Equation

$$\tilde{x} + \tilde{c}_1 \geq x$$ (3.13)
take values in the space to the right of the line

\[ \tilde{x} + \tilde{c}_1 = x \]  

(3.14)

as depicted by the shaded area in Figure 3.5. Thus

\[ \text{EL}_1(x) = P(\tilde{x} + \tilde{c}_1 > x) \]

\[ = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{l}_1(x', c) dx' dc \]  

(3.15)

However, the necessary and sufficient condition that random variables \( \tilde{x} \) and \( \tilde{c}_1 \) are independent [60] is

\[ \text{l}_1(x, c) = \text{l}(x) \text{q}_1(c) \]  

(3.16)

where \( \text{l}(x, c) \) is the combined frequency function of \( x \) and \( c_1 \) (see Equation (3.12))

\( \text{l}(x) \) is the frequency function of \( \tilde{x} \) (see Equation (3.12))

\( \text{q}_1(c) \) is the frequency function of \( \tilde{c}_1 \) (see Equation (3.3))

\( x \) is the load in MW, and

\( c \) is the outage capacity of unit one, in MW

Substituting Equation (3.16) and (3.3) in (3.15) yields

\[ \text{EL}_1(x) = \int_{-\infty}^{\infty} \text{q}_1(c) dc \int_{x-c}^{\infty} \text{l}(x') dx' \]

\[ = \int_{-\infty}^{\infty} \text{q}_1(c) L(x-c) dc \]  

(3.17)
Figure 3.5 Graphic Representation of the Sum of Independent Random Variables $x$ and $c$
Since \( q_1(c) = 0 \) for \( c < 0 \) and \( c > C_1 \), where \( C_1 \) is the maximum capacity of the first unit,

\[
EL_1(x) = \int_0^{C_1} L(x-c) q_1(c) dc \quad (3.18)
\]

which is the familiar convolution equation of random variables \( x \) and \( C_1 \).

When \( q_1(c) \) is discrete, Equation (3.18) becomes

\[
EL_1(x) = p_1 L(x) + \sum_{j=1}^J L(x-c_j)q_{1j} \quad (3.19)
\]

where \( q_{1j} \) is the discretized outage capacity frequency function for the first unit defined by

\[
q_{1j} = q_1(c_j) \quad j = 1, 2, \ldots, J \quad (3.20)
\]

In the two state representation [cf. equations (3.6) to (3.8)], we have

\[
EL_1(x) = p_1 L(x) + q_1 L(x-C_1) \quad (3.21)
\]

where \( EL_1(x) \) is the equivalent load probability function of the first generating unit (see Equation (3.10));

\( L(x) \) is the load probability function (see Equation (3.1));

\( p_1 \) is the availability probability of the first unit;

\( q_1 \) is the forced outage rate of the first unit;

\( C_1 \) is the first unit's capacity, in MW;

\( x \) is the load, in MW.
This form of the convolution equation has been suggested by Balereux [50] and later used by Booth [52-54] and others [13-15, 55-57].

The equivalent load probability function, \( EL_n(x) \), that results from the convolution of the first \( n \) generating units with the system load is defined as

\[
EL_n(x) = P(\tilde{x}_n > x) \tag{3.22}
\]

where

\[
\tilde{x}_n = \tilde{x} + \sum_{i=1}^{n} \tilde{c}_i \tag{3.23}
\]

\( \tilde{x} \): random load demand

\( \tilde{c}_i \): random outage capacity of unit \( i \)

\( \tilde{x} \): system load

\( EL_n(x) \) is found by recursive application of Equation (3.18), i.e.,

\[
EL_n(x) = \int_{0}^{x} EL_{n-1}(x-c) q_n(c)dc \tag{3.24}
\]

where \( EL_n(x) \) : the equivalent load probability function of the \( n \)-th unit, defined by Equations (3.22) and (3.23)

\( EL_0(x) \) : the (original) load demand \( L(x) \) as defined in Equation (3.1)

\( C_n \) : is the capacity of unit \( n \)

\( q_n(c) \) : the outage capacity frequency function of unit \( n \)

\( c \) : is the outage capacity in MW, and

\( x \) : is the load in MW
Since the sum of random variables is commutative (cf. Equation (3.13) and Reference [62]), the order with which units 1 through n are convolved is immaterial. Therefore, if we know the equivalent load probability function EL\(_n\)(x) of the first n units and we want to find the equivalent load probability function, EL\(_k\)(x), of n-1 units that does not incorporate the effects of outages of any unit k (k<n), we must solve the equation

\[
EL_n(x) = \int_0^{C_k} EL_{n-1}(x-c) q_k(c) dc
\]  

(3.25)

where EL\(_{n-1}\)(x) is the equivalent load probability function of units 1, 2, ..., k-1, k+1, ..., n

q\(_n\)(c) is the k-th unit's outage capacity frequency function, and

C\(_k\) is the k-th unit's capacity, in MW

This process is called deconvolution and is used when generating units are represented with more than one capacity blocks. In this case, the recursive application of the convolution Equation (3.18) for the two capacity blocks of the same generating unit is not possible because the outages of the two blocks are not independent and Equation (3.16) is not valid. Therefore, if unit k is represented with the two capacity blocks C\(_{k1}\) and C\(_{k2}\) with loading order 1, 2, ..., k\(_1\), ..., n-1, k\(_2\), n+1, ..., N, where N is the total number of units of the generating system, EL\(_{k2}\)(x) is found by first deconvoluting capacity block k\(_1\) from EL\(_{n-1}\)(x) using Equation (3.25) and then
convolving both capacity blocks $C_{k_1}$ and $C_{k_2}$ as one block. Thus

$$EL_{k_2}(x) = \int_0^{C_{k_1} + C_{k_2}} EL_{n-1}(x-c) q_{k_1 k_2}(c) dc$$ (3.26)

where $EL_{n-1}(x)$ is found by solving the equation

$$EL_{n-1}(x) = \int_0^{C_{k_1} k_1} EL_{n-1}(x-c) q_{k_1}(c) dc$$ (3.27)

where $q_{k_1 k_2}(c)$ is the outage capacity frequency function for both capacity blocks of unit $k$.

### 3.2.3 Energy and Reliability Calculations

The energy $E_n$ generated by the $n$-th unit in the loading order is given by Equation (3.9), if $L(x)$ is replaced by $EL_{n-1}(x)$ and the limits of integration are changed:

$$E_n = T p_n \int_{SC_{n-1}}^{SC_n} EL_{n-1}(x) dx$$ (3.28)

where $E_n$: expected energy generation, in MWH, by the $n$-th unit

$T$: simulation time period, in hours

$p_n$: $n$-th unit availability probability (cf. Equations (3.4-3.8))

$EL_{n-1}(x)$: $(n-1)$th equivalent load probability function

$x$: equivalent load in MW, and

$SC_n$ is defined as
where \( C_i \) is the capacity of the \( i \)-th unit in MW. \( E_n \) is depicted by the shaded area in Figure 3.6.

Assume that all the emergency actions that a utility can take to avoid loss of load (e.g. to interrupt the interruptible load customers, purchase emergency power) have been incorporated in the last equivalent load probability curve either by modifying the original load probability function \([57]\), or as pseudo generating units \([46]\). The probability that the generating system (plus emergency power) will not satisfy the load demand is called loss-of-load-probability, and is found from the last equivalent load probability function, as

\[
\text{LOLP} = E_L(NSC_N) 
\]

where \( \text{LOLP} \) is the generating system's loss-of-load probability

- \( E_L(N) \) is the last (N-th) load probability function
- \( SCN \) is the total system capacity (cf. Equation (3.29))

The unserved energy (expected energy deficiency), \( UE \), is

\[
UE = T \int_{SC_N}^{\infty} E_L(x)dx 
\]

Both LOLP and UE are depicted in Figure (3.7) for a typical generating system.
Figure 3.6 Equivalent Load Probability Function for the (n-1)th Unit. The Shaded Area Depicts the Expected Energy Generation by the n-th Unit.
Figure 3.7 Final Equivalent Load Probability Curve Depicting the Loss-of-Load-Probability and Expected Unserved Energy for a Typical Generating System
3.3 Approximations in Probabilistic Simulation

In deriving the convolution and deconvolution formulas in Section 3.2, it was assumed that the load probability function \( L(x) \) is continuous. However, this assumption cannot be satisfied in the discrete environment of digital computers. Therefore, \( L(x) \) is approximated in some discrete form. The outage capacity frequency function \( q(c) \) is also discretized. In this section three methods of approximating \( L(x) \) and \( q(c) \) are presented. These are:

a) piecewise linear approximation

b) derate approximation

c) cumulant approximation

3.3.1 Piecewise Linear Approximation

In the piecewise linear approximation the load probability function, \( L(x) \), is represented with a piecewise linear polynomial. The load axis \( x \) is divided in grid points \( x_i, x_{i+1} = i \Delta x, i=1, 2, \ldots, I \), where \( \Delta x \) is the interval between two consecutive grid points, and \( i \) is the grid number. Given the values \( L_i = L(x_i) \) of the load probability function for all the grid points, the approximate representation of \( L(x) \) is given by \( L_a(x) \)

\[
L_a(x) = \frac{x + x_{i+1}}{\Delta x} L_i + \frac{x - x_i}{\Delta x} L_{i+1}
\]  

(3.32)

The value of \( L(x) \) at the intervals between grid points is found by linear interpolation between the grid values. The representation of the
The discrete forms of the convolution and deconvolution Equations (3.24) and (3.25) are

\[
EL_n(x_i) = \sum_{j=0}^{J} EL_{n-i}(x_i - c_j)q_j
\]

(3.33)

\[
EL_{n-1}(x_i) = \frac{1}{p} \{EL_n(x_i) - \sum_{j=1}^{J} EL_{n-1}(x_i - c_j)q_j\}
\]

(3.34)

The values of \(EL(x_i - c_j)\) between the grid points \(EL_i = EL(x_i)\) are approximated by \(L_a(x)\) from Equation (3.32). Both equations (3.33) and (3.34) are applied recursively for \(i=1\) to \(i=I\), where \(I\) is the total number of grid points. In equation (3.34) we set

\[
EL_{n-1}(x_i - c_j) = 1, \text{ for any } x_i - c_j \leq 0.
\]

The piecewise linear method has been shown to represent the equivalent load probability curve \(EL(x)\) (or \(L(x)\)) with error smaller than 0.0001 when 500 or more grid points are used [16]. Therefore, the LOLP is accurate to within 0.0001 (or 0.036 days per year) and expected generating unit energy is accurate to within four significant digits. It is noted that many electric utilities use the 0.1 day per year LOLP value as their planning target [63].
3.3.2 Derate Method

In the derate method only the load demand is treated as a random variable. The load probability function is represented by piecewise linear polynomial as in the piecewise linear method (see Equations (3.32), (3.33)). However, it is assumed that the generating units can be treated as firm units of reduced capacity. Thus, if unit \( n \) has capacity \( C_n \) and forced outage rate \( q_n \) the equivalent unit capacity is

\[
C_n' = (1-q_n) C_n
\]  

(3.35)

Since \( C_n' \) is firm capacity with availability \( p_n' = 1 \) and forced outage \( q_n' = 0 \), the convolution Equation (3.24) becomes

\[
E L_n(x) = p_n' E L_{n-1}(x) + q_n' E L_{n-1}(x-C_n')
\]

\[= E L_{n-1}(x)\]

\[= E L_0(x)\]

\[= E(x)\]

(3.36)

Thus no convolutions or deconvolutions are required in this method. The energy generation of each unit is given by

\[
E_n = T \int_{SC_{n-1}}^{SC_n} L(x)dx
\]  

(3.37)
where

\[ E_n : \text{expected energy generation by unit } n \text{ in MW} \]

\[ L(x) : \text{load probability function} \]

\[ T : \text{period duration in hours} \]

\[ SC : \text{is defined by Equation (3.29) if } C \text{ is replaced by } C' \text{ from Equation (3.35)} \]

Calculation of meaningful LOLP is impossible, since this method fails to recognize the random nature of the generating unit forced outages [52]. Therefore, wherever this method is used for production costing (mainly in linear programming models [27, 28]), LOLP is calculated by different methods (e.g. LOLP is found from capacity outage tables).

3.3.3 Cumulant Approximation

In the cumulant approximation both the load, \( L(x) \), and the equivalent load probability function, \( EL(x) \), and the outage capacity frequency function, \( q(c) \), are represented with their cumulants.

The Gram-Charlier and Edgeworth series [7, 60, 64] are used for the reconstruction of \( EL(x) \) from its cumulants. When the cumulant representation of \( EL(x) \) is used in probabilistic simulation, the convolution and deconvolution equations take very simple forms as explained in the remainder of this section.

Let \( m_r \) be the \( r \)th moment of frequency function \( f(x) \)

\[
m_r = \int_{-\infty}^{\infty} x^r f(x) dx \quad (3.38)
\]
The cumulants $K_r$ of $f(x)$ are defined by the identity in $t$ [62].

$$
\exp[K_1 t + \frac{K_2 t^2}{2} + \ldots + \frac{K_r t^r}{r!} + \ldots]
$$

$$
= 1 + m_1 t + \frac{m_2 t^2}{2} + \ldots + \frac{m_r t^r}{r!} + \ldots \quad (3.39)
$$

If the left hand side of Equation (3.39) is expanded to form a polynomial in $t$ and the coefficients of the same powers of $t$ on both sides are equated, the cumulants are defined in terms of the moments of the frequency function. The defining equations for the first eight cumulants and the numerical algorithms with which the moments are obtained from system load and outage capacity data are given in Appendix B.

The Fourier transform $CF(t)$ of the frequency function $f(x)$ of random variable $x$ is known as the characteristic function of $f(x)$

$$
CF(t) = \int_{-\infty}^{\infty} f(x)e^{-jxt}dt \quad (3.40)
$$

Expanding $CF(t)$ in MacLaurin's series and using Equation (3.39) results in

$$
\log[CF(t)] = K_1(jt) + K_2 \left(\frac{jt}{2}\right)^2 + \ldots + K_r \left(\frac{jt}{r!}\right)^r + \ldots \quad (3.41)
$$
Differentiation with respect to \( x \) of both sides of Equation (3.24) gives

\[
\ln(x) = \int_{-\infty}^{\infty} \ln(x-c) q_n(c) dc \tag{3.42}
\]

where \( \ln(x) \) is the frequency function of \( \ln(x) \) (c.f Equation (3.3)) and the limits of integration were extended to \( \pm \infty \). Let \( CL_n(t) \), \( CL_{n-1}(t) \) and \( Q_n(t) \) be the Fourier transforms (characteristic functions) of \( \ln(x) \), \( \ln(x-c) \) and \( q_n(c) \) respectively. Thus

\[
CL_n(t) = \int_{-\infty}^{\infty} \ln(x)e^{-jxt} dx
\]

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \ln(x-c)q_n(c)e^{-jxt} dc dx
\]

\[
= \left[ \int_{-\infty}^{\infty} \ln(x-c)e^{j(x-c)t} d(x-c) \right] \left[ \int_{-\infty}^{\infty} q_n(c)e^{-jct} dc \right]
\]

\[
= CL_{n-1}(t) Q_n(t) \tag{3.43}
\]

and

\[
\log CL_n(t) = \log CL_{n-1}(t) + \log Q_n(t) \tag{3.44}
\]

However Equation (3.41) implies that Equation (3.44) can be written as

\[
\sum_{r=1}^{n} \frac{(jt)^r}{r!} = \sum_{r=1}^{n-1} \frac{(jt)^r}{r!} + \sum_{r=1}^{n} a_n \frac{(jt)^r}{r!} \tag{3.45}
\]
where $K_{n,r}$, $K_{n-1,r}$ and $a_{n,r}$ are the $r$th cumulants of $l_n(x)$, $l_{n-1}(x)$ and $q_n(c)$ respectively. The last Equation suggests that

$$K_{n,r} = K_{n-1,r} + a_{n,r} \quad (3.46)$$

Thus it is shown that the cumulants of the sum (convolution) of two random variables ($x$ and $c$) are the sum of the same order cumulants of the two random variables. Therefore, the convolution Equation (3.24) is reduced to summation of cumulants. Similarly the deconvolution Equation (3.25) is reduced to

$$K_{n-1,r} = K_{n,r} - a_{n,r} \quad (3.47)$$

where $K_{n,r}$ and $a_{n,r}$ are defined as in Equation (3.45), and $K_{n-1,r}$ are the cumulants of the frequency function $l_{n-1}(x)$ of the equivalent probability function $EL_{n-1}(x)$ as defined in Equation (3.25).

The simple form to which the convolution and deconvolution of random variables is reduced with the use of cumulants would have been of little practical value if the original load probability distributions could not be recovered from their cumulants. Fortunately this is possible through the Gram-Charlier or Edgeworth series expansion in terms of the standard normal distribution and its derivatives [60, 62]. Since the accuracy of these series is one of the focal points of the sensitivity analysis presented in the next section, the presentation of both the expansion
formulas and the algorithms that have been used for energy and LOLP calculations is deferred to the next section.

3.4 Sensitivity Analysis

This section presents a comprehensive sensitivity analyses in which both the accuracy and computational efficiency of the cumulant method are evaluated by comparison with the piecewise linear and derate methods.

3.4.1 Methodology

In this evaluation of the cumulant method, the piecewise linear method is considered as the benchmark for accuracy, and the derate method the benchmark for computational speed (see Sections 3.3.1 and 3.3.2)

The following three cumulant series [60] were evaluated:

1) 4-Cumulant Gram-Charlier or Edgeworth series

\[
EL_4(z) = 1 - \int_{-\infty}^{\infty} zN(z)dz + \frac{g_3}{3!} N^2(z) - \frac{g_4}{4!} N^3(z) \\
- \frac{10g_3^2}{6!} N^5(z) \quad (3.48)
\]
2) 5-Cumulant Edgeworth series

\[ EL_5(z) = EL_4(z) + \frac{g_5}{5} N_4(z) + \frac{35g_3^2g_4}{7} N_6(z) + \]
\[ + \frac{280g_3^2}{9!} N_8(z) \]  
(3.49)

3) 8-Cumulant Gram-Charlier series

\[ EL_8(z) = EL_4(z) + \frac{g_5}{5!} N_4(z) - \frac{g_6}{6!} N_5(z) + \]
\[ + \frac{(g_7 + 10g_3^2g_4)}{7!} N_6(z) - \frac{(g_8 + 56g_3^2g_5 + 35g_4^2)}{8!} N_7(z) \]  
(3.50)

where \( EL_1(z) \) is the equivalent load probability distribution represented with \( i \) cumulants (\( i = 4, 5, \) or \( 8 \))

\( g_i \) is the \( i \)-th standardized cumulant

\( N(z) \) is the standard probability distribution

\( N^n(z) \) is the \( n \)-th derivative of \( N(z) \)

\( z \) is the standardized variable.
\[ g_1, z \text{ and } N(z) \text{ are given by} \]

\[ z = \frac{(x-m)}{\sigma} \]  

\[ N(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2/2) \]  

\[ g_1 = K_{i+1}/\sigma^{i+2} \]

where \( m, \sigma, \) and \( K_i \) are the mean, standard deviation, and cumulants of the equivalent load probability function \( EL(x) \), and \( x \) is the load in MW, and where \( m \) and \( \sigma \) are the mean and standard deviation of the equivalent load probability \( EL(x) \), and \( x \) is the load in MW.

The cumulant series, the piecewise linear and derate approximations were applied to twenty two combinations between load demand distributions and generating system data. In each case the three methods are compared for (a) accuracy of representation of the load probability distribution, (b) accuracy of LOLP and energy calculations, and (c) computing time requirements.

In both the piecewise linear and derate methods the original load probability distribution was approximated with a piecewise linear polynomial of 500 grid points, so that the LOLP values of 0.0001 could be calculated accurately (cf. Section 3.3.1). In the piecewise linear method the size of the grid was kept constant and the number of grid points was increased when necessary, so that equivalent load probability values \( >10^{-7} \) were included.

In the derate and piecewise linear methods the expected energy generation by the generating units was calculated by numerical evaluation of the integrals in Equations (3.28) and (3.37) respectively, using the trapezoidal rule. In the case of the cumulant method, Equation (3.28) is reduced to
\[
E_n = p_n \frac{T}{2} \left[ EL_{n-1}(SC_n) + EL_{n-1}(SC_{n-1}) \right] C_n \tag{3.54}
\]

where \( E_n \) is the expected energy generation of unit \( n \) in MW

\( p_n \) is the availability probability of unit \( n \)

\( C_n \) is the capacity of the \( n \)-th unit in MW

\( EL_{n-1}(SC_n) \) is the value of equivalent load probability

\( T \) is the period length in hours, and

\( SC_n \) as defined in Equation (3.29)

\( EL_{n-1}(SC_n) \) was evaluated through Equations (3.48) to (3.50).

The total energy generation was found in all three methods by summing the energy generated by all system units

\[
E_T = \sum_{n=1}^{N} E_n \tag{3.55}
\]

where \( E_T \) is the total expected energy generation (MWH)

\( E_n \) is the \( n \)-th unit expected energy generation (MWH)

\( N \) is the total number of units in the generating system.

Two reliability indexes were calculated: a) the Loss-of-Load Probability (LOLP) and b) expected Unserved Energy (UE). LOLP was calculated from Equation (3.30) in both piecewise linear and cumulant methods.

In all three methods the UE is calculated from the difference between the energy demand (the area under the original load probability curve) and the total energy generation:
\[ UE = E_D - E_T \]  

where \( UE \) : expected unserved energy (MWH)  
\( E_D \) : energy demand (MWH)  
\( E_T \) : total expected energy generation (MWH)

In all three probabilistic simulation methods the generating units were simulated with one capacity block* and were assumed to be either available with probability \( p \) and capable of full power generation, or on forced outage with probability \( q \) and zero power output. Therefore, the outage capacity frequency function consisted of two spikes at zero and full capacity MW levels as shown in Figure 3.3

Further computational details can be found in the source listing of the PROSES program, that was written and used for this sensitivity analysis. Appendix C contains the fortran listing of the \textit{PRQbabilistic Simulation of Electrical Systems} (PROSES) program.

3.4.2 Load Characteristics and Generating Systems Considered

The accuracy and computational efficiency of the cumulant method were evaluated by comparison with the piecewise linear and derate

* In the simulation of expected energy production the generating units were not represented with multiple capacity blocks, because such representation would only amplify the large computing time advantage of the cumulant method, as noted in Section 3.5
methods in 22 case studies. The hourly load data that were used to construct the load probability function, \( L(x) \), for each case study are not listed here, because such listings are too voluminous. However, the peak load, the energy demand and the load factor of the load data for each case study are listed in the 2nd, 3rd, and 4th column in Table 3.1. The first column of Table 3.1 contains the case study identification number, which is of the form \( nA \) or \( nB \), where \( n \) is an integer from 1 to 11. The last three columns of Table 3.1 contain the number of units, the total capacity, and the effective capacity of the generating system for each of the 22 case studies. The effective capacity is the total of the unit capacity times \((1 - \text{forced outage rate})\). It should be observed in Table 3.1 that each pair of cases \( nA \) and \( nB \) has the same load characteristics (peak load, energy demand and load factor) and total capacity but a different number of generating units, and effective capacity. In all the pairs, the number of generating units in case \( nB \) is less than that in the corresponding case \( nA \) (this means the average capacity of generating units in case \( nB \) is assumed to be greater than that in case \( nA \)). Cases with different \( n \) are characterized by different load characteristics and a different generating system, except that cases 1A, 2A, 3A, 4A, and 5A have an identical generating system and, that Cases 1B, 2B, 3B, 4B, and 5B have another identical generating system.

The capacities and forced outage rates of all the units of the generating system that was used in cases 1A, 2A, 3A, 4A, and 5A are shown in the first six columns of Table 3.2. The last three columns of
### TABLE 3.1 Load and Generating System Characteristics

<table>
<thead>
<tr>
<th>CASE NUMBER</th>
<th>LOAD DATA</th>
<th>GENERATING SYSTEMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PEAK LOAD (MW)</td>
<td>ENERGY DEMAND (GWH)</td>
</tr>
<tr>
<td>1A</td>
<td>7,902</td>
<td>38,631</td>
</tr>
<tr>
<td>1B</td>
<td>7,902</td>
<td>38,631</td>
</tr>
<tr>
<td>2A</td>
<td>6,988</td>
<td>10,304</td>
</tr>
<tr>
<td>2B</td>
<td>6,988</td>
<td>10,304</td>
</tr>
<tr>
<td>3A</td>
<td>5,661</td>
<td>8,371</td>
</tr>
<tr>
<td>3B</td>
<td>5,661</td>
<td>8,371</td>
</tr>
<tr>
<td>4A</td>
<td>7,902</td>
<td>10,854</td>
</tr>
<tr>
<td>4B</td>
<td>7,902</td>
<td>10,854</td>
</tr>
<tr>
<td>5A</td>
<td>6,599</td>
<td>9,102</td>
</tr>
<tr>
<td>5B</td>
<td>6,599</td>
<td>9,102</td>
</tr>
<tr>
<td>6A</td>
<td>6,152</td>
<td>32,115</td>
</tr>
<tr>
<td>6B</td>
<td>6,152</td>
<td>32,115</td>
</tr>
<tr>
<td>7A</td>
<td>4,134</td>
<td>23,935</td>
</tr>
<tr>
<td>7B</td>
<td>4,134</td>
<td>23,935</td>
</tr>
<tr>
<td>8A</td>
<td>3,381</td>
<td>19,098</td>
</tr>
<tr>
<td>8B</td>
<td>3,381</td>
<td>19,098</td>
</tr>
<tr>
<td>9A</td>
<td>2,841</td>
<td>14,312</td>
</tr>
<tr>
<td>9B</td>
<td>2,841</td>
<td>14,312</td>
</tr>
<tr>
<td>10A</td>
<td>1,992</td>
<td>10,501</td>
</tr>
<tr>
<td>10B</td>
<td>1,992</td>
<td>10,501</td>
</tr>
<tr>
<td>11A</td>
<td>1,393</td>
<td>8,091</td>
</tr>
<tr>
<td>11B</td>
<td>1,393</td>
<td>8,091</td>
</tr>
</tbody>
</table>
## TABLE 3.2 Generating Unit Characteristics for Cases 1A through 5B

<table>
<thead>
<tr>
<th>UNIT NO.</th>
<th>CAPACITY (MW)</th>
<th>FOR&lt;sup&gt;α&lt;/sup&gt;</th>
<th>UNIT NO.</th>
<th>CAPACITY (MW)</th>
<th>FOR&lt;sup&gt;α&lt;/sup&gt;</th>
<th>UNIT NO.</th>
<th>CAPACITY (MW)</th>
<th>FOR&lt;sup&gt;α&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>346</td>
<td>.18</td>
<td>30</td>
<td>143</td>
<td>.13</td>
<td>1</td>
<td>1,000</td>
<td>.30</td>
</tr>
<tr>
<td>2</td>
<td>342</td>
<td>.15</td>
<td>31</td>
<td>143</td>
<td>.13</td>
<td>2</td>
<td>530</td>
<td>.25</td>
</tr>
<tr>
<td>3</td>
<td>342</td>
<td>.16</td>
<td>32</td>
<td>143</td>
<td>.13</td>
<td>3</td>
<td>530</td>
<td>.25</td>
</tr>
<tr>
<td>4</td>
<td>322</td>
<td>.21</td>
<td>33</td>
<td>143</td>
<td>.13</td>
<td>4</td>
<td>530</td>
<td>.25</td>
</tr>
<tr>
<td>5</td>
<td>300</td>
<td>.12</td>
<td>34</td>
<td>143</td>
<td>.13</td>
<td>5</td>
<td>530</td>
<td>.25</td>
</tr>
<tr>
<td>6</td>
<td>273</td>
<td>.08</td>
<td>35</td>
<td>103</td>
<td>.20</td>
<td>6</td>
<td>524</td>
<td>.20</td>
</tr>
<tr>
<td>7</td>
<td>273</td>
<td>.08</td>
<td>36</td>
<td>100</td>
<td>.0885</td>
<td>7</td>
<td>524</td>
<td>.20</td>
</tr>
<tr>
<td>8</td>
<td>273</td>
<td>.08</td>
<td>37</td>
<td>100</td>
<td>.09</td>
<td>8</td>
<td>346</td>
<td>.18</td>
</tr>
<tr>
<td>9</td>
<td>273</td>
<td>.08</td>
<td>38</td>
<td>100</td>
<td>.09</td>
<td>9</td>
<td>342</td>
<td>.15</td>
</tr>
<tr>
<td>10</td>
<td>265</td>
<td>.10</td>
<td>39</td>
<td>100</td>
<td>.09</td>
<td>10</td>
<td>342</td>
<td>.16</td>
</tr>
<tr>
<td>11</td>
<td>265</td>
<td>.10</td>
<td>40</td>
<td>100</td>
<td>.09</td>
<td>11</td>
<td>322</td>
<td>.21</td>
</tr>
<tr>
<td>12</td>
<td>265</td>
<td>.10</td>
<td>41</td>
<td>50</td>
<td>.08</td>
<td>12</td>
<td>300</td>
<td>.12</td>
</tr>
<tr>
<td>13</td>
<td>265</td>
<td>.10</td>
<td>42</td>
<td>50</td>
<td>.08</td>
<td>13</td>
<td>257</td>
<td>.12</td>
</tr>
<tr>
<td>14</td>
<td>265</td>
<td>.10</td>
<td>43</td>
<td>50</td>
<td>.08</td>
<td>14</td>
<td>257</td>
<td>.12</td>
</tr>
<tr>
<td>15</td>
<td>265</td>
<td>.10</td>
<td>44</td>
<td>50</td>
<td>.08</td>
<td>15</td>
<td>257</td>
<td>.12</td>
</tr>
<tr>
<td>16</td>
<td>265</td>
<td>.10</td>
<td>45</td>
<td>32</td>
<td>.06</td>
<td>16</td>
<td>257</td>
<td>.12</td>
</tr>
<tr>
<td>17</td>
<td>265</td>
<td>.06</td>
<td>46</td>
<td>32</td>
<td>.06</td>
<td>17</td>
<td>250</td>
<td>.09</td>
</tr>
<tr>
<td>18</td>
<td>262</td>
<td>.06</td>
<td>47</td>
<td>32</td>
<td>.06</td>
<td>18</td>
<td>250</td>
<td>.13</td>
</tr>
<tr>
<td>19</td>
<td>262</td>
<td>.06</td>
<td>48</td>
<td>16</td>
<td>.06</td>
<td>19</td>
<td>250</td>
<td>.13</td>
</tr>
<tr>
<td>20</td>
<td>262</td>
<td>.06</td>
<td>49</td>
<td>16</td>
<td>.06</td>
<td>20</td>
<td>250</td>
<td>.13</td>
</tr>
<tr>
<td>21</td>
<td>262</td>
<td>.06</td>
<td></td>
<td></td>
<td></td>
<td>21</td>
<td>250</td>
<td>.13</td>
</tr>
<tr>
<td>22</td>
<td>257</td>
<td>.12</td>
<td></td>
<td></td>
<td></td>
<td>22</td>
<td>247</td>
<td>.20</td>
</tr>
<tr>
<td>23</td>
<td>257</td>
<td>.12</td>
<td></td>
<td></td>
<td></td>
<td>23</td>
<td>200</td>
<td>.065</td>
</tr>
<tr>
<td>24</td>
<td>257</td>
<td>.12</td>
<td></td>
<td></td>
<td></td>
<td>24</td>
<td>192</td>
<td>.09</td>
</tr>
<tr>
<td>25</td>
<td>257</td>
<td>.12</td>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td>158</td>
<td>.60</td>
</tr>
<tr>
<td>26</td>
<td>200</td>
<td>.065</td>
<td></td>
<td></td>
<td></td>
<td>26</td>
<td>120</td>
<td>.08</td>
</tr>
<tr>
<td>27</td>
<td>143</td>
<td>.13</td>
<td></td>
<td></td>
<td></td>
<td>27</td>
<td>70</td>
<td>.08</td>
</tr>
<tr>
<td>28</td>
<td>143</td>
<td>.13</td>
<td></td>
<td></td>
<td></td>
<td>28</td>
<td>50</td>
<td>.08</td>
</tr>
<tr>
<td>29</td>
<td>143</td>
<td>.13</td>
<td></td>
<td></td>
<td></td>
<td>29</td>
<td>50</td>
<td>.08</td>
</tr>
</tbody>
</table>

<sup>α</sup> Forced Outage Rate
the same table provide the same information for the generating system used in cases 1B, 2B, 3B, 4B, and 5B. The generating systems for cases 6A through 11B are similarly described in Tables 3.3 to 3.5. The average forced outage rate of the generating system of case nB is intentionally made greater than for the corresponding case nA, so the system reliabilities in cases nB's are lower than the corresponding cases nA's.

The load data used are representative of a broad spectrum of United States utilities with peak load ranging from 1393 MW to 7902 MW and load factors from 55.8 percent to 68.2 percent. These load data are typical of small to large utilities and small power pools. Since preliminary results of the sensitivity analysis showed no correlation between the magnitude of load demand and the accuracy of the cumulant method, utilities and big power pools that have very large peak load demand are not considered here.

3.4.3 Sensitivity Analysis Results

The results of the sensitivity analysis are presented in Figures 3.8 to 3.15 and D.1 to D.40 in Appendix D and are summarized in Tables 3.6 and 3.8. The results are analyzed in the next section. The terms "load probability curve" or "original load probability curve" are used here to denote the graphic representation of the load probability function, L(x), and the term "final equivalent load probability curve" represents the the last equivalent load probability function, EL_N(x). L(x) is defined by Equation (3.1) and EL_N(x) by
<table>
<thead>
<tr>
<th>UNIT NUMBER</th>
<th>CASE 6A (MW)</th>
<th>CASE 6B (MW)</th>
<th>CASE 7A (MW)</th>
<th>CASE 7B (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000. .05</td>
<td>1200. .30</td>
<td>800. .05</td>
<td>1200. .30</td>
</tr>
<tr>
<td>2</td>
<td>500. .20</td>
<td>1000. .30</td>
<td>500. .20</td>
<td>800. .25</td>
</tr>
<tr>
<td>3</td>
<td>400. .20</td>
<td>800. .20</td>
<td>400. .20</td>
<td>600. .25</td>
</tr>
<tr>
<td>4</td>
<td>400. .20</td>
<td>600. .20</td>
<td>400. .20</td>
<td>400. .20</td>
</tr>
<tr>
<td>5</td>
<td>400. .15</td>
<td>600. .20</td>
<td>200. .15</td>
<td>200. .20</td>
</tr>
<tr>
<td>6</td>
<td>400. .15</td>
<td>600. .20</td>
<td>200. .15</td>
<td>200. .20</td>
</tr>
<tr>
<td>7</td>
<td>400. .15</td>
<td>600. .20</td>
<td>200. .15</td>
<td>200. .20</td>
</tr>
<tr>
<td>8</td>
<td>300. .15</td>
<td>400. .20</td>
<td>200. .15</td>
<td>200. .20</td>
</tr>
<tr>
<td>9</td>
<td>300. .15</td>
<td>200. .15</td>
<td>100. .15</td>
<td>160. .20</td>
</tr>
<tr>
<td>10</td>
<td>300. .15</td>
<td>200. .15</td>
<td>100. .15</td>
<td>100. .15</td>
</tr>
<tr>
<td>11</td>
<td>200. .15</td>
<td>100. .10</td>
<td>100. .15</td>
<td>100. .15</td>
</tr>
<tr>
<td>12</td>
<td>200. .15</td>
<td>100. .10</td>
<td>100. .15</td>
<td>100. .15</td>
</tr>
<tr>
<td>13</td>
<td>200. .15</td>
<td>100. .10</td>
<td>100. .15</td>
<td>100. .15</td>
</tr>
<tr>
<td>14</td>
<td>200. .15</td>
<td>100. .10</td>
<td>100. .15</td>
<td>60. .10</td>
</tr>
<tr>
<td>15</td>
<td>200. .15</td>
<td>100. .10</td>
<td>100. .15</td>
<td>60. .10</td>
</tr>
<tr>
<td>16</td>
<td>200. .10</td>
<td>50. .08</td>
<td>100. .10</td>
<td>60. .10</td>
</tr>
<tr>
<td>17</td>
<td>200. .10</td>
<td>50. .08</td>
<td>100. .10</td>
<td>60. .10</td>
</tr>
<tr>
<td>18</td>
<td>200. .10</td>
<td>100. .08</td>
<td>100. .10</td>
<td>60. .10</td>
</tr>
<tr>
<td>19</td>
<td>200. .10</td>
<td>100. .08</td>
<td>100. .10</td>
<td>60. .10</td>
</tr>
<tr>
<td>20</td>
<td>200. .10</td>
<td>100. .08</td>
<td>100. .10</td>
<td>60. .10</td>
</tr>
<tr>
<td>21</td>
<td>100. .10</td>
<td>40. .10</td>
<td>40. .10</td>
<td>40. .10</td>
</tr>
<tr>
<td>22</td>
<td>100. .10</td>
<td>40. .10</td>
<td>40. .10</td>
<td>40. .10</td>
</tr>
<tr>
<td>23</td>
<td>100. .10</td>
<td>40. .10</td>
<td>40. .10</td>
<td>40. .10</td>
</tr>
<tr>
<td>24</td>
<td>100. .10</td>
<td>40. .10</td>
<td>40. .10</td>
<td>40. .10</td>
</tr>
<tr>
<td>25</td>
<td>100. .10</td>
<td>40. .10</td>
<td>40. .10</td>
<td>40. .10</td>
</tr>
<tr>
<td>26</td>
<td>100. .05</td>
<td>40. .05</td>
<td>40. .05</td>
<td>40. .05</td>
</tr>
<tr>
<td>27</td>
<td>50. .05</td>
<td>40. .05</td>
<td>40. .05</td>
<td>40. .05</td>
</tr>
<tr>
<td>28</td>
<td>50. .05</td>
<td>40. .05</td>
<td>40. .05</td>
<td>40. .05</td>
</tr>
<tr>
<td>29</td>
<td>50. .05</td>
<td>40. .05</td>
<td>40. .05</td>
<td>40. .05</td>
</tr>
<tr>
<td>30</td>
<td>50. .05</td>
<td>40. .05</td>
<td>40. .05</td>
<td>40. .05</td>
</tr>
<tr>
<td>31</td>
<td>50. .05</td>
<td>20. .05</td>
<td>20. .05</td>
<td>20. .05</td>
</tr>
<tr>
<td>32</td>
<td>50. .05</td>
<td>20. .05</td>
<td>20. .05</td>
<td>20. .05</td>
</tr>
<tr>
<td>33</td>
<td>50. .05</td>
<td>20. .05</td>
<td>20. .05</td>
<td>20. .05</td>
</tr>
<tr>
<td>34</td>
<td>50. .05</td>
<td>20. .05</td>
<td>20. .05</td>
<td>20. .05</td>
</tr>
<tr>
<td>35</td>
<td>50. .05</td>
<td>20. .05</td>
<td>20. .05</td>
<td>20. .05</td>
</tr>
<tr>
<td>36</td>
<td>30. .03</td>
<td>20. .03</td>
<td>20. .03</td>
<td>20. .03</td>
</tr>
<tr>
<td>37</td>
<td>30. .03</td>
<td>20. .03</td>
<td>20. .03</td>
<td>20. .03</td>
</tr>
<tr>
<td>38</td>
<td>30. .03</td>
<td>20. .03</td>
<td>20. .03</td>
<td>20. .03</td>
</tr>
<tr>
<td>39</td>
<td>60. .03</td>
<td>20. .03</td>
<td>20. .03</td>
<td>20. .03</td>
</tr>
</tbody>
</table>

*Forced Outage Rate*
TABLE 3.4 Generating Unit Characteristics for Cases 8A, 8B, 9A, and 9B

<table>
<thead>
<tr>
<th>UNIT NUMBER</th>
<th>CAPACITY (MW)</th>
<th>FORα</th>
<th>CAPACITY (MW)</th>
<th>FORα</th>
<th>CAPACITY (MW)</th>
<th>FORα</th>
<th>CAPACITY (MW)</th>
<th>FORα</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>800. .05</td>
<td></td>
<td>1000. .30</td>
<td></td>
<td>500. .05</td>
<td></td>
<td>800. .30</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>500. .20</td>
<td></td>
<td>700. .25</td>
<td></td>
<td>300. .20</td>
<td></td>
<td>500. .25</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>500. .20</td>
<td></td>
<td>400. .25</td>
<td></td>
<td>200. .20</td>
<td></td>
<td>300. .25</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>200. .20</td>
<td></td>
<td>200. .20</td>
<td></td>
<td>200. .20</td>
<td></td>
<td>200. .20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>200. .15</td>
<td></td>
<td>200. .20</td>
<td></td>
<td>200. .15</td>
<td></td>
<td>200. .20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>200. .15</td>
<td></td>
<td>200. .20</td>
<td></td>
<td>200. .15</td>
<td></td>
<td>200. .20</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>120. .15</td>
<td></td>
<td>200. .20</td>
<td></td>
<td>200. .15</td>
<td></td>
<td>150. .20</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>100. .15</td>
<td></td>
<td>180. .20</td>
<td></td>
<td>100. .15</td>
<td></td>
<td>100. .15</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>100. .15</td>
<td></td>
<td>100. .20</td>
<td></td>
<td>100. .15</td>
<td></td>
<td>100. .15</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>100. .15</td>
<td></td>
<td>100. .20</td>
<td></td>
<td>100. .15</td>
<td></td>
<td>100. .15</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>100. .15</td>
<td></td>
<td>100. .20</td>
<td></td>
<td>100. .15</td>
<td></td>
<td>100. .15</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>100. .15</td>
<td></td>
<td>100. .20</td>
<td></td>
<td>100. .15</td>
<td></td>
<td>100. .15</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>100. .15</td>
<td></td>
<td>100. .20</td>
<td></td>
<td>100. .15</td>
<td></td>
<td>100. .15</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>100. .15</td>
<td></td>
<td>100. .15</td>
<td></td>
<td>50. .15</td>
<td></td>
<td>100. .15</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50. .15</td>
<td></td>
<td>100. .15</td>
<td></td>
<td>50. .15</td>
<td></td>
<td>50. .15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>50. .10</td>
<td></td>
<td>100. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
<td>50. .10</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
<td>30. .05</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
<td>20. .05</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>20. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>20. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>20. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>20. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
<td>10. .03</td>
<td></td>
</tr>
</tbody>
</table>

αForced Outage Rate
<table>
<thead>
<tr>
<th>UNIT NUMBER</th>
<th>CASE 10A CENTER CAPACITY (MW)</th>
<th>CASE 10B CENTER CAPACITY (MW)</th>
<th>CASE 11A CENTER CAPACITY (MW)</th>
<th>CASE 11B CENTER CAPACITY (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>400.05</td>
<td>600.30</td>
<td>300.05</td>
<td>400.30</td>
</tr>
<tr>
<td>2</td>
<td>250.20</td>
<td>400.25</td>
<td>200.20</td>
<td>300.25</td>
</tr>
<tr>
<td>3</td>
<td>200.20</td>
<td>300.25</td>
<td>100.20</td>
<td>200.20</td>
</tr>
<tr>
<td>4</td>
<td>200.20</td>
<td>200.25</td>
<td>100.25</td>
<td>100.20</td>
</tr>
<tr>
<td>5</td>
<td>100.15</td>
<td>100.20</td>
<td>50.15</td>
<td>50.20</td>
</tr>
<tr>
<td>6</td>
<td>100.15</td>
<td>100.20</td>
<td>50.15</td>
<td>50.20</td>
</tr>
<tr>
<td>7</td>
<td>100.15</td>
<td>100.20</td>
<td>50.15</td>
<td>30.20</td>
</tr>
<tr>
<td>8</td>
<td>100.15</td>
<td>100.20</td>
<td>30.15</td>
<td>30.20</td>
</tr>
<tr>
<td>9</td>
<td>100.15</td>
<td>100.20</td>
<td>30.15</td>
<td>30.20</td>
</tr>
<tr>
<td>10</td>
<td>50.15</td>
<td>50.20</td>
<td>30.15</td>
<td>30.20</td>
</tr>
<tr>
<td>11</td>
<td>50.15</td>
<td>50.15</td>
<td>30.15</td>
<td>20.15</td>
</tr>
<tr>
<td>12</td>
<td>50.15</td>
<td>50.15</td>
<td>30.15</td>
<td>20.15</td>
</tr>
<tr>
<td>13</td>
<td>50.15</td>
<td>50.15</td>
<td>30.15</td>
<td>10.15</td>
</tr>
<tr>
<td>14</td>
<td>50.15</td>
<td>30.15</td>
<td>20.15</td>
<td>10.15</td>
</tr>
<tr>
<td>15</td>
<td>50.15</td>
<td>30.15</td>
<td>20.10</td>
<td>10.10</td>
</tr>
<tr>
<td>16</td>
<td>50.10</td>
<td>30.10</td>
<td>20.10</td>
<td>10.10</td>
</tr>
<tr>
<td>17</td>
<td>50.10</td>
<td>30.10</td>
<td>20.10</td>
<td>10.10</td>
</tr>
<tr>
<td>18</td>
<td>50.10</td>
<td>30.10</td>
<td>20.10</td>
<td>10.10</td>
</tr>
<tr>
<td>19</td>
<td>50.10</td>
<td>30.10</td>
<td>20.10</td>
<td>10.10</td>
</tr>
<tr>
<td>20</td>
<td>50.10</td>
<td>10.10</td>
<td>10.10</td>
<td>10.10</td>
</tr>
<tr>
<td>21</td>
<td>20.10</td>
<td>10.10</td>
<td>10.10</td>
<td>10.10</td>
</tr>
<tr>
<td>22</td>
<td>20.10</td>
<td>10.10</td>
<td>10.10</td>
<td>10.10</td>
</tr>
<tr>
<td>23</td>
<td>20.10</td>
<td>10.10</td>
<td>10.10</td>
<td>10.10</td>
</tr>
<tr>
<td>24</td>
<td>20.10</td>
<td>10.10</td>
<td>10.10</td>
<td>10.10</td>
</tr>
<tr>
<td>25</td>
<td>20.10</td>
<td>10.10</td>
<td>10.10</td>
<td>10.10</td>
</tr>
<tr>
<td>26</td>
<td>20.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>27</td>
<td>20.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>28</td>
<td>20.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>29</td>
<td>20.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>30</td>
<td>20.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>31</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>32</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>33</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>34</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>35</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>36</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
<td>10.05</td>
</tr>
<tr>
<td>37</td>
<td>10.03</td>
<td>10.03</td>
<td>10.03</td>
<td>10.03</td>
</tr>
<tr>
<td>38</td>
<td>10.03</td>
<td>10.03</td>
<td>10.03</td>
<td>10.03</td>
</tr>
<tr>
<td>39</td>
<td>20.03</td>
<td>20.03</td>
<td>20.03</td>
<td>20.03</td>
</tr>
</tbody>
</table>

\(^a\) Forced Outage Rate
Figure 3.8 Load Probability Curves for Case 1A, for the 4-Cumulant Approximation and the Piecewise Linear Method. Both the Original Load Probability Curves and the Final Equivalent Load Probability Curves are Shown.
Figure 3.9 Differences of Original Load Probability Curves for Cases 1A and 1B, between each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure 3.10 Differences of Final Equivalent Load Probability Curves for Case 1A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method.
Figure 3.11  Differences of Final Equivalent Load Probability Curves for Case 1B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure 3.12  Percentage Differences of Original Load Probability Curves for Cases 1A and 1B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure 3.13  Percentage Differences of Final Equivalent Load Probability Curves in Case 1A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure 3.14 Percentage Differences of Final Equivalent Load Probability Curves in Case 1B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure 3.15  Percentage Differences in Plant Energy Production for Cases 1A and 1B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Table 3.6 LOLP as Calculated by the Piecewise Linear and Cumulant Methods

<table>
<thead>
<tr>
<th>CASE NUMBER</th>
<th>PIECEWISE LOLP (x.0001)</th>
<th>4 CUMULANTS LOLP DIFF. b</th>
<th>5 CUMULANTS LOLP DIFF. b</th>
<th>8 CUMULANTS LOLP DIFF. b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>.9</td>
<td>1.5</td>
<td>66.6</td>
<td>1.0</td>
</tr>
<tr>
<td>3A</td>
<td>0. a</td>
<td>0. a</td>
<td>-</td>
<td>0. a</td>
</tr>
<tr>
<td>4A</td>
<td>34.</td>
<td>55.</td>
<td>61.7</td>
<td>64.</td>
</tr>
<tr>
<td>5A</td>
<td>0. a</td>
<td>0. a</td>
<td>-</td>
<td>0. a</td>
</tr>
<tr>
<td>6A</td>
<td>11.</td>
<td>7.6</td>
<td>-30.1</td>
<td>8.</td>
</tr>
<tr>
<td>7A</td>
<td>167.7</td>
<td>174.7</td>
<td>4.2</td>
<td>179.</td>
</tr>
<tr>
<td>8A</td>
<td>107.</td>
<td>102.</td>
<td>-4.8</td>
<td>111.</td>
</tr>
<tr>
<td>9A</td>
<td>39.</td>
<td>39.</td>
<td>-</td>
<td>36.</td>
</tr>
<tr>
<td>10A</td>
<td>60.</td>
<td>57.</td>
<td>-5.</td>
<td>61.</td>
</tr>
<tr>
<td>11A</td>
<td>542.</td>
<td>570.</td>
<td>5.2</td>
<td>564.</td>
</tr>
<tr>
<td>1B</td>
<td>151.2</td>
<td>144.2</td>
<td>-4.63</td>
<td>152.7</td>
</tr>
<tr>
<td>2B</td>
<td>127.2</td>
<td>128.2</td>
<td>-7.9</td>
<td>128.2</td>
</tr>
<tr>
<td>3B</td>
<td>3.1</td>
<td>2.1</td>
<td>-32.26</td>
<td>2.5</td>
</tr>
<tr>
<td>4B</td>
<td>443.2</td>
<td>437.8</td>
<td>-1.22</td>
<td>454.1</td>
</tr>
<tr>
<td>5B</td>
<td>29.6</td>
<td>27.5</td>
<td>-7.09</td>
<td>29.3</td>
</tr>
<tr>
<td>6B</td>
<td>452.0</td>
<td>448.2</td>
<td>-0.84</td>
<td>454.7</td>
</tr>
<tr>
<td>7B</td>
<td>1572.9</td>
<td>1560.0</td>
<td>-8.2</td>
<td>1544.7</td>
</tr>
<tr>
<td>8B</td>
<td>1014.4</td>
<td>995.9</td>
<td>-1.82</td>
<td>995.9</td>
</tr>
<tr>
<td>9B</td>
<td>586.9</td>
<td>568.6</td>
<td>-3.12</td>
<td>592.3</td>
</tr>
<tr>
<td>10B</td>
<td>902.5</td>
<td>879.8</td>
<td>-2.52</td>
<td>885.1</td>
</tr>
<tr>
<td>11B</td>
<td>3181.5</td>
<td>3185.3</td>
<td>.13</td>
<td>3173.3</td>
</tr>
</tbody>
</table>

a Zero means values < 10^-4

b The percentage differences are calculated with respect to the LOLP values calculated by the piecewise linear method.
<table>
<thead>
<tr>
<th>CASE NUMBER</th>
<th>PIECEWISE ENERGY (GWH)</th>
<th>4 CUMULANTS ENERGY DIFF. (%)</th>
<th>5 CUMULANTS ENERGY DIFF. (%)</th>
<th>8 CUMULANTS ENERGY DIFF. (%)</th>
<th>DERATE METHOD ENERGY DIFF. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>38629.</td>
<td>38621. -0.02</td>
<td>38624. -0.01</td>
<td>38624. -0.01</td>
<td>38557. -0.02</td>
</tr>
<tr>
<td>2A</td>
<td>10304.</td>
<td>10303. -0.01</td>
<td>10302. -0.02</td>
<td>10302. -0.02</td>
<td>10591. 2.89</td>
</tr>
<tr>
<td>3A</td>
<td>8370.</td>
<td>8370. 0.00</td>
<td>8370. 0.00</td>
<td>8370. 0.00</td>
<td>9125. 9.02</td>
</tr>
<tr>
<td>4A</td>
<td>10852.</td>
<td>10848. -0.04</td>
<td>10850. -0.02</td>
<td>10851. -0.01</td>
<td>10832. 0.30</td>
</tr>
<tr>
<td>5A</td>
<td>9102.</td>
<td>9101. -0.01</td>
<td>9101. -0.01</td>
<td>9102. 0.00</td>
<td>9411. 3.42</td>
</tr>
<tr>
<td>6A</td>
<td>32112.</td>
<td>32080. -0.10</td>
<td>32081. -0.10</td>
<td>32077. -0.11</td>
<td>32059. 0.50</td>
</tr>
<tr>
<td>7A</td>
<td>23899.</td>
<td>23866. -0.14</td>
<td>23863. -0.15</td>
<td>23863. -0.15</td>
<td>23880. 2.83</td>
</tr>
<tr>
<td>8A</td>
<td>19077.</td>
<td>18994. -0.44</td>
<td>18991. -0.45</td>
<td>18986. -0.48</td>
<td>19065. 1.60</td>
</tr>
<tr>
<td>9A</td>
<td>14306.</td>
<td>14284. -0.15</td>
<td>14284. -0.15</td>
<td>14284. -0.15</td>
<td>14280. 0.80</td>
</tr>
<tr>
<td>10A</td>
<td>10495.</td>
<td>10468. -0.26</td>
<td>10467. -0.27</td>
<td>10467. -0.27</td>
<td>10481. 1.55</td>
</tr>
<tr>
<td>11A</td>
<td>8044.</td>
<td>8038. -0.07</td>
<td>8037. -0.08</td>
<td>8042. -0.03</td>
<td>7894. 5.53</td>
</tr>
<tr>
<td>1B</td>
<td>38564.</td>
<td>38522. -0.11</td>
<td>38525. -0.10</td>
<td>38525. -0.10</td>
<td>38610. 0.05</td>
</tr>
<tr>
<td>2B</td>
<td>10293.</td>
<td>10286. -0.07</td>
<td>10285. -0.08</td>
<td>10285. -0.08</td>
<td>10406. 0.99</td>
</tr>
<tr>
<td>3B</td>
<td>8370.</td>
<td>8357. -0.15</td>
<td>8358. -0.14</td>
<td>8357. -0.15</td>
<td>8507. 1.64</td>
</tr>
<tr>
<td>4B</td>
<td>10800.</td>
<td>10790. -0.09</td>
<td>10788. -0.11</td>
<td>10793. -0.06</td>
<td>10878. 0.24</td>
</tr>
<tr>
<td>5B</td>
<td>9100.</td>
<td>9089. -0.12</td>
<td>9089. -0.12</td>
<td>9089. -0.12</td>
<td>9156. 0.59</td>
</tr>
<tr>
<td>6B</td>
<td>31899.</td>
<td>31799. -0.31</td>
<td>31792. -0.36</td>
<td>31802. -0.30</td>
<td>32208. 0.30</td>
</tr>
<tr>
<td>7B</td>
<td>23222.</td>
<td>23100. -0.53</td>
<td>23090. -0.57</td>
<td>23094. -0.55</td>
<td>23936. 0.15</td>
</tr>
<tr>
<td>8B</td>
<td>18764.</td>
<td>18650. -0.61</td>
<td>18632. -0.70</td>
<td>18650. -0.61</td>
<td>19216. 0.73</td>
</tr>
<tr>
<td>9B</td>
<td>14166.</td>
<td>14080. -0.61</td>
<td>14070. -0.68</td>
<td>14090. -0.54</td>
<td>14328. 0.15</td>
</tr>
<tr>
<td>10B</td>
<td>10322.</td>
<td>10235. -0.84</td>
<td>10228. -0.91</td>
<td>10239. -0.80</td>
<td>10535. 0.38</td>
</tr>
<tr>
<td>11B</td>
<td>7480.</td>
<td>7432. -0.64</td>
<td>7430. -0.66</td>
<td>7425. -0.74</td>
<td>8076. 0.40</td>
</tr>
</tbody>
</table>

The percentage differences are calculated with respect to the energy values calculated by the piecewise linear method.
Table 3.8 Computing Time for LOLP and Energy Calculations\(^a\)
(AMDahl 470V6, Cpu Time in System Seconds)

<table>
<thead>
<tr>
<th>CASE NUMBER</th>
<th>SYSTEM LOLP PIECEWISE CUMULANTS(^b)</th>
<th>TOTAL SYSTEM ENERGY PIECEWISE CUMULANTS(^b) DERATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>.342 .008</td>
<td>.353 .039 .0027</td>
</tr>
<tr>
<td>2A</td>
<td>.363 .008</td>
<td>.375 .039 .0028</td>
</tr>
<tr>
<td>3A</td>
<td>.378 .008</td>
<td>.390 .039 .0030</td>
</tr>
<tr>
<td>4A</td>
<td>.358 .008</td>
<td>.369 .039 .0027</td>
</tr>
<tr>
<td>5A</td>
<td>.374 .008</td>
<td>.386 .039 .0030</td>
</tr>
<tr>
<td>6A</td>
<td>.307 .007</td>
<td>.316 .032 .0025</td>
</tr>
<tr>
<td>7A</td>
<td>.330 .007</td>
<td>.337 .032 .0023</td>
</tr>
<tr>
<td>8A</td>
<td>.352 .007</td>
<td>.361 .032 .0023</td>
</tr>
<tr>
<td>9A</td>
<td>.328 .007</td>
<td>.337 .032 .0023</td>
</tr>
<tr>
<td>10A</td>
<td>.332 .007</td>
<td>.340 .032 .0024</td>
</tr>
<tr>
<td>11A</td>
<td>.327 .007</td>
<td>.336 .032 .0032</td>
</tr>
<tr>
<td>1B</td>
<td>.237 .005</td>
<td>.244 .024 .0019</td>
</tr>
<tr>
<td>2B</td>
<td>.257 .005</td>
<td>.267 .024 .0020</td>
</tr>
<tr>
<td>3B</td>
<td>.278 .005</td>
<td>.286 .024 .0021</td>
</tr>
<tr>
<td>4B</td>
<td>.246 .005</td>
<td>.253 .024 .0019</td>
</tr>
<tr>
<td>5B</td>
<td>.254 .005</td>
<td>.267 .024 .0020</td>
</tr>
<tr>
<td>6B</td>
<td>.178 .004</td>
<td>.183 .016 .0015</td>
</tr>
<tr>
<td>7B</td>
<td>.186 .004</td>
<td>.191 .016 .0015</td>
</tr>
<tr>
<td>8B</td>
<td>.183 .004</td>
<td>.188 .015 .0015</td>
</tr>
<tr>
<td>9B</td>
<td>.181 .004</td>
<td>.186 .016 .0015</td>
</tr>
<tr>
<td>10B</td>
<td>.186 .004</td>
<td>.191 .016 .0015</td>
</tr>
<tr>
<td>11B</td>
<td>.186 .004</td>
<td>.191 .016 .0014</td>
</tr>
</tbody>
</table>

\(^a\) One simulation with one block representation

\(^b\) 4-Cumulant series
Equation (3.22) when the unit number $n$ is replace by the generating system's total number of units, $N$. It is noted that since the same load data are used for case studies $nA$ and $nB$ characterized by the same $n$, their load probability curves, $L(x)$, are identical, e.g. Case 1A has the same $L(x)$ as Case 1B. However, the final load curves for $nA$ and $nB$ are different because different generating units are convolved with the $L(x)$.

The load probability curve $L(x)$ for case studies 1A and 1B is shown in Figure 3.8 for both the piecewise linear polynomial and the 4-cumulant series representations. Both the piecewise linear and derate methods used the same piecewise linear polynomial to represent $L(x)$ (cf. Section 3.3.2). The final equivalent load probability curve $E_L(x)$ for case 1A is also depicted in Figure 3.8 for the piecewise and 4-cumulant approximation. Since in the derate method no equivalent load curves are calculated, the cumulant representation of the equivalent load probability curves are compared only with those of the piecewise linear method. It should be observed that the cumulant curve oscillates around the curve calculated by the piecewise method.

The oscillatory behavior of the cumulant curves shown in Figure 3.8 is more evident in Figures 3.9, 3.10, and 3.11, which show the differences between the load curves from the two methods. The differences are found by substraction the values of $L(x)$ (or $E_L(x)$) calculated by the piecewise linear method from those calculated by the cumulant method. The solid line in Figure 3.8 shows the load probability differences between the piecewise linear method and the 5-cumulant series approximation, as a function of the system load, for case studies.
1A and 1B. The dashed and dotted lines refer to the differences between the linear and the 4-, and 8-cumulant series approximations respectively. Similarly, Figures 3.10 and 3.11 show the differences between the final equivalent load probability curves \((EL_N(x))\) calculated by the two methods. Figure 3.9 depicts differences in \(L(x)\) for both case studies 1A and for 1B, since in both cases the same load data are used. However, the final \(EL_N(x)\) for case studies 1A (Figure 3.10) and 1B (Figure 3.11) are different since the generating systems are different. It should be noted that the magnitude of the differences decreases from Figure 3.9 to 3.11 indicating that the accuracy of the cumulant method increases with the convolutions of the generating units.

Figures 3.12, 3.13, and 3.14 show the same results as Figures 3.9, 3.10, and 3.11, respectively, except in the percentage differences of the cumulant method relative to the piecewise method. The dashed, solid, and dotted lines refer to percentage differences between the 4-, 5-, and 8-cumulant series approximation, respectively, and the piecewise linear method. The differences are shown as a function of the system load. Figures D.1 to D.30 in Appendix D refer to case studies 2A to 11B and are similar to Figures 3.12 to 3.14. The large relative error, at the maximum system capacity MW level, should be noted.

Figures 3.15 and D.31 to D.40 in Appendix D show the percentage differences in plant energy calculations between the piecewise method and the

a) 4-cumulant series
b) 5-cumulant series
c) 8-cumulant series
d) de rate method

for all case studies. The differences are shown for each plant. The word "plant" is used here as synonymous to "generating unit".

Table 3.6 provides the LOLP results for the 22 case studies. Column 2 shows the LOLP values as calculated by the piecewise linear method. Columns 3, 5, and 7 show the LOLP results from the 4-, 5-, and 8-cumulant series approximations. The LOLP percentage difference of the cumulant approximations listed in columns 4, 6, and 8 are calculated with respect to the LOLP values of the piecewise linear method. It should be noted that cases nA have smaller LOLP values than nB, because the generating system of cases nA have more and more reliable generating units (see Tables 3.2 to 3.5).

Table 3.7 gives the total energy generation, in GWH, for each of the 22 case studies. The structure of this Table is identical with that of Table 3.7 except that two more columns are added. These columns show the energy values as calculated by the de rate method, and their percentage energy differences from the piecewise method.

Finally, Table 3.8 shows the computing time requirements for LOLP and plant energy calculations. LOLP time requirements are provided for the piecewise linear and 4-cumulant series approximation, but not for the de rate method, since the latter is not suitable for LOLP calculations (see Section 3.3.2). The time required for the calculation of energy production by all the generating units in each of the case studies is shown for each of the three methods considered. The time requirements
for LOLP and energy calculations by the 5- and 8-cumulant series approximations are not shown, for reasons that will be apparent in the next section.

3.4.4 Analysis of the Results

From Figures 3.9 to 3.15, D.1 to D.40, and Tables 3.6 and 3.7 it is evident that the accuracy of the cumulant method does not increase with the increase in the number of cumulants from 4 to 5 and 8. However, since there is an increase in the calculations and more core memory is required when more than 4 cumulants are used, the 4-cumulant series is recommended for use in probabilistic simulation.

The error in the load probability curve represented with the cumulant method (Figures 3.9, 3.10, and 3.11) decreased from approximately ±0.020 in the original load curve to ±0.010 in the final curve of Case 1A and ±0.003 in Case 1B. Similar behavior of the relative error is evidenced in Figures 3.12 to 3.14, and D.1 to D.30 for all case studies.

These trends are in agreement with the previous observation [30], that the accuracy of the cumulant method increases (a) with the increase in the number of generating units convolved and (b) with the increase in the forced outage rate of the units convolved. The relative error of the cumulant representation of the load probability curve increases as the probability values become smaller. When the load is approximately equal to the total system capacity, the error becomes greater than 50 percent. The LOLP errors of the cumulant method are shown in Table 3.6. These errors are in the
order of 0.1 and 1 day per year which are intolerable to system planners of electric utilities [63]. Thus, we conclude that the cumulant method (in its present form) is not suitable for LOLP calculations.

The relative error in the calculation of plant energy generation (Figures 3.15, D.31 to D.40) with the cumulant method is less than ±5 percent for most of the plants, but increases to more than ±50 percent for the last few plants in the loading order. The reason for this inaccuracy is the small amount of energy generated by the peaking units. However, the total energy calculated by the cumulant method has a very small error as shown in Table 3.7 (the maximum difference is within 1%). The large relative error in energy generation of very small units is insignificant in most expansion planning studies, although the system reliability is significantly affected by the existence of small units.

The relative error of the derate method in plant energy calculations follows the same trends as the cumulant method error. However, the magnitude of the derate method error is 10 to 20 percent larger than for the cumulant method and increases to over 50 percent for peaking units much faster than in the cumulant method. The relative error in total energy generation by the derate method is over 1.0 percent.

The accuracy of the unserved energy calculations by both the cumulant and derate methods is the same as the accuracy of total energy calculations, since the unserved energy is calculated by subtracting the total energy generation from the energy demand.
It should be noted that the higher accuracy of energy calculations by the cumulant method as compared to LOLP calculations is due to the averaging effect that is introduced in the energy calculation, at the numerical integration of the equivalent load probability function. The oscillation of the cumulant curve around the piecewise curve is shown amplified in the schematic diagram of Figure 3.16. Since the cumulant curve oscillates around the piecewise curve, integration through Equation (3.55) gives the area \( X_1^{CDX_2} \) which is a better approximation to \( X_1^{ABX_2} \) calculated by the piecewise method than \( X_1^{CFDX_2} \). Since LOLP is calculated from \( X_1^{CFDX_2} \), its values are not as accurate as those of expected energy production.

Finally, Table 3.8 compares the CPU time in second, for the piecewise, cumulant, and derate methods. It is seen that the, cumulant, method is 10 to 100 times faster than the piecewise linear approximation and approximately 10 times slower from the derate method. The time advantage of the derate method does not compensate for the large accuracy disadvantage in energy calculations. Therefore, except for the first two or three base loaded plants the derate method should not be used for plant energy calculation in expansion planning studies.

Shenk [65] has suggested, that the LOLP accuracy of the cumulant method will increase if instead of using Equation (3.30) for LOLP calculation, the final equivalent load frequency curve, \( l_1(x) \), is recovered using the Gram-Charlier series and then integrated it, i.e.
Figure 3.16  Averaging Effect of Energy Integration in the Cumulant Method
LOLP = \int_0^{SC} l_N(x) dx \quad (3.57)

where \( l_N(x) \): final equivalent load frequency curve

SC : system capacity

When the above integration is done numerically, the LOLP calculation has the averaging advantage that is evident in the calculation of plant energy generation. However, a comparison between the time required to calculate LOLP and energy by the piecewise and cumulant methods (see Table 3.8), indicates that while in the piecewise linear method most of the time is consumed in the convolution of the generating units, in the cumulant method most time is used for the recovery of the equivalent load probability curve values (through the Gram-Charlier series). Therefore, if only 50 grid points are used to ensure reasonable accuracy of the numerical integration of \( l_N(x) \), the computing time requirement will become comparable to that of the piecewise method.

3.5 Hybrid Use of Cumulant and Piecewise Linear Approximation

Based on the advantages and disadvantages of the cumulant, and piecewise linear methods, a hybrid method was developed in which the piecewise linear approximation is used exclusively for LOLP calculations while cumulants are used only in energy calculation. When the piecewise linear method is used only for LOLP calculations, it can be used without sacrificing the
computing time for the following reasons. First, the LOLP calculations need only the last equivalent load curve, so all the units convolved can be handled as one block units. Second, once the last equivalent load probability curve is obtained for the existing system, or one candidate configuration, the last equivalent load for any other candidate configuration can be obtained by convolving or deconvolving only a few units. This is because the candidate configurations in one year are different only by a few units and also they vary little from year-to-year. Thus, when used only for LOLP calculations, the computing time for the piecewise linear method becomes comparable to that of the cumulant method.

Energy calculation using the piecewise linear method in expansion planning studies is much more time consuming than for LOLP for two reasons. Firstly, the plant energy calculations necessitate multiple capacity block representation for large and medium size units, which increases the computing time both because the number of convolutions is proportional to the total number of blocks and because deconvolution becomes necessary. Secondly, the addition of new generating units requires the recalculation of the energy generation of all units with lower loading priority. On the other hand, our sensitivity analysis shows that the accuracy in energy calculations with the cumulant methods is satisfactory. In addition, the cumulant method provides the capability to simulate partial unit deratings with no significant computing time increase.

The hybrid method of alternating use of piecewise linear and cumulant approximations is compared qualitatively in
Table 3.9 with the strict use of the piecewise linear, cumulant, and derate methods. The accuracy of the new method is satisfactory in both energy and LOLP calculations without any significant computing time sacrifice.

3.6 Load Forecast Uncertainty

3.6.1 Introduction

In many forecasting models the aggregate energy demand for each future simulation period is forecasted independently of the load probability curve [66]. The magnitude of the forecast error is usually significant, so that in many expansion planning studies forecast uncertainty is considered in sensitivity analyses for a few energy demand scenarios e.g. high, medium, and low energy demand. [19, 20, 58]. However, this treatment of uncertainty is deterministic and costly since it requires a considerable number of runs of the same expansion planning program. Furthermore, the deterministic nature of these sensitivity analyses does not allow consideration of the increase of uncertainty in the future and therefore weighting probabilities cannot be assigned to the final results. The "Over/Under Capacity Planning" model [59], that is popular among the United States utilities, uses probability tree analysis to find the probability weights. However, the number of possible alternatives required for realistic consideration of forecast uncertainty is so large, that meaningful use of the model becomes prohibitively expensive.

A new method for incorporating energy demand uncertainty
TABLE 3.9 Qualitative Comparison of Hybrid, Piecwise, Cumulant, and Derate Methods for Expansion Planning

<table>
<thead>
<tr>
<th>ATTRIBUTES</th>
<th>DERATE</th>
<th>PIECEWISE</th>
<th>CUMULANT</th>
<th>HYBRID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACCURACY</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOLP Energy</td>
<td>- a</td>
<td>Good</td>
<td>Poor Satisfactory</td>
<td>Good Satisfactory</td>
</tr>
<tr>
<td>Energy</td>
<td>Poor</td>
<td>Good</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>COMPUTER EXPENSE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOLP Energy</td>
<td>- Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Energy</td>
<td>Low</td>
<td>Very High</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

*a LOLP calculation is not possible with the derate method.*
in capacity expansion planning is suggested in this section. The
effects of uncertainty on energy and LOLP calculations are studied via
sensitivity analysis.

3.6.2 Uncertainty Algorithm

In the treatment of energy forecast uncertainty the cumulant
approximation of probabilistic simulation is used with the following
additional assumptions

a) The forecasted energy demand, denoted here with E, is the expected
value of the energy demand

b) The relative error of E, is independent of the magnitude of E
c) The error frequency function is known

Assumption (b) is a necessary requirement for the validity of the
uncertainty algorithm and is explained later in this section.

Let the expected energy demand E in a period of T hours have maximum
relative error ±e%. The maximum error \( x_m \) in the forecasted average
system load is

\[ x_m = \frac{(E \cdot e)}{100 \cdot T} \]  \( (3.58) \)

where \( x_m \) is the maximum error in forecasted average system
load, in MW

E is the forecasted expected energy demand, in MWh

e is the maximum percentage error of E, and

T is the simulation time period in hours

Let \( L(x) \) be the forecasted load probability function. The upper
\( L^+(x) \) and lower \( L^-(x) \) bounds of \( L(x) \) are
\[ L_+ (x) = L(x-x_m) \]  \hspace{1cm} (3.59) \\
\[ L_- (x) = L(x+x_m) \]  \hspace{1cm} (3.60)

\( L(x), L_+(x), \) and \( L_-(x) \) are schematically shown in Figure 3.17.

A typical error frequency function \( e(x) \) is shown in Figure 3.18 with \( e(x)=0 \) for \( x<0 \) and \( x>2x_m \), where \( x_m \) is the mean of \( e(x) \).

The load probability function of random load \( \tilde{x}^- \) with no uncertainty is \( L_-(x) \), i.e. the lower bound of \( L(x) \). The combined random load, \( x \), that contains forecast uncertainty is

\[ \tilde{x} = \tilde{x}^- + x_e \]  \hspace{1cm} (3.61)

where \( x \) is the random load with forecast uncertainty

\( \tilde{x}^- \) is the random load without forecast uncertainty

\( x_e \) is the random forecast error

The load probability function, \( L_c(x) \), of the load demand with uncertainty\([67]\) is given by the convolution Equation

\[ L_c(x) = \int_0^{2x_m} L_-(x-x_e)e(x_e)dx_e \]  \hspace{1cm} (3.62)

in accordance with Equations (3.11) and (3.18) in Section 3.2. Note that the convolution of \( \tilde{x}^- \) and \( x_e \) requires that \( \tilde{x}^- \) and \( x_e \) are independent random variables, i.e. the validity of assumption (b) is necessary.
Figure 3.17 Upper and Lower Bounds of the Load Probability Curve
Figure 3.18 Typical Load Error Frequency Function
Let \( k_{ic}, k_i, k'_l \), and \( k_{le} \) be the \( i \)th cumulant of \( L_c(x) \), \( L(x) \), \( L^-(x) \), and \( e(x) \) respectively. Equations (3.46) and (3.62) lead to

\[
k_{ic} = k'_l + k_{le} \quad i = 1, 2, \ldots \quad (3.63)
\]

However, since all cumulants except the first remain the same if the origin of \( x \)-axis is changed \([62]\) we have

\[
k'_l = k_l \quad \text{for } i \geq 2 \quad (3.64)
\]

The first cumulant is the mean. Therefore,

\[
k_1 = k'_l + x_m \quad (3.65)
\]

If we substitute Equations (3.64) and (3.65) in (3.63) we have

\[
k_{ic} = k'_l + k_{le} \quad \text{for } i \geq 2 \quad (3.66)
\]

\[
k_{lc} = k'_l - x_m + k_{le} \quad (3.67)
\]

In the case that the error frequency function is symmetric,

\[
k_{le} = x_m \quad (3.68)
\]

and Equation (3.67) becomes
Equations (3.66) to (3.69) define the cumulants of the combined load probability function that incorporates energy forecast uncertainty. The combined load probability function $L_c(x)$ can be recovered from its cumulants with the Gram-Charlier or Edgeworth series, as was shown in Section 3.3.3. When $L_c(x)$ is used instead of $l(x)$ in probabilistic simulation, the uncertainty of the energy forecast is taken into consideration in the LOLP and plant energy calculations.

3.6.3. Effects of Uncertainty on Energy and LOLP

The effects of energy demand forecast uncertainty on the system reliability and plant energy generation are studied in a sensitivity analysis of eleven combinations of load demand distributions and generating systems. In each case the load demand is convolved with the forecast uncertainty, in the manner described in Section 3.6.2, and the resulting combined load demand serves as the starting point for the probabilistic simulation of the generating system.

The uncertainty of the forecast energy demand was specified by assuming that,

a) The relative error of the energy forecast is independent of the forecasted expected energy demand.

b) The uncertainty frequency function $e(x)$ is "normal" with zero mean ($\mu = 0$).

c) The probability $P_e$ that the energy forecast
relative error is within specific limits of plus or minus some percentage of the expected energy demand, is given.

Assumption (a) is a necessary condition for the validity of the uncertainty algorithm described in Section 3.6.2. Assumption (b) implies that all the cumulants of e(x), except the second (i.e. the variance), are zero. The variance is found using assumptions (b) and (c) as follows: Normal probability distribution tables give the probability $P_e$ that a random error is between specified limits $\pm z_e$, where $z_e$ is standardized variable. Therefore, given probability $P_e$ from assumption (b), $z_e$ is read from standard normal probability function tables [68]. However,

$$z_e = \frac{x - \mu}{\sigma} \quad (3.70)$$

and

$$x_e = \frac{\langle E \rangle}{T} \cdot \frac{A}{100} \quad (3.71)$$

where $\mu$ is the mean of the error frequency function (MW)

$\sigma$ is the standard deviation (MW)

$\langle E \rangle$ is the expected energy demand (GWH)

$T$ is the hour in forecast period

$A$ is the forecast uncertainty (percent)

Since $\mu = 0$, and $\langle E \rangle$, $T$, and $A$ are given and $z_e$ is read from the standard normal tables, the variance, $v$, or the uncertainty frequency function is
The cumulants of the uncertainty distribution function $e(x)$ are added to the cumulants of the original load probability function $l(x)$ in order to find the cumulants of the combined load probability function $L_c(x)$ that incorporates load uncertainty. The 5-cumulant Edgeworth series is used to estimate the system reliability and plant energy generation as described in Section 3.3.

The combinations of load demand and generating system data used for this analysis are the same as in cases nB ($n = 1$ to 11) of the sensitivity analysis presented in Section 3.4. Cases nA were not used here because the LOLP values are smaller than the cumulant method accuracy (as shown in Section 3.4).

Figure 3.19 shows the percentage differences of the final load probability curve between the piecewise linear method and the 5-cumulants series approximation for case 1B. The dashed line shows the differences between the two methods when uncertainty is not included in either of the methods. The solid line shows the differences when the piecewise method does not consider forecast uncertainty, while in the cumulant method it is assumed that there is a $60\%$ probability that the energy demand forecast error is within $\pm10\%$ of the forecasted mean energy demand. The dashed line (both methods without uncertainty) is identical to the solid line of Figure 3.14 and is included in order to show the relative error introduced by the use of the cumulant method.

Figure 3.20 depicts the percentage differences in plant energy production between the values calculated by the piecewise linear method
Figure 3.19  Percentage Difference of the Last Equivalent Load Probability Curves for Case 1B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
Figure 3.20 Percentage Difference in Plant Energy Production for Case 1B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
and those of the cumulant method in case study IB. The meaning of the solid and dashed lines is the same as in Figure 3.19, except that the differences refer to the energy production for each of the generating system's units. Figures D.41 to D.50 in Appendix D are identical to Figure 3.20, except that they refer to case studies 2B, 3B, ..., 11B.

Finally, Tables 3.10 and 3.11 give the effects of forecast uncertainty to system LOLP and total energy generation for case studies 1B, 2B, ..., 11B. The second column in Table 3.10 gives the LOLP values, as calculated by the piecewise linear method with no forecast uncertainty. Columns 3 and 4 give the LOLP values and LOLP differences from the piecewise method for the 5-cumulant series approximation without forecast uncertainty. Columns 2, 3, and 4 are identical to the bottom half of columns 2, 5, and 6 in Table 3.6. These entries are repeated here for comparison purposes. Columns 5 to 8 provide the LOLP values and LOLP differences from the piecewise method for 80% and 60% probabilities that the energy demand forecast error is +10% of the forecasted mean energy demand. Table 3.11 is identical in structure with Table 3.10, but provides the total energy production for case studies 1B to 11B. Columns 3, 4, and 5 are reproduced from the bottom half of columns 2, 5, and 6 of Table 3.7.

As seen in Figure 3.19 the equivalent load probability curve $\text{EL}_c(x)$ with uncertainty considerations (solid line) is lower in the middle and higher at the end than the equivalent load curve $\text{EL}(x)$ without uncertainty. This results in higher expected energy generation for peaking units and lower for intermediate units, as is evident in Figures 3.20 and D.41 to D.50 (in Appendix D)
Table 3.10  LOLP With and Without Forecast Uncertainty.\(^a\)
(80 and 60 percent probability for \(\pm 10\%\) error in total energy demand)

| CASE NUMBER | PIECEWISE METHOD
|--------------|----------------------------------|
|              | NO UNCERTAINTY LOP (X.001) | CUMULANT METHOD
|              | NO UNCERTAINTY LOP DIFF.\(^b\) | 80% UNCERTAINTY LOP DIFF.\(^b\) | 60% UNCERTAINTY LOP DIFF.\(^b\) |
|              | (X.001) (%) | (X.001) (%) | (X.001) (%) |
| 1B           | 15.1      | 15.3 | 1.3 | 17.4 | 15.2 | 20.4 | 35.1 |
| 2B           | 12.7      | 12.8 | 0.8 | 16.5 | 29.9 | 21.8 | 71.6 |
| 3B           | 0.\(^c\)  | 0.\(^c\) | -  | 0.\(^c\) | -  | 0.\(^c\) | -  |
| 4B           | 44.3      | 45.4 | 2.5 | 50.2 | 13.3 | 56.4 | 27.3 |
| 5B           | 3.0       | 3.0  | -  | 3.8  | 26.7 | 5.3  | 76.7 |
| 6B           | 45.2      | 45.5 | 0.7 | 49.0 | 8.4  | 53.6 | 18.6 |
| 7B           | 157.3     | 154.5| -1.8| 159.9| 1.6  | 166.7| 6.0  |
| 8B           | 101.4     | 99.6 | -1.8| 104.2| 2.8  | 110.3| 8.8  |
| 9B           | 58.7      | 59.2 | 0.8 | 62.2 | 6.0  | 66.0 | 12.4 |
| 10B          | 90.2      | 88.5 | -1.8| 92.0 | 2.0  | 96.6 | 7.1  |
| 11B          | 318.1     | 317.3| -0.2| 321.6| 1.1  | 326.8| 2.7  |

\(^a\) 5-Cumulant series is used. Standard normal distribution for the energy demand forecast error.

\(^b\) Differences from piecewise linear method

\(^c\) LOLP \(\leq 10^{-3}\) is not reported since it is smaller than the cumulant method accuracy.
Table 3.11 Total Energy Generation With and Without Forecast Uncertainty. (80 and 60 percent probability for 10% forecast error in total energy demand)

<table>
<thead>
<tr>
<th>CASE NUMBER</th>
<th>TOTAL ENERGY DEMAND (GWH)</th>
<th>TOTAL ENERGY GENERATED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PIECEWISE LINEAR NO UNCERTAINTY (GWH)</td>
</tr>
<tr>
<td>1B</td>
<td>38631.</td>
<td>38564.</td>
</tr>
<tr>
<td>2B</td>
<td>10304.</td>
<td>10293.</td>
</tr>
<tr>
<td>3B</td>
<td>8370.</td>
<td>8370.</td>
</tr>
<tr>
<td>4B</td>
<td>10854.</td>
<td>10800.</td>
</tr>
<tr>
<td>5B</td>
<td>9102.</td>
<td>9100.</td>
</tr>
<tr>
<td>6B</td>
<td>32115.</td>
<td>31899.</td>
</tr>
<tr>
<td>7B</td>
<td>23935.</td>
<td>23222.</td>
</tr>
<tr>
<td>8B</td>
<td>19098.</td>
<td>18764.</td>
</tr>
<tr>
<td>9B</td>
<td>14312.</td>
<td>14166.</td>
</tr>
<tr>
<td>10B</td>
<td>10501.</td>
<td>10322.</td>
</tr>
<tr>
<td>11B</td>
<td>8091.</td>
<td>7480.</td>
</tr>
</tbody>
</table>

a 5-Cumulant Edgeworth series approximation was used. The probability distribution of the energy demand forecast error is the "standard normal".

b Differences from energy generation calculated by the piecewise linear method.
for Cases 1B to 11B. The higher EL (x) tail values result in significantly increased LOLP values as shown in Table 3.10. However, the total expected energy generation (and consequently the unserved energy) are not affected significantly by forecast uncertainty, as indicated by the small change in the relative difference of the total energies in all cases shown in Table 3.11. This happens because the energy demand is shifted from the intermediate to the peaking units while the total expected energy demand remains unchanged (assumption b of mean error μ=0). This shift of energy production from the intermediate to the peaking units may cause significant changes in the types of generating units included in the optimum solution in capacity expansion studies. Therefore, it is recommended that this, or a similar uncertainty algorithm, be utilized in capacity expansion planning studies.

3.7 Conclusions

From the evaluation of the cumulant method presented in this chapter the following conclusions are drawn:

1) Use of more than 4 cumulants does not improve the cumulant method accuracy.

2) Generating systems that contain larger number of less reliable units are better simulated by the cumulant method than systems with fewer and more reliable units.

3) The error in the LOLP calculation with cumulants ranges from 0.1 to 1 days per year.
4) The total energy generation calculated with cumulants is accurate to four significant digits.

5) The cumulant method is 10 to 100 times faster than the piecewise method when energy and LOLP are both calculated. The high LOLP error of the cumulant method led to the development of a hybrid simulation method in which the piecewise linear method is used for LOLP calculations and cumulants are used for the calculation of plant energy production. In this method the accuracy advantage of the piecewise linear method is retained with insignificant sacrifice in computing time.

From the evaluation of the derate method it is concluded that calculation of plant energy by this method is faster than the cumulant method by a factor of ten. However, it is not recommended for calculation of plant energy generation, because only the energy generated by the base loaded plants can be calculated accurately.

Finally, a new algorithm was presented, which incorporates energy forecast uncertainty into the simulation of the operations of the electric generating systems. This uncertainty algorithm is easily adaptable to the cumulant method.

Traditionally, forecast uncertainty in expansion planning studies is considered via sensitivity analysis, which consist of running the same expansion planning model for several energy demand scenarios. The uncertainty algorithm, described in this chapter, provides the possibility of incorporating forecast uncertainty into the expansion planning model, and thus avoiding the many model runs required for the various energy demand scenarios. Furthermore, the forecast scenarios
are weighted with their probability of occurrence, so that the effect of the most improbable scenarios to the optimal solution is the smallest.

In the sensitivity analysis that was performed using the cumulant method with the uncertainty algorithm, it was found that the energy forecast uncertainty shifts some of the plant energy generation from the intermediate to peaking units and, although it increases LOLP, the total expected energy generation remains unchanged. It is recommended that this algorithm be used in capacity expansion planning models.
4.1 Introduction

In capacity expansion planning of electric power generating systems the optimum expansion plan is found by minimizing the sum of the discounted construction and operating costs (known as the "objective function"), under prescribed plant type availability and system reliability constraints. The optimization period (also called "study horizon") is limited by the availability of forecasted load and cost data and is usually taken to be 20 to 30 years.* The operating and construction costs for each year of the study horizon are discounted to a base year and then summed for all the study horizon years.

* It is the author's opinion that the accuracy of the forecasted load and cost data is so poor that the study period should be even smaller than 20 years. However, the study period should be a few years longer than the longest construction time of the plant types considered, in order to avoid optimization end effects (see Section 4.5).
As is obvious from Chapter 3, both the system operating costs and reliability (measured by the system LOLP) are nonlinear functions of the system capacity. The nonlinear aspect of the objective function and the reliability constraints, along with a large number of possible combinations of expansion plans, for even medium to small size electric utilities, and reasonably long study horizon, make the capacity expansion problem extremely difficult. The dynamic programming (DP) optimization method is well suited for capacity expansion problems because it does not restrict the form of the objective function and constraints. Also, it can be easily formulated for the discrete nature of the capacity expansion problems. However, although dynamic programming substantially reduces the number of expansion plant configurations when compared to direct enumeration methods, the number of configurations that must be considered is still very large for even modern computers [69]. This "curse of dimensionality" problem [70] has been the focus of a lot of research in recent years [40].

Among the techniques that have been suggested for the reduction of the dimensionality problem, the "Fathoming" technique [8, 12, 38-40] is very promising and has been adopted in the development of CERES. In capacity expansion planning the number of plant type configurations that must be considered increases exponentially with the number of years of the study horizon. Therefore, even with fathoming, the size of the capacity expansion problem for medium to large utilities and power pools is too large for most modern computers. (If off-core storage is used to store large amounts of frequently used
data, the core storage requirements decrease but input/output operations become too costly).

For the above reasons two iterative methods have been tried here in order to overcome the dimensionality problem. The first is based on tunnel iterations. In WASP, where this method first appeared, the iterations were done manually [13]. The second method is called here "stage iterative dynamic programming" and preliminary research shows promising results. Both methods, when combined with the fathoming technique, substantially reduce the computer core memory requirements.

The author's contributions in this chapter relate to the iterative use of dynamic programming. The tunnel iteration method is used here for the first time in an automatic manner and in such a way that, with very little man-machine interaction, it can provide substantially increased benefits over the manual method used in WASP. The stage iterative DP is developed here in a new form so that its applicability is not restricted to continuous and convex objective functions. Modifications to the stage iterative DP method are discussed and applications to capacity expansion planning are proposed.

In this chapter the DP formulation of the capacity expansion planning problem is presented in Section 4.2. Section 4.3 describes the tunnel iterative method. The stage iterative DP method is described in Section 4.4 while its proposed application to generating system expansion planning problems is outlined in Section 4.5.
4.2 Application of Dynamic Programming to Capacity Expansion Planning

4.2.1 Problem Definition

In electric power generating system capacity expansion studies the planning period is usually 20 to 30 years. This period is called here the "planning horizon". We assume that the generating system is expanded only at the beginning of each year. Therefore, the year number is used as the stage variable, and is denoted here by n. The first year of the planning horizon is \( n = 1 \), the last is \( n = N \), where \( N - 1 \) is the length of the planning horizon in years.

The state vector \( \bar{x}_n \) describes the generating system configuration in year \( n \).

\[
\bar{x}_n = [x_{k,n}]
\]

\[
= [x_{1,n}, x_{2,n}, \ldots, x_{k,n}, \ldots]^T
\]  

(4.1)

where \( x_{k,n} \) is the number of generating units of plant type \( k \) in stage \( n \).

The generating system expansion policy is described by the number of units \( u_{k,n} \) of each plant type \( k \) that are added to the system in year \( n \). This set of numbers constitutes the control vector \( \bar{u}_n \) in stage \( n \); i.e.
\[ \bar{u}_n = [u_{k,n}] \quad (4.2) \]

The state and control vectors are obviously related through the equation

\[ \bar{x}_{n+1} = \bar{x}_n + \bar{u}_n \quad (4.3) \]

which is called the "system equation".

The state variable \( x_{k,n} \) is subject to the constraint in the form

\[ a_{k,n} \leq x_{k,n} \leq b_{k,n} \quad (4.4) \]

where \( a_{k,n} \) and \( b_{k,n} \) are the lower and upper bounds of units of type \( k \) in the year \( n \).

There are two main reasons why this constraint is important. First, there are several possible reasons why the permissible number of units of a particular type may be restricted:

1. Diversity of unit types and sizes is desired
2. Manufacturing of a type may be limited
3. Policy constraint
4. Limitation due to fuel availability
5. Already committed without optimization

Second, the constraints in the form of Equation (4.4) are used to reduce the overall computing time. This use of the constraint is discussed in Section 4.3 where "tunnel" iterations are discussed.
The optimization requires the objective function to be minimized. The objective function to be minimized in CERES in the traditional definition is given by

\[
L_k = \sum_{j=1}^{k} (C_j - R_j + O_j)
\]  

(4.5)

where

- \(L_k\) is the objective function at stage \(k\),
- \(C_j\) is the total construction cost for the units that enter the system in year \(j\),
- \(R_j\) is the salvage value of all the units that exist at the end of year \(j\), and
- \(O_j\) is the total operating cost (of both old and new units) in year \(j\).

The salvage value \(R_j\) is included to account for the finite length of the study horizon. More details on how to calculate \(C_j\), \(R_j\), and \(O_j\) are provided in Reference [13, 19-20, 29]. The objective function in the case of a fixed charge rate is calculated according to Reference [42].

### 4.2.2 Fathoming Technique (*)

Fathoming is a technique to disqualify some trajectories (history) of system expansion without completing the DP optimization. Suppose

* The author was introduced to this technique by T. Morin. More details on Fathoming are included in Morin’s works [8, 12, 39-40] and in [6, 38].
we can find an upper bound for the objective function at the final stage (year). Denote this upper bound of the objective function with UB. Suppose the DP procedure reached year \( n \) at which the objective function of a trajectory at state \( 1 \) is \( \text{OBJ}_1 \). Then, the lower bound of the objective function for the trajectory passing state \( 1 \) may be written as

\[
\text{LB}_{1n} = \text{OBJ}_{1n} + \sum_{i=n+1}^{N} \text{LOBC}_i + \text{CC}_{n1}
\]  

(4.6)

where \( \text{LB}_{1n} \) is the objective function lower bound for state \( 1 \) in year \( n \)

\( \text{OBJ}_{1n} \) is the value of the objective function for state \( 1 \) in year \( n \)

\( \text{LOBC}_i \) is the lower bound of operating cost of a system that satisfies the user constraints in stage \( i \)

\( \text{CC}_{n1} \) is the minimum possible construction cost that satisfies the user constraints after the addition of state \( 1 \) and for all remaining stages \( n + 1, n + 2, \ldots, N \)

\( N \) is the number of stages (years) in the study period

If \( \text{LB}_{1n} \) satisfies the inequality

\[
\text{LB}_{1n} > \text{UB} \]  

(4.7)

where \( \text{LB}_{1n} \) is the objective function lower bound for state \( 1 \) in year \( n \),

\( \text{UB} \) is the objective function upper bound
then the state $l$ in year $n$ and the trajectory containing this state are rejected.

$CC_{nl}$ is estimated by first calculating the MW capacity that should be added in stages $n+1$ to $N$ in order to reach the minimum reserve margin in the last stage of the study period ($N$). This MW capacity is then multiplied by the minimum construction cost and is properly adjusted for escalation and discounting to base year.

$LOBC_i$ is the operating cost for year $i$ calculated for the hypothetical generating system that consists of the scheduled units and pseudoexpansion units, which are explained next. The pseudoexpansion units are added in each year $i$ so that the total system capacity takes the minimum value needed to satisfy the system minimum reserve margin requirement $RMIN$ in years $n+1$ to $N$. The calculation of the capacity, $CA_i$, of the pseudounits in year $i$ and $RMIN$ is discussed in Section 2.4.1. The operating cost of the pseudoexpansion units are equal to the lowest among all the expansion plant types that are allowed for year $i$. The forced outage rates and the maintenance requirements of the pseudoexpansion units are assumed to be equal to the lowest among all the expansion plant types. In year $i$, there are at most $i$ pseudoexpansion units, the capacities of which are $CA_1$, $CA_2$, ..., $CA_i$. In this hypothetical system, the pseudounits are assigned a higher loading order than the scheduled units, i.e., the pseudounits are loaded before the other units of the generating system. Therefore, the operating cost of the hypothetical system is lower than any generation configuration consisting of real expansion units and so represents a lower bound of the operating
cost for the system satisfying the minimum reserve margin requirement for the year. Thus, if the calculated annual operating cost of an arbitrary system is lower than $\text{LB}_{i} \text{C}$, it does not satisfy the $\text{RMIN}_{i}$ requirement for year $i$.

Since the effectiveness of the fathoming technique depends on how small an upper bound UB can be obtained, a good original estimate of UB is essential. Quick and approximate methods such as a simplified linear programming algorithm may be used [6,12] A very effective method is described in Section 4.3.

It should also be noted that $\text{LOBC}_{\text{n}}$ is merely the lower bound of the system operating costs in years $n+1$ to $N$; thus it may be much smaller than the actual minimum operating costs during these years. However, the difference between $\text{LOBC}_{\text{n}}$ and the actual minimum operating cost is reduced as the difference $N-(n+1)$ becomes smaller. Therefore, fathoming is more effective near the end of the planning horizon.

4.3 Iterative Dynamic Programming with Tunnel Iterations

When $a$ and $b$ in Equation (4.4) are set only in order to temporarily reduce the space requirements for storing all the admissible states and reduce the overall computing, the range $a<x<b$ is called the "tunnel". It is easy to see that the computational time quickly increases as the tunnel width for every stage increases. Conversely, the computational time is reduced as the tunnel width for every stage decreases. In using this device, it is important that the constraints do not distort the solution, which is achieved by an iterative scheme. First,
the dynamic programming is run with an appropriate tunnel for each candidate at each stage. If the solution of the dynamic programming optimization is on the upper bound \((x_{k,n} = b_{k,n})\), this indicates that the truly optimized \(x_{k,n}\) may still be above \(b_{k,n}\). Therefore, \(a\) and \(b\) are both increased. A similar procedure is applied when \(x_{k,n}\) is on the lower bound except that \(a\) and \(b\) are decreased. The dynamic programming is run with the new constraints. (If the constraints are set for a legitimate reason other than reducing computing time, \(a\) or \(b\) or both are not altered.) The above procedure is repeated until \(x_{k,n}\) does not fall on the lower or upper bound.

There is a trade-off between the tunnel width and the number of iterations required. As the tunnel width is reduced, it is more likely that \(x_{k,n}\) will be on the tunnel boundaries. Therefore more iterations become necessary. On the other hand, in order to reduce the iteration number, the tunnel width must be increased; thus, computational time for each iteration is increased. An optimum width to minimize the overall computational time is utilized in DYNO, based on past experience and preliminary results.

The tunnel iteration technique described above does not guarantee that the optimum solution is found. As suggested by T. Morin [71], the optimum state history may be completely out of the tunnel boundaries, in which case, there will be no tunnel violations and the solution will be suboptimum. The author had also experienced some cases in which the tunnel iterations did not result in the optimum solution. In these cases, the problem size was small enough for use of traditional Dynamic Programming.
An example of an optimum solution that will not be obtained by the tunnel iteration technique is the following. Suppose that the optimum solution consists of 20 gas turbines of 50 MW capacity each that are added in the last year of a five year long period. Also suppose the following:

(1) the available plant types are gas turbines of 50 MW, coal plants of 500 MW, and nuclear plants of 1000 MW

(2) two coal or one nuclear plants satisfy the system reliability requirements.

(3) the system reserve margins require the addition of at least 800 MW in the last year.

(4) the tunnel constraints restrict the number of units for each plant to a maximum of 3 units.

Under these conditions, the acceptable states in the fifth year consist of (a) one nuclear plant, or (b) two coal units, or (c) either (a) or (b) and up to three gas turbines. Obviously (a) or (b) will be preferable to (c) and, although the chosen solution does not violate the tunnels, the 20 gas turbines state for the fifth year can never be obtained.

The above disadvantage can be circumvented in the following way. First apply the tunnel iterative technique for very few (2 or 3) expansion plant types. The minimum objective function found can now be used as the upper bound of the objective function so that the fathoming technique becomes more effective in disqualifying possible state trajectories. Since the number of states considered in each stage is now reduced, the tunnel width may increase and more
expansion plant types may be considered. This process can be repeated until the tunnel constraints are no longer needed.

4.4 **Stage Iterative Dynamic Programming**

4.4.1 **Algorithm description**

The stage iterative dynamic programming (SIDP) algorithm is based on the progressive optimality principle [D1]. This is deduced from Bellman's Principle of Optimality [70] and states that:

"The optimal path has the property that each pair of decision sets is optimal in relation to its initial and terminal values"

Howson and Sancho [72] devised a two-stage iterative dynamic programming algorithm that finds the optimal trajectory between two specified states at the initial and final stages (two-point boundary value problem). The algorithm is proven to find the global minimum only for continuous and convex objective functions. [73-74] However, when the objective function is not convex, or inherently discrete*, local minima may result in suboptimal solutions as shown in the sample numerical application discussed in Sections 4.4.2 and 4.4.3. The stage iterative DP algorithm described here is a generalization of

* Discrete, here, refers to physically discrete systems and not discrete systems obtained as an approximation to continuous systems. In the former, convexity cannot be assured, while in the latter convexity can be determined [75-78].
the two stage iterative method and is applicable to non convex inherently discrete objective functions.

Assume that the optimization problem is defined as in Section 4.2, with the additional restriction that in the final stage \( N \) the state vector \( \bar{x}_N \) is fixed. The SIDPDP algorithm is described in the following steps:

Step 1. Assign an original "best guess" \( \bar{x}_0^n \) to the state vector \( \bar{x}_n \) for each stage \( n < N \) and find the value of the objective function \( \text{OBJ}_N^0 \). Note that \( \bar{x}_1 = \bar{x}_f \) and \( \bar{x}_N = \bar{x}_f \) are fixed.

Step 2. Set the "stage step" to \( m \), where \( 1 \leq m \leq N-2 \).

Step 3. Find the optimal trajectory (set of controls \( \bar{u}_n \)) between state state \( \bar{x}_1 \) and \( \bar{x}_{m+1}^0 \) by applying dynamic programming between them. Let this trajectory go through some state, call it \( \bar{x}_2^1 \), in stage 2.

Step 4. Repeat step 3 between stage \( n \) and \( n+m+1 \) for \( n=2, 3, \ldots, N-2 \), and states \( \bar{x}_n^t \) and \( \bar{x}_{n+m+1}^t \), where iteration number \( t=1 \). If \( n+m+1 > N \), set \( \bar{x}_{n+m}^t = \bar{x}_f \) and optimize between stages \( n \) and \( N \).

Step 5. Let the value of the objective function after completion of the step 4 be \( \text{OBJ}_N^t \) (\( t=1 \)) and the set of "best" state vectors be \( \bar{x}_n^t \), \( n=1, 2, \ldots, N-1 \). Repeat steps 3 and 4 by replacing the original guess \( \bar{x}_{n-t}^{t-1} \) with \( \bar{x}_n^t \) and \( \text{OBJ}_N^t \) with \( \text{OBJ}_N^t \). Note that the original \( \bar{x}_1 \) and final \( \bar{x}_f \) states remain fixed.

Step 6. Continue step 5 until either an iteration limit is reached or \( \bar{x}_n^t \) remains unchanged between two successive iterations. In each iteration \( t \), the fathoming upper bound is set to \( \text{OBJ}_{n+m+1}^t \).
Step 7. Increase the "stage step" to $m+1$.

Repeat steps 2 to 6 until there is no change in the optimal trajectory between iterations with different stage step size.

It should be noted that the convergence criterion given in step 7 does not guarantee convergence to the global optimum, because an increased step size may result in the same suboptimum solution (see next section). Therefore, the sufficient step size for convergence to the global optimum must be found through sensitivity analyses for each specific application of SIDP.

The proper size of the stage step $m$ depends on the type of the objective function and the available computer core-memory. When $m = 1$, the algorithm reduces to the two-stage algorithm suggested in References [72] and the core-memory requirements are minimal, i.e., only the one state vector $\bar{x}_n$ needs to be stored at each stage. However, when $m$ is increased, the number of state vectors that must be stored increases in the same way as in traditional DP [69]. When $m = N-2$, the algorithm becomes the traditional dynamic programming.

The two-stage iterative dynamic programming has been proven to converge to the optimal solution only for continuous, strictly convex objective functions [73]. Also it has been shown to work in cases where a continuous strictly convex objective function is approximated with discrete functions [74] However, to the best of the author's knowledge, neither a multi-stage iterative dynamic programming, such as the one described above, has been proposed
nor an application of stage iterative DP on non convex, inherently (see footnote in page 113) discrete functions, has been attempted.

4.4.2 Test Problem Definition

The stage iterative dynamic programming algorithm described in Section 4.4.1 was applied to a test optimization problem under various discrete objective function definitions. In this one-dimensional problem the system equation is:

\[ x_{n+1} = x_n + u_n \]  \hspace{1cm} (4.8)

The objective function is

\[ I = \sum_{n=1}^{N} A_n \hspace{1cm} n=1, 2, \ldots, N \]  \hspace{1cm} (4.9)

where \( N \) is the total number of stages considered

\( A \) is given in Table 4.1 for 12 case studies.

The constraints are

\[ 1 \leq x_n \leq N \] \hspace{1cm} (4.10)

\[ 2 \leq u_n \leq 2 \] \hspace{1cm} (4.11)

where both \( x_n \) and \( u_n \) are integers. The boundary conditions were set to \( x_1 = x_1 \) and \( x_N = x_f \). In all case studies the following constants were used:
TABLE 4.1 Various Objective Functions for the Test Problem

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Objective Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(x_n - 1)^2 + u_n^2$</td>
</tr>
<tr>
<td>2</td>
<td>$(x_n - 1)^2 - u_n^2$</td>
</tr>
<tr>
<td>3</td>
<td>$[(x_n - 1)^2 + u_n^2]^{1/2}$</td>
</tr>
<tr>
<td>4</td>
<td>$(x_n - 1) + u_n$</td>
</tr>
<tr>
<td>5</td>
<td>$(x_n - 1)^3 + u_n^3$</td>
</tr>
<tr>
<td>6</td>
<td>$(x_n - 1)^3 - u_n$</td>
</tr>
<tr>
<td>7</td>
<td>$(x_n - 1) - u_n^3$</td>
</tr>
<tr>
<td>8</td>
<td>$x_n^2 + (u_n + 10)^{-2}$</td>
</tr>
<tr>
<td>9</td>
<td>$[x_n^2 + (u_n + 10)^{-2}]^{1/2}$</td>
</tr>
<tr>
<td>10</td>
<td>$x_n^{-1} + (u_n + 10)^{-1}$</td>
</tr>
<tr>
<td>11</td>
<td>$[x_n^{-1} + (u_n + 10)^{-1}]^{1/2}$</td>
</tr>
<tr>
<td>12</td>
<td>$x_n^{-3} + (u_n + 10)^{-3}$</td>
</tr>
</tbody>
</table>
This optimization problem, with the objective function defined as in case 1 of Table 4.1 and \( N = 10 \), is taken from Reference [69].

The objective functions chosen in this study are arbitrary except that they are defined only for discrete values of the state variables. Therefore, in all cases listed in Table 4.1 the objective function is discrete and non-convex [75,77].

The computer program that was written to solve the optimization problem stated above is listed in Appendix E.

4.4.3 Results and Analysis

Table 4.2 shows for each case study the value of the objective function to which the SIDP algorithm converged. The number of iterations required for convergence is indicated for each stage step size. The optimum objective function that was found by applying traditional dynamic programming is also listed for comparison purposes. It should be noted that the iterative algorithm did not always converge to the optimum solution. However, when the stage step size was increased, the optimum was obtained. In only cases 6 and 12 a stage step size greater than three was required for convergence to the optimum solution. Increased step size always resulted in a smaller number of iterations.

Figures 4.1 to 4.12 show the trajectory of the initial guess and the optimum objective function trajectories for all 12 case studies.
TABLE 4.2 Objective Function Value and Number of Iterations Required for Convergence of SIDP for the Test Problem

<table>
<thead>
<tr>
<th>CASE NUMBER</th>
<th>TRADITIONAL DP</th>
<th>STAGE STEP SIZE OF SIDP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>OBJ. $F_n$</td>
</tr>
<tr>
<td>1</td>
<td>138.00</td>
<td>149.00</td>
</tr>
<tr>
<td>2</td>
<td>72.00</td>
<td>76.00</td>
</tr>
<tr>
<td>3</td>
<td>23.87</td>
<td>36.69</td>
</tr>
<tr>
<td>4</td>
<td>14.00</td>
<td>14.00</td>
</tr>
<tr>
<td>5</td>
<td>771.00</td>
<td>771.00</td>
</tr>
<tr>
<td>6</td>
<td>24.00</td>
<td>39.00</td>
</tr>
<tr>
<td>7</td>
<td>806.00</td>
<td>806.00</td>
</tr>
<tr>
<td>8</td>
<td>0.37</td>
<td>0.38</td>
</tr>
<tr>
<td>9</td>
<td>2.57</td>
<td>2.60</td>
</tr>
<tr>
<td>10</td>
<td>3.53</td>
<td>3.53</td>
</tr>
<tr>
<td>11</td>
<td>8.13</td>
<td>8.13</td>
</tr>
<tr>
<td>12</td>
<td>0.04</td>
<td>0.04</td>
</tr>
</tbody>
</table>
Figure 4.1 Initial Guess, Optimal, and Suboptimal Trajectories for Case 1. The Stage Step Size is Denoted by $m$. 
Figure 4.2 Initial Guess, Optimal, and Suboptimal Trajectories for Case 2. The Stage Step Size is Denoted by $m$. 

- ○ Initial guess
- ○ Suboptimal $m=1$
- ● Optimal $m>2$
Figure 4.3 Initial Guess, Optimal, and Suboptimal Trajectories for Case 3. The Stage Step Size is Denoted by $m$.
Figure 4.4 Initial Guess and Optimal Trajectories for Case 4. The Stage Step Size is Denoted by $m$. 

○ Initial guess
● Optimal $m > 1$
Figure 4.5 Initial Guess and Optimal Trajectories for Case 5. The Stage Step Size is Denoted by m
Figure 4.6 Initial Guess, Optimal, and Suboptimal Trajectories for Case 6. The Stage Step Size is Denoted by m
Figure 4.7 Initial Guess and Optimal Trajectories for Case 7. The Stage Step Size is Denoted by $m$. 

- Initial guess
- Optimal $m > 1$
Figure 4.8  Initial Guess, Optimal, and Suboptimal Trajectories for Case 8. The Stage Step Size is Denoted by m
Figure 4.9  Initial Guess, Optimal, and Suboptimal Trajectories for Case 9. The Stage Step Size is Denoted by m
Figure 4.10 Initial Guess and Optimal Trajectories for Case 10. The Stage Step Size is Denoted by $m$. 

- Diamond: Initial guess
- Circle: Optimal $m>1$
Figure 4.1: Initial Guess and Optimal Trajectories for Case 11. The Stage Step Size is Denoted by m.
Figure 4.12  Initial Guess, Optimal, and Suboptimal Trajectories for Case 12. The Stage Step Size is Denoted by m
The trajectory of each objective function (or the initial guess) is depicted by a heavy solid line that connects the state values $x_n$ for stages $n=1$ to $n=20$. In cases 1, 2, 3, 6, 8, 9, and 12 the suboptimum trajectories to which the algorithm converged for small stage step sizes ($n$) are also depicted. It should be noted that even when the SIDP did not converge to the optimal trajectory for small stage step sizes, the suboptimal trajectory was very close to the optimal. Case 12 should also be noted because although the optimal objective function was found with the desired precision (two decimal places at step size $m=1$, see Table 4.2) the optimal trajectory required $m=4$.

Figure 4.13 depicts the trajectories for the five different initial guesses that were tried in case 1. Table 4.3 shows the results of the application of the SIDP algorithm with the initial guesses shown in Figure 4.13. The trajectories of the initial guesses, as they are identified by the sequence numbers in column one, are shown in Figure 4.13. The objective function, number of iterations, and the computing time requirements are shown, in groups of three columns, for stage step sizes $m=1$, 2, 3, 4, 5, 10, and 15. It should be observed that unless the initial guess is almost identical to the optimal solution, SIDP will converge to the same optimal or suboptimal solution for a given stage step size. In other words, convergence to the optimum is almost independent of the initial guess. However, the number of iterations and the computing time required for convergence become smaller as the initial guesses approach the optimum trajectory (see the optimal trajectory in Figure 4.1).
Figure 4.13 Initial Guesses for Case 1
TABLE 4.3 Number of Iterations and Computing Time for the Test SIDP Problem

<table>
<thead>
<tr>
<th>Initial Guess Number</th>
<th>Stage Step Size $m$</th>
<th>OBJ. ITER</th>
<th>OBJ. ITER</th>
<th>OBJ. ITER</th>
<th>OBJ. ITER</th>
<th>OBJ. ITER</th>
<th>OBJ. ITER</th>
<th>OBJ. ITER</th>
<th>OBJ. ITER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>149.9 b 235 b</td>
<td>149.5 b 508 b</td>
<td>138.3 677</td>
<td>138.3 830</td>
<td>138.2 769</td>
<td>138.2 531</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>148.1 262 b</td>
<td>149.6 61 b</td>
<td>138.5 840</td>
<td>138.4 905</td>
<td>138.3 835</td>
<td>138.2 770</td>
<td>138.2 537</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>138.2 47</td>
<td>138.2 203</td>
<td>138.2 336</td>
<td>138.2 455</td>
<td>138.2 553</td>
<td>138.2 772</td>
<td>138.2 535</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>149.5 132 b</td>
<td>149.3 306 b</td>
<td>138.3 673</td>
<td>138.3 835</td>
<td>138.2 775</td>
<td>138.2 533</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>149.4 105 b</td>
<td>149.3 312 b</td>
<td>138.3 511</td>
<td>138.3 686</td>
<td>138.2 557</td>
<td>138.2 775</td>
<td>138.2 534</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a Time is measured in CPU microseconds of the AMDAHL 470 V Ohio State University computer.

The objective function and control vector are defined as in Case One in Table 4.1.

The computing time required by traditional DP is 83 CPU microseconds.

b The iterative algorithm converged to suboptimum solution.
It should also be noted in Table 4.3 that the computing time increases when the stage step size increases until the step size becomes greater than \((N-2)/2\), where \(N\) is the total number of stages. At this point, the computing time decreases and, unless the initial guess is almost identical to the optimum trajectory, the computing time of traditional dynamic programming is the smallest (83 CPU microseconds). The reason for the decrease of computing time when the stage step \(m\) becomes greater than \((N-2)/2\) is that \(m\) is forced to be \(M = N-2-n < (N-2)/2\) when stage \(n = ((N-2)/2)+1\) is reached. (See step 4 in the SIDP description in Section 4.4.1). Otherwise \(n+m > N\), which is impossible.

The longer computing time that is the obvious time disadvantage of the SIDP algorithms is compensated for by a significant reduction in core memory requirements. When the number of admissible control values is \(k\), the number of storage locations required by dynamic programming to solve the sample problem for \(m+2\) stages is given in Table 4.4. It is assumed that 3 words are required for each admissible state in order to store the value of the objective function, the state description, and the optimum control value. The number of storage locations required by DP for an \(N\) stage, one dimensional, problem is at most \((2N-1)K\). When the problem is multidimensional, the number of admissible controls \(k\) should be multiplied by the number of dimensions. The number of admissible states (and required memory) increases exponentially with the number of dimensions [69].
Table 4.4 Core Memory Required by Stage Iterative Dynamic Programming for the Test Problem

<table>
<thead>
<tr>
<th>STAGE STEP SIZE</th>
<th>CORE MEMORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3(k+1)</td>
</tr>
<tr>
<td>3</td>
<td>3(2k+1)</td>
</tr>
<tr>
<td>4</td>
<td>3(2k+2k-1+1)</td>
</tr>
<tr>
<td>5</td>
<td>3(2k+2(2k-1)+1)</td>
</tr>
</tbody>
</table>

Although the core memory requirements are significantly reduced when the stage step is small, storage requirements are further reduced when the fathoming technique is used. Fathoming is most effective (in reducing the admissible states) in the last few stages i.e. stage N-1, N-2, N-3, etc. in an N stage problem (see Section 4.2.2). Therefore, when the stage step is m and DP is applied between stages n and n+m+1, fathoming is most effective in reducing the number of admissible states in stages n+m, n+m-1, n+m-2 etc. Since m is much smaller than N, fathoming reduces the number of admissible states significantly in all the range of DP application for stage step m (i.e. n to n+m+1).
Although the new iterative algorithm was tested only with a limited range of problems, the consistency with which the new algorithm converged to the optimal solution as the stage step increased, strongly suggests that this algorithm will work well for other optimization problems. It is pointed out that substantial more research is needed in order to find the proper stage step size* and verify the SIDP's usefulness in applications to specific optimization problems. Some guidelines for the research needed to apply SIDP to capacity expansion planning are suggested in the next section.

4.5 Suggested Application of Stage Iterative Dynamic Programming to Capacity Expansion Planning

The introduction of the cumulant method in capacity expansion planning reduced substantially the computing time required to calculate the plant energy production for each DP state. Since the energy calculation is the most time consuming calculation required for each state, the application of DP to expansion planning is not limited by the computing time requirements, but by the core memory needed. Therefore application of SIDP to capacity expansion planning of electric power generating systems is of particular interest.

The most direct application of the new algorithm in capacity expansion planning is for cases where a proposed expansion plan needs to be investigated for its optimality. In these cases the state

* See discussion on the choice of stage step size in Section 4.4.1.
vector (generating unit configuration) for the final stage (year) of the planning horizon is known. Therefore, the stage iterative dynamic programming algorithm can be applied exactly as described in Section 4.4. The known expansion policy may be used as the initial guess.

However, the new algorithm (as described in Section 4.4), does not solve the open ended capacity planning optimization problem (the case where the final state is not known). Therefore, the following modifications in steps 3 and 4 of the SIDP algorithm, described in Section 4.4.1, are suggested. Let \( \bar{x}_n \) be the \( k \)-dimensional initial guess state vector in stage \( n \) (see Equation 4.1). The system capacity \( C_n^t \) in state \( \bar{x}_n \) is

\[
C_n^t = \sum_{k=1}^{k} x_{k,n} C_k
\]  

(4.12)

where \( C_n^t \) is the system capacity in MW for state \( x \) in stage \( n \) for iteration \( t \)

\( x_{k,n} \) is the number of generating units of plant type \( k \) for state \( \bar{x}_n \) in stage \( n \)

\( C_k \) is the \( k \)th unit capacity in MW

A "capacity window" \( C^n_w \) is defined as the system capacity range between \( C_{n}^{t-w/2} \) and \( C_{n}^{t+w/2} \), where \( w \) is the user specified window width in MW, and \( t \) is the iteration number. The SIDP algorithm should be modified so that in each application of dynamic programming (steps 3 and 4) between stages \( n \) and \( n+m+1 \) in iteration \( t \), \( m \) being the stage step size, the target state \( \bar{x}_{n+m+1}^{t-1} \) should be replaced by all states admissible in the capacity
window \( C^{t-1}_{n+m+1} \); i.e. any state \( x_n \), that results in system capacity \( C_n \), which satisfies the inequality

\[
C_n^t - w/2 \leq C_n \leq C_n^t + w/2
\]

is an admissible target state. \( C_n^t + w/2 \) and \( C_n^t - w/2 \) serve as temporary upper and lower reserve margins, so that the number of admissible target states is restricted by the user through the width of the capacity window. Thus, the DP application between stages \( n \) and \( n+m+1 \) does not only determine the optimum trajectory, but also the optimum target state. When stage \( n+m+1 \) becomes the final stage \( N \) in iteration \( t \), through SDIP steps 5, 6, and 7, the objective function of the optimum target state is the optimum objective function for iteration \( t \). Steps 1, 2, 5, 6, 7, in this modified SIDP algorithm are the same as in the original SIDP algorithm described in Section 4.4.1.

The advantage of the modified SIDP is that, besides the optimum trajectory, the optimum target state and objective function are determined. The modified SIDP algorithm requires the specification of a target system capacity range in the final stage instead of a specific plant configuration. This capacity range can be provided by the user specified minimum and maximum reserve margins (capacity window). However, if the window width becomes too large, the number of admissible states will be too many to store in the computer core memory. On the other hand, if the window width is too small, more iterations will be required for convergence.
In order to avoid excessive core memory requirements, the capacity window must be kept small. However, SIDP may not find the true optimum objective function if the capacity window (reserve margins) for the final year of the study horizon is too restrictive. A prohibitively large capacity window at the final year \( N \) of the study horizon may be avoided by extending the study horizon by a few years. The extended study horizon is called the "planning horizon". An extended horizon is also necessary to avoid the end effects created by the addition of large and expensive power plants at the final years of the planning horizon [43,44]. Naturally, the optimal expansion policy is valid only during the study horizon. This suggested use of a narrow capacity window with an extended study horizon should be investigated as to whether it inhibits the finding of the true optimum.

It should be noted that the capacity window modification is not limited only to open ended problems. It can also be used in two-boundary multidimensional problems in order to reduce the number of iterations required for convergence.
5.1 Summary and Conclusions

The objectives of this research were (a) to evaluate the cumulant approximation of probabilistic simulation of electric generating systems for the purpose of capacity expansion planning, and (b) to develop new, fast, and efficient algorithms for capacity expansion planning.

The first objective was accomplished through a comprehensive sensitivity analysis of a number of combinations of load demand and generating system characteristics. This evaluation of the cumulant method led to the following conclusions:

1) Use of more than 4 cumulants does not improve the cumulant method accuracy.

2) Generating systems that contain less reliable units are better simulated by the cumulant method than systems with more reliable units.

3) The error in the LOLP calculation with cumulants ranges from 0.1 to 1 days per year.

4) The total energy generation calculated with cumulants is accurate to four significant digits.

5) The cumulant method is 10 to 100 times faster than
the piecewise method when energy and LOLP are both calculated.

The computational speed of the cumulant method is its most important advantage. The low accuracy of this method in LOLP calculations had not been recognized until now. However, the accuracy of plant energy calculations is acceptable.

The second dissertation objective was accomplished through the development of the following algorithms:

(a) Hybrid approximation of probabilistic simulation
(b) Energy forecast uncertainty
(c) Automated tunnel iterations
(d) Stage iterative dynamic programming.

In the hybrid approximation, cumulants are used to calculate plant energy production while piecewise linear polynomials are used for the LOLP calculation. Thus, both plant energy and LOLP are calculated accurately without significant sacrifice in computing time.

The energy forecast uncertainty algorithm makes possible the inclusion of forecast uncertainty into the calculations of LOLP and plant energy production. This algorithm provides the means to study the effects of forecast uncertainty in capacity expansion planning not only accurately but with significant reduction in computing time. Computing time is reduced because multiple runs of the same planning model for various forecast scenarios are avoided.

The effects of incorporating the forecast uncertainty in the probabilistic simulation of electric generating systems were studied in a limited sensitivity analysis. It was shown that when the uncertainty
algorithm is incorporated in the probabilistic simulation, the LOLP values increase, and, although the total energy generation remains unchanged, some of the energy generated by intermediate units is shifted to peaking units.

Algorithms (c) and (d) relate to the iterative use of dynamic programming in capacity expansion planning models. Tunnel iterations were performed manually in WASP [13]. The new algorithm developed not only automates the tunnel iterations, but also incorporates the fathoming technique in dynamic programming. This algorithm is faster and easier to use than the one used in WASP.

The stage iterative dynamic programming algorithm was developed and shown to work for inherently discrete and non-convex objective functions. This algorithm requires significantly less computer memory than the traditional dynamic programming algorithm. It is noted that the "curse of dimensionality" (i.e., excessive memory requirements) is the most important disadvantage of dynamic programming. Therefore, the stage iterative dynamic programming algorithm may prove to be very useful. The potential application of the stage iterative dynamic programming algorithm in capacity expansion planning was analyzed and details on how this can be accomplished were provided.

Finally, it is noted that the hybrid approximation of probabilistic simulation and the automated tunnel iterations have been incorporated in the CERES program. These algorithms, along with the plain English interactive dialogue for man-machine interaction, made the CERES program fast, accurate, and very simple to use.
5.2 Recommendations for Further Research

It is recommended that the following research be undertaken:

(a) incorporate in CERES the energy forecast uncertainty algorithm
(b) evaluate the application of the stage iterative dynamic programming method in capacity expansion planning models, as suggested in Chapter 4
(c) include in CERES a more sophisticated financial analysis algorithm
(d) simulate in CERES hydroelectric and pump storage units
(e) develop a data management system for the integration of load forecasting, expansion planning, load management and financial analysis models. The input/output data of all these models would be stored in a common database through which the models would interact. Thus, feedback effects (such as the effect of high electricity prices, that may result from an ambitious capacity expansion plan, on forecasting electric energy demand) can be studied.
APPENDIX A

OPTIM MODULE SOURCE PROGRAM LISTING

In this Appendix the Fortran source program of the OPTIM module of CERES is listed. It should be noted that the development of CERES was a group effort. The author was responsible for the development of most of the PREP submodule of OPTIM although J. Shih did most of the coding. DYNO was developed and written by the author with some help from J. Shih and M. Harrunuzzaman in the maintenance and report writing subroutines. PREP submodule is listed in the first section of this Appendix and DYNO in the second section.
**PREP SUBMODULE**

This submodule determines:

1. Minimum Reserve Margin
2. Economic Loading Order
3. Lower Bound of Operation Costs

For each year in the study period.

**Input:**

1. Plant Data (Unit 11)
2. Normal Distribution Table (Unit 15)

**Output:**

1. Final Load After the Convolution of Scheduled System for Each Year (Unit 22)
2. Economic Loading Order and Scheduled System Maintenance Allocation for Each Year (Unit 24)
3. Hourly Load Cumulants and Scheduled System Outage Cumulants for Each Year (Unit 27)
4. Plant Data (Unit 23)
5. Other Information Required by Dyno-Submodule (Unit 10).

---

**Main Program**

```plaintext
define file 22(80,5004,1,122)
define file 23(250,80,1,123)
define file 24(100,10004,1,124)
define file 27(80,6436,1,127)

common/curves/el(4,1250)
cmmon/arrays/x1(1250),plant(200,14),emdr(220,4),
*peak(20,4),rsv(20),1up(20,20),numf(200,20),ibk(500,3),
*nx(20),nt(20),mx(20),main(200)
common/one/yrb,pmp,tmk,ntyr,mxeldc(20,4),ec(20),rsmx(20),
custom(20),perene(20,4),avspe(20,4),nstpre(20,10)
cmmon/var1ab/n1,dax,ipt1,pmax,txp,npe,nyr,kmax,add
common/table/ty(120),tim(1,120)
double precision ty,tim
cmmon/two/placa(20,16)
cmmon/three/ptcum(111,16),hrcum(4,8)
double precision ptcum,hrcum
data k11,k15,11,15/
write(6,1080)
1080 format(1h,"INPUT THE CRITICAL LOLP VALUE IN DAYS PER 10 YEARS")
read(5,*) pmp
pmp = pmp/3650.
write(6,1090) pmp
1090 format(1h,"CRITICAL LOLP ASSIGNED = "e14.7)
np=4
ipt=500
ncum=4
jnum=14
mx=1250
```
DXNORM = 1.0/FLOAT(IP)
DO 25 I = 1,1250
25 X(I) = FLOAT(I) * DXNORM
READ PLANT DATA
READ(K11,103) IPMAX,NPEXP,IYKB,IYRE
NYK=IYRE-IYKB+1
DO 10 IP=1,IPMAX
READ(K11,104) (NJF(IP,N),N=1,20)
READ(K11,105) (PLANT(IP,J),J=1,JNUM)
PLANT(IP,9)=30.
I23 IP
WRITE(23,123,904) (PLANT(IP,J),J=1,JNUM)
904 FORMAT(F3.0,2F5.0,F3.0,F8.6,3F7.2,F3.0,2F7.3,E10.3,2F4.2)
10 CONTINUE
STORE INFORMATION FOR EXP. SYSTEM
DO 15 IP=1,NPEXP
PLACAI1P=J)— PLANT(IP,J)
15 CONTINUE
READ(K11,104) (IUP(IP,N),N=1,20)
20 CONTINUE
1030 FORMAT(4110)
1040 FORMAT(4012)
1050 FORMAT(5E15.5)
READ NORMAL DISTRIBUTION TABLE
READ(K15,106) (TY(I),TINT(I),I=1,120)
1060 FORMAT(F10.4,F12.8)
DO 30 J=1,NTVR
JYR=J
IYEAR=IYKB+JYR-1+1900
PRINT 1070, IYEAR
1070 FORMAT(///,I15,' STUDY YEAR = ',I5//)
CALL RINPUT
CALL MINRSV
CALL LOADER
CALL OPCODE(NCUM)
30 CONTINUE
CALL INOUT
STOP
END
SUBROUTINE MINRSV

COMMON/CURVES/EL(4,1250)
COMMON/ARRAYS/X(1250),PLANT(200,14),EMOR(220,4),
*PEAK(20,4),RSMX(20),IUP(20,20),NUM(20,20),IBM(500,3),
*NX(20),NT(20),IX(20),IBNS(200)
COMMON/ONE/1YRB*PMP,T,MX,NTYR,EXELDC(20,4),EC(20),RSMX(20),
*COSTM(20),PERENE(20,4),AVSP(20,4),NSTPRE(20,10)
COMMON/VARIAB/NP,ID,MAX1,1P1,1PMAX,NPEXP,JYR,KMAX,ADD

C-----------------------------

C THIS ROUTINE COMPUTES THE MINIMUM RESERVE MARGIN FOR EACH
C YEAR. IT IS OBTAINED BY REQUIRING THAT THE CRITICAL
C SEASONAL LOLP IS LESS THAN OR EQUAL TO THE CRITICAL
C VALUE ASSIGNED BY THE USER AT THE BEGINNING.
C
C OBTAINED INFORMATION ARE EXPRESSED IN TERMS OF:

RSV(YEAR) : AMOUNT IN MW
R    : IN PERCENT
NX(1) : NUMBER OF UNIT OF TYPE IX(1)
       (EXPANSION SYSTEM ONLY)
NT(1) : TOTAL NO OF ADDITION OF
       TYPE IX(1)
       (EXPANSION SYSTEM ONLY)
IX(1) : TYPE OF UNIT
       (EXPANSION SYSTEM ONLY)

SUBROUTINE CALLED:
LPD

C***DECLARATION OF LOCAL VARIABLE***

DIMENSION IUT(20)
IYR=IYRB+JYR+1*1900
G=0.4

C FIND YEARLY PEAK AND MAX RES. MARGIN

PMAX=.0
DO 200 I=1,4
IF(PEAK(JYR,I).GT.PMAX) PMAX=PEAK(JYR,I)
200 CONTINUE
RSMX(JYR)=PMAX*(1.0+G)

C IF(JYR.GT.1) GO TO 400
IX(1)=NPEXP
IUT(1)=IUP(1,JYR)
NX(1)=0
NT(1)=0
DO 410 I=2,NPEXP
IX(I)=IX(I-1)-1
210 NT(I)=0
410 IUT(I)=IUP(I,JYR)
GO TO 420

C DETERMINE NO OF UNITS AVAILABLE FOR THE PRESENT STUDY
C YEAR AND INITIALIZE NX(I) ARRAY

C-----------------------------
SCHEDULED SYSTEM CONVOLUTION

! ID=1
! IDL=0
! ADD = 0.0
! CALL LPD(CLP,PLDP,ECP,IDL)
! IF(CLP.LE.PMP) GO TO 1001

! OTHERWISE PUT THE SMALLEST CANDIDATE ON LINE
! ONE AT A TIME
! ID=2
! IDL=1
! NX(1)=1
! ADD=ADD+PLANT(IX(1),3)
! CALL LPD(CLP,PLDP,ECP,IDL)
! IF(CLP.LE.PMP) GO TO 100

! NX(1)=NX(1)+1
! GO TO 10

! IF(NX(1).LE.IUT(IX(1))) GO TO 1001

! DISTIBUTE THIS AMOUNT BETWEEN OTHER AVAILABLE
! CANDIDATES

! IDL=2
! RV=ADD
! DO 110 J=2,NPEXP
! NX(J)=1
! IF(NX(J).LE.IUT(IX(J))) GO TO 115
! GO TO 110

! 115 IE=J-1
! 170 ADD=0.0
! DO 120 I=1,IE
! 120 NX(I)=0
! ADD=ADD+FLOAT(NX(I))*PLANT(IX(I),3)
! IF(ADD.GE.RV) GO TO 20

! NX(I)=NX(I)+1
! GO TO 130

! 130 CONTINUE
! 140 ADD=ADD+PLANT(IX(I),3)
! IF(ADD.GE.RV) GO TO 20

! NX(I)=NX(I)+1
! GO TO 160

! CONTINUE
! 150 NX(J)=NX(J)+1
! IF(NX(J).GT.IUT(IX(J))) GO TO 110
! GO TO 170

! CALL LPD(CLP,PLDP,ECP,IDL)
! IF(CLP.LE.PMP) GO TO 1001

! START ADDING THE SMALLEST CANDIDATE AGAIN
! GO TO 15

! 110 CONTINUE

! PRINT 1020
! 1020 FORMAT(IX* NO CONVERGED SOLUTION */)
! GO TO 1100

! RSV(IXR,IUT)=ECP-PMAX
R=(11ECP/PMAX)-1)*100.

DO 900 I=1,NPEXP
   II=NPEXP-1+1
900 NSTPRE(JYR,I)=NX(11)

Determined Pseudo-Plant Parameters

II=IPMAX+1
PLANT(II,2)=ADD
PLANT(II,3)=ADD
PLANT(II,4)=10.0
PLANT(II,5)=0.02
IP=1BK(1,1)

DO 505 J=6,14
505 PLANT(II,J)=PLANT(1P, J)

DO 510 I=1,NPEXP
   J=NPEXP-1+1
510 IF(INT(J).LE.0) GO TO 510
   IP=1X(J)

PLANT(1P,2)=PLANT(1P,3)
DO 520 K=4,14
520 PLANT(1P,K)=PLANT(II,K)
510 CONTINUE

Allocate Maintenance

ID=3
CALL MAINT(1,ECP)

PRINT 3,RSV(JYR)*R
PRINT 6,RSMX(JYR)
PRINT 8,CLP

Find Total No of Units Added Up to the Present Year

STUDY YEAR

DO 500 I=1,NPEXP
   NT(1)=NT(1)+NX(1)
500

PRINT 4,(IX(I),I=1,NPEXP)
PRINT 5,(NX(I),I=1,NPEXP)
PRINT 7,(NT(I),I=1,NPEXP)

3 FORMAT(1H6*MIN. RES. MARGIN = *1*
*F7.1* MW * ("*F5.2* %")
6 FORMAT(1H6*MAX. RES. MARGIN = *1*
*F8.1* MW * )
4 FORMAT(1H8*PLANT NO. *12X,1013)
5 FORMAT(1H8*NO. OF UNIT ADDED THIS YEAR *11X,1013)
7 FORMAT(1H8*TOTAL UNITS ADDED UP TO THIS YEAR *5X,1013)
8 FORMAT(1H8*LOLP = *1,1E4+7/)

1100 RETURN
END

Subroutine LPD(CL Pé, PLOP, ECP, IDL)

Common/Curves/EL(4,1250)
Common/Arrays/X(1250),PLANT(200,14),EMOR(220,4),
*PEAK(200,4),RSV(20),LUP (20,20),NUFI (200,20),1BK (500,3),
*NX (20),NT (20),IX (20),MAINS (200)

Common/One/IYR,PMP,TMAX,NTY,MAXELDC(20,4),EC(20),RSMX(20),
*COSTM(20),PERENE(20,4),AVSPE(20,4),NSTPRE(20,10)

Common/Variab/NP,1D,MAX,1P,IPMAX,NPEXP,JYR,KMAX,ADD

Common/Twog/PLACA(20,14)

Common/Three/PCUM(211,16),HRCUM(4,8)
DOUBLE PRECISION PTCUM,HRCUM

COMPUTES SEASONAL LOLP USING PIECEWISE LINEAR METHOD. MAINTENANCE OUTAGE IS CONSIDERED USING EQUIVALENT CAPACITY ALGORITHM.

SUBROUTINE CALLED:
MAINT1, PWADD, PLOLP

C

C

C

C

DIMENSION P(4), ELDC(4, 1250)
EQUIVALENCE (PTCUM(1), ELDC(1))
K22=22
ISTAR=(JYR-1)*4+1

CALL MAINT1(ECP)

GO TO (100, 200), ID

100 CONTINUE

110 1=P MAX
1L=1 PAX-1+-1
FI I G.T. NPEXP) GO TO 120
IF (N T(1L), LE.0) GO TO 110
JE=NT(1L)
IP=IX(1L)
IN=1P*NPEXP

120 IF (N U F(1L, JYR).LE.O) GO TO 110
JL=NUF(1L, JYR)
IP=IP+NPEXP

130 DO 140 J=1, JE
Y=PLANT(IP, 3)*1.0-EMOR(IN, N)/PK
P=1.0-PLANT(IP, 5)
140 CALL PWADD(Y, P, NMAX1)
IF (1L-1.EQ.NPEXP) GO TO 150
GO TO 110

150 CONTINUE

STORE INFORMATION AFTER THE CONVOLUTION OF THE ORIGINAL SCHEDULED SYSTEM

MAXELDC(JYR, N)=NMAX1
ISTAR+N-1
WRITE(K22*ISTAR+1, 1) (EL(N, K), K=1, MX)

110 CONTINUE

STORE 'EL' AFTER THE CONVOLUTION OF THE SCHEDULED SYSTEM (ORIGINAL SCHEDULED+NEW
CANDIDATES DETERMINED FROM PREVIOUS STUDY)

DO 160 I=1,1250
160 ELDC(N,I)=EL(N,I)
GO TO 500

C

DO 210 I=1,1250
210 EL(N,I)=ELDC(N,I)

C

DO 220 I=1,NPEXP
220 EL(N,I)=ELDC(N,I)

C

IF(INX(I).LE.0) GO TO 220
JE=NX(I)
IP=IX(I)
IN=2*IP
JB=1
IF(IDL.EQ.1.AND.NX(I).GT.1) JB=JE
DO 230 J=JB,JE
230 Y=PLANT(IP,3)*((1.0-EMOR(IN,N))/PK)
220 CONTINUE

FIND LOP

DO 500 A=ECP/PK
500 A=ECP/PK
PLP(N)=PLP(A,N)
IF(PLP(N).GE.CLPA) CLP=PLP(N)
20 CONTINUE

FIND SYSTEM LOP

PLP=PP/FLOAT(NP)

RETURN

END

SUBROUTINE MAINT(ECP)

COMMON/CURVES/EL(4,1250)
COMMON/ARRAYS/X(1250),PLANT(200,14),EMOR(220,4),
*PEAK(120,4),RSV120),IUP(20,20),NUF(200,20),18K(500,3),
*NX(20),NT(20),IX(20),MAINS(200)
COMMON/VARY1,RHS,P,T,NX,NXTYR,MXEDLC(20,4),EC(20),RSMX(20),
*CostM(20),PERE(120,4),AVSPE(20,4),NSTPRE(20,10)
COMMON/VARIAB/NP,IP,MAXI,IPT,IPMAX,NPEXP,JYR,KMAX,ADD

C
C

ALLOCATES MAINTENANCE FOR SCHEDULED SYSTEM AND
C EXPANSION SYSTEM ACCORDING TO FLAG CONTROL ID.
C MAINTENANCE OUTAGE IS STORED IN EMOR(IP,NP),
C WHERE IP=PLANT INDEX, NP=SEASON INDEX.
C
C Definition:
ID=1, FOR 'SCHEDULED' SYSTEM (ORIGINAL SCHEDULED SYSTEM EXP. CANDIDATES DETERMINED FROM PREVIOUS STUDY YEARS).

ID=2, FOR EXPANSION SYSTEM WITH ONE-BLOCK REPRESENTATION.

ID=3, RESCHEDULE MAINTENANCE (ALLOCATE ORIGINAL SCHEDULED SYSTEM FIRST THEN NEW EXP. CANDIDATES).

C

C **DECLARATION OF LOCAL VARIABLES**

DIMENSION AVSP(4), AP(4), EP(20)

RLT=365./FLOAT(NP)

GO TO (100,200,500), ID

C

C ALLOCATE MAINTENANCE FOR SCHEDULED SYSTEM

C AND STORE IN APPROPRIATE SPACE

100 EP(JYR)=0.0

EC(JYR)=0.0

DO 105 J=1,NP

DU 105 I=1,NP

105 EMDR(I,J)=0.0

IB=1

IE=1MAX

DO 110 I=IB,IE

IF(I.GT.NP) GO TO 111

EP(JYR)=EP(JYR)*FLOAT(NP(I))*PLANT(I,3)

110 GO TO 110

111 EP(JYR)=EP(JYR)*FLOAT(NP(I))*PLANT(I,3)

EC(JYR)=EC(JYR)*FLOAT(NP(I))*PLANT(I,3)

110 CONTINUE

120 AVSP(N)=ECP—PEAK(JYR,N)

130 DO 140 L=1,NP

IF(AVSP(L).GT.EMXAS) GO TO 150

GOTO 140

150 EMXAS=AVSP(L)

150 CONTINUE

160 AP(N)=AVSP(N)

200 CONTINUE

C

C ALLOCATE MAINTENANCE FOR EXPANSION SYSTEM
C USING 1-BLOCK REPRESENTATION

ECP = EP(JYR)
DO 210 I = 1, NP
210 ECP = ECP + FLOAT(NX(I)) * PLANT(IX(I), 3)
DO 220 N = 1, NP
AVSP(N) = AP(N)
DO 230 I = 1, NP
IF (NX(I) .EQ. 0) GO TO 220
230 AVSP(N) = AVSP(N) + FLOAT(NX(I)) * PLANT(IX(I), 3)
220 CONTINUE
DO 240 I = 1, NPEXP
IF (NX(IX(I)) .EQ. 0) GO TO 240
240 EMXAS = 0.0
DO 250 N = 1, NP
IF (AVSP(N) .GT. EMXAS) GO TO 250
250 CONTINUE
EMXAS = AVSP(N)
260 CONTINUE
ALLOCATE MAINTENANCE FOR A SINGLE * PSEUDO-PLANT *
WHOSE CAP. IS EQUAL TO THE VALUE OF MIN. RES.
C
PRINT 1
C
INITIALIZATION
DO 405 J = 1, 4
DO 405 I = 1, 220
405 EMOR(J, I) = 0.0
ECP = EP(JYR)
ECP = ECP * PLANT(IPMAX + 1, 3)
DO 410 N = 1, NP
AVSP(N) = ECP - PEAK(JYR, N)
410 CONTINUE
IP = IPMAX + 1
IN = IP + NPEXP
IF (PLANT(IP, 3) .LE. 0.0) GO TO 435
EMXAS = 0.0
DO 420 N = 1, NP
IF (AVSP(N) .GT. EMXAS) GO TO 430
420 CONTINUE
EMXAS = AVSP(N)
430 CONTINUE
EMOR(IN, NN) = PLANT(IP, 4) / RLT
PRINT 3, IP, NN, EMOR(IN, NN)
1 FORMAT(/IH ** MAINTENANCE ALLOCATION ***/)
2 FORMAT(/1X * PLANT NUMBER SEASON EMOR*/)
3 FORMAT(5X*13*11X*I2*4X+F7.5)
AVSP(NN) = AVSP(NN) - PLANT(IP, 3) * EMOR(IN, NN)
435 CONTINUE
440 CONTINUE
630. IB=NPEXP+1
631. IE=IPMAX
632. 495 CONTINUE
633. C
634. DO 440 I=IB,IE
635. IF( I.GT.NPEXP) GO TO 450
636. N=NPEXP-I+1
637. IF(NT(I).EQ.0) GO TO 440
638. IP=1X(I)
639. IN=2*IP
640. GO TO 460
641. 450 IP=1
642. IN=1*NPEXP
643. C
644. 460 IF(PLANT(IP,4).LE.0.0) GO TO 440
645. EMXAS=0.0
646. DO 470 N=1,NP
647. IF(AVSP(N).GT.EMXAS) GO TO 480
648. GO TO 470
649. 480 EMXAS=AVSP(N)
650. NN=N
651. 470 CONTINUE
652. IF(1.LE.NPEXP) GO TO 475
653. K=1-NPEXP
654. MAINS(K)=NN
655. 475 EMOR(IN,NN)=PLANT(IP,4)/RLT
656. C
657. PRINT 3,IP,NN,EMOR(IN,NN)
658. IF(1.GT.NPEXP) GO TO 490
659. AVSP(NN)=AVSP(NN)-PLANT(IP,3)*FLOAT(NT(I))*EMOR(IN,NN)
660. GO TO 440
661. 490 AVSP(NN)=AVSP(NN)-PLANT(IP,3)*FLOAT(NF(IP,JYR))*EMOR(IN,NN)
662. 440 CONTINUE
663. C
664. IF(IB.Eq.1) GO TO 1060
665. IP=IPMAX+1
666. IN=1+NPEXP
667. DO 510 N=1,4
668. AE=AVSP(N)
669. DO 520 I=1,NPEXP
670. 520 AE=AE-FLOAT(IN)*PLANT(IP,3)
671. AVSP(NN)=AE-PLANT(IP,3)*(1.0-EMOR(IN,NN))
672. 510 CONTINUE
673. C
674. C
675. IB=1
676. IE=NPEXP
677. GO TO 495
678. C
679. C
680. 1000 RETURN
681. END
682. C
683. C
684. C
685. C
686. C
687. SUBROUTINE PHADD(Y,P,N,NEWMAX)
688. C
689. C
690. C
691. COMMON/CURVES/EL(4,1250)
692. COMMON/ARRAYS/X(1250),PLANT(200,14),EMOR(220,4),
693. *PEAK(20,4),RSV(20),IUP(20,20),NUF(200,20),IK(500,3),
694. *NX(20),NT(20),I(20),MINS(200)
695. COMMON/UNE/1YR,B,W,TS,NS,NTYR,NSXLOC(20,4),EC(20),RSN(20),
696. *COSTM(20),SOMENW(20,4),AVSP(20,4),NSTPRE(20,10)
697. COMMON/VARIAB/NP,ID,MAX,IP,NPEXP,NEWMAX,ADD
698. C
699. C ****DECLARATION OF LOCAL VARIABLES ***
DIMENSION ELF(1250)
DATA PMAX/0.999/, ELMIN/0.000001/, ONE/1.0/
IF(P.GE.PMAX) GO TO 100
Q=ONE-P
J=1
DO 50 I=1,MAXI
XY=X(I)-Y
IF(XY.LE.X(1)) GO TO 30
J=J+1
50 CONTINUE
25 IF(ABS(XY-X(J))*LE.ELMIN) GO TO 40
ELY=(X(J-1)-XY)/(X(J-1)-X(J))*EL(N,J)+
* (XY-X(J))/(X(J-1)-X(J))*EL(N,J-1)
GO TO 45
30 ELY=ONE
40 ELY=EL(N,J)
45 CONTINUE
ELF(I)=P*EL(N,1)*Q*ELY
MAXIM=MAXI
IF(ELY.LE.ELMIN) GO TO 55
50 CONTINUE
NEWMAX=MAXIM
IF (NEWMAX.GT.MAXI) NEWMAX=MAXI
DO 60 I=1,NEWMAX
EL(N,I)=ELF(I)
60 CONTINUE
100 CONTINUE
RETURN
END

FUNCTION PLOLP(Y,N)
COMMON/CURVES/EL(4,1250)
COMMON/ARRAYS/X(1250), PLANT(200,14), EMOR(220,4),
* PEAK(20,4), RSV(20), IUP(20,20), NUF(200,20), IBK(500,3),
* MX(20), NT(20), IIX(20), MAINS(200)
* COMM/ONE/IYRB, PMP, T, MX, NTYR, MXELDC(20,4), EC(20), RSMX(20),
* COSTM(20), PERENE(20,4), AVSPE(20,4), NSTPRE(20,10)
COMM/VAR1AB/NP, 10, MAXI, 1PT, IPMAX, NPEXP, JYR, KMAX, ADD
YY=Y*1PT
NY=YY
MAXI=MAXI-1
IF(NY.GT.MAXI) GO TO 100
DY=NY-YY
PLOLP=(EL(N,NY)-EL(N,NY+1))*DY + EL(N,NY+1)
RETURN
100 PLOLP=0.0
RETURN
END

SUBROUTINE LOADER

COMMON/CURVES/EL(4,1250)
COMMON/ARRAYS/X(1250), PLANT(200,14), EMOR(220,4),
* PEAK(20,4), RSV(20), IUP(20,20), NUF(200,20), IBK(500,3),
* MX(20), NT(20), IIX(20), MAINS(200)
* COMM/ONE/IYRB, PMP, T, MX, NTYR, MXELDC(20,4), EC(20), RSMX(20),
* COSTM(20), PERENE(20,4), AVSPE(20,4), NSTPRE(20,10)
COMMON/VAR1AB/NP, 10, MAXI, 1PT, IPMAX, NPEXP, JYR, KMAX, ADD
C

DIMENSION M(3),B(500)

DATA K24/24/

C

II=1

DO 30 I=1,IPMAX

B(I)=PLANT(I,1)

IBK(I,I)=1

IBK(I,I+1)=1

IBK(I,I+2)=1

IBK(I,I+3)=-1

IF(I.GT.NPEXP) IBK(I,3)=NUF(I,JYR)

II=II+1

C=PLANT(I,3)-PLANT(I,2)

IF(C.EQ.0.0) GO TO 30

B(I)=PLANT(I,8)

IBK(I,I)=1

IBK(I,I+2)=2

IBK(I,I+3)=-1

IF(I.GT.NPEXP) IBK(I,3)=NUF(I,JYR)

II=II+1

30 CONTINUE

KMAX=II-1

C

ARRANGE *IBK* ARRAY ACCORDING TO FUEL COST

C

N=KMAX

NE=N-1

DO 40 I=1,NE

40 CONTINUE

WRITE(K24*124,901) ((IBK(I,J),J=1,3),I=1,420)

WRITE(K24*124,902) MAINSJ,J=1,200)

901 FORMAT(1Z5*13,12,13))

902 FORMAT(200I2)

RETURN

END

C

SUBROUTINE OPCSTINCUM

C

COMMON/CURVES/EL14

COMMON/ARRAYS/XI1250,PLANT120014),EMOR2204)

COMMON/PEAK20,4),RSV20,1UP20,20),NUF20020),IBK5003)

COMMON/NXI200)1X200),MAINS1200)

COMMON/ONE/JYR,PMP,T,MX,MAX,EC1200),RSM1200)

COMMON/PERENE20,4),AVSPE20,4),NSTPRE20,10)

COMMON/VARIAB/NP,ID,MAXI,IPMAX,NPEXP,MAXKMAX,ADD

COMMON/TABLE/TY120),TINT120)

DOUBLE PRECISION TY,TINT

COMMON/TWDO/PLACA2014)

COMMON/THREE/PICUM121116),HRCUM48)

DOUBLE PRECISION PICUM,HRCUM
840. C
841. DIMENSION E1R(220,2),ITK(500,3),PCUM(201,8,1)
842. DOUBLE PRECISION SYSLUM(8),COEFF(8),SIGMA,SIGIN,
843. *ZL1,ZL2,ZV1,ZV2
844. T=8760./NP
845. IYR=IYRB+JYR-1+900
846. ISTAR=(JYR-1)*4+1
847. C
848. C INITIALIZATION
849. C
850. DO 5 J=1,16
851. DO 5 1=1,211
852. 5 PTCUM(1,J)=0.0
853. C
854. DO 50 J=1,2
855. DU 50 1=1,220
856. 50 ENR(I,J)=0.0
857. C
858. COST(I,JYR)=0.0
859. C
860. C COMPUTE THE MIN. POSSIBLE OPERATION COST USING
861. C A SINGLE 'PSEUDO-PLANT' WHOSE CAPACITY IS
862. C DETERMINED BY THE MIN. RES. MARGIN CALCULATION
863. C AND ALL OTHER PARAMETERS ARE MIN. THESE QUANTITIES
864. C ARE STORED IN THE LAST SPACE OF 'PLANT' ARRAY.
865. C
866. C
867. C DETERMINE EFFECTIVE LOADING ORDER AND
868. C PARAMETERS OF 'PSEUDO'-PLANTS FOR PREVIOUS
869. C STUDY YEARS
870. C
871. C
872. ITK(1,1)=IPMAX+1
873872. ITK(1,2)=1
874. ITK(1,3)=1
875. I1=IPMAX+1
876. KK=1
877. DO 10 I=1,NPEXP
878. J=NPEXP-I+1
879. IF((NT(J)-NX(J)).LT.0)GO TO 10
880. IP=IX(J)
881. KK=KK+1
882. ITK(KK,1)=IP
883. ITK(KK,2)=1
884. ITK(KK,3)=NT(J)-NX(J)
885. 10 CONTINUE
886. C
887. C
888. DO 15 K=1,KMAX
889. IF(IBK(K,3).LE.0)GO TO 15
890. KK=KK+1
891. DO 16 J=1,3
892. 16 ITK(KK,J)=IBK(K,J)
893. 15 CONTINUE
894. NMAX=KK
895. C
896. C
897. C START ENERGY & COST CALCULATION FOR EACH SEASON
898. C
899. DO 60 NSA=1,NP
900. N=NSA
901. C
902. C
903. C CALCULATE PLANT CUMULANTS. CUMULANTS FOR BASE
904. C PIECE ARE STORED IN THE FIRST 8 SPACES,
905. C CUMULANTS FOR THE PEAKING PIECE ARE STORED IN THE
906. C NEXT 8 SPACES
907. C
908. C
909. C CALCULATE PLANT CUMULANTS FOR SCHEDULED SYSTEM
C IFLAG=1
C CALL CUCAL(IFLAG,N)
DO 64 I=1,201
DO 64 J=1,8
PCUM(J,J+1)=0.0
DO 65 K=1,8
PCUM(K,J+1)=SNGL(HRCUM(N,K))
DO 69 L=1,IE
DO 70 K=1,8
PCUM(I,K+1)=SNGL(PCUM(I,K))
GO TO 62
63 KK=K+4
64 PCUM(K,J+1)=SNGL(PCUM(K,J+1))
65 CONTINUE
66 CONTINUE
67 127=ISTAR+N-1
WRITE(27,127) PCUM
C CALCULATE PLANT CUMULANTS FOR SINGLE 'PSEUDO'-PLANT
C IFLAG=2
C CALL CUCAL(IFLAG,N)
C.......INITIALIZE SYSTEM CUMULANTS
DO 70 I=1,8
SYSCUM(I)=HRCUM(N,I)
DO 75 I=1,8
COEFF(I)=0.0
SUMP=0.0
DO 80 KC=1,NMAX
K=KC
C.......IDENTIFY FOR BASE & PEAK BLOCKS
IF(ITK(K,2).EQ.2) GO TO 90
C.......INDICES FOR BASE PIECE
ID1=2
ID3=1
GO TO 100
C.......INDICES FOR PEAKING PIECE
90 ID1=3
93 ID2=2
97 IP=ITK(K,1)
98 IF(IP.GT.NPEXP) GO TO 110
IM=2*(I+1)-ID1
GO TO 120
110 IM=IP+NPEXP
111 IM=IP+NPEXP
112 IM=IP+NPEXP
114 IM=2*(I+1)-ID1
GO TO 120
120 IM=IP+NPEXP
124 IM=IP+NPEXP
127 IM=IP+NPEXP
130 I=1,12
SUMP=SUMP+Y
                   Y1=SUMP-Y
                   Y2=SUMP
                   IF(I\(=\)1) GO TO 140
                   IQ=IP
                   ID2=ITK(K,2)
                   GO TO 150
                   IF(K\(\neq\)1) GO TO 200
                   IQ=ITK(K-1,1)
                   ID2=ITK(K-1,2)

                   C
                   C PLANT CONVOLUTION
                   C
                   150 DO 160 J=1,NCUM
                   J1=J
                   IF(ID2\(\neq\)2) J1=J+8
                   SYSCUM(J)=SYSCUM(J)+PTCUM(IQ,J1)
                   160 CONTINUE
                   197 IF(ID1\(\neq\)3) GO TO 200
                   C
                   C PLANT DECONVOLUTION
                   C
                   180 DO 200 J=1,NCUM
                   J1=J
                   SYSCUM(J)=SYSCUM(J)-PTCUM(IP,J1)
                   200 CONTINUE
                   C
                   200 COEFF(1)=SYSCUM(1)
                   COEFF(2)=SYSCUM(2)
                   SIGMA=D SQRT(SYSCUM(2))
                   SIGIN=SIGMA*SIGMA
                   DO 210 J=3,NCUM
                   SIGIN=SIGIN*SIGMA
                   210 COEFF(J)=SYSCUM(J)/SIGIN
                   Z1=(DBLE(Y1)-COEFF(1))/SIGMA
                   Z2=(DBLE(Y2)-COEFF(1))/SIGMA
                   C
                   EVALUATE VALUES AT Z1 & Z2
                   C
                   CALL VALUE(Z1,COEFF,NCUM,ZV1)
                   CALL VALUE(Z2,COEFF,NCUM,ZV2)
                   C
                   ENR(IP,1U3)=ENR(IP,103)+P*Y*(ZV1+ZV2)/2.0*T
                   130 CONTINUE
                   60 CONTINUE
                   60 CONTINUE
                   C PRINT 1001
                   1001 FORMAT(1H4,* ENERGY PRODUCTION BY PLANT TYPE (MWH))
                   C PRINT 1002
                   1002 FORMAT(1H4,* BASE-BLOCK PEAK-BLOCK*,
                      ** BASE-BLOCK PEAK-BLOCK*/)
                   1031 IE=IPMAX+1
                   1032 DO 290 L=1,IE,2
                   1033 LL=L+1
                   1034 C PRINT 1003, ((ENR(IJ,J),J=1,2),L=L,LL)
                   1035 1003 FORMAT(2X,E14.7,3X,E14.7))
                   1036 290 CONTINUE
                   C
                   1037 C
                   1038 L=0.0
                   1039 DO 1008 J=1,2
                   1040 DO 1008 I=1,41
                   1041 IEF=IE+ENR(I,J)
                   1042 PRINT 1009, IEF
                   1043 1009 FORMAT(1H4,* TOTAL ENERGY PRODUCTION = E14.7, MWH*/
                   C
                   1044 C
                   1045 C CALCULATE COST
                   1046 C
                   1047 DO 300 IE=1,IE
                   1048 FIX=PLANT(1,10)
                   1049 VAB=PLANT(1,11)
1050. DD 310 J=1,2
1051. IF(J.GT.1) GO TO 315
1052. CFL=PLAN1(1,7)
1053. C=PLAN(1,2)
1054. GO TO 320
1055. 315 CFL=PLAN1(1,8)
1056. C=PLAN(1,3)-PLAN(1,2)
1057. 320 FAC=IF(L.GT.LMUT)/1.0+(FAC(1,L)+0.5)
1058. COSTM(JYR)=COSTM(JYR)+COSTM(JYR)*(CFL+VAB+C*FIX)*FAC/1.0E+06
1059. 310 CONTINUE
1060. 300 CONTINUE
1061. PRINT 1004, COSTM(JYR)
1062. 1004 FORMAT(1H , 9  LONER 80UN0 OF OP. COST =  ' ,E 1 4 .7 ,
1063. * MILLION DOLLARS*)
1064. C
1065. C RETRIEVE INFORMATION FOR EXP. CANDIDATES
1066. C
1067. DO 330 J = 1 ,1 4
1068. 330 PLANT(1,J)=PLACA(1,J)
1069. C
1070. C
1071. C
1072. C
1073. RETURN
1074. END
1075. C
1076. C
1077. C
1078. C
1079. C SUBROUTINE JUCAL(IFLAG,N)
1080. C
1081. C
1082. C
1083. COMMON/CURVES/EL(4,1250)
1084. COMMON/ARRAYS/A(X(1250),PLANT(200,14),EMOR(220,4),
1085. *PEAK(20,4),RSV1(20,20),NUF(200,20),IBK(500,3),
1086. *NX(20),NT(20),IX(20),MAINS(200)
1087. COMMON/ONE/YKB,PMP,T,NU,T,NTYR,KELDC(20,4),EC(20),KSMX(20),
1088. *COSTM(20),PERENE(20,6),AVSP(20,4),NSTPRE(20,10)
1089. COMMON/VARIB/NP,10,MAXI,1PT,1PMAX,NPEXP,JYR,KMAX,ADD
1090. COMMON/TABLE/ SY(120),TINT(120)
1091. DOUBLE PRECISION TV,TINT
1092. COMMON/TWO/PLACA(20,14)
1093. COMMON/THREE/PTCUM(211,16),HRCUM(4,8)
1094. DOUBLE PRECISION PTCUM,HRCUM
1095. C
1096. C
1097. C
1098. C
1099. GO TO (100,200),IFLAG
1100. 100 LB=1
1101. LE=1PMAX
1102. GO TO 500
1103. 200 LB=1
1104. LE=1
1105. C
1106. C
1107. 500 CONTINUE
1108. DO 510 L=LB,LE
1109. IF(L.GT.NPEXP) GO TO 515
1110. IF(IFLAG.LE.2) GO TO 517
1111. IF(NT(L)=NX(L),LE.0) GO TO 510
1112. 1=I(L)
1113. GO TO 516
1114. 517 I=IPMAX+1
1115. GO TO 516
1116. 517 I=L
1117. 516 Q=PLANT(1,5)
1118. DO 520 J=1,2
1119. IF(L.GT.NPEXP) GO TO 530
IF (IFLAG.EQ.2) GO TO 530
C=PLANT(1,2)*(1.0-EMOR) + (1,3)*(1.0-EMOR)
GO TO 540
530 C=PLANT(1,2)*(1.0-EMOR(NP&NP))
540 CONTINUE
R(1)=C*U
R(2) = C*C*(1.0-Q)
R(3) = C*C*C*Q*I (1.0 - 0.0 - 2.0*Q)*Q)
R(4) = C*C*C*C*U*(1.0-(2.0 - 6.0*Q)*Q)*Q)
R(5) = C*C*C*C*C*Q*(1.0-(1.0 - 3.0*Q)*Q)*Q)
R(6) = C*C*C*C*C*C*U*(1.0-(10.0 - 20.0*Q)*Q)*Q)
R(7) = C*C*C*C*C*C*C*Q*(1.0-(1.0 - 3.0*Q)*Q)*Q)
R(8) = C*C*C*C*C*C*C*C*Q*(1.0-(1.0 - 3.0*Q)*Q)*Q)
R(9) = C*C*C*C*C*C*C*C*C*Q*(1.0-(1.0 - 3.0*Q)*Q)*Q)
R(10) = C*C*C*C*C*C*C*C*C*C*Q*(1.0-(1.0 - 3.0*Q)*Q)*Q)
R(11) = C*C*C*C*C*C*C*C*C*C*C*Q*(1.0-(1.0 - 3.0*Q)*Q)*Q)

RETURN
END

C

C

C

C

C

SUBROUTINE VALUE(Z,C,NCUM,FRE)

DOUBEL PRECISION Z(ZN(8),F(5,F3,F4,F5,F6,F7,F8,F9,E))
ZINT=Z(#,Z(8),F(5,F3,F4,F5,F6,F7,F8,F9,E))
Z(1)=Z(9.getValue())
Z(2)=Z(3.getValue())
Z(3)=Z(4.getValue())
Z(4)=Z(5.getValue())
Z(5)=Z(6.getValue())
Z(6)=Z(7.getValue())
Z(7)=Z(8.getValue())
Z(8)=Z(9.getValue())

RETURN
END

C

C

C

C

COMMON/CURVES/EL(4,1250)
COMMON/ARKAYS/X(1250),PLANT(200,14),EMOR(220,4),

F3=6.0D00
F4=2.0D00
F5=1.0D00
F6=7.2D00
F7=5.0D00
F8=4.0D00
F9=3.6288D00
NCUM=NCUM-5

CALL TALOOK(Z,ZNT)
C FRE = 1.0000 - ZINTEG
C FRE = FRE + C(3)*ZN(2)/F3
C + - C(4)*ZN(3)/F4
C + -C(13)*C(3)*ZN(5)*10.0000/F6
C
C IF(NUMCU) 5,10,15
C
C FOUR CUMULANT EXPANSION
C
C IF(FRE.LT.0.0000) FRE=0.0000
C IF(FRE.GT.1.0000) FRE=1.0000
C RETURN
C
C FIVE CUMULANT EXPANSION
C
C 10 FRE = FRE + C(5)*ZN(4)/F5
C + + C(3)*C(4)*ZN(6)*35.0000/F7
C + + C(3)*C(3)*ZN(8)*280.0000/F9
C IF(FRE.LT.0.0000) FRE=0.0000
C IF(FRE.GT.1.0000) FRE=1.0000
C RETURN
C
C EIGHT CUMULANT EXPANSION
C
C 15 FRE = FRE + C(5)*ZN(4)/F5
C + + C(6)*ZN(5)/F6
C + + C(7)+10.0000*C(3)*C(4)*ZN(6)/F7
C + + C(8)+50.0000*C(3)*C(13)*C(5)+35.0000*C(4)*C(4)*ZN(7)/F8
C IF(FRE.LT.0.0000) FRE=0.0000
C IF(FRE.GT.1.0000) FRE=1.0000
C RETURN
C
C IF(Z.GE.-5.9D00) GO TO 5
C
C SUBROUTINE TAL00(Z,ZINTEG)
C
C COMMON/TABLE/ Y(120),YINT(120)
C DOUBLE PRECISION AVZ,Y,YINT,Z,ZINTEG,DY,PK,ZINT,FACTOR,
C + R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17
C
C LOOKUP IN NORMAL DISTRIBUTION FUNCTION THE VALUE OF
C THE INTEGRAL OF EXP(-Z^2/2) FROM MINUS INFINITY TO Z.
C INTERPOLATE WITH CUBIC FIT BETWEEN THE 120 DATA POINTS
C COPIED FROM TABLES. THE REFERENCE USED IS: TABLES OF
C NORMAL PROBABILITY FUNCTIONS BY THE U.S. DEPARTMENT
C OF COMMERCE.
C
C DEFINITION OF KEY VARIABLES:
C
C NAME TYPE SIZE MEANING
C Z REAL - NORMALIZED INDEPENDENT VARIABLE
C ZINTEG REAL - INTEGRAL VALUE
C Y REAL 120 NORMALIZED INDEPENDENT VARIABLE
C YINT REAL 120 INTEGRAL VALUES CORRESPONDING TO Y (INPUT DATA)
C K INT. - GRID POINT CORRESPONDING TO Z
C DY REAL - Y INCREMENT
C IF(Z.GE.-5.9D00) GO TO 5
ZINTEG = 0.0000
RETURN
5 CONTINUE
IF ( DABS(Z).GT.1.00-10 ) GO TO 10
ZINTEG = 0.5000
RETURN
10 CONTINUE
DY = Y(Z) - Y(I)
PK = DABS(DY) + 1.0000
K = PK
AVZ = DABS(Z)
R1 = 0.0000
R2 = 0.0000
R3 = 0.0000
R4 = 0.0000
IF ( AVZ.NE.Y(K+1))R1 = AVZ - Y(K+1)
IF ( AVZ.NE.Y(K+2))R2 = AVZ - Y(K+2)
IF ( AVZ.NE.Y(K+3))R3 = AVZ - Y(K+3)
IF ( AVZ.NE.Y(K+4))R4 = AVZ - Y(K+4)
Q12 = Y(K) - Y(K+1)
Q13 = Y(K) - Y(K+2)
Q14 = Y(K) - Y(K+3)
Q23 = Y(K+1) - Y(K+2)
Q24 = Y(K+1) - Y(K+3)
Q34 = Y(K+2) - Y(K+3)
Q21 = -Q12
Q31 = -Q13
Q41 = -Q14
Q23 = -Q23
Q24 = -Q24
Q34 = -Q34
IF ( DABS(R1).GT.1.00-10) GO TO 20
ZINT = YINT(K)*R2*R3*R4/(Q12*Q13*Q14)
GO TO 30
IF ( DABS(R2).LT.1.00-10) GO TO 22
ZINT = YINT(K)*R1*R3*R4/(Q21*Q23*Q24)
GO TO 30
22 IF ( DABS(R3).LT.1.00-10) GO TO 24
ZINT = YINT(K)*R1*R2*R4/(Q31*Q32*Q34)
GO TO 30
24 IF ( DABS(R4).LT.1.00-10) GO TO 26
ZINT = YINT(K+1)*R1*R2*R3/(Q41*Q42*Q43)
GO TO 30
26 ZINT = YINT(K)*R2*R3*R4/(Q12*Q13*Q14)+
YINT(K+1)*R1*R3*R4/(Q21*Q23*Q24)+
YINT(K+2)*R1*R2*R4/(Q31*Q32*Q34)+
YINT(K+3)*R1*R2*R3/(Q41*Q42*Q43)
GO TO 30
30 FACTOR = 1.0000
IF (Z.LT.0.0000) FACTOR = -1.0000
ZINTEG = 0.5000 * ( 1.0000 + FACTOR * ZINT )
IF ( ZINTEG.GT.1.0000 ) ZINTEG = 1.0000
RETURN
END

-----

SUBROUTINE RINPUT

COMMON/CURVES/EL(4,1250)
COMMON/ARRAYS/X1(12250),PLANT(200,14),EMDR(220,4),
PENK(20,4)
COMMON/ONE/IY,KPMP,T,MX,NTYR,MXEDC(20,4),EC(20),RSMX(20),
COMMON/GC(20),PEREN(E(20,4),AVSPE(20,4),NSTRK(20,10)
COMMON/VARIAB/\text{NP, IU, MAXI, IPT, IPMAX, NPEXP, JYR, KMAX, ADD}

COMMON/THREE/PTCUM(211,16), HRCUM(4,8)

DOUBLE PRECISION PTCUM, HRCUM

DATA K25, K26, 25, 26 /

READ LOAD CUMULANTS

DO 10 \text{I} = 1, 4

READ(K26, 1000) \text(HRCUM(I, J), J = 1, 8) 1000 FORMAT(1H7, 13E23.14)

10 CONTINUE

READ PEAK AND ORIGINAL LDC

INITIALIZATION

DO 20 \text{I} = 1, 4

DO 20 \text{J} = 1, MX

20 \text{EL(I, J)} = 0.0

DO 30 \text{I} = 1, 4

READ(K25) \text{1CY, KS}

READU25) \text{DUM, OUM, PEAK(JYR, I), PEAK(JYR, I)}

PRINT 1011, PEAK(JYR, I), PERENE(JYR, I)

1011 FORMAT(1H8, \text{PEAK=}, E12.5, TOTAL ENG=, E12.5)

READ(K25) \text{EL(I, J), J = 1, 500)

30 CONTINUE

NORMALIZATION

DO 40 \text{I} = 1, 4

\text{FNUR} = \text{EL(I, 1)}

DO 45 \text{J} = 1, 500

45 \text{EL(I, J)} = \text{EL(I, J)}/\text{FNUR}

40 CONTINUE

RETURN

END

SUBROUTINE INOUT

COMMON/CURVES/\text{EL(<*, 1250)}

COMMON/ARRAYS/X(1250), PLANT(200, 14), EMOR(220, 4),

\text{PEAK(20, 4), RSV(20), IUP(20, 20), NUF(200, 20), IBR(500, 3),}

\text{MAX(20, 1), JX(20), MAINS(200)}

COMMON/ONE/JYRB, PMP, JX, NTRYR, MXELDC(20, 4), EC(20), RSMX(20),

\text{COSTM(20), PERENE(20, 4), AVSPE(20, 4), NSTPRE(20, 10)}

\text{COMMON/VARIAB/\text{NP, IU, MAXI, IPT, IPMAX, NPEXP, JYR, KMAX, ADD}

\text{DATA L10/10/}

WRITE(L10, 902) PMP, I

902 FORMAT(5E16.8)

IBASYR = JYRB - 1*1900

MAXPLA = IPMAX - NPEXP

WRITE(L10, 900) IBASYR, MX, NPEXP, MAXPLA, IPT, NTRYR, KMAX

900 FORMAT(20I4)

DO 10 \text{J} = 1, NTRYR

WRITE(L10, 900) \text{MXELDC(J, I), I = 1, 4}

10 \text{DO 20 \text{J} = 1, NTRYR}

WRITE(L10, 902) EC(J), RSV(J), RSMX(J), COSTM(J)

20 \text{DO 30 \text{J} = 1, NTRYR}

30 \text{WRITE(L10, 902) \text{PEAK(J, I), I = 1, 4)}

35 \text{WRITE(L10, 902) PERENE(J, I), I = 1, 4)
1400. DO 40 IP=1,NPEXP
1401. 40 WRITE(L10,904) (PLANT(IP,J), J=1,IP)
1402. 904Format(F3.0,F5.0,F3.0,F8.6,F3.0,F7.2,F3.0,F7.2,E10.3,F4.2)
1403. DO 60 J=1,IP
1404. WRITE(L10,900) (NUF(IP,J), IP=1,NPEXP)
1405. 60 WRITE(L10,900) (1UP(IP,J), IP=1,NPEXP)
1406. DO 70 J=1,IP
1407. WRITE(L10,900) (NSTPRE(J,I), I=1,NPEXP)
1408. 70 WRITE(L10,900) (NSTPRE(I,J), I=1,NPEXP)
1409. DO 80 J=1,IP
1410. 80 WRITE(L10,902) (AVSPE(J,N), N=1,4)
1411. RETURN
1412. END
1. DYNO is the optimization submodule of the optimizer module in the CERES code.

2. **Definition of Common Variables**

3. **NAME** | **TYPE** | **SIZE** | **DEFINITION**
--- | --- | --- | ---
4. AVSP | REAL | 20 | Space available for scheduled maintenance
5. CAPABS | REAL | 20 | Scheduled system capacity for each year
6. CLOLP | REAL | 20 | Critical loss-of-load-probability (LOLP)
7. CPLOLP | REAL | 20 | Critical loss-of-load-probability (LOLP)
8. DISRAT | REAL | 20 | Discount rate
9. DX | REAL | 1250 | Normalized increment (DX = 1 / MAXPO)
10. EL | REAL | 4,1250 | Final load duration curve after all
11. EDC | REAL | 1,250 | Auxiliary EDC for LOLP calculation
12. EDC | REAL | 1,250 | Normalized M increment (DX = 1 / MAXPO)
13. ELDC | REAL | 4,1250 | Final load duration curve after all
14. ELDC | REAL | 1,250 | Auxiliary EDC for LOLP calculation
15. ENEDEM | REAL | 20 | Energy demand for each season
16. FATOPT | REAL | 20 | Lower bound for energy cost for each
17. FCR | REAL | 4,1250 | Financial charge rate
18. FCR | REAL | 4,1250 | Fixed charge rate
19. HOURS | REAL | 1,000 | Number of hours per simulation period
20. IBASYS | INT | 10 | Base year calendar year next to
21. IBASYS | INT | 10 | Which the study period begins.
22. IDEXP | INT | 10 | IDEXP[N]=IP means that the I-th plant
23. IPCH | INT | 10 | Number of units by which the artificial
24. ISDOL | INT | 20,10 | ISDOL[N]=IP means that the I-th plant
25. ITIN | INT | 10 | Number of tunnel iterations
26. ITMAX | INT | 10 | Maximum number of units that can be added
27. ITOOT | INT | 10 | Number of Dyno sensitivity analyses
28. LIST | INT | 10,1000 | List of states generated within tunnels
29. MAXADD | INT | 10 | Maximum number of units that can be added
30. MAXAUX | INT | 10 | Each year.
31. MAXALL | INT | 10 | Maximum number of expansion candidates
32. MAXINP | INT | 10 | Defined by input module
33. MAXI | INT | 10 | Maximum number of points in the EDC and
34. MAXNEW | INT | 10 | Maximum number of new candidates.
35. MAXPLA | INT | 10 | Maximum number of plants in the
36. SCHEDULED SYSTEM.
37. MAXPO | INT | 10 | Maximum number of points in the normal
c38. MAYSISI1 | INT | 10 | Number of plant blocks in the loading order
39. MAYSIS11 | INT | 10 | Maximum allowed number of units for each
40. MAYSIS2 | INT | 10 | Year and exp. candidate
41. MAYSIS2 | INT | 10 | Maximum allowed number of units for each
42. MAYSIS3 | INT | 10 | Year and exp. candidate
43. MXELDC | INT | 20,4 | Maximum number of points in each of the
44. MXYEAR | INT | 20,4 | Scheduled system EDC's for the study period
45. MXYEAR | INT | 20,4 | Maximum number of years in study period
46. MXEXP | INT | 10 | MEANS THAT THE IP-TH INPUT
47. NEXPID | INT | 10 | MODULE PLANT IS THE I-TH PLACA PLANT.
48. IP | INT | 10 | If I<0 the plant is not used in the
49. IP | INT | 10 | Current sensitivity analysis
NORDER INT. 420,3
NORDER(I,J): PLANT LOADING ORDER.

I=1,420: PLANT BLOCK LOADING ORDER
J=1: INDICATES THE POSITION OF THE PLANT
IN THE SCHEDULED OR NEW CANDIDATE FILES (PLANTS & PLACA RESPECTIVELY)
J=2: PLANT BLOCK
J=3: IF >=0, IT INDICATES THE NUMBER OF UNITS OF PLANT (I,J=1) IN THE SCHEDULED SYSTEM.
J=3: IF < 0 (USUALLY -1), INDICATES THAT THE CORRESPONDING PLANT (I,J=1) IS A NEW CANDIDATE.
J=2: THE NUMBER OF UNITS FOR THIS PLANT IS FOUND FROM THE STATE UNDER EXAMINATION.

NSTPEK INT. 20,10
YEARLY STATES USED FOR THE MINIMUM RESERVE MARGIN CALCULATION IN PREP SUBMODULE. THEY ARE USED HERE FOR THE FIRST ESTIMATE OF UBOUND.
PLANTS REAL 20,10
YEARLY PEAKS FOR THE 20 YEARS OF THE STUDY PERIOD

PLACA REAL 10,14
PLANT CANDIDATES.
I: PLANT NUMBER IN THE SAME ORDER AS IN THE NEW CANDIDATE PLANTS FILE.
J=1 NUMBER OF UNITS BELONGING TO THIS PLANT CODE
J=2 BASE CAPACITY IN MW
J=3 MAX CAPACITY IN MW
J=4 MAINTENANCE REQUIREMENT IN DAYS PER YEAR
J=5 FORCED OUTAGE RATE
J=6 CAPITAL COST IN $/KW
J=7 BASE FUEL COST IN $/MWH
J=8 MAX OPERATING FUEL COST IN $/MWH
J=9 ECONOMIC PLANT LIFE IN YEARS
J=10 FIXED OPERATION AND MAINTENANCE COSTS IN $/MW.YEAR
J=11 VARIABLE OPERATION AND MAINTENANCE COSTS IN $/MWH
J=12 SALVAGE VALUE IN THOUSAND DOLLARS
J=13 ANNUAL CAPITAL COST ESCALATION RATE
J=14 FUEL COST ESCALATION RATE

PLANTS(1,J): SCHEDULED SYSTEM PLANTS
I: PLANT NUMBER IN THE SAME ORDER AS IN THE SCHEDULED SYSTEM FILE.
J=1 NUMBER OF UNITS BELONGING TO THIS PLANT CODE
J=2 BASE CAPACITY IN MW
J=3 MAX CAPACITY IN MW
J=4 MAINTENANCE REQUIREMENT IN DAYS PER YEAR
J=5 FORCED OUTAGE RATE
J=6 CAPITAL COST IN $/KW
J=7 BASE FUEL COST IN $/MWH
J=8 MAX OPERATING FUEL COST IN $/MWH
J=9 ECONOMIC PLANT LIFE IN YEARS
J=10 FIXED OPERATION AND MAINTENANCE COSTS IN $/MW.YEAR
J=11 VARIABLE OPERATION AND MAINTENANCE COSTS IN $/MWH
J=12 SALVAGE VALUE IN THOUSAND DOLLARS
J=13 ANNUAL CAPITAL COST ESCALATION RATE
J=14 FUEL COST ESCALATION RATE

PLANTS(1,I,P): SCHEDULED SYSTEM PLANT CUMULANTS
I=1,4 BASE BLOCK CUMULANTS
I=5,6 PLANT CUMULANTS
J=1,4 YEARLY SEASON
FILE DESCRIPTION

NAME UNIT TYPE DEFINITION

INOUT 10 SEQUEN. INPUT DATE READ BY READIN SUBROUTINE

DPOUT 11 SEQUEN. DEBUG OUTPUT

STATES 14 DlR.AC. YEARLY ACCEPTED STATES

LDFIN 22 DIR.AC. SCHEDULED SYSTEM FINAL ELD.0'S

PSCH 23 DIR.AC SCHEDULED PLANT DATA

LORD 24 DIR.AC. ECONOMIC LOADING ORDER

PTCUM 27 DIR.AC. SCHEDULED PLANT CUMULANTS

SOL 28 DIR.AC. FINAL SOLUTIONS AFTER EACH TUNNEL ITERATION

DEFINE FILE 14(20,3000,L,114),22(80,5004,L,122),23(250,80,L,123),
+ 24(100,1000,L,124),27(80,6430,L,127),20(400,50,L,129),
+ 15(120,4000,L,115)

COMMON /ONE/ CLOLP,CPL0LP,CLOLP,DISKAT,DX,FCR,HOURS,IBASYR,ITIN,ITMAX,
+ ITOUT,MAXALL,MAX1,MAXINP,MAXOR,MAXNEW,MAXPLA,
+ MAXQ,MAXALL,MAYEAR,RESMAR,UBOUND

COMMON /THO/ AVSP(20*4),CAPABS(20),ENEDEM(20*4),FATQPE(20),
+ IDEXP(10),1PCH(10),1SOL(20,10),1N1DTH(10),
+ L0BH(20,10),MAXUN(20,10),MAXXD(10),MINUN(20,10),
+ NEXP1(10),NORDER(420,3),NSTPRE(20,10),LLOBH(20,10),
+ NUPB(20,10),PSEAKS(20,4),PLAC0(10,14),PLANT(20,14),
+ RESMAX(20),RESMIN(20),SOL(20,2),UNE(20),AINS(200)

COMMON /THREE/ EL(1250),ELDC(4,1250),ELF(1260),X(1250),
+ MXELDC(20,4)

COMMON /FOUR/ LIST(10,1000)

COMMON /FIVE/ PTCUM(20,8,4),PTCUXM(10,8),ROM(20,4)

COMMON /SIX/TY(120),TINT(120)

DOUBLE PRECISION TY,TINT

DIMENSION NTCUR(10),NSTSAE(10),ITEMP(3,1000),ITUN(10,3),
+ NONPT(20,3),NUMACX(20),OPTIMA(20,4),RESCAP(20,10),TEMP4(1000),
+ KS1(250,3),KS2(250,3),
+ SK1(250,4),SK2(250,4)

EQUIVALENCE (LIST(1),ITEMP(1),LIST(3)),ITEMP(1),
WRITE(6,926)

926 FORMAT(1H1//)
READ INPUT DATA
CALL READIN
IRES = 0
ITOUT = 0
IPFLAG = 0
ICFLAG = 0
CALL UFIRST

CALL CHOOSE(ISTOP)
IF (ISTOP.GT.0) GO TO 600

ITIN = 1
CALL CHANEL(MAXSTA,ICFLAG)
IF (ICFLAG.GT.0) GO TO 5

CONTINUE

DO 12 ITIN = 1,20
ONE(N) = 0.0
DO 13 I = 1,10
RESCAP(N) = 0.0
13 ISOL(N) = 0
CONTINUE

DEFINE THE MAXIMUM NUMBER OF STATES THAT CAN BE GENERATED EACH YEAR BASED ON THE MAXIMUM PLANT ADDITION FOR THE STUDY PERIOD.
MAX14 = 250

SET BASE YEAR ORIGIN STATE
DO 10 I = 1,MAXNEW
NSTATE(I) = 0
CONTINUE

PACK NSTATE
NSTATE = 0
OBJOK1 = 0.0
NOR1 = 1

SET BEGINNING YEAR
NY = 1
KYEAR = IBSYR + NY
WRITE(6,900) KYEAR
FORMAT(120012)
WRITE(11,902) NY,14)
WRITE(11,902) (11NORDER(J,K),K=1,3) ,J=1,100)
FORMAT(20012)
WRITE(11,902) (11NORDER(J,K),K=1,3) ,J=1,100)

SET TUNNELS FOR NY
CALL TUNNEL(NSTATE,NY,ITUN)
GENERATE ALL STATES ALLOWABLE BY THE TUNNELS AND STORE THEM IN THE "LIST" ARRAY.
CALL STAGEN(ISTUN,MXSTA)

INITIALIZE AUXILIARY VARIABLES
NSTOLD = 0
OLOLP = 1.0
C FIND THE ACCEPTABLE STATES FOR THE FIRST YEAR OF
C THE STUDY PERIOD.

C DO 100 N=1,MAXSTA
C WRITE(6,903) NY,N
C WRITE(11,903) NY,n
C 903 FORMAT(5H YEAR ,14,7H STATE ,14)

C COPY STATE N FROM THE "LIST" ARRAY INTO "NSTCUR"

DO 150 N=1,MAXNEW
150 NSTCUR(N) = LIST(IN)

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)

C CALCULATE CURRENT SYSTEM CAPACITY
SYSCAP = CAPABS(NY)

DO 20 1=1,MAXNEW
20 SYSCAP = SYSCAP+PLACA(1)*NSTCUR(1)

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)

C CHECK RESERVE MARGINS AND REJECT UNACCEPTABLE STATES.
IF (SYSCAP .LT. RESMIN(NY)) GO TO 100
IF (SYSCAP .GT. RESMAX(NY)) GO TO 100

C CALCULATE AND CHECK LOLP. REJECT UNACCEPTABLE STATES.
CALL FLOLP(NSTCUR, CULOLP, NSTOLD, OLOLP, NY)

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)

C CALCULATE ENERGY COST
CALL OPERC(NSTCUR, NY, OPEC, IRES)

C CALCULATE THE OBJECTIVE FUNCTION
CALL OBJFUN(NSTCUR, NY, OPEC, NSTATE, CAPCO, OBJOR1, OBJ,
  + SALV, RESCAP, IRES)

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)

C EXERCISE FATHOMING
CALL FATHOM(NSTCUR, NY, OPEC, IFATH)

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)
WRITE(11,*) (NY,OPEC,CAPCO,OBJOR1,OBJ

C REACHING THIS POINT MEANS THAT THE STATE UNDER CONSIDERATION
C IS ACCEPTABLE.

C PACK AND STORE IN DIRECT ACCESS FILE 14.
PACK = PACK+1

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)

CALL PACK(NSTCUR, NST, NY)

C REACHING THIS POINT MEANS THAT THE STATE UNDER CONSIDERATION
C IS ACCEPTABLE.

PACK = PACK+1

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)

CALL PACK(NSTCUR, NST, NY)

C REACHING THIS POINT MEANS THAT THE STATE UNDER CONSIDERATION
C IS ACCEPTABLE.

PACK = PACK+1

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)

CALL PACK(NSTCUR, NST, NY)

C REACHING THIS POINT MEANS THAT THE STATE UNDER CONSIDERATION
C IS ACCEPTABLE.

PACK = PACK+1

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)

CALL PACK(NSTCUR, NST, NY)

C REACHING THIS POINT MEANS THAT THE STATE UNDER CONSIDERATION
C IS ACCEPTABLE.

PACK = PACK+1

WRITE(11,*) (NSTCUR(INN),INN=1,MAXNEW)

CALL PACK(NSTCUR, NST, NY)
350. SK2(K4,4) = OBJ
351. C
352. C STORE THE CURRENT STATE L0LP FOR OLD L0LP USAGE
353. C
354. C STORE THE CURRENT STATE LOLP FOR OLD L0LP USAGE
355. C
356. C
357. C
358. C
359. C
360. C
361. C
362. C
363. C
364. C
365. C
366. C
367. C
368. C
369. C
370. C
371. C
372. C
373. C
374. C
375. C
376. C
377. C
378. C
379. C
380. C
381. C
382. C
383. C
384. C
385. C
386. C
387. C
388. C
389. C
390. C
391. C
392. C
393. C
394. C
395. C
396. C
397. C
398. C
399. C
400. C
401. C
402. C
403. C
404. C
405. K4 = N-NBEG+1
406. NSTORI = KSL(K4,1)
407. OBJSTORI = SK1(K4,4)
408. CALL UNPACK1NSTORI(NSTATE,NY,IPFLAG)
409. C
410. C
411. C
412. C
413. C
414. C
415. C
416. NSTOLD = NSTORI
417. NORTI = N-NBEG+1
418. CALL STAGEN(NITUN, MAXSTA)
419. C
420. 901 FORMAT(3110, F11.6, 3F13.2)
421. C
422. WRITE(14*114) KS2
423. 115 = 1
424. WRITE(15*115) SK2
425. C
426. C
427. C
428. C
429. C
430. C
431. C
432. C
433. C
434. C
435. C
436. C
437. C
438. C
439. C
440. C
441. C
442. C
443. C
444. C
445. C
446. C
447. C
448. C
449. C
500. 904 FORMAT(5H YEAR=14, 27H NUMBER OF ACCEPTED STATES *14)
501. C
502. WRITE(11,904) NY,NACCEP
503. C
504. C
505. C
506. C
507. C
508. C
509. C
510. C
511. C
512. C
513. C
514. C
515. C
516. C
517. C
518. C
519. C
520. C
521. C
522. C
523. C
524. C
525. C
526. C
527. C
528. C
529. C
530. C
531. C
532. C
533. C
534. C
535. C
536. C
537. C
538. C
539. C
540. C
541. C
542. C
543. C
544. C
545. C
546. C
547. C
548. C
549. C
550. C
551. C
552. C
553. C
554. C
555. C
556. C
557. C
558. C
559. C
560. C
561. C
562. C
563. C
564. C
565. C
566. C
567. C
568. C
569. C
570. C
571. C
572. C
573. C
574. C
575. C
576. C
577. C
578. C
579. C
580. C
581. C
582. C
583. C
584. C
585. C
586. C
587. C
588. C
589. C
590. C
591. C
592. C
593. C
594. C
595. C
596. C
597. C
598. C
599. C
600. C
601. C
602. C
603. C
604. C
605. K4 = N-NBEG+1
606. NSTORI = KSL(K4,1)
607. OBJSTORI = SK1(K4,4)
608. CALL UNPACK1NSTORI(NSTATE,NY,IPFLAG)
609. C
610. C
611. C
612. C
613. C
614. C
615. C
616. NSTOLD = NSTORI
617. NORTI = N-NBEG+1
618. CALL STAGEN(NITUN, MAXSTA)
619. C
620. WRITE(11,905) NSTORI,MADTA
C CHECK RESERVE MARGINS
IF (SYSCAP .LT. RESMIN(NYNEXT)) GO TO 250
IF (SYSCAP .GT. RESMAX(NYNEXT)) GO TO 250
C PACK CURRENT STATE
CALL PACK(NSTCUR, NST, NYNEXT)
C IF THIS IS NOT THE FIRST TIME STATES IN STAGE NYNEXT ARE
SIMULATED, CHECK IF THE SAME STATE WAS SIMULATED BEFORE.
IF (IN.EQ.BADBEGIN beginner.GT.EQ.1) GO TO 240
C SEARCH PREVIOUSLY GENERATED STATES IN NYNEXT STAGE
NSTART = 1
NNS1P = NUMACC(NYNEXT)
DO 230 NSTART = NNS1P
K4 = I
NST = KS2(K4*1)
NSTOR = KS2(K4*2)
NO1 = KS2(K4*3)
COLUMNS = SK2(K4*1)
OPECNN = SK2(K4*2)
CAPNN = SK2(K4*3)
OBJNN = SK2(K4*4)
IF (NST .GT. NST) GO TO 235
230 CONTINUE
C DROPPING THROUGH THE LOOP MEANS THAT CURRENT STATE WAS NOT
GENERATED OR ACCEPTED FOR RESERVE MARGINS BEFORE
GOTO 240
235 CONTINUE
C CHECK FOR LOLP
IF (COLUMNS .GT. CLULP) GO TO 250
C THIS STATE'S LOLP IS ACCEPTABLE
CALL OBJFUN(NSTCUR, NYNEXT, OPEC, NST, CAPCO, OBJOR1,
OBJ, SALV, RESCAP, IRES)
C IF OBJ .GT. OBJNN GO TO 250
KS2(K4*1) = NST
KS2(K4*2) = NSTOR
KS2(K4*3) = NO1
SK2(K4*1) = COLUMNS
SK2(K4*2) = OPEC
SK2(K4*3) = CAPCO
SK2(K4*4) = OBJ
GOTO 250
240 CONTINUE
C CHECK LOLP
CALL FLOLP(NSTCUR, COLUMNS, NSTOLD, OLOLP, NYNEXT)
IF (COLUMNS .GT. OLOLP) GO TO 245
C CALCULATE OBJECTIVE FUNCTION
CALL OPERCO(NSTCUR, NYNEXT, OPEC, NST, CAPCO, OBJOR1,
OBJ, SALV, RESCAP, IRES)
C GO TO 247
OPEC = UNBOUND*2.0
CAPCO = OPEC
OBJ = OPEC
247 CONTINUE
REACHING THIS POINT MEANS THAT THE STATE UNDER CONSIDERATION IS ACCEPTABLE EXCEPT FOR FATHOMING. STATES WITH BAD RELIABILITY WILL BE FATHOMED. STORE THIS STATE AND ITS CHARACTERISTIC VARIABLES IN DIRECT ACCESS FILE 14.

```
NACCEP = NACCEP + 1
K4 = NACCEP
KS2(K4,1) = NST
KS2(K4,2) = NSTORI
KS2(K4,3) = NURI
SK2(K4,1) = CULOLP
SK2(K4,2) = OPEC
SK2(K4,3) = CAPCO
SK2(K4,4) = OBJ
NSTOLD = NST
OLOLP = CULOLP
200 CONTINUE
C RECORD THE NUMBER OF ACCEPTED STATES
NUMACC(NYNEXT) = NACCEP
WRITE(6,908) N, NYNEXT, NACCEP
908 FORMAT(3H N = ,I4, 8H NYNEXT =, 14, 17H NACCEP,PAST 250 =, 14)
C 300 CONTINUE
C EXERCISE FATHOMING FOR THE STATES IN YEAR NYNEXT
WRITE(11,912) NYNEXT, NACCEP
912 FORMAT(1H THE UNFATHOMED STATES FOR YEAR ',14,' ARE ',14)
1F (NACCEP .LT. 1) GO TO 460
NSTART = 1
NEND = NUMACC(NYNEXT)
NALL = 0
DO 350 N=NSTART, NEND
K4 = N
NST = KS2(K4,1)
NSTORI = KS2(K4,2)\nNURI = KS2(K4,3)
CULOLP = SK2(K4,1)
OPEC = SK2(K4,2)
CAPCO = SK2(K4,3)
OBJ = SK2(K4,4)
CALL UNPACK(NST, NSTCUR, NYNEXT, 1PFLAG)
CALL FATHOM(NSTCUR, NYNEXT, OBJ, IFATH)
IF (IFATH .EQ. 0) GO TO 350
WRITE THE ACCEPTABLE STATES IN THE "ITEM" AND "TEMP" ARRAYS THAT OCCUPY THE SAME CORE SPACE AS THE "LIST" ARRAY
NALL = NALL + 1
ITEM(1,NALL) = NST
ITEM(2,NALL) = NSTORI
ITEM(3,NALL) = NURI
TEMP(1,NALL) = CULOLP
TEMP(2,NALL) = OPEC
TEMP(3,NALL) = CAPCO
TEMP(4,NALL) = OBJ
350 CONTINUE
C STATES ACCEPTED AFTER FATHOMING
WRITE(6,909) NALL, NYNEXT
909 FORMAT(1H ,14, STATES WERE ACCEPTED AFTER FATHOMING ,14)
IF (NALL .EQ. 1) GO TO 480
DO 360 N=1,NALL
WRITE(11,910) (ITEM(I,N), I=1,3), (TEMP(I,N), I=1,4)
360 CONTINUE
```

560. SK2(N+2) = TEMP(2*N)
561. SK2(N+3) = TEMP(3,N)
562. SK2(N+4) = TEMP(4,N)
563. 360 CONTINUE
564. II4 = NYNEXT
565. WRITE(14*114) KS2
566. 115 = NYNEXT
567. WRITE(15*115) SK2
568. C
569. C
570. 910 FORMAT(I1M ,3110, F11*8, 3F13.2)
571. C SET THE FINAL VALUE FOR THE NUMBER OF ACCEPTED STATES UP TO
572. C STAGE NYNEXT
573. NUMACC(NYNEXT) = NALL
574. C
575. C GO TO NEXT STAGE
576. NY = NY+1
577. IF (NY .LT. MXYEAR) GO TO 200
578. C
579. C REACHING THIS POINT MEANS ALL STAGES(YEARS) WERE EXAMINED.
580. C FIND THE MINIMUM OBJECTIVE FUNCTION AND TRACE BACK THE
581. C OPTIMUM SOLUTION
582. C
583. OBJMIN = UBOUND
584. DO 370 N=1,NALL
585. IF (OBJMIN .LT. TEMP(4,N)) GO TO 370
586. OBJMIN = TEMP(4,N)
587. NFIND = N
588. 370 CONTINUE
589. C
590. DO 380 I=1,4
591. OPTIMA(MXYEAR*I) = TEMP(1,NFIND)
592. IF (I .GT. 3) GO TO 360
593. NOPTIMA(MXYEAR*I) = TEMP(1,NFIND)
594. 380 CONTINUE
595. WRITE(6,911) TEMP(4,NFIND)
596. WRITE(6,910) (TEMP(I,NFIND), I=1,3), (TEMP(I,NFIND), I=1,4)
597. WRITE(11*910) (TEMP(I,NFIND), I=1,3), (TEMP(I,NFIND), I=1,4)
598. 911 FORMAT(I1M ,*/ THE FINAL YEAR OBJECTIVE FUNCTION IS: **E13.6)
599. C
600. C TRACE BACK THE OPTIMUM SOLUTION
601. NSTOP = MXYEAR-1
602. DO 390 N=1,NSTOP
603. NN = NSTOP - N + 1
604. 115 = NN
605. READ(15*115) SK1
606. C
607. 114 = NN
608. READ(14*114) KS1
609. KOR1 = NOPTIMA(NN+1,3)
610. DU 385 I=1,4
611. OPTIMA(NN,1) = SK1(KOR1,1)
612. IF(I .GT. 3) GO TO 385
613. NOPTIMA(NN,1) = KS1(KOR1,1)
614. 385 CONTINUE
615. 390 CONTINUE
616. 390 CONTINUE
617. C
618. C WRITE THE OPTIMUM SOLUTION
619. WRITE(6,917) ITIN
620. WRITE(11*917) ITIN
621. C
622. I28 = (ITIN-1) * 20 + 1
623. DU 400 N=1,MXYEAR
624. NSF = NOPTIMA(N,1)
625. CALL UNPACK(NST, NSTATE,N,IPFLAG)
626. DU 395 I = 1, MAXNEW
627. 395 ISOL(N,1) = NSTATE(1)
628. SOL(N,1) = OPTIMA(N,1)
629. SOL(N,2) = OPTIMA(N,4)
C WRITE(6,*)(NSTATE(I), I=1,MAXNEW), (NOPTIM(N,1), I=1,3),
C +(OPTIMA(N,1), I=1,4)
C WRITE(6,*)(NSTATE(I), I=1,MAXNEW), (NOPTIM(N,1), I=1,3),
C +(OPTIMA(N,1), I=1,4)
C WRITE(28*128,913)(ISOL(N,1), I=1,6), SOL(N,1), SOL(N,2)
C 400 CONTINUE

C 913 FORMAT(614,F13.9,E13.7)
C
C UDJUST THE OBJECTIVE FUNCTION UPPER BOUND
C UBOUND = OPTIMA(MAXYEAR,4) * 1.1
C WRITE(6,920) UBOUND
C WRITE(6,920) UBOUND
C 920 FORMAT(* NEW UPPER BOUND FOR FATHOMING IS ,F13.2/)
C
C C CHECK WHETHER THE OPTIMUM SOLUTION IS CONSTRAINED BY THE ARTIFICAL LOWER OR UPPER BOUNDS FOR EACH YEAR IN THE STUDY PERIOD
C C STORE UNCHANGED LOWER IN LLowb
C DO 406 N = 1,MAXYEAR
C DO 404 I = 1,MAXNEW
C LLowb(N,1) = LOMB(N,1)
C 404 CONTINUE
C NYFLAG = 21
C DO 450 N = 1,MAXYEAR
C NY = MAXYEAR-N +1
C LOFLAG = 0
C NUFLAG = 0
C NST = NOPTIM(NY,1)
C CALL UNPACK(NST,NSTATE,NY,IPFLAG)
C WRITE(6,*)(NY,LOFLAG,NUFLAG,NYFLAG)
C WRITE(6,*)(NSTATE(I), I=1,MAXNEW)
C WRITE(6,*)(LLowb(NY,1), I=1,MAXNEW)
C WRITE(6,*)(NUPB(NY,1), I=1,MAXNEW)
C C CHECK ARTIFICIAL LOWER BOUND
C DO 420 IP=1,MAXNEW
C IP = IDEXP(IP)
C IF (LLowb(NY,IP) .LE. MINUN(NY,IP)) GO TO 410
C IF (NSTATE(IP) .LE. MINUN(NY,IP)) GO TO 410
C IF (NSTATE(IP) .NE. LLowb(NY,IP)) GO TO 410
C LOdB(NY,IP) = LOdB(NY,IP) - IPCH(IP)
C IF (LOdB(NY,IP) .LT. MINUN(NY,IP)) LOdB(NY,IP) = MINUN(NY,IP)
C LOFLAG = IP
C NUPb(NY,IP) = LOdB(NY,IP) + IWIDTH(IP) -1
C IF (NUPb(NY,IP) .GT. MAXUN(NY,IP)) NUPb(NY,IP) = MAXUN(NY,IP)
C 410 CONTINUE
C IF (LOFLAG.GT.0) WRITE(6,915) NY
C IF (NUFLAG.GT.0) WRITE(6,916) NY
C IF (LOFLAG.GT.0.OR.NUFLAG.GT.0) WRITE(6,919) NY,
C💚 (LLowb(NY,1), I=1,MAXNEW), (NUPB(NY,1), I=1,MAXNEW)
C 420 CONTINUE
C IF (LOFLAG.GT.0) WRITE(6,915) NY
C IF (NUFLAG.GT.0) WRITE(6,916) NY
C IF (LOFLAG.GT.0.OR.NUFLAG.GT.0) WRITE(6,919) NY,
C💚 (LLowb(NY,1), I=1,MAXNEW), (NUPB(NY,1), I=1,MAXNEW)
REPACK STOR1 D STATES WITH THE NEW LOWB
NY = NYFLAG - 1
IPFLAG = 1
DO 465 N = 1,NY
K4 = N
READ(14*114) KS1
II5 = N
READ(15*115) SK1
NBEG = 1
NSTOP = NUMACC(N)
DO 460 LL = NBEG,NSTOP
K4 = LL
NST = KS1(K4*1)
NSTORI = KS1(K4*2)
NORI = KS1(K4*3)
CULOP = SK1(K4*1)
OPEC = SK1(K4*2)
CAPCO = SK1(K4*3)
OBJ = SK1(K4*4)
CALL UNPACK(NST,NSTCUR,N,IPFLAG)
CALL UNPACK(INSTOR1,NSTATE,N,IPFLAG)
WRITE(11*,#) NST,NSTCUR
C WRITE(11*,#) NSTOR1,NSTATE
CALL PACK(NSTCUR,NST)
CALL PACK(NSTATE,NSTOR1,N)
K52(K4+1) = NST
K52(K4+2) = NST
770. KS2(K4,3)  =  NDI
771. SK2(K4,1) =  CLOLP
772. SK2(K4,2) =  OPEC
773. SK2(K4,3) =  CAPO
774. SK2(K4,4) =  OBJ
775. C  WRITE(11,*1) NST,NSTOK
776. 400 CONTINUE
777. 114 = N
778. WRITE(14,**114) KS2
779. 115 = N
780. WRITE(15,**115) SK2
781. 465 CONTINUE
782. IFLAG = 0
783. GO TO 200
784. C
785. 470 WRITE(6,**918) ITMAX
786. 917 FORMAT(' THE NUMBER OF ITERATION IS:*14*)
787. 918 FORMAT(' MORE THAN:*14,* ITERATIONS HAVE BEEN RUN. PROGRAM STOPS*)
788. GO TO 500
789. 480 NACALL = NACEP
790. IF ( NACEP.GE.1 ) NACALL = NALL
791. NXYEAR = NNEXT + IBASYR
792. WRITE(6,**922) NXYEAR,NACALL
793. 922 FORMAT(' THE NUMBER OF STATES DEFINED IN YEAR:*15,
794. * IS *15'/' PROBABLE REASONS */
795. * A. SMALL MINIMUM RESERVE MARGIN*/
796. * B. THE SIZE OF THE EXPANSION PLANT TYPES SELECTED IS /
797. * TOO SMALL TO PROVIDE ADEQUATE RELIABILITY*/
798. ** **** RERUN WITH PROPER ADJUSTMENTS *****)
799. ITOUT = 0
800. GO TO 5
801. 500 CONTINUE
802. C
803. IF ( ISTOP.LE.0 ) GO TO 5
804. C
805. 600 STOP
806. END
807. C
808. C
809. C
810. C
811. SUBROUTINE CHANEL (MAXSTA,ICFLAG)
812. C
813. C
814. C
815. C
816. C
817. C
818. C
819. C
820. C
821. C
822. C
823. C
824. C
825. C
826. C
827. C
828. C
829. C
830. C
831. C
832. C
833. 5 IWIDTH(1) = 3
834. C
835. NID = MAXNEW
836. C
837. C
838. 60 TO (10,20,30,40,50,60), NID
839. 10 IWIDTH(1) = 6
840.        MAXSTA = 6
841.        GO TO 70
842.        20 DO 25 I=1,MAXNEW
843.        25 WIDTH(I) = 6
844.        MAXSTA = 30
845.        GO TO 70
846.        30 WIDTH(I) = 4
847.        WIDTH(2) = 4
848.        WIDTH(3) = 5
849.        MAXSTA = 80
850.        GO TO 70
851.        40 WIDTH(I) = 3
852.        WIDTH(2) = 3
853.        WIDTH(3) = 4
854.        WIDTH(4) = 5
855.        MAXSTA = 160
856.        GO TO 70
857.        50 WIDTH(I) = 3
858.        DO 55 J=2,MAXNEW
859.        55 WIDTH(MAXNEW) = 5
860.        MAXSTA = 960
861.        GO TO 70
862.        DO 65 I = 1,MAXNEW
863.        65 WIDTH(I) = 3
864.        60 WIDTH(MAXNEW) = 4
865.        MAXSTA = 972
866.        70 CONTINUE
867.        C DEFINE CHANNEL INCREMENT
868.        DO 80 I = 1,10
869.        80 IPC(I) = WIDTH(I) - 2
870.        C DEFINE THE ORIGINAL ARTIFICIAL VCHANNELS.
871.        DO 100 NY=1,MAXYEAR
872.        90 NUPB(NY,I) = IWIDTH(I) - 1
873.        100 CONTINUE
874.        C RESTRUCTURE AND PRINT THE EXPANSION CANDIDATE
875.        DO 104 IP = 1,MAXNEW
876.        104 NEXP(I) = -2
877.        DO 120 IP = 1,MAXNEW
878.        120 IF (LOWB(NY,IP).GE.MAXMIN(NY,IP)) GO TO 125
879.        LOWB(NY,IP) = LOWB(NY,IP) + NEXP(I)
880.        IF (SYSDIF.GT.RM) GO TO 124
881.        GO TO 120
882.        PK = 0.0
883.        DO 108 J = 1,4
884.        IF (PK.GT.PEAKS(NY,J)) GO TO 108
885.        PK = PEAKS(NY,J)
886.        108 CONTINUE
887.        C YY = NY
888.        SYSDIF = CAPABS(NY) - PK
889.        DO 109 I = 1,MAXNEW
890.        109 CONTINUE
891.        RM = RESMIN(NY) * 0.95
892.        IF (SYSDIF.GT.RM) GO TO 130
893.        DO 120 IP = 1,MAXNEW
894.        120 IF (LOWB(NY,IP).GE.MAXMIN(NY,IP)) GO TO 125
895.        LOWB(NY,IP) = 1 + LOWB(NY,IP)
896.        SYSDIF = SYSDIF + NEXP(I) * LOWB(NY,IP)
897.        IF (SYSDIF.GT.RM) GO TO 124
898.        GO TO 120
910.
911.
912.
913.
914.
915.
916.
917.
916.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.

1 2 5 IF1IP.LT.MAXNEW) GO TO 1 20
GO TO 2 0 0
1 2 0 CONTINUE
GO TO 1 10
1 2 4 0 0 12 8 N = NYYgMXVEAR
0 0 1 2 6 I = 1 (MAXNEM
L O N B (N tl) = LOWb(NYY « 1 )
1 2 6 N U PB !N »I) = L Q H b i N g l) « 1U 1DTHI1) 128 CONTINUE

1

C
1 3 0 CONTINUE
C
C

C
c
c
c
c

c

c

c
c
c
c

c
c
c
c
c
c
c
c
c
c
c

CALCULATE MAXIMUM UNIT ADDITION PER YEAR BY CANDIDATE TYPE
0 0 1 3 5 1P=1»MAXNEW
135 MAXADD(IP) = 1 W IU T H I1 P I/2 + 1
PRINT THE RESULTS OF TH IS SUBROUTINE
U R lT E (6 f9 0 0 )
WRITE!1 1 ( 9 0 0 )
9 0 0 FORMAT(1H / / 8 CHANNEL WIDTHg INCREMENT AND MAXIMUM » ,
NUMBER OF STATES' / )
H R I T £ ! 6 , 9 0 1 ) 1W1DIH
WRITE!6 ( 9 0 1 ) 1PCH
WRITE!6 ( 9 0 1 ) MAXSTA
WRITE!1 1 ( 9 0 1 ) IWIOTH
WRITE!1 1 ( 9 0 1 )
1PCH
WR1TE!11 v9 0 1 ) MAXSTA
9 0 1 FORMAT!1H ( 2 0 1 4 )
WRITE!6 ( 9 0 2 )
9 0 2 FORMAT 11H ✓ /" LOWER AND UPPER CHANNEL BOUNDS PER YEAR8)
0 0 1 4 0 NY = IgMXYEAR
NVY - IBASYR
NY
W R 1 T E I6 .9 0 1 ) NYY«IL0WB!NY(1)( I=1(MAXNEW)
W R lT E !6 e 9 0 3 ) ( N U P B l N Y d ) , 1 = 1 , MAXNEW)
W R 1 T E ! 1 1 ( 9 0 1 ) NYY, i LOWBINY,1 ) , 1 = 1 , MAXNEW)
WR1TEI1 1 ( 9 0 3 ) I N U P U i N Y ( I ) . 1 = 1 VMAXNEW)
1 4 0 CONTINUE
9 0 3 FORMATI1H t 4 X ( 1 9 l 4 )
904 F 0 R M A T IF 3 .0 (2 F 5 .0 (F 3 .0 (F 8 .6 (3 F 7 .2 tF 3 .0 (2 F 7 .3 (E 1 0 .3 (2 F 4 .2 )
9 0 5 FORMAT11H ( F 3 . 0 ( 2 F 5 . U ( F 3 . U ( F 8 . 6 ( 3 F 7 . 2 ( F 3 . 0 ( 2 F 7 . 3 ( E 1 0 . 3 « 2 F 4
1CFLAG = 0
GO TO 3 0 0
2 0 0 1CFLAG = 1
WRITE!6 ( 9 0 6 )
9 0 6 FORMAT 11H /* EXPANSION PLANT TYPES SELECTED ARE TOO SMALL
COVER THE MINIMUM RESRVt MARGIN.8/
* * * * * RERUN WITH MORE OR LARGER PLANTS * * * » * 8 )
ITOUT = 0
3 0 0 RETURN
END

SUBROUTINE C H O IC E !1 S T 0P )

USER INTERACTION SUBROUTINE. THE USER CHOOSES UP TO SIX
EXPANSION PLANT CANDIDATES THAT WILL BE USED IN OPTIMIZA­
TIO N . THIS PLANT COMBINATION CAN BE CHANGED AFTER EACH
OPTIMIZATION. THE MAXIMUM NUMBER OF ITERATIONS I S ALSO
DEFINED BY THE USER.
AT THE END OF EACH OPTIMIZATION THE
TO TEN BEST OPTIMAL SOLUTIONS WILL BE PRINTED DEPENDING ON
USER REQUEST.
DEFINITION OF LOCAL VARIABLES


NAME: TYPE SIZE    DEFINITION
ISTOP      INT.    -  PROGRAM STOP FLAG.
IF ISTOP = 0 PROGRAM CONTINUES
IF ISTOP = 1 PROGRAM STOPS
NBEST      INT.    -  NUMBER OF BEST SOLUTIONS REQUESTED
FOR REPORT

COMMON /ONE/  CLOLP,LPLDLP,PDISRAT,DX,FCR,HOURS,IBASY,ITIN,ITMAX,
             ITOUT,MAXALL,MAX1,MAXINP,MAXOR,MAXNEW,MAXPLA,
             MAXQ,MAXALL,MAXYEAR,RESMAR,UBOUND

COMMON /TWO/  AVSPI(20,4),CAPABS(20),ENEDEM(20,4),FAKTOPE(20),
               IDEXP(10),IPCH(20,10),ISOL(20,10),IWIDTH(10),
               LOMB(20,10),MAXUN(20,10),MAXADD(10),MINUN(20,10),
               NEXP(10),NORDER(420,3),NSPRE(20,10),LLEE(20,10),
               NPEAK(20,4),PLACAI(10,14),PLANT(20,14),
               RMAX(20),RSM(20,10),SOL(20,2),UVEA(20,10),

COMMON /THREE/ ELI(1250),ELDC(4,1250),ELF(1260),X(1250),
               XELDC(20,4)

COMMON /FOUR/  LIST(10,1000)

COMMON /FIVE/  PTCUM1(20,8,4),PTCUMX(10,8),ROM(20,4)

COMMON /SIX/  TY(120),TINT(120)

DOUBLE PRECISION  TY,TINT

DIMENSION KEEP(10)
IF (ITOUT .GT. 0) GO TO 10

WRITE(*,900)  // CERES OPTIMIZATION BEGINS
          /* CERES HAS A BUILT IN SENSITIVITY ANALYSIS PROCESS*/
          /* FOR EACH SENSITIVITY ANALYSIS ITERATION (CALLED HERE*/
          /* "MAIN ITERATION"), THE USER MUST SPECIFY THE EXPANSION*/
          /* PLANT CANDIDATE COMBINATION THAT WILL BE USED IN*/
          /* OPTIMIZATION. SINCE THE CERES OPTIMIZATION ALGORITHM*/
          /* IS OF ITERATIVE NATURE THE MAXIMUM NUMBER OF "TUNNEL*/
          /* ITERATIONS" MUST BE PROVIDED*/
          /* CERES WILL STOP WHEN THE USER INSTRUCTS TO DO SO.*/
          /* 4. FIXED CHARGE RATE OR SALVAGE VALUE OPTION FOR COST*/
          /* 5. PLUS 50% OF THE COST OF THE PRODUCT*/

WRITE(*,901)  // FOR EACH MAIN ITERATION THE USER MUST SPECIFY:
          /* 1. THE DESIRED EXPANSION PLANT COMBINATION.*/
          /* 2. THE MAXIMUM NUMBER OF TUNNEL ITERATIONS MUST BE*/
          /* 3. THE NUMBER OF BEST SOLUTIONS FOR WHICH A REPORT*/
          /* IS DESIRED (MAXIMUM IS 10 OR THE NUMBER OF TUNNEL*/
          /* ITERATIONS REQUIRED TO REACH THE OPTIMUM SOLUTION*/
          /* IF THIS IS LESS THEN 10 DEFAULT IS 1")*/
          /* THERE IS NO LIMIT ON THE NUMBER OF MAIN ITERATIONS*/
          /* CERES WILL STOP WHEN THE USER INSTRUCTS TO DO SO*/
          /* 4. FIXED CHARGE RATE OR SALVAGE VALUE OPTION FOR COST*/
          /* 5. PLUS 50% OF THE COST OF THE PRODUCT*/

GO TO 100
10 CONTINUE

WRITE(*,902) ITOUT
902 FORMAT(* THE NUMBER OF MAIN ITERATIONS IS:*),
         * THE NUMBER OF TUNNEL ITERATIONS WERE REQUIRED*,
         * TO FIND THE OPTIMAL SOLUTION*/

WRITE(*,903) ITIN
903 FORMAT(* THE NUMBER OF MAIN ITERATIONS IS:*),
         * THE NUMBER OF TUNNEL ITERATIONS WERE REQUIRED*,
         * TO FIND THE OPTIMAL SOLUTION*/

NNIT = 10
NIT = MINO(NNIT,11)
WRITE(6,904) NIT
104 FORMAT(' INPUT THE NUMBER OF BEST SOLUTIONS YOU WANT*,
+* REPORTED**, THIS SHOULD BE LESS OR EQUAL TO*,14)
105 READ(*,*)NBEST
106 C
107 C CHECK NBEST AND SET TO 1 IF OUT OF RANGE
108 C IF (NBEST .LT. 1 .OR. NBEST .GT. NIT) NBEST = 1
109 C CALL RESIMULATION AND REPORT ROUTINES NBEST TIMES.
110 IRES = 1
111 DO 20 N = 1,NBEST
112 NB = NBEST
113 CALL RESIM(NN,NB,IRES,KEEP)
114 20 CONTINUE
115 IRES = 0
116 C
117 WRITE(6,905)
118 905 FORMAT(' IF YOU WANT TO STOP THE MAIN ITERATIONS*,
+* SENSITIVITY ANALYSIS), TYPE 1 */
119 +* IF YOU DESIRE TO CONTINUE, TYPE 0 */
120 READ(5,*) ISTUP
121 IF (ISTUP .LT. 1) GO TO 100
122 RETURN
123 C
124 100 ITOUT = ITOUT+1
125 C ASK FOR EXPANSION CANDIDATE COMBINATION
126 WRITE(6,906)
127 906 FORMAT(1H / ' ENTER THE NUMBER OF EXPANSION CANDIDATES*,
128 +* YOU WILL CONSIDER IN THIS ITERATION*/
129 +* IT SHOULD NOT BE MORE THEN 6. IT IS RECOMMENDED TO BE*,
130 +* 4 OR 5*)
131 READ(5,*) MAXNEW
132 C CALCULATE NUMBER OF ALL PLANTS
133 MAXALL = MAXPLA + MAXNEW
134 WRITE(6,907) MAXNEW
135 907 FORMAT(1H / ' ENTER THE EXPANSION PLANT ID'S*/
136 +* THE PLANT ID'S ARE DEFINED FROM THE PLANT ORDER IN*,
137 +* THE NUMBERS ENTERED SHOULD BE SEPARATED BY BLANKS*)
138 READ(5,*) (IDEXP(I),I=1,MAXNEW)
139 C
140 C ASK FOR ITMAX
141 WRITE(6,908)
142 908 FORMAT(1H / ' ENTER THE MAXIMUM NUMBER OF TUNNEL*,
143 +* IT SHOULD BE BETWEEN 10 AND 50*)
144 READ(5,*) ITMAX
145 C CHECK ITMAX
146 IF (ITMAX .GE. 10 .AND. ITMAX .LE. 50) GO TO 110
147 ITMAX = 10
148 110 CONTINUE
149 C
150 WRITE(6,909)
151 909 FORMAT(1H / ' FOR SALVAGE VALUE OPTION ENTER -1.0*/
152 +* FOR FIXED CHARGE RATE ENTER THE FCR VALUE*)
153 READ(5,*) FCR
154 C
155 GLOLP = CPLOLP
156 WRITE(6,910) CPLOLP
157 910 FORMAT(' THE CRITICAL LQLP VALUE IN PREP MODULE IS*,F9.6/
158 +* IF YOU WANT TO INCREASE IT, ENTER THE NEW NUMBER. OTHER*,
159 +* WISE ENTER -1.0*)
160 READ(5,*) CHLOLP
161 IF (CHLOLP .GT. -1.0) GLOLP = CHLOLP
162 C
163 WRITE(6,911)
164 911 FORMAT(1H / ' ENTER THE MAXIMUM RESERVE MARGIN IN %*/
165 +* IT SHOULD BE BETWEEN 20 AND 50%. DEFAULT IS 40%*)
166 READ(5,*) RESMAR
1120. WRITE(6,912)
1121. 912 FORMAT(' ENTER THE DISCOUNT RATE IN %.  DEFAULT IS 15%')
1122. READ(5,9) DISRAT
1123. IF (DISRAT .LT. 5.0) DISRAT = 5.0
1124. IF (DISRAT .GT. 50.0) DISRAT = 50.0
1125. RESMAR = RESMAR/100.0
1126. C FIND THE YEARLY MAXIMUM RESERVE MARGINS
1127. C DO 130 NY = 1,MAX YEAR
1128. 130 PVR = 0.0
1129. DO 180 J = 1,4
1130. 180 CONTINUE
1131. RESMAX(NY) = PVR*(1.04 RESMAR)
1132. 190 CONTINUE
1133. RETURN
1134. END
1135. C SUBROUTINE CUGAL1(NSCUR,N)
1136. C
1137. C COMMON /ONE/ CLOLP,CPLQR,P,DISRAT,DX,FCS,HOURS,IBASYR,I ITL,ITMAX,
1138. * ITOUT,MAXALL,MAXIN,MAXM,MAXNEW,MAXPLAQ,
1139. * MAXPG,MXX,MAXY,RESMAR,UBOUND
1140. C COMMON /TWO/ AVSP(20,4),CAPASS(20),EMEDEN(20,4),FATOPE(20),
1141. * IDEP(10),IPCH(10),ISOL(20,10),IWIDTH(10),
1142. * LOMB(20,10),MAXUN(20,10),MAXAUD(10),MINUN(20,10),
1143. * NEXTL(10),NORDER(20,3),NTPH(20,10),LLOMB(20,10),
1144. * NUPB(20,10),PEAKS(20,4),PLACA(10,14),PLANT(210,14),
1145. * RESMAX(20),RESMIN(20),SOL(20,2),UNR(20),MAINS(20)
1146. C COMMON /FIVE/ PTCUM(201,8,4),PTCUMX(10,8),ром(20,4)
1147. C
1148. C DIMENSION R(4), NSTCUR(10)
1149. C
1150. C INITIALIZATION
1151. C
1152. DO 10 I=1,MAXINP
1153. DO 10 J=1,8
1154. DO 10 PTUM(K,K)=0.0
1155. IF (NSTCUR( K ) .LE. 0) GO TO 20
1156. IP=IDEP(I)
1157. IN=2*IP
1158. IF(J.GT.1) IN=2*IP
1159. C=PLACA(I,2)*(1.0-RRM(1N,NN))
1160. GO TO 40
1161. 35 IN=2*IP
1162. C=PLACA(I,3)*(1.0-RRM(IN,NN))
1163. 40 R(1)=C*Q
1164. R(2)=C*Q*(1.0-Q)
1165. R(3)=C*Q*(1.0-(3.0-2.0*Q)*Q)
1166. R(4)=C*Q*(1.0-(7.0-12.0-6.0*Q)*Q)
1167. DO 45 K=1,4
1168. KK=K
1169. IF(J.GT.1) KK=KK+4
1170. 45 PTCUMX(IP,KK)=R(K)
1171. 30 CONTINUE
1172. 20 CONTINUE
1173. RETURN
SUBROUTINE FATHQM(NS1CUR,NY,OBJ,IFATH)

C                          CALCULATES THE MINIMUM CONSTRUCTION COST FOR REACHING THE
C                          FINAL YEARS MINIMUM RESERVE MARGIN. THE MINIMUM YEARLY
C                          OPERATING COST FOR ALL FUTURE STUDY YEAR PERIODS IS ADDED
C                          TO THE ABOVE CONSTRUCTION COST. BOTH THESE ARE ADDED TO THE
C                          STATE'S OBJECTIVE FUNCTION AND COMPARED WITH THE UPPER BOUND
C                          FOR THE OBJECTIVE FUNCTION. THE STATE "NSTCUR" IS ACCEPTABLE
C                          IF THE UPPER BOUND IS NOT EXCEEDED BY THE ABOVE COSTS. THEN
C                          FLAG IFATH = 1. OTHERWISE IFATH = 0.

C                          COMMON /ONE/ CLOLP,CLOLP,DISLAT,DX,FCR,HOURS,IBASYR,ITIN,ITMAX,
C                          +           IOUT,MAXALL,MAX1,MAX1P,MAXOR,MAXNEH,MAXPLA,
C                          +           MAXPQ,MXALL,MYEAR,RESMAR,UBOUND
C                          +           COMMON /THW/ AVSP(20,4),CAPABS(20),ENEDEN(20,4),FAT0PE(20),
C                          +         IEXP(10),IPCH(10),ISOL(20,10),IWIDTH(10),
C                          +         LOUB(20,10),MAXUN(20,10),MAXAD(10),MINUN(20,10),
C                          +         NEXPH(10),NORDER(20,3),NSTPRE(20,10),LOWB(20,10),
C                          +         NUPB(20,10),PEAKS(20,4),PLACA(10,14),PLANT(20,14),
C                          +         RESMAX(20),RESMIN(20),SOL(20,2),UNE(20),MINS(200)
C                          DIMENSION NSTCUR(IO)

   IF  (OBJ . LT . UBOUND) GO TO 5
   IFATH = 0
   RETURN
   5 CONTINUE
   IF (NY .EQ . MXYEAR) GO TO 100
   DO 10 1=1,MAXNEH
       C = 0.0
   10   C = C + PLACA(1,3) * NSTCUR(I)
   IFACH = (OBJ + C) / (1000.0 * FCR)
   IF (IFACH .LT . 0.0) GO TO 12
   12 IFACH = 0.0
   DO 20 N=NS,MXYEAR
        C = C + PLACA(N)/D1SFAC
   20   C = C * FCR * PWFC * (1.0*D1SFAC)
   15 DO 20 N=NS,MXYEAR
        DISFAC = 1.0/((1.0+DISRAT)**(FLOAT(NY)+1.0))
    20   DISFAC = 1.0/((1.0+DISRAT)**(FLOAT(N)+0.5))
   19 C = C * DISFAC
   RETURN
C CHECK THE TOTAL COST AGAINST THE OBJECTIVE FUNCTION UPPER
C BOUND
IF (COST .GT. UBOUND) GO TO 30
IFATH = 1
RETURN
C
30 IFATH = 0
RETURN
C 100 COST = 0.0
IF (NY .LT. MXYEAR) GO TO 15
IFATH = 1
RETURN
C END
C
SUBROUTINE FLOLP(NSTCUR, CULOLP, NSTOLD, OLOLP, NYNEXT)
C
FLULP FINDS THE LULP FOR THE STATE UNDER EXAMINATION
(NSTCUR). IF THE LAST STATE EXAMINED (NSTOLD) WAS RELIABLE
AND THE CURRENT STATE HAS THE SAME OR MORE UNITS FOR EACH
PLANT CANDIDATE, THEN THIS STATE IS ALSO RELIABLE AND THE
CURRENT STATE LOLP(CULOLP) IS EQUATED TO THE OLD STATE LOLP
(OLOLP). IF THE ABOVE CONDITIONS ARE NOT SATISFIED THE LOLP
IS CALCULATED STARTING FROM THE ELDC OF THE SCHEDULED
SYSTEM.
C
NOTE THAT LOLP SERVES ONLY AS A CONSTRAINTED OPTIMIZATION
FOR RELIABILITY IS BASED ON UNSERVED ENERGY AND NOT LOLP.
C
COMMON /ONE/ CLOLP, CPLOLP, DISRAT, DX, FCR, MUURS, BASYR, ITIN, ITMAX,
+ ITOUT, MAXALL, MAX, MAXNP, MAXUR, MAXNEW, MAAXPLA,
+ MAXPO, MAAXALL, MXYEAR, RESMAR, UBOUND
C
COMMON /TWO/ A V S P (104), CAPAB (20), ENEDM (204), FATOPE (20),
+ IDEP (10), IPCM (10), ISOL (204), ISTDTH (10),
+ LOH (20), MAXUN (20), MAXADD (10), MNU (20),
+ NEXP (10), NURDER (4203), NSTPRE (2010), LLONB (20),
+ NPEAK (20), PEAKS (204), PLACA (10), PLANT (20),
+ RESM (20), RESMIN (20), SOL (20), UNE (20), MAINS (200)
C
COMMON /THREE/ EL (1250), EDLC (1250), ELF (1260), X1250,
+ MXELDC (204)
C
COMMON /FOUR/ LIST (10, 1000)
C
COMMON /FIVE/ PTCUM (201, 84), PTCUMX (108), ROM (204)
C
DIMENSION NSTCUR (10), NSTOLD (10)
C
*** DEFINITION OF LOCAL VARIABLES ***
C
NAME TYPE SIZE DEFINITION
C
ISTOLD INT 10 UNPACKED NSTOLD STATE
C
C CHECK LOLP OF THE LAST EXAMINED STATE
C IF OLOLP .GT. CLOLP) GO TO 20
C
C COMPARE THE NUMBER OF UNITS IN CURRENT AND OLD STATE
C UNPACK NSTOLD
C IFLAG = 0
C CALL UNPACK(NSTOLD, NSTOLD, NYNEXT, IFLAG)
C DO 10 I = 1, MAXNEW
C IF (NSTCUR(I) .LT. NSTOLD(I)) GO TO 20
10 CONTINUE
DROPPING THROUGH THE LOOP MEANS THAT THE PREREQUISITS FOR
EQUATING THE OLD STATE ULOP TO THE CURRENT ONE ARE SATISFIED

CULOP = OLOP
RETURN

20 CONTINUE

CALL MAINT2(NSTCUR,NYNEXT)

CULOP=0.0
DO 40 N=1,L
PK=PEAKS(NYNEXT,N)
MAX=MAXELDC(NYNEXT,N)
DO 50 I=1,MAX
EL(I)=ELDC(N,N)
SUMY=0.0
DO 60 J=1,MAX
IF (NSTCUR(J).LE.0) GO TO 60
JSTOP=NSTCUR(J)
IN = 2*IDEXP(J)
DO 50 J=1,JSTOP
IF (PLA(J).GT.0.0) GO TO 50
MAX=1.0-PLA(J)/PK
SUMY=SUMY+PLA(J)/PK
50 CONTINUE
PLP=(PLP+SUMY)/PK

CALL PADD(Y,P,MAXPL)

CONTINUE

RETURN
END

COMMON /ONE/ CULOP,PLPLOP,DISRAT,DX,FCR,HOURS,IBASYR,ITIN,ITMAX,
+ IOUT,MAXALL,MAX1,MAXINP,MAXOK,MAXNEW,MAXPLA,
+ MAXPO,MXALL,MXYEAR,RESMAR,UBOUND

COMMON /THREE/EL(1250),POPEC(250,20),ELF(1250),
+ X1(1250),MXELDC(20,4)

COMMON /SEVEN/ NP,MX1

DIMENSION A1(250),A2(20),FMT1(18),FMT2(18),FMT3(18),
+ FMT4(18),FMT5(18),FMT6(18),FMT7(18),C(60),IYR(20)

LHX=51
M=NP+1
N1=M*11/2+1
N2=M/2+1
N3=M+N1/2+1
N4=M+N1/2+1
FMT1(3)=C(N1)
FMT2(3)=C(N1)
FMT1(14)=C(N2)
FMT2(14)=C(N2)
FMT3(3)=C(N3)
FMT4(5)=C(N)
FMT4(5)=C(N)
FMT5(5)=C(N)
FMT5(5)=C(N)
IF (LN.EQ.0) GO TO 350
LK=LHX-LN*2+1
IF (I.EQ.NP) K=1
IF (NP.EQ.MAXINP) GO TO 355
GO TO 360

355 WRITE (6,355)
365 FORMAT (1H //)
366 FORMAT (36H *PLANT COSTS ARE GIVEN AS THE TOTAL,
+48H WORTH OF THE PLANT AS IT COMES ON LINE LESS THE //
+46H SALVAGE VALUE AT THE END OF THE STUDY PERIOD, //
+44H IF THE FIXED CHARGE RATE OPTION IS USED, //
CONSTRUCTION COSTS REPRESENT THE FIXED,
+24H CHARGES FOR EACH PLANT.)

LK=LK-8

CONTINUE

DO 10 1K=1,LK

WRITE (6,30)

330 FORMAT (1H )

320 CONTINUE

WRITE (6,330)

330 CUNTINUE

IF (NP.EQ.NAX1N P) WRITE !6,FMT 1) 1PAGE

IF (NP.EQ.MXALL) WRITE (6,FMT 2) 1PAGE

WRITE (b ,F M T 3 ) IBASYR


RETURN

END

SUBROUTINE MAINT2( NSTCUR, NYNEXT)

COMMON (ONE/ CLOLP,CPLOLP,DISRAT,DX,FCH,HOURS,IBASYR,ITIN;ITMAX,
* ITOUT,MAXALL,MAXI,MAXINF,MAXOR,MAXNEW,MAXPLA,
+ MAXPO,MXALL,MXYEAR,RESMAR,UBOUND

COMMON /TWO/ AVSP(20,4),CAPAH(20),ENEDEN(20,4),FATUP(20,4)
+ IDEXP(I10),ISOL(20,10),1WIDTH(110),
+ LONB(20,10),MAXUN(20,10),MAXADD(10),MINUN(20,10),
+ NEXP10,1MDER(420,3),NSTRE(26,16),LLONB(20,10),
+ NUP8(20,10),PEAKS(20,4),PLACA(10,14),PLANT(10,14),
+ RESMAX(20),RESMIN(20),SOL(20,2),UNE(20),MAINS(200)

COMMON /FIVE/ PTCUM(201,8,4),PTCUM(10,8,RM(20,4)

DIMENSION NSTCUR(10),ASP(4)

RLT=363./4.

DU 50 J=1,4

DO 50 I=1,20

50 ROM(I,J)=0.0

DO 100 NP=1,4

ASP(NP)=AVSP(NYNEXT,N)

N=NP

ASP(N)=AVSP(NYNEXT,N)

DO 110 I=1,MAXNEW

110 ASP(N)=ASP(N)+FLOAT(NEXP)*PLACA(I,3)

100 CONTINUE

C

DU 120 I=1,MAXNEW

IF(NSTCUR(I).LT.0) GO TO 120

120 NEXP=NSTCUR(I)

IP=I

DO 140 J=1,2

140 IF(J.GT.1) GO TO 140

C=PLACA(IP,2)

150 IN=2*IDEXP(I)

GO TO 150

140 C=PLACA(IP,3)-PLACA(IP,2)

150 EMXAS=0.0

160 EMXAS=ASP(I)+

GO TO 170

160 EMXAS=ASP(I)

LL=LL

170 CONTINUE

180 ROM(NP,LL)=PLACA(IP,4)/RLT
130 ASP(ILL)=ASP(ILL)-FLOAT(NEXP)*ROM(IN,LL)*C
120 CONTINUE
1472. RETURN
1473. END

SUBROUTINE OBJFUN(NSTCUR, NY, OPEC, NSTATE, CAPCO, OBJRI, OBJ, +S, RESCAP, IRES)

C CALCULATES THE DISCOUNTED CONSTRUCTION COST "CAPCO" AND THE
C OBJECTIVE FUNCTION (INCLUDING SALVAGE VALUE) FOR STATE
C "NSTCUR" THAT ORIGINATED FROM STATE "NSTATE"

C COMMON /ONE/ CLDLP, CPLOLP, DISKAT, DX, FCR, HOURS, IBASRY, ITIN, ITMAX,
+ ITOUT, MAXALL, MAX1, MAXINP, MAXOR, MAXNEN, MAXPLA,
+ MAXPO, MAXALL, MAXYEAR, RESMAR, UBOUND

C COMMON /TWO/ AVSP(20,4), CAPABS(20), ENEDEM(20,4), FATOPE(20),
+ IDEXP(10), IPCHE(10), ISOL(20,4), IWIDTH(10),
+ LWD(20,10), MAXUM(20,10), MAXADD(10), MINUM(20,10),
+ NEXP(10), NORDER(420,3), NSTPRE(20,10), LLOW(20,10),
+ NUPB(20,10), PEAKS(20,4), PLACA(10,16), PLANT(210,14),
+ RESMAX(20), RESMIN(20), SOL(20,21), UNE(20), MAINS(20)

C COMMON /THREE/ LUb(20,10), MAXUN(20,10), MINUN(20,10),
+ NEXP10(10), ORDER(420,3), NSTPRE(20,10), LOWB(20,10),
+ NUPB(20,10), PEAKS(20,4), PLACA(10,16), PLANT(210,14),
+ RESMAX(20), RESMIN(20), SOL(20,21), UNE(20), MAINS(20)

C DIMENSION NSTCUR(10), NSTATE(10), NADD(10), CONCO(10), RESCAP(20,10)

1500. DO 10 I=1,MAXNEH
1501. NADD(I) = NSTCUR(I) - NSTATE(I)
1502. C CALCULATE THE PLANT ADDITIONS

1503. C CALCULATE THE UNDISCOUNTED CONSTRUCTION COST FOR EACH
1504. C CANDIDATE TYPE IN MILLION OF DOLLARS. ALSO CALCULATE
1505. C THE TOTAL CONSTRUCTION COST
1506. C = 0.0
1507. DO 20 I=1,MAXNEH
1508. CONCO(I) = NADD(I)*PLACA(1,3)*PLACA(1,6)/1000.0*
1509. + (1.0*PLACA(I,13))/NY)
1510. 20 C = C+CONCO(I)

1511. C CALCULATE THE DISCOUNT FACTOR
1512. DISFAC = 1.0/(1.0+DISRAT)*NY
1513. C
1514. C CALCULATE THE SALVAGE VALUE FOR EACH CANDIDATE USING
1515. C STRAIGHT LINE DEPRECIATION. ALSO CALCULATE TOTAL SALVAGE
1516. C VALUE.
1517. C = S=0.0
1518. DO 30 I=1,MAXNEH
1519. NYLEFT = MXYEAR-NV+1
1520. C PLANT SALVAGE VALUE
1521. SYALV = CONCO(I)*(1.0-FLOAT(NYLEFT))/PLACA(1,9))
1522. RESCAP(NY,1) = (CONCO(I) - SYALV)*DISFAC
1523. C = S=SYALV*DISFAC
1524. 30 S = S+SYALV*DISFAC
1525. C
1526. C IF (FCR .GT. 0.0) GO TO 40
1527. C CALCULATE DISCOUNTED CAPITAL COST AND OBJECTIVE FUNCTION
1528. CAPCO = C*DISFAC
1529. GO TO 50
1530. 40 NYLEFT = MXYEAR-NV+1
1531. PWFC = (1.0-(1.0+DISRAT)*(1-NYLEFT))/DISRAT
1532. DISFAC = 1.0/(1.0+DISRAT)**(NY*0.5)
1533. C = C*FCR*PWFC
1534. CAPCO = C*DISFAC
1535. S = 0.0
1536. C
1537. C
1538. 50 OBJ = OBJRI+CAPCO=5+UPEC
1539. IF (FCR .LE. 0.0 .OR. IRES .LT. 1) GO TO 100
DISFAC = 1.0/(1.0+DISKAL)**NY
DO 60 I=1,MAXNEW
    CNEW = CONCUR(I)*FCR*DISFAC
60 RESCAP(NY,1)=RECAP(NY,1)*CNEW
100 RETURN
END

SUBROUTINE OPERC(NSTCUR, NYNEXT, OPEC, IRES)

PURPOSE: CALCULATES THE DISCOUNTED OPERATING COSTS OPEC FOR STATE NSTCUR IN STUDY YEAR NYNEXT. THE METHOD USED IS "PLANT DERATINGS" WITH 100 POINT P-W REPRESENTATION OF THE ORIGINAL LOAD PROBABILITY CURVE (INVERTED LOAD DURATION CURVE).

THE COMMON VARIABLES ARE DEFINED IN THE MAIN PROGRAM

*** DEFINITION OF LOCAL VARIABLES ***

FAC REAL - DISCOUNT FACTOR
IDP INT - PLANT ID IN SCHEDULED OR CANDIDATE PLANT FILES
P REAL - PLANT ID IN SCHEDULED OR CANDIDATE PLANT FILES
Y REAL - NORMALIZED AND DERATED PLANT CAPACITY FOR ALL MEMBER UNITS
COST REAL - PLANT OPERATING COST IN $/Mh

COMMON /ONE/ CLOLP,CPLULP,DISRAT,DX,FCR,HOURS,IBASYR,ITIN,ITMAX,
     1101,MAXALL,MAX1,MAXINC,MAXOR,MAXNEW,MAXPLA,
     1110,MAXP,MXALL,MXYEAR,RESMAR,UBOUND

COMMON /TWO/ AVSPI(20,2),CAPARS(20),ENEDEN(20,4),FATOE(20),
     1111,IPCH(10),ISOL(20,10),WIDTH(10),
     1120,MAX(20,10),MAXADD(10),MINUNI(20,10),
     1130,NEXP(10),NUDER(420,3),NSTRP(20,10),LLOWB(20,10),
     1140,NUPt(20,10),PLAKS(20,4),PLAA(10,14),PLANT(210,14),
     1150,REMAX(20),RESMIN(20),SOL(20,2),UNE(20),MAINS(200)

COMMON /THREE/ EL(1250),ELUC(4,1250),ELF(1260),EX(1250),
     1131,NXELDC(20,4)

COMMON /FOUR/ LIST(10,1000)

COMMON /FIVE/ PTUM(201,8,4),PTUCX(10,8),ROM(20,4)

COMMON /SIX/ITY(120),TINT(120)

DOUBLE PRECISION TY,TINT,ZL1,ZLZ2

DIMENSION SYSCUM(4),C0EFL(4),NSTCUR(101),ITK(420,3),OPEC(250,20)

COMMON /SIX/ EL(1250),ELUC(4,1250),ELF(1260),EX(1250),
     1131,NXELDC(20,4)

COMMON /FOUR/ LIST(10,1000)

COMMON /FIVE/ PTUM(201,8,4),PTUCX(10,8),ROM(20,4)

COMMON /SIX/ITY(120),TINT(120)

DOUBLE PRECISION TY,TINT,ZL1,ZLZ2

DIMENSION SYSCUM(4),C0EFL(4),NSTCUR(101),ITK(420,3),OPEC(250,20)

EQUALVENCE (ELF(I),ITK(I)), (ELDC(I),OPEC(I))

DO 100 I=1,MAXNEW
    IP=10EXP(I)
100 DO 110 K=1,MAXOR
    IF(NORDER(K,1)=NE.1P) GO TO 110
    NORDER(K,3)=NSTCUR(I)
110 J=J+1
1010 DO 120 J=J,MAXOR
    IF(NORDER(J,1)=NE.1P) GO TO 120
    NORDER(J,3)=NSTCUR(I)
120 GO TO 100
120 CONTINUE
GO TO 100
100 CONTINUE
K1=0
DO 130 K=1,MAXORD
IF (NORDER(K,3)>LE.0) GO TO 130
K1=K1+1
DO 140 J=1,3
140 ITK(K1,J)=NORDER(K,J)
130 CONTINUE
KNMAX=K1
CALL MAINT2(INSTCVR,NEXT)
UNE(NEXT)=0.0
OPEC=0.0
DO 150 NS=1,4
N=NS
E=0.0
CALL DCAL1(INSTCUR,N)
DO 160 I=1,4
160 SYSCUM(I)=PTCUM(I,1,N)
DO 170 I=1,4
170 COEFF(I)=0.0
SUMP=0.0
DO 180 KC=1,KNMAX
K=KC
I01=2
IF (ITK(K,2)>EQ.2) I01=3
200 IP=ITK(K,1)
I1=1
I2=ITK(K,3)
IF (IP>MAX-INP) GO TO 210
DO 220 IPL=1,MAXNEW
IF (IDEXP(IPL)>EQ.IP) GO TO 230
220 CONTINUE
230 IX=1PL
P=1.0-PLACA(IX,5)
IF (ID1>EQ.3) GO TO 240
1M=2*IP
Y=PLACA(IX,ID1)*(1.0-KOM(IM,N))
COST=PLACA(IX,7)
GO TO 250
240 1M=2*IP-1
Y=(PLACA(IX,3)-PLACA(IX,2))*(1.0-KOM(IM,N))
COST=PLACA(IX,5)
250 FAC=((1.0+PLACA(IX,14))/(1.0+DISRAT))*((FLOAT(NEXT)+0.5)
VAB=PLACA(IX,11)
FIX=PLACA(IX,10)
6U TO 300
210 JP=IP-MAXINP
P=1.0-PLANT(IP,5)
IF (ID1>EQ.3) GO TO 260
Y=PLANT(IP,ID1)
IF (MAINS(JP)>EQ.N) Y=PLANT(IP,ID1)*(1.0-PLANT(IP,4)/(365.0/4.0))
COST=PLANT(IP,7)
GO TO 270
260 Y=PLANT(IP,3)-PLANT(IP,2)
IF (MAINS(JP)>EQ.N) Y=Y*(1.0-PLANT(IP,4)/(365.0/4.0))
COST=PLANT(IP,8)
270 FAC=((1.0+PLANT(IP,14))/(1.0+DISRAT))*((FLOAT(NEXT)+0.5)
VAB=PLANT(IP,11)
FIX=PLANT(IP,10)
300 DO 310 I=1,12
SUMP=SUMP+Y
V1=SUMP+Y
Y2=SUMP
1681. IF(I.EQ.1) GO TO 320
1682. IQ=IP
1683. ID2=ITK(K,2)
1684. GO TO 330
1685. 320 IF(K.EQ.1) GO TO 400
1686. IQ=ITK(K-1,1)
1687. ID2=ITK(K-1,2)
1688. 330 DO 340 J=1,J*
1689. J1=J
1690. IF(ID2.EQ.J) J1=J*
1691. IF(IQ.GT.MAXINP) GO TO 350
1692. SYSCUM(J)=SYSCUM(J)+PTCUM(IQ,J1)
1693. GO TO 340
1694. 350 SYSCUM(J)=SYSCUM(J)+PTCUM(IQ-9,J1,N)
1695. 340 CONTINUE
1696. 3
1697. C
1698. IF(ID1.NE.3) GO TO 400
1699. DO 370 J=1,J*
1700. J1=J
1701. IF(IP.GT.MAXINP) GO TO 380
1702. SYSCUM(J)=SYSCUM(J)+PTCUM(IP,J1)
1703. GO TO 370
1704. 380 SYSCUM(J)=SYSCUM(J)+PTCUM(IP-9,J1,N)
1705. 370 CONTINUE
1706. 3
1707. C
1708. 400 COEFF(1)=SYSCUM(1)
1709. COEFF(2)=SYSCUM(2)
1710. SIGMA=SQR(T(SYSCUM(2)))
1711. SIGIN=SIGMA*SIGMA
1712. DO 410 J=3,J*
1713. SIGIN=SIGIN*SIGMA
1714. 410 COEFF(J)=SYSCUM(J)/SIGIN
1715. 4L1=(Y1-COEFF(1))/SIGMA
1716. 4L2=(Y2-COEFF(2))/SIGMA
1717. 4L1 = DBLE(4L1)
1718. 4L2 = DBLE(4L2)
1719. CF3 = COEFF(3)
1720. CF4 = COEFF(4)
1721. CALL VALUE(ZZL1,CF3,CF4,ZV1)
1722. CALL VALUE(ZZL2,CF3,CF4,ZV2)
1723. IF(NRES.NE.1) GO TO 420
1724. OPPEC(IP,NYNEXT1)=OPPEC(IP,NYNEXT1)+
1725. *(1HOURS*P*Y*ZV1*ZV21/2.0)*(COST+VAB)*(Y*FIX*FAC/1.0)*E*06
1726. 420 E=E+HOURS*P*Y*(ZV1+ZV21/2.0)*(COST+VAB)*(Y*FIX)*FAC/1.0*E*06
1727. CONTINUE
1728. 310 CONTINUE
1729. 180 CONTINUE
1730. UNE(0YNEXT1)=UNE(0YNEXT1)+(ENEDIM(0YNEXT1,N)-E)
1731. IF(UNE(0YNEXT1).LT.0.0) UNE(0YNEXT1)=0.0
1732. 150 CONTINUE
1733. DO 500 I=1,MAXNEW
1734. IP=1DEXP(1)
1735. DO 510 K=1,MAXOR
1736. IF(NORDER(K,1).NE.1) GO TO 510
1737. NORDER(K,3)=1
1738. J=K+1
1739. DO 520 J=J,J,MAXOR
1740. IF(NORDER(J,1).NE.1) GO TO 520
1741. NORDER(J,3)=1
1742. GO TO 500
1743. 520 CONTINUE
1744. GO TO 500
1745. 510 CONTINUE
1746. 500 CONTINUE
1747. C
1748. C
1749. RETURN
SUBROUTINE PACK NSTATE, NST, NY

PACKS ARRAY NSTATE INTO A SINGLE WORD NST. CURRENTLY USE PACKING BASE 10 AND DO NOT ALLOW MORE THAN 9 PLANTS TO BE PACKED, i.e., ARRAY NSTATE WILL HAVE AT MOST THE FIRST NINE ENTRIES NON ZERO AND THEIR VALUES SHOULD BE LESS OR EQUAL TO 9.

DIMENSION NSTATE(10), LSTATE(10)

COMMON /ONE/ CLOLP, CLOLP, DISRAT, DX, FCR, N, B, Y, ITIN, ITMAX,
+ ITOUT, MAXALL, MAXI, MAXIMP, MAXUR, MAXNEH, MAXPLA,
+ MAXPO, MAXALL, MAXYEAR, RESMAR, UBOUND

COMMON /TWO/ AVSP(20,4), CAPABS(20), ENEDER(20,4), FATUPE(20),
+ IDEXP(10), IPC(10), ISOL(20,10), IWIDTH(10),
+ LQBMI(20,10), MAXWIN(20,10), MAXADD(10), MINWIN(20,10),
+ MEXPID(10), NORDER(14,20,3), NSTPRE(20,10), LLOWB(20,10),
+ NUPB(20,10), PEAKS(20,4), PLAC(10,14), PLANT(210,14),
+ RESMAX(20), RESMIN(20), SOL(20,2), UNE(20), MAINS(1200)

DO 5 I = 1, MAXNEH
5 LSTATE(I) = NSTATE(I) - LOWB(NY, I)

NST = 0
MULT = 1
DO 10 I=1, MAXNEH
10 NST = NST + LSTATE(I) * MULT
MULT = MULT * 6
RETURN
END

FUNCTION PLOLP(Y)

FIND LOLP, i.e., the E{i[1]} point that corresponds to Y
Y: NORMALIZED WAH CAPACITY

COMMON /ONE/ CLOLP, CLOLP, DISRAT, DX, FCR, N, B, Y, ITIN, ITMAX,
+ ITOUT, MAXALL, MAXI, MAXIMP, MAXUR, MAXNEH, MAXPLA,
+ MAXPO, MAXALL, MAXYEAR, RESMAR, UBOUND

COMMON /THREE/ E{1}(1250), ELDC(4, 1250), ELF(1260), X(1250),
+ MAXELDC(20, 4)

YY = Y * MAXPO
IV = YY
MAXIM = MAXI - 1
IF (IV + GT MAXIM) GO TO 100
DY = IV + IY
PLOLP = (E{IY} - EL(IY+1)) * DY * EL(IY+1)
RETURN
100 PLOLP = 0.0
RETURN
END

END
SUBROUTINE PWADDi (y, p, newmax)

DETERMINES THE NEW ELD CURVE AFTER THE ADDITION OF THE UNIT WITH NORMALIZED MW CAPACITY Y AND RELIABILITY P.

COMMON /THREE/ EL(1250), ELDC(4,1250), ELF(1260), X(1250), MAXELDC(20,4)

DATA PMAX/0.999/, ELMIN/0.000001/, ONE/1.0/

DEFINITION OF TERMS:

X: X-AXIS I.E. NORMALIZED MW AXIS
EL: LOAD PROBABILITY MATRIX
ELF: TEMPORARY LOAD PROBABILITY MATRIX, STORES THE NEW EL TEMPORARILY.

Y: NORMALIZED MW CAPACITY OF ADDED UNIT: Y=MWC/PERPK
P=1-FOR LOAD PROBABILITY
Q=FORCED OUTAGE RATE
MAX1: EXPECTED MAXIMUM DIMENSION OF EL MATRIX.
PMAX: =0.999 IS CONSIDERED A REALISTIC LIMIT OF UNIT AVAILABILITY ESTIMATE. I.E. IF P>PMAX IT IS ASSUMED THAT P=1.0 AND THE ELD CURVE DOES NOT change.
ELMIN=0.0000001, IS THE LIMIT OF LOLP ACCURACY.
IF LOLP < ELMIN THEN LOLP = 0.0

IF (P.GE.PMAX) GO TO 100

FORM THE NEW ELD CURVE. STORE IN ELF

Q=ONE-P
J=1
DO 50 I=1, MAX1
XY=X(I)-Y
IF (XY.LE.X(1)) GO TO 30
J=J+1

25 IF (ABS(XY-X(J)) .LE. ELMIN) GO TO 40
ELY=(X(J-1)-XY)/(X(J-1)-X(J)) * ELF(J)
ELY=(XY-X(J))/(X(J-1)-X(J)) * ELF(J-1)
GO TO 45

30 ELY=ONE
GO TO 45

40 ELF=EL(J)

45 CONTINUE

ELF(I)=PMEL(I)+Q*ELY
MAXIMU=I
IF (ELF(I).LE.ELMIN) GO TO 55

50 CONTINUE

55 NEWMAX=MAXIMU
IF (NEWMAX.GT.MAX1) NEWMAX=MAX1

ALL ELD VALUES FOR WHICH I SATISFIED THE :
/NEWMAX<1<MAX1, HAVE EL(I)<ELMIN AND ARE LEFT TO BE ZERO. (NOTE THAT THE EL MATRIX IS INITIALIZED TO ZERO IN LOADSY)

REDEFINE THE EL MATRIX AS THE NEW ELD CURVE THAT IS TEMPORARILY STORED IN ELF.

DO 60 I=1, NEWMAX
EL(I)=ELF(I)
60 CONTINUE

100 CONTINUE
RETURN
END

SUBROUTINE READIN

COMMON /ONE/ COLP(L), CPLDLP,L, DISRP, DX, FCR, HOURS, IBASYS, ITIN, ITMAX,
  • ITOUT, MAXALL, MAXI, MAXIMP, MAXOR, MAXNEM, MAXPLA,
  • MAXP, MAXALL, MXYEAR, RESMAR, UBOUND

COMMON /THO/ AVSP(20,4), CAPAB(20), ENEDEM(20,4), FATOPE(20),
  • IDEMP(10), IPCH(10), ISOL(20,10), MWIDTH(10),
  • LOHBO(20,10), MAXUN(20,10), MAXADD(10), MINUN(20,10),
  • NEXPID(10), ORDER(420,3), NSTPRE(20,10), LLNWBO(20,10),
  • NUPB(20,10), PEAKS(20,4), PLA(10,14), PLANT(20,14),
  • RESMAX(20), RESMAIN(20), SOL(20,2), UNE(20), MAINS(200)

COMMON /THREE/ EL(1250), ELDC(4,1250), ELF(1260), X(1250),
  • HAXELDC(20,4)

COMMON /SIX/TY(120), TINT(120)

DOUBLE PRECISION TY, TINT

C CRITICAL LOLP, HOURS PER SEASON

C READ(10,902) CPLP, HOURS

C WRITE(11,901) CPLDLP, HOURS

C BASE YEAR (CALENDAR), ELDC DIMENSION, NUMBER OF NEW PLANTS

C NUMBER OF SCHEDULED PLANTS, ORIGINAL ELDC DIMENSION,

C NUMBER OF YEARS IN THE STUDY PERIOD

C READ(10,901) IBASYS, MAXI, MAXIMP, MAXP, MAXALL, MXYEAR, MAXOR

C WRITE(11,900) IBASYS, MAXI, MAXIMP, MAXP, MAXALL, MXYEAR, MAXOR

C MAXIMUM NUMBER OF POINTS OF SCHEDULED SYSTEM FINAL

C ELDC'S. (BY SEASON PER YEAR)

C DO 10 NY = 1, MXYEAR

C READ(10,901) (HMLDC(NY), I = 1, 4)

C WRITE(11,900) (HMLDC(NY), I = 1, 4)

C READ FOR EACH YEAR: SCHEDULED SYSTEM CAPACITY,

C MIN. & MAX. RES. MARGINS; MINIMUM POSSIBLE YEARLY

C PRODUCTION COST

C DO 20 NY = 1, MXYEAR

C READ(10,902) CAPAB(NY), RESMIN(NY), RESMAX(NY), FATOPE(NY)

C 20 WRITE(11,902) CAPAB(NY), RESMIN(NY), RESMAX(NY), FATOPE(NY)

C DEFINE AND PRINT THE NORMALIZED X-AXIS.

C DX = 1.0 / MAXPO

C DO 30 I = 1, MAXP

C X(I) = I * DX

C WRITE(11,906) (X(I), I = 1, MAXI)

C PEAK LOAD DEMAND (BY SEASON PER YEAR).

C DO 40 NY = 1, MXYEAR

C READ(10,902) (PEAKS(NY), I = 1, 4)

C 40 WRITE(11,902) (PEAKS(NY), I = 1, 4)

C SCHEDULED SYSTEM TOTAL ENERGY DEMAND (BY SEASON PER YEAR).

C DO 50 NY = 1, MXYEAR

C READ(10,902) (ENEDEM(NY), I = 1, 4)

C 50 WRITE(11,902) (ENEDEM(NY), I = 1, 4)

C EXPANSION CANDIDATE DATA

C DO 60 IP = 1, MAXIMP

C READ(10,904) (PLCA(IP), J = 1, 14)

C 60 WRITE(11,904) (PLCA(IP), J = 1, 14)

C MIN. & MAX. NUMBER OF UNITS ALLOWED FOR EACH CANDIDATE PER YEAR.

C DO 80 NY = 1, MXYEAR

C READ(10,900) (MINUN(NY,IP), IP = 1, MAXIMP)

C 80 WRITE(11,900) (MINUN(NY,IP), IP = 1, MAXIMP)

C READ PREP MODULE SOLUTION FOR CALCULATION OF OBJ UPPER BOUND
1960. DO 90 NY=1,MXYEAR
1961. READ(10,900) (NSTP(NY,IP),IP=1,MAXINP)
1962. 90 WRITE(11,901) (NSTP(NY,IP),IP=1,MAXINP)
1963. C READ SPACE AVAILABLE FOR MAINTENANCE
1964. DO 100 NY=1,MXYEAR
1965. READ(10,902) (AVSP(NY,J), J=1,4)
1966. 100 WRITE(11,903) (AVSP(NY,J), J=1,4)
1967. C
1968. C READ NORMAL DISTRIBUTION TABLE
1969. READ(9,907) (TY(K),TINT(K),K =1,120)
1970. Q WRITE(11,908) (TY(K),TINT(K),K =1,120)
1971. C
1972. 900 FORMAT(14)
1973. 901 FORMAT(1H * 20)
1974. 902 FORMAT(1E16.8)
1975. 903 FORMAT(1H * E16.8)
1976. 904 FORMAT(1H * 5I * F10.4 * F12.8)
1977. 905 FORMAT((1H * 5I * F10.4 * F12.8))
1978. C
1979. RETURN
1980. END
1981. C
1982. RETURN
1983. END
1984. C
1985. SUBROUTINE REPORTSALV, RESCAP)
1986. COMMON /ONE/ CLULOPIPLOP, DISRAT, DX, FCR, HOURS, IBASYR, ITIN, ITMAX,
1987.  + ITOUT, MAXALL, MAXI, MAXINP, MAXOUT, MAXNEW, MAXPLA,
1988.  + MAXPO, MAXALL, MXYEAR, RESMAR, UBOUND
1989. C
1990. COMMON /TWO/ AVSP(20,4), CAPAB(20,1), ENEDEM(20,4), FATUPE(20),
1991.  + IDEXP(10), IPCHI(10), ISOL(20,10), INITH(10),
1992.  + LLHAV(20,10), MAXUN(20,10), MINUN(20,10),
1993.  + NEXPD(10), NORDER(4,20,3), NSTPRE(20,10),
1994.  + NUPC(20,10), PEAKS(20,4), PLAF(10,14), PLANT(210,14),
1995.  + RESMAX(20), RESMIN(20), SOL(20,2), UNE(20), RAINES(200)
1996. COMMON /THREE/ EL(1250), PUPEC1(250,20), ELF(1250),
1997.  + EL(1250), WELDC(20,4)
1998. COMMON /SEVEN/ NP, MX1
1999. DIMENSION SALV(20), RESCAP(20,10), OPTOT(20), CAP(20),
2000.  + NTABLE(20,10), RSCAP1(20,10), POPEC1(10,20),
2001.  + IPL(10)
2002. DIMENSION A(250), A2(20), FMT1(18), FMT3(18), FMT5(18),
2003.  + FMT7(18), C(60), IYR(20)
2004. EQUIVALENCE (EL(1), A11111), (EL(251), NTABLE(1)),
2005.  + (EL(451), PUPEC1(1)), (EL(651), RSCAP1(1)),
2006.  + (EL(651), OPTOT(1)), (EL(871), CAP(1)),
2007.  + (EL(891), IYR(1)), (EL(911), IPL(1)), (EL(921), A2(1))
2008. EQUIVALENCE (EL(999), C(1)), (EL(1001), FMT1(1)),
2009.  + (EL(1021), FMT2(1)), (EL(1041), FMT3(1)),
2010.  + (EL(1061), FMT4(1)), (EL(1081), FMT5(1)),
2011.  + (EL(1101), FMT6(1)), (EL(1121), FMT7(1))
2012. DATA NONE/11X/
2013. REWIND 19
2014. MX1=MXYEAR-MXYEAR/2
2015. 15 CONTINUE
2016. DO 13 I=1,MAXINP
2017. 13 IPL(1)=1
2018. DO 15 NY=1,MXYEAR
2019. 15 CONTINUE
2020. DO 21 NY=1,MXYEAR
2021. UPTOT(NY)=0
2022. CAP(NY)=0
2023. IYR(NY)=IBASYR+NY
2024. DO 25 NY=1,MXYEAR
2025. CONTINUE
2026. DO 23 NY=1,MXYEAR
2027. 23 MAXNEW=0
2028. 25 CONTINUE
2029. RSCAP1(NY,KK)=RESCAP(NY,1)
2030. NTABLE(NY,1)=1SOL(NY,1)
2030. 23 CONTINUE
2031. 21 CONTINUE
2032. DO 25 NY=1,MXYEAR
2033. DO 26 I=1,MAXINP
2034. 26 OPTOT(NY)=OPTOT(NY)+POPEC(I,NY)
2035. DO 27 I=1,MAXINP
2036. 27 CAP(NY)=CAP(NY)+RSCAP1(NY,I)
2037. 25 CONTINUE
2038. WRITE (6,41)
2039. 41 FORMAT (1H1)
2040. DO 35 L=1,10
2041. WRITE (6,71)
2042. 35 CONTINUE
2043. WRITE (6,43)
2044. 43 FORMAT (1H,30X,23H CHARACTERISTICS OF THE,
2045. +3IH OPTIMAL OR SUBOPTIMAL SOLUTION//)
2046. WRITE (6,46)
2047. 46 FORMAT (1H,15X,11H PLANT TYPE,24X,9H UNSERVED,4X,6H TOTAL)
2048. WRITE (6,51)
2049. 51 FORMAT (1H,51X,7TH ENERGY,4X,10H OPERATING,
2050. +1X,6H CAPITAL,9X,6H SALVAGE,2X,10H OBJECTIVE)
2051. WRITE (6,55) I1P
2052. 55 FORMAT (1H,5H YEAR,1X,10(1X,12),6X,5H LQDP,6X,6H (MWH),5X,
2053. +6H COSTs,4X,7H COSTs,4X,7H VALUE,3X,10H FUNCTION)
2054. DO 60 NY=1,MXYEAR
2055. WRITE (6,61) IYR(NY),NNTABLE(NY,1),I=1,MAXINP),SOL(NY,1),
2056. +ONE(NY),OPTOT(NY),CAP(NY),SALV(NY),SOL(NY,2)
2057. 61 FORMAT (/1H,15,1X,10(1X,12),1X,F15.0,1X,E12.5,4(1X,F10.3))
2058. 60 CONTINUE
2059. WRITE (6,67) IBASYR
2060. 67 FORMAT (//1H,16H * MILLIONS OF ,14,8H DOLLARS)
2061. WRITE (6,69)
2062. 69 FORMAT (//1H,38H * PLANT COSTS ARE GIVEN AS THE TOTAL,
2063. +4HZMTH WORTH OF THE PLANT AS IT COMES ON LINE LESS THE //
2064. +4GH SALVAGE VALUE AT THE END OF THE STUDY PERIOD,
2065. +4IH IF THE FIXED CHARGE RATE OPTION IS USED, //
2066. +34H CAPITAL COSTS REPRESENT THE FIXED,
2067. +24H CHARGES FOR EACH PLANT.)
2068. WRITE (6,41)
2069. DO 74 I=1,10
2070. WRITE (6,71)
2071. 74 CONTINUE
2072. 71 FORMAT (1H /)
2073. 72 CONTINUE
2074. READ (19,99) C
2075. 99 FORMAT (3OA2)
2076. WRITE (6,130)
2077. 10 FORMAT (1HL)
2078. READ (19,100) FMT1
2079. 10 FORM1 (19,100) FMT2
2080. READ (19,105) FMT3
2081. READ (19,110) FMT4
2082. READ (19,110) FMT5
2083. READ (19,115) FMT6
2084. READ (19,116) FMT7
2085. 100 FORMAT (2A4,A2,A10A4,A2,A4A4)
2086. 105 FORMAT (A3,A2,A2,15A4)
2087. 110 FORMAT (4A4,A2,13A4)
2088. 115 FORMAT (3A4,A1,A2,13A4)
2089. 116 FORMAT (5A4,A2,12A4)
2090. NP=MXALL
2091. DO 122 NY=1,MXYEAR
2092. 122 ZNY=OPTOT(NY)
2093. CALL TABLE(FMT1,FMT2,FMT3,FMT4,FMT5,FMT6,
2094. +FMT7,C,A1,A2,1YK)
2095. NP=MAXINP
2096. DO 130 I=1,NP
2097. DO 140 J=1,MXYEAR
2098. 140 POPEC(I,J)=RSCAP1(J,1)
2099. 130 CONTINUE
2010. DO 125 NY=1,MXYEAR
SUBROUTINE RESIM(N, NBEST, IRES, KEEP)

RESIMULATES THE NBEST SOLUTIONS

COMMON /ONE/ CLOLP, CPLOLP, DISRAT, DXVAR, HOURS, IBASYS, ITIN, ITMAX,
  ITOUT, MAXALL, MAXI, MAXINP, MAXOR, MAXNEW, MAXPLA,
  MAXPO, MXAUX, MXYEAR, RESMAR, UBOUND

COMMON /TWO/ AVSP(20,4), CAPABS(20), ENEDEM(20,4), FATOP(20),
  IDXP(10), IPCH(10), ISUL(20,10), IWIDTH(10),
  LDHd(20,10), MAXUN(20,10), MAXADD(10), MINUN(20,10),
  NEXP10(10), NORDER(4,20,3), NSTPRE(20,10), LLOUB(20,10),
  NUPBI(20,10), PLACA(10,14), PLANT(210,14),
  RESMAX(20), RESSH(20), SOL(20,2), UNE(20), MAINS(200)

COMMON /THREE/ EL(1250), POPEC(250,20), ERF(1260), X(1250),
  MAXDLC(20,4)

COMMON /FOUR/ LIST(10,1000)

COMMON /FIVE/ PTCUM(40,8,4), PTCUMX(10,8), ROM(20,4)

COMMON /SIX/ TV(120), TINT(120)

DOUBLE PRECISION TV, TINT

DIMENSION KEEP(15), SALV(20), NSTATE(10), NSTCUR(10), RESCAP(20,10)
  + RESMAR(50)

FIND THE NBEST SOLUTIONS THE FIRST TIME CALLED FOR EACH MAIN

ITERATION

KEEP(I) = ITIN

WRITE(*,60) ITIN, N, NBEST, KEEP(I)

DO 5 NY = 1, MXYEAR

128 = NY

5 READ(*,128,900) (ISOL(NY,1)*I=1,6), SOL(NY,1), SOL(NY,2)

WRITE(*,60) (ISOL(NY,1)*I=1,6), SOL(NY,1), SOL(NY,2)

5 WRITE(*,110) (ISOL(NY,1)*I=1,6), SOL(NY,1), SOL(NY,2)

IF (NBEST .LE. 1) GO TO 100

IF (NY .NE. 1) GO TO 100

500 FORMAT(6I4,F13.9,E13.7)

FFIND = UBOUND*100.0

KSTOP = ITIN-1

IF (KSTOP .LT. 1) GO TO 100

DO 10 K=1,KSTOP

128 = (K-1)*20 + MXYEAR

READ(*,128,900) (ISOL(MXYEAR,1)*I=1,6), (SOL(MXYEAR,1),J=1,2)

10 REFIN(K) = SOL(MXYEAR,2)

100 FORMAT(5I4,F13.9,E13.7)

CONTINUE

DO 20 M=2,NBEST

IF (REFIN(K) .GE. FFIND) GO TO 20

20 CONTINUE

FFIND = REFIN(K)
RESIMULATE THE N-TH BEST SOLUTION

ISTAR = (KEEP(N)-1)*20 + 1
WRITE(6,9) N,KEEP(N),ISTAR
WRITE(6,10) OBJ = 0.0
C INITIALIZE RESCAP & POPEC.
DO 200 NY=1,MXYEAR
ISTAR = NY - 1
READ(28,128,900) (ISOL(NY,1),I=1,6),SOL(NY,1),SOL(NY,2)
WRITE(6,11) (PTCUM(I,J,N),I=1,201),J=1,8)
CONTINUE
DO 215 JSTAR=(NY-1)**41
READ(24*124,902) (NURES(NY,K),K=1,10)
READ(24*124,921) (MAIN(NY,J),J=1,MAXPLA)
FORMAT(1125(13,12,11))>
FORMAT(20012)
DO 230 N=1,MAXNEW
NSTATE=N
IF (NY .NE. 1) NSTATE=1
CONTINUE
CALL UPERCO(NSTCUR,NY,OPEC,IRES)
CALL UBjFUN(NSTCUR,NY,OPEC,NSTATE,CAPCO,OBJ,OBJJ,S,
+RESCAP,IRES)
SALV(NY) = S
C WRITE(6,9) NY,INES
C WRITE(6,10) NY,INES
C WRITE(6,10) NY,INES
C WRITE(11,9) NY,INES
C WRITE(11,9) NY,INES
C WRITE(11,9) NY,INES
C WRITE(11,9) NY,INES
C WRITE(11,9) NY,INES
C DO 300 NY = 1,MXYEAR
C WRITE(11,9) (RESCAP(NY,K),K=1,10)
C WRITE(11,9) (POPEC(1,NY),I=1,40)
C 300 CONTINUE
CALL REPORT(SALV,RESCAP)
RETURN
END

SUBROUTINE STAGENITUN, MAXSTA)

GENERATE THE STATES WITHIN THE "ITUN" TUNNEL, STORE THE
STATES IN COMMON BLOCK FOUR
COMMON /UNE/ CLOLP,CPLOLP,DISRT,DX,FCR,HOURS,IBASYR,ITIN,ITMAX,
+ ITOUT,MAXALL,MAX1,MAXINP,MAD,MAXNEW,MAXPLA,
+ MAXPU,MXALL,MXYEAR,RESMAR,UBOUND
COMMON /FOUR/ LIST(10,1000)

DIMENSION ITUN(10,3)

INITIALIZE LIST
DO 10 I=1,10
   LIST(I,N) = 0
10 CONTINUE

FIND THE NUMBER OF STATES THAT WILL BE GENERATED
MAXSTA = 1
DU 20 I=1,MAXNEW
MAXSTA = MAXSTA*ITUN(1,3)
IF (MAXSTA .LE. 1000) GO TO 22
EVENTUALLY THE PROGRAM MAY BE CHANGED TO ACCOMODATE
MORE STATES SINCE A LOT OF THE STATES GENERATED BY
THIS SUBROUTINE WILL BE REJECTED IN THE RESERVE MARGIN
AND LOLP CHECKS
WRITE(11,901) MAXSTA
MAXSTA = 1000

GENERATE THE STATES
MAXD = 1
MD = 1
DU 60 I=1,MAXNEW
MAXD = MAXD*ITUN(1,3)
MD = MD*ITUN(1,3)
25 CONTINUE
DO 50 N1=1,MAXSTA,MAXD
NO = 0
DO 40 N2=1,MAXD,MD
NO = NO+1
DO 30 N3=1,MD
NF = N1*N2*N3-2
30 LIST(1,NF) = ITUN(I,1)+NO-1
40 CONTINUE
50 CONTINUE
60 CONTINUE

WRITE(11,900) ((LIST(I,J),I=1,MAXNEW,J=1,MAXSTA)
900 FORMAT(I1*24,I1*40,13/))
901 FORMAT(I1,' THE NUMBER OF STATES GENERATED IS ',16F10.0)
'THIS EXCEEDS THE PROGRAM CAPACITY. ONLY THE FIRST',
'1000 STATES WILL BE CONSIDERED'
RETURN
END

SUBROUTINE TABLE(FMT1,FMT2,FMT3,FMT4,FMT5,FMT6,
+ FMT7,C1,A1,A2,YR)

COMMON ONE/CLOLP,CPLOLP,DISRAT,DX,FCR,HOURS,BASYR,IT1N,ITMAX,
+ ITOUT,MAYALL,MAX1,MAXIN,MAXOK,MAXNEW,MAXPLA,
+ MAXPU,MAYALL,MAYYEAR,RESMAR,UBOUND

COMMON THREE/EL(1250),PDPEC(1250,20),ELF(1250),
+ X(1250),MXcLDC(20,4)
COMMON NP,NX1
DIMENSION A1(250),A2(20),FMT1(118),FMT2(18),FMT3(18),
+FMT4(18),FMT5(18),FMT6(16),FMT7(18),C1(60),IYR(20)
LMX=51
LN=0
K=1
N=NX1
TOT=0.0
IF (MAYYEAR.LT.10) N=MAYYEAR
IPAGE=1
DO 170 I=1,NP
   DO 190 I=1,NP
SUBROUTINE TALK(IZ,ZINTEG)

COMMON /SIX/ Y(120),YINT(120)
DOUBLE PRECISION AVZ,Y,YINT,Z,ZINTEG,DY,PK,ZINT,FACTOR,
+ R1,R2,R3,R4,U12,Q14,Q14,Q14,Q24,Q34,Q21,Q31,Q41,Q32,Q42,Q43

C
C LOOKUP IN NORMAL DISTRIBUTION FUNCTION THE VALUE OF
C THE INTEGRAL OF EXP(-Z^2/2) FROM MINUS INFINITY TO Z.
C INTERPOLATE WITH CUBIC FIT BETWEEN THE 120 DATA POINTS
C COPIED FROM TABLES. THE REFERENCE USED IS: TABLES OF
C NORMAL PROBABILITY FUNCTIONS BY THE U.S. DEPARTMENT
C OF COMMERCE.
C
C DEFINITION OF KEY VARIABLES:
C NAME TYPE SIZE MEANING

SUBROUTINE MINTABLE(Z,ZINTEG)

COMMON /SIX/ Y(120),YINT(120)
DOUBLE PRECISION AVZ,Y,YINT,Z,ZINTEG,DY,PK,ZINT,FACTOR,
+ R1,R2,R3,R4,U12,Q14,Q14,Q14,Q24,Q34,Q21,Q31,Q41,Q32,Q42,Q43

C
C LOOKUP IN NORMAL DISTRIBUTION FUNCTION THE VALUE OF
C THE INTEGRAL OF EXP(-Z^2/2) FROM MINUS INFINITY TO Z.
C INTERPOLATE WITH CUBIC FIT BETWEEN THE 120 DATA POINTS
C COPIED FROM TABLES. THE REFERENCE USED IS: TABLES OF
C NORMAL PROBABILITY FUNCTIONS BY THE U.S. DEPARTMENT
C OF COMMERCE.
C
C DEFINITION OF KEY VARIABLES:
C NAME TYPE SIZE MEANING
END
RETURN
IF (ZINT=.5) ZINT = .4999999999999999
ZINT = 0
RETURN
ENDIF

A INCREMENT
6 REAL POINT CORRESPONDING TO Z
DO REAL = A(1), A(2), A(3), A(4), A(5), A(6)

INTERNAL VALUES CORRESPONDING TO A
(INPUT DATA)
INTEG. REAL VALUES

NORMALIZE INDEPENDENT VARIABLE
INTEGRAL VALUE
NORMALIZED INDEPENDENT VARIABLE
Z
REAL

Z
REAL
SUBROUTINE TUNNELINSTATE, NYNEXT, ITUN

DEFINES THE TUNNEL FOR ANY "ORIGIN" STATE "NSTATE" IN STAGE (YEAR) "NY". THE TUNNEL LOWER BOUND FOR CANDIDATE PLANT 1 IS RETURNED IN ITUN(1,1), THE UPPER BOUND IN ITUN(1,2), AND THE TUNNEL WIDTH IN ITUN(1,3). THESE TUNNELS ARE USED IN GENERATING THE POSSIBLE STATES IN STAGE(YEAR) NYNEXT = NY+1.

COMMON /ONE/ CLOLP, CPLULP, DISRAT, DX, FCR,OURS, IBASYR, ITIN, ITMAX,
+ ITOUT, MAXALL, MAXI, MAXINP, MAXOR, MAXNEW, MAXPLA,
+ MAXPO, MXALL, MXYEAR, RESMAR, UBOUND

COMMON /TWO/ AUSI(20, 4), CAPABS(20), ENDEM(20, 4), FATOPE(20),
+ IDEXP(10), IPCH(10), ISOL(20, 10), IWIDTH(10),
+ JOUB(20, 10), MAXUN(20, 10), MAXADD(10), MINUN(20, 10),
+ NEXPIDI(10), NORDER(420, 3), NSTPRE(20, 10), LLQHB(20, 10),
+ NUPB(20, 10), PEAKS(20, 4), PLACA(10, 14), PLANT(210, 14),
+ RESMAX(20), RESMIN(20), SOL(20, 2), UNE(20), RAINS(200)

COMMON /THREE/ ELI1250, ELC(4, 1250), ELF(1260), X(1250),
+ MXELDC(20, 6)

COMMON /FOUR/ LIST(10, 1000)

COMMON /FIVE/ PTCUM(20, 8, 4), PTCUMX(10, 8), ROM(20, 4)

COMMON /SIX/TY(120), TINT(120)

DOUBLE PRECISION TY, TINT

DIMENSION NSTATE(10), NSTCUR(10), RESCAP(20, 10)

DO 10 J=1, MAXNEW
  IPP = IDEXP(I)
  ITUN(1,1) = MAX(NSTATE(J), LOWB(NYNEXT, J), MINUN(NYNEXT, IPP))
  ITUN(1,2) = MIN(ADD, NUPB(NYNEXT, J), MAXUN(NYNEXT, IPP))
  ITUN(1,3) = ITUN(1,2)-ITUN(1,1)+1
  CONTINUE

WRITE(I1, *) ((ITUN(I,J), I=1,MAXNEW), J=1,3)

RETURN
END

SUBROUTINE UFIRST

CALCULATES THE FIRST ESTIMATE OF THE OBJECTIVE FUNCTION BASED ON TRADITIONAL OBJ DEFINITION AND DISCOUNT FACTOR OF 10%.

COMMON /ONE/ CLOLP, CPLULP, DISRAT, DX, FCR,OURS, IBASYR, ITIN, ITMAX,
+ ITOUT, MAXALL, MAXI, MAXINP, MAXOR, MAXNEW, MAXPLA,
+ MAXPO, MXALL, MXYEAR, RESMAR, UBOUND

COMMON /TWO/ AUSI(20, 4), CAPABS(20), ENDEM(20, 4), FATOPE(20),
+ IDEXP(10), IPCH(10), ISOL(20, 10), IWIDTH(10),
+ JOUB(20, 10), MAXUN(20, 10), MAXADD(10), MINUN(20, 10),
+ NEXPIDI(10), NORDER(420, 3), NSTPRE(20, 10), LLQHB(20, 10),
+ NUPB(20, 10), PEAKS(20, 4), PLACA(10, 14), PLANT(210, 14),
+ RESMAX(20), RESMIN(20), SOL(20, 2), UNE(20), RAINS(200)

COMMON /THREE/ ELI1250, ELC(4, 1250), ELF(1260), X(1250),
+ MXELDC(20, 6)

COMMON /FOUR/ LIST(10, 1000)

COMMON /FIVE/ PTCUM(20, 8, 4), PTCUMX(10, 8), ROM(20, 4)

COMMON /SIX/TY(120), TINT(120)

DOUBLE PRECISION TY, TINT

DIMENSION NSTATE(10), NSTCUR(10), RESCAP(20, 10)

END
C

IF (ITOUT .LE. 0) GO TO 10
RETURN
C

10 CONTINUE
OBJ = 0.0
C
FILL THE IDEAP AND NEXPID ARRAYS
C
DO 20 IP = 1,MAXINP
IDEAP(IP) = 1
C
20 IDEAP(1) = 1
C
MAXNEW = MAXINP
MXXALL = MAXPLA + MAXINP
C
DO 40 IP = 1,MXXALL
I23 = IP
C
40 CONTINUE
C
WRITE(11,901) (PLANT(IP,J),J=1,14)
C
901 FORMAT(1H ,F3.0 ,2F5.0,F3.0,F8.6,F3.0,F7.2,F3.0,F7.3,E10.3,F4.2)
C
FLK = -1.0
C
DISRAT = 0.10
C
DO 50 NY = 1,MXYEAR
C
50 CONTINUE
C
ISTAR = (NY-1)*4 + 1
C
ISTAR = (NY-1)*5 + 1
C
DO 55 N = 1,4
I27 = ISTAR + N - 1
C
55 CONTINUE
C
READ(27*127,902) ((PTCUM(I,J,N),I=1,201),J=1,8)
C
902 FORMAT(125 (13,12,131))
C
WRITE(11,*)(PTCUM(I1,1,1),I=1,31)
C
C
WRITE(11,921) (MAINS(J),J=1,MAXPLA)
C
921 FORMAT(20012)
C
WRITE(11,*)(MAINS(J),J=1,MAXPLA)
C
CALL OPEC(IRES)
C
CALL OBJFUN(IRES)
C
+RESCAP, IRES)
C
WRITE(6,*)(NSCUR(I),I=1,MAXNEW,NSSTATE,RESMAR)
C
WRITE(6,*)(OPEC,CAPOC,OBJ1,OBJ,RESMAR)
C
SUBROUTINE UNPACK(NST,NSTATE, NY,IPFLAG)
C
UNPACKS STATE "NST" AND STORES IT IN "NSTATE" ARRAY
C
COMMON /ONE/ CLOLP,CPLOLP,DISRAT,FLK,HOURS,INASYR,ITIN,ITMAX,
C
ITOUT,MAXALL,MAXI,MAXINP,MAXOR,MAXNEW,MAXPLA,
C
MAXPO,MAXALL,MXYEAR,RESCAP,UBOUND
C
C
SUBROUTINE UNPACK(NST,NSTATE, NY,IPFLAG)
COMMON /TWO/ AVS(20,4),CAPAB(20),ENEDEM(20,4),FATOPE(20),
+ IDEXP(10), ICHD(10), ISOFL(20,10),IWIDTH(10),
+ LOWF(20,10),MAXUN(20,10),MAXADD(10),MINUN(20,10),
+ NEXP(10),NORDER(420,3),NSTPRE(20,10),LLOWB(20,10),
+ NUPB(20,10),PEAKS(20,4),PLAC(10,14),PLANT(210,14),
+ RESMAX(20),RESMIN(20),SOL(20,2),UNE(20),MAINS(200)

2590. C
2591. C
2592. C
2593. C
2594. C
2595. C
2596. C
2597. C DIMENSION NSTATE(10), LSTATE(10)
2598. C
2599. C IF (NST .LE. 0) GO TO 20
2600. MULT = 6**((MAXNEW-1)
2601. NSTEMP = NST
2602. DO 10 I=1,MAXNEW
2603. I1 = MAXNEW+1
2604. IR = MOD(NSTEMP,MULT)
2605. LSTATE(I1) = (NSTEMP-IR)/MULT
2606. MULT = MULT/6
2607. NSTEMP = IR
2608. 10 CONTINUE
2609. GO TO 35
2610. C
2611. 20 CONTINUE
2612. DO 30 I=1,MAXNEW
2613. 30 LSTATE(I) = 0
2614. 35 DO 40 I=1,MAXNEW
2615. IF (IFLAG.GT.0) NSTATE(I) = LSTATE(I) + LLOWB(NY,1)
2616. IF (IFLAG.LT.1) NSTATE(I) = LSTATE(I) + LLOWB(NY,1)
2617. 40 CONTINUE
2618. RETURN
2619. END
2620. C
2621. C
2622. C
2623. SUBROUTINE VALUE(Z,C3,C4,FRESGL)
2624. C
2625. C
2626. C
2627. COMMON /SIX/ TY(120), TINT(120)
2628. DOUBLE PRECISION TV, TINT
2629. C
2630. C
2631. C DOUBLE PRECISION Z,C3,C4,F3,F4,F6,E,ZINTEG,ZSQ
2632. IF (Z.GT.5.90D0) GO TO 40
2633. ZSQ = Z*Z/2.0
2634. E = DEXP(-ZSQ)/2.506628275000
2635. ZN(1) = -Z*E
2636. ZN(2) = E*(Z*Z - 1.0D00)
2637. ZN(3) = -2*E*(Z*Z*Z - 3.0D00)
2638. ZN(4) = 6*E*(Z*Z*Z*Z - 6.0D00) - 3.0D00)
2639. ZN(5) = -24*E*(Z*Z*Z*Z*Z - 10.0D00) + 15.0D00)
2640. F3 = 6.0D00
2641. F4 = 24.0D00
2642. F6 = 720.0D00
2643. C
2644. CALL TALOOK(Z,ZINTEG)
2645. C
2646. C FREQ = 1.0D00 - ZINTEG
2647. FREQ = FREQ + DBLE(C3)*ZN(2)/F3
2648. + DBLE(C4)*ZN(3)/F3
2649. + DBLE(C3)*DBLE(C3)*ZN(5)*10.0D00/F6
2650. IF(FREQ.LT.0.0000) FREQ = 0.0000
2651. IF(FREQ.GT.1.0000) FREQ = 1.0000
2652. C
2653. C FRESGL = SNGL(FREQ)
2654. C
2655. RETURN
2656. 40 FRESGL = 0.0
2657. RETURN
2658. END
The plant and load cumulants are calculated from their central moments [68]. The plant moments are:

\[ m_{1ij} = (1 - \text{FOR}_i) \cdot \text{EC}_{ij} \]

\[ m_{rij} = (1 - \text{FOR}_i) \cdot (\text{EC}_{ij} - m_{1ij})^r, \quad r = 2, \ldots, 8 \]

where \( m_{rij} \) is the \( r \)-th moment of plant type \( i \) in season \( j \)

\( m_{1ij} \) is the average capacity for plant type \( i \) in season \( j \)

\( \text{FOR}_i \) is the forced outage rate of plant \( i \)

\( \text{EC}_{ij} \) is the capacity of plant \( i \) in season \( j \) or equivalent plant capacity when the plant is derated for maintenance [41].

For plants with two blocks of capacity, the central moments for the first capacity block are calculated by the above equation, by replacing \( \text{EC}_{ij} \) with \( \text{ECB}_{ij} \), defined as

\[ \text{ECB}_{ij} = (1 - \text{MOR}_{ij}) \cdot \text{CB}_i \]

where \( \text{MOR}_{ij} \) is the \( i \)-th plant maintenance outage rate in season \( j \)

\( \text{CB}_i \) is the capacity of the base block of unit \( i \)
The load moments are calculated from the hourly load demand data.

They are

\[ \text{LM}_1 = \frac{1}{T} \sum_{t=1}^{T} L_t \]

\[ \text{LM}_r = \frac{1}{T} \sum_{t=1}^{T} (L_t - \text{LM}_1)^r \quad r = 2, \ldots, 8 \]

where \( \text{LM}_1 \) is the average load

\( \text{LM}_r \) is the \( r \)-th load moment

\( L_t \) is the load demand at the \( t \)-th hour, in MW

\( T \) is the simulation period length in hours

The formulas for calculating the first eight cumulants are

\[ k_1 = m_1 \]

\[ k_2 = m_2 \]

\[ k_3 = m_3 \]

\[ k_4 = m_4 - 3m_2^2 \]

\[ k_5 = m_5 - 10m_3 m_2 \]

\[ k_6 = m_6 - 15m_4 m_2 - 10m_3^2 + 30m_2^3 \]
\[ k_7 = m_7 - 21m_5m_2 - 35m_4m_3 + 210m_3^2 \]
\[ k_8 = m_8 - 28m_6m_2 - 56m_5m_3 - 35m_4^2 \]
\[ + 420m_4m_2^2 + 560m_3^2m_2 - 630m_2^4 \]

where \( k_r \) is the \( r \)-th cumulant

\( m_r \) is the \( r \)-th central moment

The plant cumulants are calculated by replacing \( m_r \) with the plant central moments \( m_{rij} \) as defined above. Similarly, for the calculation of the load cumulants \( m_r \) should be substituted by the load central moments \( LM_r \).
APPENDIX C
PROSES SOURCE LISTING

This Appendix contains the Fortran source listing of the PROduction Simulation for Electrical Systems (PROSES) code that was developed to analyze the cumulant, derate, and piecewise linear approximations of probabilistic simulation of electric generating systems. The load forecast uncertainty algorithm is also included.
**Production Simulation of Electrical Systems**

The PROSES program is designed to test the accuracy and speed of the following production simulation methods for electricity generating systems:

- Probabilistic simulation using piece-wise linear representation of the load duration curves.
- Probabilistic simulation using cumulants for the representation of the load duration curves.
- Plant derating.

This purpose is accomplished by comparing the results of the cumulant and derating methods with the results obtained by piece-wise linear representation of the load duration curves. The later method has been extensively tested at the Ohio State University and the National Regulatory Research Institute and it has proven to give "LOLP" results accurate to less than one day in 10 years (Ref). Therefore the piece-wise linear method can be safely considered the "reference method".

The "PROSES" code calculates by each of the following:

A. Equivalent load duration curves (EL).
B. Expected energy generation by each unit.
C. System loss of load probability (LOLP) and expected unserved energy.

The results are printed and/or plotted at various levels of detail.

**Program Structure**

2. Main Program

The main program reads the following input data:

A. Generating units capacities and forced outage rates
B. System hourly loads

The main program prints tables for comparison of the results obtained by the three methods tested. This is done when test option 10 is invoked.

The main program controls the execution of the "PROSES" subroutine according to the table listed below:

<table>
<thead>
<tr>
<th>Option</th>
<th>Calculation Method</th>
<th>Subroutine</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>END</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>LD AND LF</td>
<td>FRED</td>
</tr>
<tr>
<td>2</td>
<td>LD FROM INPUT</td>
<td>FRED</td>
</tr>
<tr>
<td>3</td>
<td>ELS</td>
<td>P-W LINEAR</td>
</tr>
<tr>
<td>4</td>
<td>ELS</td>
<td>P-W LINEAR, CUMULANTS</td>
</tr>
<tr>
<td>5</td>
<td>ELS</td>
<td>CUMULANTS, HRL LOADS</td>
</tr>
<tr>
<td>6</td>
<td>ELS</td>
<td>CUMULANTS, LD DIFF.</td>
</tr>
<tr>
<td>7</td>
<td>PLANT'S ENERGY</td>
<td>CUMULANTS</td>
</tr>
<tr>
<td>8</td>
<td>PLANT'S ENERGY</td>
<td>P-W LINEAR</td>
</tr>
<tr>
<td>9</td>
<td>PLANT'S ENERGY</td>
<td>DERATE</td>
</tr>
<tr>
<td>10</td>
<td>CHECKS</td>
<td>MAIN</td>
</tr>
</tbody>
</table>

**Legend:**
- LD: Original load duration curve
- LF: Original load frequency function
- EL: Equivalent load duration curve
- ELF: Equivalent load frequency function
2.2 SUBROUTINE "FRED" (TEST=LIST). FREQUENCY AND DURATION CURVES

A. TEST = 1. Calculates the load duration (LD) and load frequency (LF) curves of the system hourly load (HL) data. Applicable printout options are "LIST" = 1, 2, 3, 4.

B. TEST = 2. Reads the original load duration curve from input data.

2.3 SUBROUTINE "LINCON" (TEST=LIST). LINEAR CONVOLUTION.

A. TEST = 3. Calculates the equivalent load duration curves (EL) by representing them with a piece-wise linear polynomial in the generating unit convolution process. Applicable printout options are "LIST" = 1 - 10.

B. TEST = 4. Calculates the EL's using piece-wise linear representation of the EL's for the convolution of the scheduled system units and cumulants for the expansion candidates.

C. TEST = 8. Calculates the energy generated by each plant and the unserved energy. Applicable printout options are "LIST" = 1 - 10.

2.4 SUBROUTINE "CUCON" (TEST=LIST). CUMULANT CONVOLUTION.

A. TEST = 5. Calculates EL's using cumulants. The original LD cumulants are calculated from the input hourly loads. Applicable printout options are "LIST" = 3, 6 or 9.

B. TEST = 6. Same as 5 but the original LD cumulants are calculated by numerical differentiation of the input LD.

C. TEST = 7. Calculates the energy generated by each plant and the unserved energy.

2.5 SUBROUTINE DERATE

TEST = 9. Calculates the energy generated by each plant and the unserved energy using the plant derating method.

2.6 "LIST" = 10. Prints tables for comparison of the results obtained by the three methods tested.

2.7 SUBROUTINE "CUCAL". CUMULANT CALCULATION.

Auxiliary subroutine. Calculates the moments and cumulants.

2.8 FUNCTION "TALQOK" (TABLE LOOK-UP)

Interpolates between the standardized normal distribution ("FLAG" = 1) or the standardized normal frequency function data in order to find and return the requested values.

2.9 SUBROUTINE "PRIPLQ". PRINT-PLOT.

Prints and plots according to the following "LIST" options:

<table>
<thead>
<tr>
<th>TEST</th>
<th>PRINTOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PRINT NOTHING</td>
</tr>
</tbody>
</table>
PRINT FINAL ARRAY, EVERY 10 POINTS
PRINT FINAL ARRAY EVERY POINT
PRINT & PLOT FINAL ARRAY, EVERY POINT
PRINT EVERY ARRAY EVERY 10 POINTS
PRINT EVERY ARRAY EVERY POINT
PRINT & PLOT EVERY ARRAY, EVERY POINT
PRINT "1"TH ARRAY (I.E. AFTER "1"TH UNIT IS CONVOLVED) EVERY 10 POINTS
PRINT "1"TH ARRAY (I.E. AFTER "1"TH UNIT IS CONVOLVED) EVERY POINT
SAME AS 9 WITH PLOTTING

TABLE 3. "LIST" OPTIONS
FOR TEST=10 AND PRINT PLOT OF ENERGY & EL DIFFERENCES

<table>
<thead>
<tr>
<th>LIST OPTION</th>
<th>PRINT</th>
<th>PLOT</th>
<th>PLOT X STEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NO</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>YES</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>YES</td>
<td>YES</td>
<td>2.0</td>
</tr>
<tr>
<td>3</td>
<td>YES</td>
<td>YES</td>
<td>2.0,1.0</td>
</tr>
<tr>
<td>4</td>
<td>YES</td>
<td>YES</td>
<td>2.0,1.0,0.4</td>
</tr>
<tr>
<td>5</td>
<td>YES</td>
<td>YES</td>
<td>2.0,1.0,0.4,0.2</td>
</tr>
</tbody>
</table>

2.10 SUBROUTINE "NDF"
FINDS THE ELDC OR ELF VALUES FOR A GIVEN STANDARDISED X VALUE, USING GRAM-CHARLIER OR EDGORTH SERIES.

2.11 SUBROUTINE "PWADD"
PLANT CONVOLUTION USING P-W LINEAR ELDC REPRESENTATION.

2.12 SUBROUTINE "PLOLP"
FINDS THE LOLP FOR A GIVEN ELDC AND X VALUE.

LOCAL VARIABLES

COMMON /CURVES/ X(3000), EL(3000), ELF(3000), NORMLD(1000)
COMMON /ARRAYS/ CHECK(100,10,2), ENERGY(100,3), HOURC(8), HRL0ADI(24),
+CUMLIN(100,8), PLANTC(100,8), PLANT(100,2), TESTAR(10,2), EUNE(100,2),
+EAST(100,2)
COMMON /VARIAB/ COUNTH, COUNTP, DX, DXNORM, INL, MAXI, MULTI, NEXIST,
+OUT, PEAK, POINTS, SYSCAP, TEN, PLT, HRT, TCE, TLPC, TLE, TLPL, TDE
COMMON /TABLE/ TY(120), TINT(120)
DOUBLE PRECISION TV, TINT

INTEGER COUNTH, COUNTP, HRL0ADI, INL, OUT, PEAK, POINTS, TESTAR
REAL NORMLD
DOUBLE PRECISION CUMLIN, PLANTC, HOURC

DECLARATION OF LOCAL VARIABLES
INTEGER INP, LIST, LISTAR(10), TEST, YES
DIMENSION ENODE(100), ELASTU(100), EPER1(100), EPER2(100),
+XAXIS(100), XPANT(100)
DATA YES/3 YES/
INL=11
INP=12
C READ NORMAL DISTRIBUTION TABLE
915 FORMAT(F10.4,F12.8)
C INITIALIZE THE CHECK ARRAY
DO 20 I=1,100
   DO 10 J=1,10
      CHECK(I,J,1) = 1.0
      CHECK(I,J,2) = 0.0
20 CONTINUE
C
READ PLANT DATA (CAPACITY & FOR)
DO 40 I=1,101
   COUNTP = I - 1
   READ(INP, 902) (PLANT(I,J), J=1,2)
40 CONTINUE
50 IF(COUNTP .GT. 10) GO TO 60
WRITE(6, 903) COUNTP
902 FORMAT(2F10.4)
903 FORMAT(37H * * * * * THE NUMBER OF PLANTS IS , I3, 
         +37H THIS IS NOT ENOUGH PROGRAM ABENDS.)
STOP
60 WRITE(6,900)
900 FORMAT(51H SHOULD THE PLANT DATA BE PRINTED? ENTER YES OR NO / 
         +4H YYY )
READ(5,901) MARK
901 FORMAT(A3)
1F (MARK.NE.YES) GO TO 75
WRITE(13,904)
DO 70 I=1, COUNTP
   WRITE(13, 905) (PLANT(I,J), J=1,2)
70 CONTINUE
75 CONTINUE
C FIND THE SYSTEM CAPACITY
SYSCAP = 0.0
77 SYSCAP = SYSCAP + PLANT(I,1)
75 CONTINUE
77 SYSCAP = SYSCAP + PLANT(I,1)
C DEFINE THE NUMBER OF PLANTS IN THE EXISTING SYSTEM
NEXIST = COUNTP
C OPTIONALLY LIST THE TEST AND LIST OPTIONS
**FOR FUTURE COMPLETION**
C INITIALIZE THE TESTAR AND LISTAR ARRAYS
DO 80 I=1,10
   TESTAR(I) = -1
   L1STAR(I) = - 1
80 CONTINUE
C PROMPT AND READ TEST AND LIST OPTIONS
85 WRITE(6, 90b)
906 FORMAT(47H ENTER THE TEST AND LIST OPTIONS IN 2 12 FORMAT/
         +5H TTL)
READ(5, 907) TEST, LIST
907 FORMAT(2 1 2 )
IF (TEST .LE . 10 .AND. TEST .GE . 0 .AND. LIST .LE . 10 .AND. 
   +LIST .GE . 0) GO TO 90
WRITE(6, 908) TEST, LIST
908 FORMAT(22H ** ERROR ** /
         +15H INVALID TEST=,I2,9H OR LIST=,I2,8H OPTIONS)
90 CONTINUE
10 WRITE(13, 913) TEST, LIST
913 FORMAT(1H///21H OPTIONS ARE: TEST= ,I2,6H LIST= ,I2///)
1F1TEST .NE. 0) GO TO 100
C STORE LIST OPTION IN THE LISTAR ARRAY
LISTAR(LIST) = LIST

C GO TO 110, 120, 130, 140, 150, 160, 170, 180, 190, 200. 100 CONTINUE

110 CALL FRED(TEST, LIST)
GO TO 85

120 CALL FRED(TEST, LIST)
GO TO 85

130 CALL LICON(TEST, LIST)
GO TO 85

140 CALL LICON(TEST, LIST)
GO TO 85

150 CALL CCN(TEST, LIST)
GO TO 85

160 CALL CCN(TEST, LIST)
GO TO 85

170 CALL CCN(TEST, LIST)
GO TO 85

180 CALL LICON(TEST, LIST)
GO TO 85

190 CALL DERATE(TEST, LIST)
GO TO 85

200 CONTINUE

C THE CHECK ARRAY CONTAINS 10 ARBITRIRLY SELECTED VALUES
C OF THE EL ARRAY AFTER EACH PLANT CONVOLUTION. THE 10
C REFERENCE POINTS ARE SELECTED AT X=(SYSTEM CAPACITY)/I
C WHERE I VARIES FROM 10 TO 1.
C THE ELDC VALUES FOR THE LINEAR AND CUMULANTS CONVOLUTION
C ARE PRINTED ALONG WITH THEIR DIFFERENCE AND % CHANGE
C WITH RESPECT TO THE LINEAR CONVOLUTION VALUE.

IF (LIST.EQ.0) GO TO 230
DO 220 IP = 1, COUNT
WRITE(13, 910)
DO 210 J = 1, 10
DIFF = CHECK(IP, J, 1) - CHECK(IP, J, 2)
IF (ABS(CHECK(IP, J, 1)) .LT. 0.000001) GO TO 203
PERCEN = DIFF / CHECK(IP, J, 1) * 100.
GO TO 206
DIFF = 0.0
GO TO 206
205 CONTINUE

206 CONTINUE

XCAP1 = SYSCAP/I0. * FLUAT(I)
XCAP2 = XCAP1 / FLOAT(PEAK)

WRITE(13, 911) IP, J, XCAP1, XCAP2, {CHECK(IP, J, K), K=1,2}, DIFF, PERCEN

220 CONTINUE

230 CONTINUE

210 WRITE(13, 910) IP, J, XCAP1, XCAP2, {CHECK(IP, J, K), K=1,2}, DIFF, PERCEN

220 CONTINUE

911 FORMAT(1H1.49H PLANT CHECK CAPACITY NORMALIZED ELDC,
+24H DIFFERENCE % CHANGE /
+50H NUMBER POINT MW NORM. LINEAR CUMULANTS //)
912 FORMAT(4H , 13, 6X, I2 , 2X, F6.0 , 4!1X, F8.6 ), 1X, F11.4)
919 FORMAT(1H //3 0 X , 13 H PLANT ENERGY /
+22H PLANT CAPACITY FOR ;5X,7H LINEAR,8X,10H CUMULANTS ,
+7X,7H DERATE,7X,7H CUM-LIN %CUM-LIN/LIN DER-LIN %DER-LIN/LIN
+ I/)
919 FORMAT(1H //3 0 X , 13 H PLANT ENERGY /
+22H PLANT CAPACITY FOR ;5X,7H LINEAR,8X,10H CUMULANTS ,
+7X,7H DERATE,7X,7H CUM-LIN %CUM-LIN/LIN DER-LIN %DER-LIN/LIN
+ I/)
SUM1 = 0.0
SUM2 = 0.0
SUM3 = 0.0
DO 250 I = 1, COUNT
DPER1 = 0.0
DPER2 = 0.0
DE1 = ENERGY(1,2) - ENERGY(1,1)
DE2 = ENERGY(1,3) - ENERGY(1,1)
213

350. IF ( ENERGY(1,1) .LT. 1.E-06 ) GO TO 240
351. IF ( ABS(DE2(1,1)) .LT. 1.E-06 ) DPER2 = DE2/ENERGY(1,1)*100.0
352. CONTINUE
353.
354. C
355. EPER(1,1) = DPER1
356. EPER(2,1) = DPER2
357. XPLAN(1) = FLOAT(1)
358. C
359. SUME1 = SUME1 + ENERGY(1,1)/1000.0
360. SUME2 = SUME2 + ENERGY(1,2)/1000.0
361. SUME3 = SUME3 + ENERGY(1,3)/1000.0
362. WRITE(13,920) I, (PLANT(I,J), J=1,2), (ENERGY(I,J), J=1,3)
363. + DE1, DPER1, DE2, DPER2
365. WRITE(15,921) SUME1, SUME2, SUME3
366. WRITE(15,921) SUME1, SUME2, SUME3
367. 921 FORMAT(1H TOTAL ENERGIES ARE: 3(2X, F12.3), 4H GMH)
368. UE1 = TEN - SUME1
369. UE2 = TEN - SUME2
370. UE3 = TEN - SUME3
371. WRITE(16,922) UE1, UE2, UE3
372. WRITE(15,922) UE1, UE2, UE3
373. 922 FORMAT(1H THE UNSERVED ENERGIES FOR P-W LINEAR*/
374. +* CUMULANTS AND DERATE METHODS RESPECTIVELY ARE*1H ,
375. +3(4X, 12E13.5), 4H GMH)
376. PERUE1 = UE1*100.0/TEN
377. PERUE2 = UE2*100.0/TEN
378. PERUE3 = UE3*100.0/TEN
379. WRITE(16,923) PERUE1, PERUE2, PERUE3
380. WRITE(15,923) PERUE1, PERUE2, PERUE3
381. 923 FORMAT(1H THE % OF UNSERVED ENERGY WRT THE TOTAL ENERGY IS*/
382. +*3(I2X, E15.7))
383. IF (UE1.LT.0.001) GO TO 260
384. EDIF2 = (UE2 - UE1)*100.0/UE1
385. EDIF3 = (UE3 - UE1)*100.0/UE1
386. WRITE(16,925) EDIF2, EDIF3
387. WRITE(15,925) EDIF2, EDIF3
388. 925 FORMAT(1H THE % OF UNSERVED ENERGY WRT LINEAR CALCULATION IS*/
389. +* CUMULANTS: 3.5F9.4/* DERATE: 3.5F9.4/*
390. GO TO 270
391. 260 WRITE(16,926)
392. WRITE(15,926)
393. 926 FORMAT(* THE UNSERVED ENERGY CALCULATED BY P-W LINEAR*,
394. +* IS LESS THEN 0.001 GMH*/)
395. 270 CONTINUE
396. WRITE(16,927) PLT, HRT, IPL, TLPC, TLE, TCE, TDE
397. WRITE(13,923) PLT, HRT, IPL, TLPC, TLE, TCE, TDE
398. 923 FORMAT(1H /* CPU TIME IN SECONDS*/
399. +* PLANT CUMULANTS: 3.5T22, E13.8/* HOURLY CUMULANTS: 3.5T22, E13.8/
400. +* LOPP P-W LINEAR*: 3.5T22, E13.8/* LOPP CUMULANTS: 3.5T22, E13.8/
401. +* ENERGY P-W LINEAR*: 3.5T22, E13.8/* ENERGY CUMULANTS: 3.5T22, E13.8/
402. +* ENERGY DERATE*: 3.5T22, E13.8/*
403. C
404. IF (POINTS.LT.100) LIST = 0
405. IF (LIST.EQ.0) GO TO 85
406. C
407. MAXI = 100
408. M5 = POINTS * 1.5
409. I5 = M5/MAXI
410. SLIM = 0.000001
411. WRITE(13,927)
412. DD 320 1 = 1, MAXI
413. J = 1 * I5
414. XXAIJ(I) = X(J) + PEAK
415. EONE(I) = (EONE(I,1) - EONE(I,2))
416. ELAST(I) = (ELAST(I,1) - ELAST(I,2))
417. IF(ABS(EONE(I)) .LT. SLIM .OR. ABS(EONE(I)) .LT. SLIM) GO TO 280
418. EONE(I) = EONE(I) * 100.0 / EONE(I,1)
419. GO TO 290
280  EDNED(1) = 0.0
290  IF(ABS(ELASTD(1)) .LT. SLM .OR. ABS(ELAST(1,1)) .LT. SLM) GO TO 300
291  ELASTD(1) = ELASTD(1) * 100.0 / ELAST(1,1)
293  GO TO 310
300  ELASTD(1) = 0.0
310  WRITE(13,928) XAXIS(1),EDONE(1,1),EDONE(1,2),EDNED(1),
320  ELAST(1,1),ELAST(1,2),ELASTD(1)
320  CONTINUE
320
329  927  FORMAT(1H //T40, F FIRST AND LAST EL IN S/ 
330  +T6, *AXIS*,T30,*FIRST EL*,T70,*LAST EL*)
331  928  FORMAT(1H,14,F6.0,2(F11.7,F11.5))
332  C
333  LI = 1
334  CL1 = 1.0
335  IF (LIST .LE. 1) GO TO 85
336  IF (LIST .GT. 5) LIST = 5
337  DO 400 L = 2, LIST
338    LL = L-1
339  GO TO (390,330,340,350,L)
340  330  LI = 2
341  CL1 = 1.0
342  GO TO 360
343  340  LI = 5
344  CL1 = 0.4
345  GO TO 360
346  350  LI = 10
347  CL1 = 0.2
348  360  CONTINUE
349  DO 370 J = 1,MAX15
350   EDNED(J) = EDNED(J) * LL
351  370  ELASTD(J) = ELASTD(J) * LL
352  DO 380 J = 1,COUNTP
353    EPER2(J) = EPER2(J) * LL
354  380  EPER2(J) = EPER2(J) * LL
355  C
356  390  WRITE(13,929) CL1
357  CALL PLOT(XAXIS,EDNED,MIX15,LI)
358  WRITE(13,930) CL1
359  CALL PLOT(XAXIS,ELASTD,MIX15,LI)
360  WRITE(13,931) CL1
361  CALL PLOT(XPLANT,EPER1,COUNTP,LI)
362  WRITE(13,932) CL1
363  CALL PLOT(XPLANT,EPER2,COUNTP,LI)
364  C
365  400  CONTINUE
366  C
367  929  FORMAT(1H /* CUMULANT DIFFERENCES FOR THE FIRST EI IN S/ * F4.1,*% STEP*/)
368  930  FORMAT(1H /* CUMULANT DIFFERENCES FOR THE LAST EI IN S/ * F4.1,*% STEP*/)
369  931  FORMAT(1H /* ENERGY DIFFERENCES FOR CUMULANTS IN S/ * F4.1,*% STEP*/)
370  932  FORMAT(1H /* ENERGY DIFFERENCES FOR DERATE METHOD IN S/ * F4.1,*% STEP*/)
371  C
372  GO TO 85
373  C
374  END
375  C
376  SUBROUTINE FRED.TEST(LIST)
377  C
378  WHEN"TEST" = 1 FRED CALCULATES THE INITIAL LOAD DURATION
379  WHEN "TEST" = 2, FRED READS THE SYSTEM PEAK AND LD C FROM FILE 14
380  C
381  LIST OF KEY VARIABLES - COMMON AND LOCAL COMMON VARIABLES
382  C
383  NAME TYPE DIMENS. VALUE MEANING
384  C
385  COUNT INT. <878 NUMBER OF HOURLY LOADS
386  C
387  DX REAL - MK INCREMENT (X-AXIS)
388  C
389  NORMLD REAL 1000 - NORMALIZED LOAD DURATION CURVE
LOCAL VARIABLES

BASE INT. -   BASE LOAD
DEX REAL -   DIFFERENTIATION INCREMENT
KMAX INT. - <=1000 MAX SIZE OF CONDESED LP, LF AND X ARRAYS
LD REAL 1000 - LOAD DURATION CURVE
LF REAL 1000 - LOAD FREQUENCY CURVE
LFF REAL 1000 LF ARRAY FROM LD DIFFERENTIATION
LIST INT. - 1 - 10 LIST OPTIONS
LISTFL INT. - 1 PRINT AND PLOT LD
MULTI INT. - 1 - 10 MULTIPLICATION FACTOR
MAXFRE REAL - MAXIMUM FREQUENCY
MAXFRL REAL - LOAD OF MAX. FREQUENCY
PEAK INT. -  PEAK LOAD
X REAL 1000 - X-AXIS* MW ARRAY

DECLARATION OF ARRAYS AND VARIABLES

COMMON/CURVES/X(3000),EL(3000),ELF(3000),NORMLD(1000)
COMMON/ARRAYS/CHECK!100,10,2),ENERGY!100,3),HOURC(18),HRLOAD!24)
COMMON/VARIAB/LCOUNTH,COUNTP,DX,KXNORM,INL*MAXI*HULT1*NEXIST*
OUT(PEAK(POINTS*SYSCAP)*TEN*PLT*
HRT,

DECLARATION OF LOCAL VARIABLES

INTEGER BASE(ORIGIN*TESTyYES
REAL MAXFRE,MAXFRL,LF(IOOO),LFF(1000),LD(1000)
EQUIVALENCE(ELF(11),LF(1)),(ELF(1001),LFF(1)),(LD(1),EL1))
DATA YES/3HYES/3H/NA1/3HNA1/
YOU 38H IN FORMAT(LESS OR EQUAL TO 1000) /
5H NNNN)
READ(5,901) POINTS
901 FORMAT(14)
900 FORMAT(44H ENTER THE DESIRED SIZE OF LD, LF, X ARRAY/
+38H IN FORMAT(LESS OR EQUAL TO 1000) /
+5H NNNN)
905 READ(5,901) POINTS
901 FORMAT(14)
549. C IF (TEST.EQ.8) GO TO 200
548. C FIND BASE AND PEAK LOADS
549. C REWIND INL
550. C COUNTH=0
551. C BASE = 1000000
552. C PEAK = 0
553. C ITEN = 0
554. C COUNTH = COUNTH + 24
555. C READ(INL,899)END=15) (HRLOAD(I),I=1,24)
556. C DO 10 I = 1, 24
557. C IF (HRLOAD(I) .GT. PEAK) PEAK = HRLOAD(I)
558. C IF (HRLOAD(I) .LT. BASE) BASE = HRLOAD(I)
559. C ITEN = ITEN + HRLOAD(I)
10 CONTINUE
20 TO 6
15 COUNTH = COUNTH - 26
20 TEN = FLOAT(ITEM)/1000.0
100 FORMAT(20X,12I5/20X > 12I5)
101 IF(COUNTH .GT. 100) GO TO 56
102 WRITE(6,898) COUNTH
103 898 FORMAT(48H THE NUMBER OF HOURLY LOADS READ IS ,
104 +14, 39H THESE ARE NOT ENOUGH. PROGRAM ABENDS.)
STOP
56 CONTINUE

C SET THE MW INCREMENT TO DX = PEAK/POINTS
57 DX = FLOAT(PEAK)/FLOAT(POINTS)

C SET THE DX INCREMENT FOR THE NORMALIZED X-AXIS.
58 DXNORMAL = 1.0 / FLOAT(POINTS)
59 C FIND THE MAXIMUM ARRAY DIMENSIONS
60 MAXI = SYSCAP / DX + 1 + POINTS
61 WRITE(13,902) DX, POINTS, MAXI, BASE, PEAK, SYSCAP,ITEM,TEN
62 WRITE(6,902) DX, POINTS, MAXI, BASE, PEAK, SYSCAP,ITEM,TEN
63 902 FORMAT(34H THE MW INCREMENT (FOR X-AXIS) IS ,
64 + F7.3, 4H FOR, 14, 27H POINTS IN LD AND LF ARRAYS/
65 +42H THE MAXIMUM DIMENSIONS OF THE ELDC'S IS ,16/
66 +18H THE BASE LOAD IS , 15, 3H MW/
67 +18H THE PEAK LOAD IS , 15, 3H MW/
68 + 24H THE SYSTEM CAPACITY IS , F7.1, 4H MW /
69 + 28H THE TOTAL ENERGY DEMAND IS,110,8H MW OR, F12.3,4H GWH)
60 DEFINE AND FILL THE X-AXIS (MW) ARRAY
61 XI(1) = DX
62 DO 20 I =2, POINTS
63 XI(I) = XI(I-1) + DX

C CALCULATE THE LD AND LF ARRAYS. FIND MAXIMUM FREQUENCY
64 REWIND INL
65 MAXFRE = 0.0
66 30 READ(INL,899,EMU=55) (HRLOAD(J),J=1,24)
67 DO 50 I = 1, POINTS
68 DO 40 J = 1, 24
69 RLOAD = FLOAT(HRLOAD(J))
70 IF (RLOAD .GT. XI(D)) LD(I) = LD(I) + 1.0
71 XM1NUS = XI(I) - DX
72 IF (RLOAD .GT. XM1NUS) AND. RLOAD .LE. XI(I) LF(I) =
73 +LF(I) + 1.0
74 40 CONTINUE

C FIND MAXIMUM FREQUENCY AND THE CORRESPONDING LOAD
75 IF (MAXFRE .GE. LF(I)) GO TO 50
76 MAXFRE = LF(I)
77 MAXFRL = XI(I)
78 50 CONTINUE
79 GOTO 20
80 CONTINUE

C OPTIONAL PRINTING OF THE ORIGINAL LD AND LFC
81 KMAX = POINTS
82 LISTFL = 2
83 WRITE(6,907)
84 907 FORMAT(46H FOR ORIGINAL LD AND LFC PRINTING ENTER: YES /4H YYY)
85 READ(5,908) ORIGIN
86 IF ( ORIGIN.EQ.YES ) CALL PRIPLO(LIST,LISTFL,KMAX)

C NORMALIZE LD TO LD/COUNTH AND LF TO LF/COUNTH
87 DD 60 I = 1, POINTS
88 LD(I) = LD(I)/FLOAT(COUNTH)
89 NORMLD(I) = LD(I)
60  \texttt{LF(1) = LF(1)/FLOAT(COUNTH)}
61  \texttt{MAXFRE = MAXFRE/FLOAT(COUNTH)}
62  \texttt{C DEFINE NORMALIZED X-AXIS}
63  \texttt{DO 70 I = 1,3000}
64  \texttt{70 X(I) = FLOAT(I) / DXNORM}
65
66  \texttt{C OPTIONAL PRINTING OF THE NORMALIZED LDC AND LFC.}
67  \texttt{C}
68  \texttt{C WRITE(6*,909)}
69  \texttt{909 FORMAT(49H FOR NORMALIZED LDC AND LFC PRINTING ENTER: YES /}
70  \texttt{+4H YYY )}
71  \texttt{READ(5*,908) NORMAL}
72  \texttt{908 FORMAT(A3)}
73  \texttt{C IF (NORMAL.EQ.YES) CALL PRIPL0I(LIST,LISTFL,KMAX)}
74  \texttt{C}
75  \texttt{C PRINT THE NORMALIZED LDC IN FILE 14}
76  \texttt{IF (NORMAL.NE.NAL) GO TO 75}
77  \texttt{DO 72 I=1,POINTS,5}
78  \texttt{72 ISTOP = I+4}
79  \texttt{WRITE(14,897) (NORMLD(J),J=I,ISTOP)}
80
81  \texttt{C FALOAD = 0.0}
82  \texttt{DO 74 I = 1,POINTS}
83  \texttt{74 FALOAD = FALOAD + NORMLD(I)}
84  \texttt{FALOAD = FALOAD / FLOAT(POINTS)}
85  \texttt{WRITE(14,897) FALOAD}
86  \texttt{897 FORMAT(5F10.8)}
87  \texttt{75 CONTINUE}
88
89  \texttt{C NOTE THAT THE TEST = 1 OPTION CALCULATIONS HAVE BEEN PERFORMED}
90  \texttt{C AND THEY ENDED SUCCESSFULLY}
91  \texttt{C}
92  \texttt{C WRITE(6*,904) TEST}
93  \texttt{WRITE(13*,904) TEST}
94  \texttt{904 FORMAT(5H SUBROUTINE FRED SUCCESSFULLY COMPLETED FOR TEST = ,}
95  \texttt{+12, 8H OPTION.)}
96  \texttt{C}
97  \texttt{C RETURN}
98  \texttt{C}
99  \texttt{C READ THE SYSTEM PEAK AND NORMALIZED LDC FROM FILE 14}
100 \texttt{C}
101 \texttt{200 CONTINUE}
102 \texttt{C READ PEAK}
103 \texttt{READ(14*,911) PEAK}
104 \texttt{911 FORMAT(1X,15)}
105 \texttt{C DEFINE MAXIMUM ARRAY SIZE}
106 \texttt{DXNORM = 1.0 / FLOAT(POINTS)}
107 \texttt{DX = DXNORM * FLOAT(PEAK)}
108 \texttt{MAXI = SYSCAP / DX + 1 + POINTS}
109 \texttt{C READ INPUT LDC.}
110 \texttt{READ(14*,912) (LDI(I),I=1,POINTS)}
111 \texttt{912 FORMAT(8(I*,F9.6))}
112 \texttt{C DEFINE X & NORMLD ARRAYS}
113 \texttt{DO 210 I = 1,3000}
114 \texttt{X(I) = FLOAT(I) / DXNORM}
115 \texttt{IF (I.GT.POINTS) GO TO 210}
116 \texttt{NORMLD(I) = LDI(I)}
117 \texttt{210 CONTINUE}
118 \texttt{C}
119 \texttt{WRITE(6*,913) PEAK,DXNORM,POINTS,MAXI,SYSCAP}
120 \texttt{WRITE(13*,913) PEAK,DXNORM,POINTS,MAXI,SYSCAP}
121 \texttt{913 FORMAT(9H PEAK IS ,15, 14H MW, DXNORM IS ,F6.6/}
122 \texttt{+20H POINTS = ,15, 8H MAXI = ,15/20H SYSTEM CAPACITY IS ,}
123 \texttt{+F8.1, 3H MW)}
124 \texttt{KMAX = POINTS}
125 \texttt{LISTFL = 1}
126 \texttt{C CALL PRIPL0I(LIST,LISTFL,KMAX)
SUBROUTINE PRIPLU(OPTION, FLAG, MAXSAV)

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

C PRINT AND OPTIONALLY PLOT Y AND Z VS. X. THE ARRAYS

C ARE CONDENSED TO LESS THAN 100 POINTS FOR PLOTTING AND

C NORMALIZED TO 1.

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

C THE MEANING OF THE CONTROL VARIABLES ("OPTION" AND

C "FLAG") AS WELL AS THAT OF THE OTHER VARIABLES IS AS

C FOLLOWS:

C CONTROL VARIABLES

C NAME TYPE DIMENS. VALUE MEANING

C OPTION INT. - 1 PRINT NOTHING

C 2, 5, 8 PRINT EVERY 10TH POINT OF X, Y,

C (AND Z, IF FLAG >1)

C 3, 6, 9 PRINT X, Y,

C (AND Z IF FLAG >1)

C 4, 7, 10 PRINT AND PLOT

C X, Y, AND Z, ACCORDING TO "FLAG" CONTROL

C FLAG INT. - 1 PRINT AND PLOT

C Y VS. X

C 2 PRINT AND PLOT

C Y AND Z VS. X

C 3 PRINT AND PLOT

C Y AND Z VS. X

C OTHER KEY VARIABLES

C CONDX INT. - - - DX INCREMENT OF CONDENSED ARRAYS

C DIMENSIONS

C DOWN REAL - - LOWER LIMIT OF ARRAY

C IN PLOTTED LINE

C MAX INT. - <=2000 ACTUAL NUMBER OF X, Y

C Z ARRAY DATA POINTS

C PRINTA REAL 100 - PRINTING ARRAY

C UP REAL - - UPPER LIMIT OF ARRAY

C IN PLOTTED LINE

C X REAL <=2000 - X ARRAY, ABSISCA

C Y REAL <=2000 - Y ARRAY, ORDINATE

C Z REAL <=2000 - Z ARRAY ORDINATE

C IN SAME GRAPH

C PLOTTING CHARACTERS

C ARROW INT. - - - ==

C BLANK INT. - - - ==

C DASH INT. - - - ==

C PLUS INT. - - - ==

C STAR INT. - - - ==

C DECLARATION OF VARIABLES

COMMON /CURVES/ XSAV(3000), YSAVE(3000), ZSAVE(3000), NORDMLD(1000)

COMMON /ARRAYS/ CHECK(100, 10, 2), ENERGY(100, 3), HOURC(8), HRLOAD(24),

*GUMLIN(100, 8), PLANTG(100, 8), PLANT(100, 2), TESTAR(10), EONE(100, 2),

*ELAST(100, 2)

COMMON /VAR1AB/ COUNTH, COUNTP, DX, DXNORM, INL, MAX1, MULTI, NEXIST,

*OUT, PEAK, POINTS, SYSCAP, TEN, PLHRT, TLC, TLPC, TLE, TLPL, TDE

INTEGER COUNTH, COUNTP, HRLOAD, INL, OUT, PEAK, POINTS, TESTAR
REAL NUMNLD
DOUBLE PRECISION CUMLT, PLANTC, HOURC

INTEGER ARROW, BLANK, CONDX, DASH, FLAG,
+OPTION, PLUS, PRINA(100), REMAIN, STAR

REAL X(3000), Y(3000), Z(3000)

C INITIALIZATION
DATA ARROW/4H1, BLANK/4H /, DASH/4H - /,
+ PLUS/4H + /, STAR/4H *

C SAVE THE VALUES OF THE ORIGINAL VARIABLES.
MAX = MAXSAV
DO 5 I = 1, MAX
X(I) = XSAVE(I)
Y(I) = YSAVE(I)
5 Z(I) = ZSAVE(I)
IF (FLAG.NE.3) GO TO 7
DO 6 I = 1, MAX
11 = 1000 + I
Y(I) = YSAVE(11)
5 Z(I) = ZSAVE(I1)
6 Z(I1) = ZSAVE(I11)
7 CONTINUE
IF (FLAG .NE. 1 .AND. FLAG .NE. 2 .AND. FLAG .NE. 3) FLAG = 0

C ARRAY PRINTING
GO TO (10, 20, 30, 30, 20, 30, 30, 20, 30, 30, 20)
OPTION
10 RETURN

C PRINT EVERY 10TH POINT OF X, Y AND Z ARRAYS
C PRINT HEADINGS AND FIRST THREE LINES
MAXPRI = 90
IF (MAX .LT. MAXPRI1) MAXPRI1 = MAX
WRITE(13, 900)
WRITE(13, 901) X(I), X(I), I = 1, MAXPRI1, 10
WRITE(13, 902) Y(I), Y(I), I = 1, MAXPRI1, 10
IF (FLAG .NE. 1) WRITE(13, 903) Z(I), Z(I), I = 1, MAXPRI1, 10
C
C PRINT THE REMAINING LINES
900 FORMAT(1H ,T14, ZH00, 10X, ZH10, 10X, ZH20, 10X, ZH30,
+10X, ZH40, 10X, ZH50, 10X, ZH60, 10X, ZH70, 10X, ZH80,
+10X, ZH90//)
901 FORMAT(1H , T6, 3H X , 10F12.6)
902 FORMAT(1H , T6, 3H Y , 10F12.6)
903 FORMAT(1H , T6, 3H Z , 10F12.6)
910 IF (MAXPRI1 .LT. 100) GO TO 26
24 DU 24 I = 100, MAX, 100
25 MAXPRI1 = I = 90
26 IF (MAXPRI .GT. MAX) MAXPRI = MAX
WRITE(13, 904) I, (X(J), J = I, MAXPRI1, 10)
904 FORMAT(1H , T6, 3H X , 10F12.6)
905 WRITE(13, 902) Y(J), J = I, MAXPRI1, 10
IF (FLAG .NE. 1) WRITE(13, 903) Z(J), J = I, MAXPRI1, 10
23 CONTINUE
24 26 RETURN
C
C PRINT EVERY POINT OF X, Y, AND Z ARRAYS
C PRINT HEADINGS AND FIRST THREE LINES
C
MAXPRI = 9
IF (MAX .LE. MAXPRI) GO TO 38
WRITE(13, 905)
905 FORMAT(1H , T15, 1H0, 11X, 1H1, 11X, 1H2, 11X, 1H3, 11X,
+1H4, 11X, 1H5, 11X, 1H6, 11X, 1H7, 11X, 1H8, 11X, 1H9//)
906 WRITE(13, 906) X(I), I = 1, 9
907 WRITE(13, 907) Y(I), I = 1, 9
908 WRITE(13, 908) Z(I), I = 1, 9
909 IF (FLAG .NE. 1) WRITE(13, 909) (Z(I), I = 1, 9)
910 WRITE(13, 906) X(I), I = 1, 9
911 WRITE(13, 907) Y(I), I = 1, 9
912 WRITE(13, 908) Z(I), I = 1, 9
913 WRITE(13, 909) (Z(I), I = 1, 9)
914 WRITE(13, 906) X(I), I = 1, 9
915 WRITE(13, 907) Y(I), I = 1, 9
916 WRITE(13, 908) Z(I), I = 1, 9
917 WRITE(13, 909) (Z(I), I = 1, 9)
918 WRITE(13, 906) X(I), I = 1, 9
919 WRITE(13, 907) Y(I), I = 1, 9
920 WRITE(13, 908) Z(I), I = 1, 9
921 WRITE(13, 909) (Z(I), I = 1, 9)
922 WRITE(13, 906) X(I), I = 1, 9
923 WRITE(13, 907) Y(I), I = 1, 9
924 WRITE(13, 908) Z(I), I = 1, 9
925 WRITE(13, 909) (Z(I), I = 1, 9)
926 WRITE(13, 906) X(I), I = 1, 9
927 WRITE(13, 907) Y(I), I = 1, 9
928 WRITE(13, 908) Z(I), I = 1, 9
929 WRITE(13, 909) (Z(I), I = 1, 9)
C
C PRINT EVERY POINT OF X, Y, AND Z ARRAYS
C PRINT HEADINGS AND FIRST THREE LINES
C
30 MAXPRI = 9
31 IF (MAX .LE. MAXPRI) GO TO 38
32 WRITE(13, 905)
33 905 FORMAT(1H , T15, 1H0, 11X, 1H1, 11X, 1H2, 11X, 1H3, 11X,
+1H4, 11X, 1H5, 11X, 1H6, 11X, 1H7, 11X, 1H8, 11X, 1H9//)
34 906 WRITE(13, 906) X(I), I = 1, 9
35 907 WRITE(13, 907) Y(I), I = 1, 9
36 908 WRITE(13, 908) Z(I), I = 1, 9
37 909 IF (FLAG .NE. 1) WRITE(13, 909) (Z(I), I = 1, 9)
38 906 WRITE(13, 906) X(I), I = 1, 9
39 907 WRITE(13, 907) Y(I), I = 1, 9
40 908 WRITE(13, 908) Z(I), I = 1, 9
41 909 IF (FLAG .NE. 1) WRITE(13, 909) (Z(I), I = 1, 9)
C PRINT REMAINING LINES
DO 35 I = 10, MAX, 10
MAXPRI = 1 + 9
IF (MAX .LT. MAXPRI) MAXPRI = MAX
IF (ABS(Y(I)) .LE. 0.00001) GO TO 37
WRITE(13, 904) I, (X(J), J = I, MAXPRI)
WRITE(13, 902) (Y(J), J = I, MAXPRI)
IF (FLAG .NE. 1) WRITE(13, 903) (Z(J), J = I, MAXPRI)
35 CONTINUE
GO TO 39
37 MAX = 1
GO TO 39
38 WRITE(6, 909) MAX
909 FORMAT(1H , 1011H* ) , 6H ERROR, 10 (1 H* )/
*14, 20H THIS IS TOO SMALL. */
STOP
39 IF (FLAG .NE. 0) GO TO 40
C PLOT IF OPTION = 4 OR 7 OR 10
40 IF (OPTION .EQ. 4 OR OPTION .EQ. 7 OR OPTION .EQ. 10) GO TO 42
RETURN
42 IF (MAX .LE. 100) GO TO 95
C CONDENSE X, Y, AND Z ARRAYS TO 100 POINTS
IF (OPTION = 4 OR OPTION = 7 OR OPTION = 10) CONDUX = MAX/100 + 1
K = 1
DO 60 1 = CONDUX, MAX, CONDUX
X(K) = X(1)
IF (FLAG .NE. 3) Y(K) = Y(1)
YSUM = 0.0
ZSUM = 0.0
JJ = 1 - CONDUX + 1
DO 50 J = JJ, 1
YSUM = YSUM + Y(J)
ZSUM = ZSUM + Z(J)
50 Y(K) = YSUM
Z(K) = ZSUM
60 K = K + 1
REMAIN = MOD(MAX, CONDUX)
IF (REMAIN) 70, 70, 80
70 MAX = K - 1
GO TO 90
80 X(K) = X(MAX)
81 IF (FLAG .NE. 3) Y(K) = Y(MAX)
82 YSUM = 0.0
ZSUM = 0.0
83 JJ = MAX - REMAIN + 1
84 DO 85 J = JJ, MAX
YSUM = YSUM + Y(J)
ZSUM = ZSUM + Z(J)
85 Z(K) = ZSUM
86 IF (FLAG .NE. 3) Y(K) = YSUM
90 Z(K) = ZSUM
C FIND THE MAXIMA OF X, Y AND Z ARRAYS AND NORMALIZE THEM TO 1.

XMAX = 0.0
YMAX = 0.0
ZMAX = 0.0

DO 100 I = 1, MAX
IF (ABS(X(I)) .GT. XMAX) XMAX = X(I)
IF (ABS(Y(I)) .GT. YMAX) YMAX = Y(I)
IF (ABS(Z(I)) .GT. ZMAX) ZMAX = Z(I)
IF (ZMAX .EQ. ABS(Z(I))) ZMAX = X(I)

100 CONTINUE

IF (FLAG .EQ. 1) GO TO 102
IF (XZMAX .LT. 10.0) GO TO 102
WRITE(6,918) ZMAX, XZMAX
WRITE(13,918) ZMAX, XZMAX
918 FORMAT(10H ZMAX IS , F10.3, 10X, 3H AT , F7.1, 3H MU)

102 CONTINUE
IF (XMAX .GT. 0.000000001 .AND. XMAX .LT. 0.000000001) XMAX = 1.0
IF (YMAX .GT. 0.000000001 .AND. YMAX .LT. 0.000000001) YMAX = 1.0
IF (ZMAX .GT. 0.000000001 .AND. ZMAX .LT. 0.000000001) ZMAX = 1.0

ZERO = 0.0
WRITE(13,910) ZERO, (XI(J), J = 1, MAX, 10)
910 FORMAT(1H Z , T10, 11(F6.2,2X))
DO 105 J = 1, MAX
IF (MOD(J, 10)) 103, 103, 104
103 PRINA(J) = ARROW
104 PRINA(J) = DASH
105 CONTINUE
WRITE(13,913) (PRINA(J), J = 1, MAX)
WRITE(13,914) (PRINA(J), J = 1, MAX)

110 CONTINUE

DO 120 J = 1, MAX
120 IF (I .GT. 10 .AND. I .LT. 100) GO TO 110
JMAX = J
111 = 1 - I
UP = FLOAT(11) * 0.01 + 0.005
DOWN = UP - 0.01
DO 116 J = 1, MAX
IF (I(J) .GT. DOWN .AND. Y(J) .LE. UP) GO TO 114
PRINA(J) = BLANK
114 PRINA(J) = STAR
115 PRINA(J) = PLUS
116 CONTINUE

117 WRITE(13,911) (PKMAX(J), J = 1, JMAX)
911 FORMAT(1H , T10, 14, 1X, 1H1, 100A1)
912 FORMAT(1H , 115, 1H1, 100A1)
120 CONTINUE
C PRINT BOTTOM LINE

C DO 130 J = 1, MAX

C IF (MOD(J, 10)) 122, 122, 124

C 122 PRINA(J) = ARROW

C GO TO 130

C 124 PRINA(J) = DASH

C 130 CONTINUE

C WRITE(13, 913) (PRINA(J), J = 1, MAX)

C 913 FORMAT(1H, TI5, 1H, 100A1)

C WRITE(13, 914) ZERU, (X(J), J = 10, MAX, 10)

C 914 FORMAT(1H, TI4, 11(F4.2, 6X))

C WRITE LEGEND

C IF (FLAG .NE. 3) GO TO 140

C WRITE(13, 915) FLAG, OPTION

C 915 FORMAT(1H, 6H FLAG=* 12H LIST OPTION= *, 12/

C +40H COMPARATIVE PLOTTING OF LOAD FREQUENCY ARRAYS /)

C RETURN

C 140 WRITE(13, 916) FLAG, OPTION

C 916 FORMAT(1H, 6H FLAG=* 12H LIST OPTION= , 12/

C +22H LOAD DURATION CURVE /)

C IF (FLAG .EQ. 2) GO TO 150

C RETURN

C PLOT Z VS. X

C CONTINUE

C DO 160 I=1,101

C JM=I

C II=101-I

C UP=FLOAT(I1)*0.01 + 0.005

C DOWN=UP-0.01

C DD 156 J=1,MAX

C IF (Z(J) .GT. DOWN .AND. Z(J) .LE. UP) GO TO 154

C PRINA(J) = BLANK

C GO TO 156

C 154 PRINA(J) = PLUS

C JM = J

C 156 CONTINUE

C III=I=1

C IF (MOD(III, 5)) 157, 157, 158

C 157 WRITE(13, 911) II, (PRINA(J), J = 1, JMAX)

C GO TO 160

C 158 WRITE(13, 912) (PRINA(J), J = 1, JMAX)

C CONTINUE

C PRINT BOTTOM LINE

C DO 170 J = 1, MAX

C IF (MOD(J, 10)) 162, 162, 164

C 162 PRINA(J) = ARROW

C GO TO 170

C 164 PRINA(J) = DASH

C 170 CONTINUE

C WRITE(13, 913) (PRINA(J), J = 1, MAX)

C WRITE(13, 914) ZERU, (X(J), J = 10, MAX, 10)

C +23H LOAD FREQUENCY CURVE /)

C RETURN
END

SUBROUTINE CUALC(FLAG)

C     CALCULATES THE FIRST EIGHT CUMULANTS OF THE SYSTEM LOADS
C AND GENERATING UNITS ACCORDING TO THE FLAG CONTROL OPTION
C     FLAG = 1  . CALCULATE PLANT CUMULANTS
C     FLAG = 2  . CALCULATE LOAD CUMULANTS FROM HOURLY LOADS
C             (HRLOAD)
C     FLAG = 3  . CALCULATE LOAD CUMULANTS FROM ORIGINAL LOAD
C             DURATION CURVE (EL).
C     FLAG = 4  . CALCULATE LOAD CUMULANTS FROM EQUIVALENT LOAD
C             DURATION CURVE (LD).
C
C IDENTIFICATION OF ALL LOCAL VARIABLES

C NAME    TYPE   SIZE     VALUE   MEANING

C FLAG    INT.-1   1  CALCULATE PLANT CUMULANTS
C         2  CALCULATE LOAD CUMULANTS FROM
C HOURLY LOADS
C         3  CALCULATE LOAD CUMULANTS FROM
C EL CURVE
C         4  CALCULATE LOAD CUMULANTS FROM
C ORIGINAL LD CURVE

C C REAL    -    PLANT CAPACITY
C Q REAL    -    PLANT FORCED OUTAGE RATE
C M REAL    8    LOAD MOMENTS

C DECLARATION OF COMMON VARIABLES

COMMON /CURVES/ X(3000), EL(3000), ELF(3000), NORMLD(1000)
COMMON /ARRAYS/ CHECK(100,10,2), ENERGY(100,3), HOURC(6), HRLOAD(24),
*CUMLIN(100,6), PLANTC(100,8), PLANT(100,2), TESTAR(100,2),
*ELAST(100,2)
COMMON /VARIABLES/ COUNTH, COUNTP, DX, DXNORM, INL, MAXI, MULTI, NEXIST,
*OUT, PEAK, POINTS, SYCAP, TEN, PLT, HAT, TCE, TLP, TLE, TPL, TDE
INTEGER COUNTH, COUNTP, DX, DXNORM, INL, MAXI, MULTI, NEXIST,
INTEGER FLAG, PRICUM, YES
REAL NORMLD
INTEGER COUNTH, COUNTP, HRLOAD, INL, OUT, PEAK, POINTS, TESTAR
DOUBLE PRECISION CUMLIN, PLANTC, HOURC

C DOUBLE PRECISION C, Q, HOURC, XL, P, DIFFL, FORM, SQM, A(8)

DATA YES/3/NO/YES/

C ACT ACCORDING TO FLAG CONTROL

GO TO (100, 200, 300, 300), FLAG

100 CONTINUE

1097 CALL SCLUK1

1098 DO 110 I = 1, COUNTP

1099 C = PLANT(I, 1)

1100 Q = PLANT(I, 2)

1102 PLANTC(I, 1) = C * Q

1103 PLANTC(I, 2) = C*C*C*Q*(1.0 - Q)

1104 PLANTC(I, 3) = C*C*C*Q*(1.0 - (3.0 - 2*Q) *Q)

1105 PLANTC(I, 4) = C*C*C*Q*(1.0 - (7.0 - (12.0 - 6.0*Q) *Q) *Q)

1106 PLANTC(I, 5) = C*C*C*C*C*Q*(1.0 + (-15.0 +

1107 + (50.0 + (-60.0 + 24.0 * Q) *Q) *Q) *Q)

1108 PLANTC(I, 6) = C*C*C*C*C*C*Q*(1.0 + (-31.0 +

1109 + (180.0 + (-390.0 + (326.0 - 120.0 * Q) *Q) *Q) *Q)

1109 PLANTC(I, 7) = C*C*C*C*C*C*C*Q*(1.0 + (-63.0 +

1111 + (602.0 + (-210.0 + (336.0 + (-252.0 + 720.0 * Q) *Q) *Q) *Q) *Q)

1111 PLANTC(I, 8) = C*C*C*C*C*C*C*C*Q*(1.0 + (-127.0 +

1112 + (1932.0 + (-1026.0 + (2520.0 + (-3192.0 +

1115 + (20160.0 - 5040.0 * Q) *Q) *Q) *Q) *Q) *Q)

1110 CONTINUE

1117 PLT = RCLD0K111.0

1118 C

1119 C
C PRINT PLANT CUMULANTS
WRITE(6,905)
905 FORMAT(40H SHOULD THE PLANT CUMULANTS BE PRINTED? , +16H ENTER YES OR NO /4H YYY )
READ(5,906) PRICUM
906 FORMAT(4H)
IF(PRICUM.NE.YES) GO TO 130
WRITE(13,900)
900 FORMAT(1H // 20X, 17H PLANT CUMULANTS / +42H PLANT CAP. FOR CUMUL. 1 CUMUL. 2 , +36H CUMUL. 3 CUMUL. 4 CUMUL. 5 , +36H CUMUL. 6 CUMUL. 7 CUMUL. 8 / +24H NUM. (M M )//)
DO 120 I = 1, COUNTP
210 A(I) = 0.0
C CALCULATE THE MEAN
REWIND INL
COUNTH=0
212 COUNTM = COUNTH+24
READ(INL,899,END=216) (HRLOAD(I),I=1,24)
899 FORMAT(20X,12F9.4,12F9.4)
DO 214 I = 1,24
214 A(I) = A(I) + DFLOAT(HRLOAD(I))
GO TO 212
216 COUNTH = COUNTH - 24
A(1) = A(1)/DFLOAT(COUNTM)
C INCLUDE LOAD UNCERTAINTY IN HOURLY CUMULANTS.
THOURS = COUNTH
CMEAN = A(1)
CALL UNCERTIMEAN.THOURS.VAR)
C CALCULATE CENTRAL MOMENTS M2-M8
REWIND INL
217 READ(INL,899,END=225) (HRLOAD(I),I=1,24)
DO 220 I = 1,24
220 A(I) = A(I)/DFLOAT(COUNTM)
225 CONTINUE
DD 230 J=2, 8
230 A(J) = A(J)/DFLOAT(COUNTM)
C PRINT CENTRAL LOAD MOMENTS
WRITE(13,907)
907 FORMAT(32H THE CENTRAL LOAD MOMENTS ARE ,/ +16H ENTER YES OR NO /4H YYY )
WRITE(13,908)
C CALCULATE THE CUMULANTS FROM CENTRAL MOMENTS
SQM= A(2)*A(2)
HOURC(1) = A(1)
HOURC(2) = A(2) + VAR
HOURC(3) = A(3)
HOURC(4) = A(4) - 3.0*SQM
HOURC(5) = A(5) - 10.0*A(3)*A(2)
HOURC(6) = A(6) - 15.0*A(4)*A(2) - 10.0*A(3)*A(3)
HOURC(7) = A(7) - 21.0*A(5)*A(2) - 35.0*A(4)*A(3)
HOURC(8) = A(8) - 28.0*A(6)*A(2) - 56.0*A(5)*A(3)
C MTR = RELOK(1,0)
GO TO 245
C CALCULATE CUMULANTS FROM ELDG MOMENTS
C 240 SQM = A(1) * A(1)
C FORM = SQM * SUM
C HURC(1) = A(1)
C HURC(2) = A(2) - SUM
C HURC(3) = A(3) - 3 * A(2) * A(1) + 2 * A(1) * SQM
C HURC(4) = A(4) - 4 * A(3) * A(1) - 3 * A(2) * A(1) * SQM
C + 12 * A(2) * SQM - 6 * FORM
C 244 HURC(5) = A(5) - 5 * A(4) * A(1) - 10 * A(3) * A(2) +
20 * A(3) * SQM + 30 * A(2) * A(2) * A(1)
C + - 60 * A(2) * SQM + 2 * FORM * A(1)
C HURC(6) = A(6) - 6 * A(5) * A(1) - 15 * A(4) * A(2) +
30 * A(4) * SQM - 10 * A(3) * A(3) +
C + 120 * A(3) * A(2) * A(1) - 120 * A(3) * SQM * A(1) +
C + 30 * A(2) * A(2) * A(2) - 270 * A(2) * A(2) * SQM +
C + 360 * A(2) * FORM - 120 * FORM * SQM
C HURC(7) = A(7) - 7 * A(6) * A(1)
C + - 21 * A(5) * A(2) + 42 * A(5) * SQM
C + - 35 * A(4) * A(3) + 210 * A(4) * A(2) * A(1)
C + - 350 * A(4) * SQM * A(1) + 140 * A(3) * A(3) * A(1) +
C + 210 * A(3) * A(2) * A(2) - 1260 * A(3) * A(2) * SQM +
C + 840 * A(3) * FORM - 630 * A(3) * A(2) * A(1) +
C + 2520 * A(2) * A(2) * SQM * A(1) - 2520 * A(2) * FORM * A(1) +
C + 720 * FORM * SQM * A(1)
C HURC(8) = A(8) - 8 * A(7) * A(1)
C + - 28 * A(6) * A(2) + 56 * A(6) * SQM
C + - 35 * A(5) * A(3) + 330 * A(5) * A(2) * A(1)
C + - 330 * A(5) * SQM * A(1) - 35 * A(4) * A(4) +
C + 560 * A(4) * A(3) * A(1) +
C + 420 * A(4) * A(2) * A(2)
C + - 2520 * A(4) * A(2) * SQM +
C + 168 * A(4) * FORM +
C + 560 * A(3) * A(3) * A(1)
C + - 160 * A(3) * A(3) * SQM
C + - 5040 * A(3) * A(2) * A(2) * A(1) +
C + 13440 * A(3) * A(2) * SQM * A(1)
C + - 6720 * A(3) * FORM * A(1)
C + - 630 * A(2) * A(2) * A(2) * A(2) +
C + 10080 * A(2) * A(2) * A(2) * SQM
C + - 2520 * A(2) * A(2) * FORM +
C + 20160 * A(2) * FORM * SQM
C + - 5040 * FORM * FORM
C
245 CONTINUE
C
V = 1. / DSQRT(HURC(2))
B = HURC(3) * V * V * V
B1 = B * B
S1 = B * (B2 + 3.1) / (5 * B2 - 5 * B1 - 9)
C WRITE(6,911) B1,b2, S1,S2
C WRITE(13,911) B1,b2,S1,S2
C 911 FORMAT(25H CHECK COEFFICIENTS ARE:,F12.6)
C
IF (FLAG .EQ. 2) WRITE(13,902)
IF (FLAG .EQ. 3) WRITE(13,903) DFX
IF (FLAG .EQ. 4) WRITE(13,908) DFX
902 FORMAT(10H LOAD CUMULANTS FROM HOURLY LOADS/)
903 FORMAT(10H LOAD CUMULANTS FROM ORIGINAL LD CURVE/)
904 FORMAT(10H DIFFERENTIATION INCREMENT IS: ,F10.6/)
905 FORMAT(10H DIFFERENTIATION INCREMENT IS: ,F10.6/)
+42H NORMALIZED DIFFERENTIATION INCREMENT IS: ,F10.6/}
DO 250 J = 1,B
C CALCULATE THE LOAD MOMENTS FROM THE ORIGINAL OR EQUIVALENT LOAD CURVE.
C
300 CONTINUE
C INQUIRE ABOUT THE MULTI IF IT IS NOT SPECIFIED BEFORE.
IF (TESTAR(2).EQ.2 .OK. TESTAR(6).EQ.6) GO TO 310
TESTAR(6) = 6
C PROMPT FOR THE DIFFERENTIATION INCREMENT
WRITE(6, 909)
909 FORMAT(14H ENTER THE NUMBER (BETWEEN 1 AND 10) BY WHICH /
$$45H THE MINCREMENT WILL BE MULTIPLIED TO GIVE/ 
$$43H THE DIFFERENTIATION INCREMENT.(12 FORMAT/3H NN)
110 FORMAT(12)
READ(5*, 110) MULTI
MAX = MAX - MULTI
1277. IF (FLAG.EQ.3) MAX = POINTS - MULTI
1278. DO 320 J = 1, 6
1279. A(J) = 0.0
1280. C DEFINE THE DIFFERENTIATION INCREMENT
1281. DFX = DXNORM * MULTI
1282. C CALCULATE MOMENTS FOR THE FIRST INTERVAL.
1283. XL = DFX / 2.0
1284. DIFFL = 1.0 - EL(MULTI)
1285. IF (FLAG.EQ.3) DIFFL = 1.0 - NORMLD(MULTI)
1286. XLOAD = 1.0
1287. DO 325 J = 1, 8
1288. XLOAD = XLOAD * XL
1289. DO 325 A(J) = A(J) * XLOAD * DIFFL
1290. C CALCULATE THE MOMENTS FOR THE REMAINING INTERVALS.
1291. DO 340 1 = MULTI * MAX * MULTI
1292. NEXT = 1 + MULTI
1293. XL = (XI(1) * XI(NEXT))/2.0
1294. DIFFL = EL(1) - EL(NEXT)
1295. IF (FLAG.EQ.3) DIFFL = NORMLD(1) - NORMLD(NEXT)
1296. XLOAD = 1.0
1297. DO 330 J = 1, 8
1298. XLOAD = XLOAD * XL
1299. DO 330 A(J) = A(J) + XLOAD * DIFFL
1300. CONTINUE
1301. C CALCULATE THE MOMENTS FOR THE LAST INTERVAL.
1302. IREST = MOD(MAX, MULTI)
1303. 1 = MAX - MULTI - IREST
1304. XL = (XI(1) + XI(MAX))/2.0
1305. DIFFL = EL(1) - EL(MAX)
1306. IF (FLAG.EQ.3) DIFFL = NORMLD(1) - NORMLD(MAX)
1307. XLOAD = 1.0
1308. DO 345 J = 1, 8
1309. XLOAD = XLOAD * XL
1310. DO 345 A(J) = A(J) + XLOAD * DIFFL
1311. C ADJUST FOR SYSTEM PEAK
1312. P = 1.0
1313. DO 350 J = 1, 8
1314. 350 A(J) = A(J) * P
1315. C
go to 240
1316. C END
1317. C SUBROUTINE CULON(TEST*, LIST)
1318. C CONVOLVE THE PLANTS WITH THE HOURLY LOADS BY
1319. C ADDING THE CORRESPONDING CUMULANTS. RESTORE
1320. C THE EQUIVALENT LOAD FREQUENCY FUNCTIONS AND
INTEGRATE THEM TO PRODUCE THE LOAD DURATION CURVES.

KEY LOCAL VARIABLES

DECLARATION OF COMMON VARIABLES

COMMON /CURVES/ X(3000),EL(3000),ELF(3000),NORMLD(1000)
COMMON /VARS/ CHECK(100,10,2),ENERGY(100,3),MOURC(8),HRLOAD(24),
+CUMLIN(100,8),PLANTC(100,8),TESTAR(10),EDONE(100,2),
+ELAST(100,2)
COMMON /VAR/ COUNTH,COUNTP,DX,DXNORM,INL,MAX8.MULTI,NEIXST,
+OUT,PEAK,POINTER,SYSCAP,PLT,HRT,TCL,TLPC,TLPL,TLE
COMMON /TABLE/ TY(120),TINT(120)
DOUBLE PRECISION TY,TINT

C

INTEGER COUNTH,COUNTP,HRLOAD,INL,OUT,PEAK,POINTS,TESTAR
REAL NORMLD
DOUBLE PRECISION CUMLIN,PLANTC,MOURC

DECLARATION OF LOCAL VARIABLES

INTEGER CUNUM,TEST,FLAG,SYSYES,YES
REAL LULP
DOUBLE PRECISION SYSTC(B),COEFF(8),SIGINV,ZN(B),ZEL,ZELF,FRE,
+ZVAL,ZVAL1,ZVAL2,ZEL1,ZEL2

DATA YES/3HYES/
PROMPT FOR THE NUMBER OF TERMS TO BE USED IN THE

GRAM-CHARLIER SERIES EXPANSION

WRITE(6,900)
900 FORMAT(3H WRITE THE DESIRED NUMBER OF TERMS
+33H TO BE USED IN THE GRAM CHARLIER /
+40H SERIES EXPANSION (2 - 10. 12 FORMAT) /
+3H NN)
901 FORMAT(12)
WRITE(6,905) CUNUM
905 FORMAT(1H ,12,20H CUMULANTS ARE USED /
+4H YYYY)
READ(5,907) SYSYES
907 FORMAT(A3)
906 FORMAT(4H FOR A LIST OF SYSTEM CUMULANTS ENTER YES:
+4H YYYY)
READ(5,907) SYSYES
908 FORMAT(12)
WRITE(6,905) CUNUM
905 FORMAT(1H ,12,20H CUMULANTS ARE USED /
+4H YYYY)
READ(5,907) SYSYES

CHECK FOR LIST OPTION

IF (LIST .LT. 8) GO TO 100
WRITE(6,902)
902 FORMAT(36H WRITE THE PLANT NUMBER AFTER WHOSE
+40H CONVOLUTION THE ELOC SHOULD BE PRINTED /
+43H FOR A LIST OF SYSTEM CUMULANTS ENTER YES:
+4H YYYY)
READ(5,901) ITH
100 ITH = 0
110 CONTINUE

C
C CALCULATE THE PLANT CUMULANTS
FLAG = 1
CALL CUCAL(FLAG)
C CALCULATE THE HOURLY LOAD CUMULANTS
IF (TEST.EQ.0 .OR. TEST.EQ.7) FLAG = 2
IF (TEST.EQ.6) FLAG = 3
IF (TEST.EQ.4) GO TO 123
C CALL CUCAL(FLAG)
C INITIALIZE THE SYSTEM CUMULANTS
DO 120 J = 1, CNUM
120 SYSTEC(J) = HOURC(J)
WRITE(13,*) SYSTEC
GO TO 127
123 CONTINUE
DO 124 J = 1, CNUM
124 SYSTEC(J) = CUMUL(NEXIST, J)
127 CONTINUE
SUMP = 0.0
ISTART = 1
IF (TEST.EQ.4 ) ISTART = NEXIST + 1
C START PLANT CUMULANT CONVOLUTION
M15 = POINTS*1.5
115 = M15/100
TCE = 0.0
TLPC = 0.0
DO 200 I = 1, POINTS
200 SYSTEC(I) = CUMUL(15, I)
201 TCE = TCE + CUMUL(I)
RETURN
C CALCULATE THE ENERGY GENERATED BY THE I-TH PLANT
IF (TEST.NE.7 ) GO TO 201
P = 1.0 - PLANT(I, 1)
Y = PLANT(I, 1)
SUMP = SUMP + Y
Y1 = SUMP - Y
Y2 = SUMP
COEFF(1) = SYSTEC(I)
COEFF(2) = SYSTEC(2)
SIGMA = DSQRT(SYSTEC(2))
DO 140 J = 3, CNUM
140 COEFF(J) = SYSTEC(J) / SIGMA**J
ZEL = (Y1 - COEFF(I)) / SIGMA
ZEL2 = (Y2 - COEFF(I)) / SIGMA
CALL NDF(ZEL, COEFF, 2, CNUM, ZVAL1, INFLAG, I)
CALL NDF(ZEL2, COEFF, 2, CNUM, ZVAL2, INFLAG, I)
ENERGY(I, 2) = P*Y*ZVAL1*ZVAL2/2.0*COUNTH
TIME1 = RCLOK(1.0)
TCE = TCE + TIME1
129 CONTINUE
C CALL SCLOK1
C DO 130 J = 1, CNUM
130 SYSTEC(J) = SYSTEC(J) + PLANT(I, J)
TIME3 = RCLOK(1.0)
TCE = TCE + TIME3
TLPC = TLPC + TIME3
C COEFF(1) = SYSTEC(1)
COEFF(2) = SYSTEC(2)
SIGMA = DSQRT(SYSTEC(2))
SIGMA = 1.0 / SIGMA
DO 140 J = 3, CNUM
140 COEFF(J) = SYSTEC(J) * SIGMA**J
1470. \[ B_1 = \text{COEFF}(3) \times \text{COEFF}(3) \]
1471. \[ B_2 = \text{COEFF}(4) + 3.0 \]
1472. \[ S_1 = \text{COEFF}(3) \times (B_2 + 3.0) / (B_2 + 6.0) \times (B_1 - 9.0) \]
1473. \[ S_2 = B_1 \times (B_2 + 3.0) / (B_2 + 3.0 - B_1) / (B_2 - B_1 + 3.0) \]
1474. IF (J.EQ.1.OR.J.EQ.2) OR (J.EQ.COUNTP) WRITE(6, 921) 1.0, B1, B2, S1, S2
1475. WRITE(13, 921) 1.0, B1, B2, S1, S2
1476. 921 FORMAT(13, 4F12.6)
1477. IF (SYSYES.EQ.NE.YES) GO TO 145
1478. WRITE(23, 896) I, (COEFF(I), I=1,CNUM)
1479. 896 FORMAT(4H PLANT=,2I6, 1H COEFFS=,10(2X,DE10.4))
1480. 145 CONTINUE
1481. IF (LIST.LT.5) GO TO 153
1482. IF (LIST.EQ.7) OR (LIST.EQ.COUNTP) GO TO 153
1483. 153 CONTINUE
1484. C RESTORE THE EQUIVALENT FREQUENCY DISTRIBUTION.*
1485. SUMELF = 0.0
1486. DO 150 IX = MAX1
1487. ZELF = ((X(IX) - 0.0) / 2.0) * DFLAT(PEAK) - COEFF(1) * SIGINV
1488. C
1489. INFLAG = 0
1490. CALL NDF(ZELF, COEFF, ZN, CNUM, FRE, INFLAG, 1)
1491. IF (FRE.LT.0.0) FRE = 0.0
1492. ELF(IX) = FRE * DNUMR
1493. 150 SUMELF = SUMELF + ELF(IX)
1494. C
1495. WRITE(6, 921) SUMELF
1496. C
1497. 150 CONTINUE
1498. 153 CONTINUE
1499. C RESTORE THE EQUIVALENT LOAD DURATION CURVE.*
1500. INFLAG = 1
1501. C
1502. DO 155 IX = MAX1
1503. ZEL = (X(IX) - 0.0) / DFLAT(PEAK) - COEFF(1) * SIGINV
1504. C
1505. CALL NDF(ZEL, COEFF, ZN, CNUM, ZVALUE, INFLAG, 1)
1506. 155 ZVALUE = ZVALUE
1507. C
1508. 150 CONTINUE
1509. C
1510. C FILL THE CHECK ARRAYS
1511. DO 160 J = 1, 10
1512. XCAP = SYSCAP / 10.0 * FLOAT(J) / FLOAT(PEAK)
1513. 160 CHECK(1, J, 2) = PLULP(XCAP)
1514. C
1515. IF (1.EQ.1) OR (J.EQ.1) GO TO 180
1516. DO 170 J = 1, 100
1517. JJ = J + 15
1518. IF (J.EQ.1) EON(1, J, 2) = EL(JJ)
1519. IF (J.EQ.COUNTP) ELAST(1, J, 2) = EL(JJ)
1520. 170 CONTINUE
1521. 180 CONTINUE
1522. C
1523. LISTFL = 2
1524. IF (LIST.LT.5) GO TO 200
1525. IF (LIST.LT.7) OR (J. EQ. 1) OR (J.EQ.COUNTP) GO TO 200
1526. C
1527. MAX = MAX1
1528. CALL PRIPL0(LIST, LISTFL, MAX)
1529. C
1530. IF (1.EQ.1) OR (J.EQ.1) GO TO 200
1531. WRITE(6, 902) 1.0, J
1532. READ(5, 901) ITH
1533. C
1534. IF (J.EQ.0) GO TO 299
1535. 200 CONTINUE
1536. C
1537. MAX = MAX1
1538. IF (LIST.EQ.2) LISTFL = 1
1539. C
1540. C FIND THE SYSTEM LOLP
1540. C
1542. C  Y = SYSCAP/FLAT(PY)
1543. LLOP = PLOP(Y)
1544. CALL SCL0K1
1545. C
1546. ZEL = (SYSCAP-COLFF(1))/SIGMA
1547. INFLAG = 1
1548. I = COUNTP
1549. CALL NDF(ZEL,COEFF,ZN,CUNUM,ZVALUE,INFLAG,I)
1550. C
1551. TIME4 = RCL0K1(1.0)
1552. TLPC = TLPC + TIME4
1553. C
1554. WRITE(6,903)LOLP
1555. WRITE(13,903)LOLP
1556. 903 FORMAT(14H CUCON LOP=, F12.9)
1557. C
1558. LOP = ZVALUE
1559. WRITE(6,903)LOLP
1560. WRITE(13,903)LOLP
1561. C
1562. IF (TEST.EQ.4) GO TO 299
1563. C
1564. C
1565. WRITE(6,904) TEST
1566. WRITE(13,904) TEST
1567. 904 FORMAT(52H SUBROUTINE CUCON SUCCESSFULLY COMPLETED FOR TEST = ,
1568. +12, 8H OPTION.)
1569. 299 RETURN
1570. END
1572. C
1573. C
1574. SUBROUTINE NDF(Z, C, ZN, CUNUM, FRE, INFLAG, IP)
1575. COMMON /TABLE/ TY(120), TINT(120)
1576. DOUBLE PRECISION TY, TINT
1577. C
1578. C
1579. INTEGER CUNUM
1580. DOUBLE PRECISION ZN9,ZN(9),E,C(8),Z,FRE,F3,F4,F5,F6,F7,F8,F9,
1581. + 2INTG,ZSQ
1582. C
1583. IF ( DABS(Z) .LT. 1.0D-10 ) GO TO 50
1584. IF ( DABS(Z) .GT. 5.0D00 ) GO TO 40
1585. ZSQ = Z*Z/2.0D00
1586. Z = DEAP(-ZSQ)/2.506626275D00
1587. ZN(1) = -Z * E
1588. ZN(2) = E * (Z**2 - 1.0D00)
1589. ZN(3) = -Z*E*(Z**2 - 3.0D00)
1590. ZN(4) = E*(Z**2*Z**2 - 6.0D00) + 3.0D00
1591. ZN(5) = -Z*E*(Z**2*Z**2 - 10.0D00) + 15.0D00
1592. ZN(6) = E*(Z**2*Z**2*Z**2 - 15.0D00) + 45.0D00 - 15.0D00
1593. ZN(7) = -Z*E*(Z**2*Z**2*Z**2 - 21.0D00) + 105.0D00 - 105.0D00
1594. ZN(8) = E*(Z**2*Z**2*Z**2*Z**2 - 28.0D00) + 210.0D00 - 420.0D00
1595. + 105.0D00
1596. C
1597. ZN9 = -Z*E*(Z**2*Z**2*Z**2*Z**2*Z**2 - 36.0D00) + 378.0D00
1598. + -1266.0D00 + 945.0D00
1599. C
1600. F3 = 6.0D00
1601. F4 = 24.0D00
1602. F5 = 120.0D00
1603. F6 = 720.0D00
1604. F7 = 5040.0D00
1605. F8 = 40320.0D00
1606. F9 = 362880.0D00
1607. C
1608. NUMCU = CUNUM - 5
1609. IF ( INFLAG.GT.0 ) GO TO 20
1610. \texttt{FRE = E}
1611. + \texttt{FRE = FRE - C(3)*ZN(3)/F3}
1612. + \texttt{C(4)*ZN(4)/F4}
1613. + \texttt{C(3)*C(3)*ZN(6) = 10.0000/F6}
1614. C
1615. \texttt{IF (NUMCU) 5,10,15}
1616. \texttt{FOUR CUMULANT SERIES}
1617. \texttt{RETURN}
1618. C
1619. \texttt{FIVE CUMULANT SERIES (CRAMER)}
1620. \texttt{10 FRE = FRE - C(5)*ZN(5)/F5}
1621. + \texttt{- C(3)*C(4)*ZN(7) = 35.0000/F7}
1622. + \texttt{C(3)*C(3)*ZN(9) = 280.0000/F9}
1623. \texttt{RETURN}
1624. C
1625. \texttt{EIGHT CUMULANT SERIES}
1626. \texttt{15 FRE = FRE - C(5)*ZN(5)/F5}
1627. + \texttt{C(6)*ZN(6)/F6}
1628. + \texttt{- (C(7) = 10.0000*C(3)*C(4)*ZN(7)/F7}
1629. + \texttt{C(8) = 56.0000*C(3)*C(5) + 35.0000*C(4)*C(4)*ZN(8)/F8}
1630. \texttt{RETURN}
1631. C
1632. \texttt{20 CONTINUE}
1633. C
1634. \texttt{REACHING 20 MEANS THAT LOC WILL BE CALCULATED}
1635. \texttt{CALL TALOK(2,ZINTEG,1P)}
1636. \texttt{22 FRE = 1.0000 - ZINTEG}
1637. \texttt{ZN9 = -ZN(8)}
1638. \texttt{DO 25 I = 1, 6}
1639. \texttt{J = 9 - I}
1640. \texttt{25 ZN(J) = -ZN(J-1)}
1641. \texttt{RETURN}
1642. C
1643. \texttt{40 FRE = 0.0000}
1644. \texttt{IF ( INFLAG .GT. 0.0000 .AND. Z.LT. 0.0000) FRE = 1.0000}
1645. \texttt{RETURN}
1646. \texttt{50 ZINTEG = 0.5000}
1647. \texttt{IF ( INFLAG .GT. 0.0000 ) GO TO 22}
1648. \texttt{FRE = 1.0000}
1649. \texttt{RETURN}
1650. \texttt{END}
1651. C
1652. \texttt{SUBROUTINE LICON( TEST, LIST )}
1653. C
1654. \texttt{PIECE-WISE LINEAR CONVOLUTION}
1655. C
1656. \texttt{COMMON /CURVES/ X(3000),EL(3000),ELF(3000),NORMLD(1000)}
1657. \texttt{COMMON /ARRAYS/CHECK(100,10,2),ENERGY(100,3),HOURC(8),HRLOAD(24),}
1658. \texttt{+CUMLIN(100,8),PLANTC(100,8),PLANT(100,2),TESTAR(10),EONE(100,2),}
1659. \texttt{+ELAST(100,2)}
1660. \texttt{COMMON /VAR1AB/ COUNTH,COUNTP,DX,DXNORM,INL,MAXI,MULTI,NEXIST,}
1661. \texttt{+ OUT, PEAK, POINTS, SYSCAP, TEN, PLT, HRI, TCE, TLPC, TLE, TLPL, TDE}
1662. \texttt{+INTEGR COUNTH, COUNTP, HRLOAD, INL, OUT, PEAK, POINTS, TESTAR}
1663. \texttt{REAL NORM}
1664. \texttt{DOUBLE PRECISION CUMLIN, PLANTC, HOURC}
1665. C
1666. C
1667. \texttt{DECLARATION OF LOCAL VARIABLES}
1668. \texttt{INTEGER LIST, TEST, FLAG}
1669. \texttt{REAL LOLP}
1670. C
1671. \texttt{NEXIST = COUNTP}
1672. \texttt{IF ( TEST .NE. 4 ) GO TO 5}
1673. \texttt{WRITE(6,916)}
1674. \texttt{916 FORMAT(42H ENTER THE NUMBER OF EXPANSION CANDIDATES /}
1675. \texttt{+4H NNN)}
1676. \texttt{READ(5,917) NUMEX}
1677. \texttt{917 FORMAT(13)}}
WRITE(6,916) NUMEX
WRITE(13,916) NUMEX
NEXIST = COUNTP - NUMEX
918 FORMAT(9H THE NUMBER OF EXPANSION CANDIDATES IS ,I3)
5 CONTINUE
C TRANSFER LD IN EL AND FILL UP WITH ZEROS
DO 10 I=1,3000
ELF(I)=0.0
IF (I.LT.POINTS) ELF(I) = NORMDL(I)
IF (I.GT.POINTS) ELF(I) = 0.0
10 CONTINUE
C CHECK THE LIST OPTIONS
901 FORMAT(12)
C IF (LIST .LT. 8) GO TO 100
WRITE(6,902)
902 FORMAT(13H WRITE THE PLANT NUMBER AFTER WHOSE,
+13H CONVOLUTION THE ELDC SHOULD BE PRINTED /
+13H (12 FORMAT) /3H NN)
READ(5,901) ITH
GO TO 110
100 ITH = 0
110 CONTINUE
C BEGIN PIECE-WISE LINEAR CONVOLUTION
M15 = POINTS*1.5
N15 = M15/100
TLE = 0.0
TLPL = 0.0
DO 200 IP=1,IPEND
WRITE(6,921) IP
WRITE(13,921) IP
921 FORMAT(1H * 13)
CALL SCLOK1
Y = PLANT(IP,1)/FLOAT(PEAK)
P = 1.0 - PLANT(IP,2)
IF ( TEST.EQ.8) GO TO 150
C CALCULATE THE ENERGY GENERATED BY THE IP-TH PLANT
SUMP = SUMP + Y
Y2 = SUMP
Y1 = SUMP - Y
INTEGRATE EL(1) FROM Y1 TO Y2
YY1 = Y1*POINTS
YY2 = Y2*POINTS
IY1 = YY1
IY2 = YY2
IF (YY1.LT.1.0. AND. YY2.LT.1.0) GO TO 130
IF (IY1.LE.0) IY1 = 1
SUMENE = 0.0
DO 120 IE = IY1, IY2
120 SUMENE = SUMENE + EL(IE)
SUMENE = SUMENE - 0.5 * (EL(IY1) + EL(IY2))
C ADJUST FOR END POINTS
DY1 = YY1 - IY1
DY2 = IY2 - YY2
1750. ADY1 = 0.0
1751. ADY2 = 0.0
1752. IF (YY1.LE.1.0) ADY1 = -1.0
1753. IF (DY1.GT.0.000001) ADY1 = DY1*EL(YY1+1) +
1754. + (EL(YY1)-EL(YY1+1))*(1.0-DY1*0.5))
1755. IF (DY2.GT.0.000001) ADY2 = DY2*(EL(YY2) +
1756. + (EL(YY2-1)-EL(YY2))*DY2*0.5))
1757. SUMENE = (SUMENE - ADY1 - ADY2) * FLOAT(PEAK)*COUNTH /
1758. + FLOAT(POINTS)
1759. GO TO 140
1760. 130 SUMENE = Y * FLOAT(PEAK) * COUNTH / FLOAT(POINTS)
1761. C 140 ENERGY(IP,1) = P * SUMENE
1762. C 150 CONTINUE
1763. C
1764. TIME1 = RCLK1(1.0)
1765. TLE = TLE + TIME1
1766. C
1767. 150 CONTINUE
1768. C
1769. C
1770. C
1771. CALL SCHK1
1772. C
1773. C CONVOLVE THE IP-TH PLANT
1774. CALL PMADD(Y,P)
1775. C
1776. TIME2 = RCLK1(1.0)
1777. TLE = TLE + TIME2
1778. TLPL = TLPL + TIME2
1779. C
1780. C FILL THE CHECK ARRAY.
1781. DO 166 J = 1,10
1782. XCAP = SYSCAP / 10. * FLOAT(J) / FLOAT(PEAK)
1783. C 160 CHECK(IP,J,1) = PLDLP(XCAP)
1784. C
1785. IF(IP.NE.1.AND.IP.NE.COUNTP) GO TO 168
1786. DO 165 J = 1,100
1787. JJ = 115*J
1788. IF(IP.EQ.1) EONE(J,1) = EL(JJ)
1789. IF(IP.EQ.COUNTP) ELAST(J,1) = EL(JJ)
1790. 165 CONTINUE
1791. 168 CONTINUE
1792. C
1793. LISTFL = 1
1794. IF (LIST.LT.5) GO TO 200
1795. IF (LIST.GT.7.AND.IP.NE.ITH) GO TO 200
1796. C
1797. MAX = MAXI
1798. CALL PRIPLQ(LIST,LISTFL,MAL)
1799. C
1800. IF(IP.NE.ITH) GO TO 200
1801. WRITE(6,902)
1802. READ(5,901) ITH
1803. C
1804. 200 CONTINUE
1805. C
1806. MAX = MAXI
1807. IF (LIST.GE.2.AND.LIST.LE.4.AND.TEST.NE.4) + CALL PRIPLQ(LIST,LISTFL,MAL)
1808. C
1809. C CALCULATE AND STORE THE CUMULANTS FOR THE NEW ELDC
1810. C
1811. IF (TEST.NE.4) GO TO 180
1812. FLAG = 4
1813. CALL CUCAL(FLAG)
1814. C
1815. DO 170 J = 1,8
1816. 170 CUMLIN(NEXIST,J) = HOUKC(J)
1817. CALL CUCON(RESULT,LIST)
1818. 180 CONTINUE
1819. C
```
1820. C     FIND THE SYSTEM LOLP
1821. C     CALL SCLOK1
1822. C     Y = SYSCAP/FLOAT(PEAK)
1823. C     LOLP = PLOLP(Y)
1824. C     TIME3 = RCLQK1(1.0)
1825. C     TLPL = TLPL + TIME3
1826. C     WRITE(6,920) LOLP
1827. C     WRITE(13, 920) LOLP
1828. C     920 FORMAT(1SH L1L0N LOLP = ,F12.9)
1829. C     IF ( TEST.EQ.3 ) TESTARI3) = 3
1830. C     IF ( TEST.EQ.4 ) TESTARI4) = 4
1831. C     WRITE(6, 904) TEST
1832. C     WRITE(13, 904) TEST
1833. C     904 FORMAT(52H SUBROUTINE LICON SUCCESSFULLY COMPLETED FOR TEST = , 
1834. C     +12, 8H OPTION.)
1835. C     RETURN
1836. C     END
1837. C     SUBROUTINE PWADD(Y,P)
1838. C     C     DECLARATION OF COMMON VARIABLES
1839. C     COMMON /CURVES/ XI3 0 0 0 ) ,EL(3000),ELF(3000),NORMLD(1000)
1840. C     COMMON /ARRAYS/CHERK(100,10,2),ENERGY(100,3),MORCH(11),MRLOAD(24), 
1841. C     *CUMLINE(100,8),PLANTC(100,8),TESTAR(10),HOURC(100,2), 
1842. C     ENERGY(1100,2)
1843. C     COMMON /VARIAB/ CUUNTH,CUUNTP,DX,DX,INL,MAXI,MINI,EXIST, 
1844. C     OUT,PEAR,POINT,SYSCAP,TEN,PLT,HAT,TEC,TLC,TLE,TLPL,TDE 
1845. C     +REAL(100,2), 
1846. C     DATA PMAX/0.999/, ELMIN/0.000001/, ONE/1.0/
1847. C     DEFINITION OF TERMS:
1848. C     X: X-AXIS I.E. NORMALIZED MW AXIS
1849. C     EL: LOAD PROBABILITY MATRIX
1850. C     ELF: TEMPORARY LOAD PROBABILITY MATRIX, STORES THE NEW 
1851. C     EL TEMPORARILY.
1852. C     Y: NORMALIZED MW CAPACITY OF ADDED UNIT: Y=MWC/PERPK
1853. C     P=1-FOR, LOAD PROBABILITY
1854. C     Q=FOR, FORCED AOUTAGE RATE
1855. C     MAXI: EXPECTED MAXIMUM DIMENSION OF EL MATRIX.
1856. C     PMAX: =0.999 IS CONSIDERED A REALISTIC LIMIT OF UNIT 
1857. C     AVAILABILITY ESTIMATE. I.E. IF P>PMAX 
1858. C     IT IS ASSUMED THAT P=1.0 AND THE ELD 
1859. C     CURVE DOES NOT CHANGE
1860. C     ELMIN=0.0000001, IS THE LIMIT OF LOLP ACCURACY.
1861. C     IF LULP < ELMIN THEN LOLP = 0.0
1862. C     DX NORMALIZED MW INCREMENT. DX=1/POINTS
1863. C     IF(P.GE.PMAX) GO TO 100
1864. C     FORM THE NEW ELD CURVE. STORE IN ELF
1865. C     Q=ONE-P
```
DO 50 I = 1, MAXI
   J = J + 1
   IF (ABS(X(I) - XY) .LE. ELMIN) GO TO 50
   ELY = (X(J - 1) - XY) / (X(J - 1) - X(J + 1)) * EL(J + 1)
   * (XY-X(J+1))/(X(J-1)X(J+1)) * EL(J - 1)
   GO TO 45
40 ELY = ELY
   CONTINUE
   45 RETURN
50 CONTINUE
   55 NEWMAX = MAX1
   IF (NEWMAX .GT. MAX1) NEWMAX = MAX1
   CONTINUE
   100 CONTINUE
   RETURN
END
SUBROUTINE TALOOK(Z,ZINTEG,IP)
COMMON /TABLE/ Y(120), YINT(120)
DOUBLE PRECISION AVZ,Y,YINT,Z,ZINTEG,DY,PK,ZINT,FACTOR,
+ R1,R2,R3,R4,Q12,Q13,Q14,Q23,Q24,Q34,Q21,Q31,Q41,Q32,Q42,Q43

LOOKUP IN NORMAL DISTRIBUTION FUNCTION THE VALUE OF
THE INTEGRAL OF EXP(-Z^2/2) FROM MINUS INFINITY TO Z.
INTERPOLATE WITH CUBIC FIT BETWEEN THE 120 DATA POINTS.
COPYED FROM TABLES. THE REFERENCE USED IS: TABLES OF
NORMAL PROBABILITY FUNCTIONS BY THEU.S. DEPARTMENT
OF COMMERCE.

DEFINITION OF KEY VARIABLES:
NAME  TYPE SIZE MEANING
Z   REAL - NORMALIZED INDEPENDENT VARIABLE
ZINTEG   REAL - INTEGRAL VALUE
Y   REAL 120 NORMALIZE INDEPENDENT VARIABLE
YINT REAL 120 INTEGRAL VALUES CORRESPONDING
K    INT. - GRID POINT CORRESPONDING TO Z
DY   REAL - Y INCREMENT

IF (DABS(Z).LT.5.70000) GO TO 5
ZINTEG = 1.00000
RETURN
5 CONTINUE
IF ( DABS(Z).GT.0.10000 ) GO TO 10
ZINTEG = 0.50000
RETURN
10 CONTINUE
DY = Y(2) - Y(1)
PK = DABS(Z)/DY + 1.00000
K = PK
AVZ = DABS(Z)
R1 = 0.00000
R2 = 0.00000
R3 = 0.00000
R4 = 0.00000
IF ( AVZ.NE.Y(K)) K1 = AVZ - Y(K)
IF ( AVZ.NE.Y(K+1)) R2 = AVZ - Y(K+1)
IF ( AVZ.NE.Y(K+2)) R3 = AVZ - Y(K+2)
IF ( AVZ.NE.Y(K+3)) R4 = AVZ - Y(K+3)
Q12 = Y(K) - Y(K+1)
Q13 = Y(K) - Y(K+2)
Q14 = Y(K) - Y(K+3)
Q23 = Y(K+1) - Y(K+2)
Q24 = Y(K+1) - Y(K+3)
Q34 = Y(K+2) - Y(K+3)
Q21 = -Q12
Q31 = -Q13
Q41 = -Q14
Q32 = -Q23
Q42 = -Q24
Q43 = -Q34
IF ( DABS(R1).GT.0.00-10)GO TO 20
ZINT = YINT(K1)*R2*R3*R4/(Q12*Q13*Q14)
GO TO 30
20 IF ( DABS(R2).GT.0.00-10)GO TO 22
ZINT = YINT(K1)*R1*R3*R4/(Q21*Q23*Q24)
GO TO 30
22 IF ( DABS(R3).GT.0.00-10)GO TO 24
ZINT = YINT(K1)*R1*R2*R4/(Q31*Q32*Q34)
GO TO 30
24 IF ( DABS(R4).GT.0.00-10)GO TO 26
ZINT = YINT(K1)*R1*R2*R3/(Q41*Q42*Q43)
GO TO 30
26
SUBROUTINE OPERATE (TEST, LIST)

THIS SUBROUTINE CALCULATES THE ENERGY GENERATED BY THE SYSTEM PLANTS USING A BLOCK REPRESENTATION OF THE LOAD DURATION CURVE AND THE DERATING METHOD.

COMMON /CURVES/ X(3000), EL(3000), ELF(3000), NORMLD(1000)
COMMON /ARRAYS/ CMLK(100,10,2), ENERGY(100,3), HOURC(8), HRLOAD(24),
CMLIN(100,6), PLANTC(100,8), PLANT(100,2), TESTAR(10), EON(100,2),
ELAST(100,2)
COMMON /VARIABLES/ COUTH, COUTP, INL, OUT, PEAK, POINTS, TESTAR
COMMON /TAB/ TY(120), INT(120)
DOUBLE PRECISION TY, INT

DIMENSION NBREAK(20)

INPUT THE NUMBER OF LOAD BLOCKS
WRITE(6,900)
900 FORMAT(33H ENTER THE NUMBER OF LOAD BLOCKS /
+ 5H NNNN)
READ(5,901) NBLOCK
901 FORMAT(20(I4,1x))

IF (NBLOCK.GT.20) GO TO 50

INPUT THE BREAK POINTS (FIRST = 1, LAST = POINTS)
WRITE(6,902) NBLOCK
902 FORMAT(8H ENTER 15,13H BREAK POINTS /20(5H NNNN))
READ(5,901) (NBREAK(N), N=1,NBLOCK)
NBREAK(1) = 1
NBREAK(NBLOCk) = POINTS

DEFINE THE NEW LOAD DURATION CURVE APPROXIMATION.

DO 30 N=1,NBLOCK
BLEVEL = 0.0
II = NBREAK(N)
I2 = NBREAK(N+1)
ISIZE = NBREAK(N+1) - NBREAK(N) + 1
30 CONTINUE

DEFINE THE BLOCK LOAD LEVEL
DO 10 I = II, I2
10 BLEVEL = BLEVEL + NORMLD(I)
BLEVEL = BLEVEL / FLOAT(ISIZE)

DO 20 I = 11, 12
20 ELF(I) = BLEVEL

30 CONTINUE
GO TO 100

50 CONTINUE

REACHING 50 MEANS THAT THE NUMBER OF BLOCKS IS GREATER THAN 20. LOAD BLOCKS ARE ASSUMED EQUAL.

DEFINE THE BLOCK SIZE
ISIZE = POINTS / NBLK

DEFINE THE BLOCK LOAD LEVELS
DO 70 N = I* POINTS t 1S12E
BLEVEL = 0.0
11 = N
IF ( 11.EQ.POINTS ) GO TO 70
12 = N + ISIZE
IF ( 12.GT.POINTS ) 12 = POINTS
DO 55 1 = 11, 12
55 BLEVEL = BLEVEL + FLOAT(12)

IBSIZE = ISIZE + 1
IF ( 12.EQ.POINTS ) IBSTE = POINTS - 11 + 1

BLEVEL = BLEVEL / FLOAT(IBSIZE)

DEFINE THE NEW LDC
DO 60 I = 11,12
60 ELF(I) = BLEVEL

70 CONTINUE

100 CONTINUE

REACHING 100 MEANS THAT A NEW LDC EXISTS.
DO 110 1 = 1,POINTS
110 ELF(I) = NORMLO(I)
LISTFL = 2
MAXP = POINTS
CALL PRIPLU(LIST,LISTFL,MXP)

CALCULATE THE ENERGY PRODUCED BY EACH PLANT USING PLANT DERATINGS
SUMP = 0.0
TDE = 0.0
CALL SCL0K1

DO 200 IP = 1, COUNTP
200 Y = PLANT(IP)/FLOAT(PEAK)
P = 1.0 - PLANT(IP*2)
Y=Y*P

CALCULATE THE ENERGY GENERATED BY THE IP-Th PLANT
SUMP = SUMP + Y
Y2 = SUMP
Y1 = SUMP - Y
INTEGRATE ELF(I) FROM Y1 TO Y2
YY1 = Y1*POINTS
YY2 = Y2*POINTS
IV1 = YY1
IV2 = YY2
2170. IY2 = IY2 + 1
2171. IF ( Y11.LT.1.0 .0 . AND. YY2.LT.1.0 ) GO TO 130
2172. IF ( IY1.LE.0 ) IY1 = 1
2173. SUMENE = 0.0
2174. DD 120 IE = IY1, IY2
2175. 120 SUMENE = SUMENE + ELF(IE)
2176. SUMENE = SUMENE - 0.5 * ( ELF(IY1)+ELF(IY2) )
2177. C
2178. C ADJUST FOR END POINTS
2179. DY1 = YY1 - IY1
2180. DY2 = IY2 - YY2
2181. ADY1 = 0.0
2182. ADY2 = 0.0
2183. IF ( YY1.LT.1.0 ) ADY1 = -1.0
2184. IF (DY1.GT.0.00001) ADY1 = DY1*( ELF(IY1+1) +
2185. + (ELF(IY1)-ELF(IY1+1))*(1.0-DY1*0.5))
2186. IF (DY2.GT.0.00001) ADY2 = DY2*(ELF(IY2) +
2187. + (ELF(IY2-1)-ELF(IY2))*DY2*0.5)
2188. SUMENE = (SUMENE - ADY1 - ADY2) * FLOAT(PEAK)*COUNTH /
2189. * FLOAT(POINTS)
2190. GO TO 140
2191. 130 SUM ENE = V  6 FLOAT(PEAK) 6 COUNTH / FLOAT(POINTS)
2192. C
2193. 140 ENERGY(IP*3) = SUMENE
2194. C
2195. 200 CONTINUE
2196. C
2197. RETURN
2198. END
2199. C
2200. C SUBROUTINE UNCERT(MEAN,THOURS,VAR)
2201. C
2202. WRITE(6*900)
2203. 900 FORMAT(AH / / •  INPUT THE EXPECTED DEVIATION FR O M  THE FORCASTED*;
2204. +* ENERGY DEMAND,*# USE A DECIMAL NUMBER FOR THE % DEVIATION*)
2205. READ(5*) DEV
2206. IF (DEV.LT.0.01) GO TO 300
2207. D = MEAN * DEV /100.0
2208. WRITE(6*901)
2209. 901 FORMAT(* PROVIDE THE CONFIDENCE LEVEL OF THE FORCAST*;
2210. +* INPUT AN INTEGER BETWEEN 50% AND 100% IN INCREMENTS OF 5*)
2211. READ(5*) LEVEL
2212. INDEX = (LEVEL - 45)/5
2213. GO TO (50,55,60,65,70,75,80,85,90,95,100), INDEX
2214. 50 S = D/0.6745
2215. GO TO 200
2216. 55 S = D/0.7555
2217. GO TO 200
2218. 60 S = D/0.8417
2219. GO TO 200
2220. 65 S = D/0.9346
2221. GO TO 200
2222. 70 S = D/1.0365
2223. GO TO 200
2224. 75 S = D/1.1303
2225. GO TO 200
2226. 80 S = D/1.2816
2227. GO TO 200
2228. 85 S = D/1.4395
2229. GO TO 200
2230. 90 S = D/1.6450
2231. C
GO TO 200
95 S = D/1.9600
GO TO 200
100 S = D/6.0000
C
200 DE = S*THOURS/1000.0
VAR = S*S
WRITE(903) LEVEL,DEV,DE,D,S,VAR
WRITE(13,903) LEVEL,DEV,DE,D,S,VAR
903 FORMAT(C6, LEVEL: ',14.6 E/ E14.8 DEVIATION : ',F4.0 ', F8.2 ', GWH: ',F6.2 '/
SIGMA: ',F8.2 ', AND VARIANCE ',F12.2'/)
RETURN
300 VAR = 0.0
RETURN
END
C
SUBROUTINE PLOT(X,Y,MAX,L1)
DIMENSION X(100),Y(100)
INTEGER*2 BLANK,DASH,STAR,PRC100),YAX
DATA BLANK/2H /,DASH/2H - /,STAR/2H* /,YAX/2H1 /
SET TO 100% ANY DIFFERENCE EXCEEDING IT.
DO 10 J = 1,MAX
IF(Y(J).GT.100.0) Y(J) = 100.0
IF(Y(J).LT.-100.0) Y(J) = -100.0
10 CONTINUE
PLOT THE DIFFERENCES
DO 120 I = 1,101
UP = 103.0 - 1*2.0
D = UP - 2.0
DO 90 J = 1,MAX
IF(Y(J).GT.D.AND.Y(J).LE.UP) GO TO 80
IF(Y(J).LT.1.0.AND.Y(J).GT.-1.0) GO TO 40
20 PRC(J) = YAX
GOTO 70
30 PRC(J) = DASH
GOTO 70
40 IF(MOD(J,10)) 50,50,60
50 PRC(J) = YAX
GOTO 70
60 PRC(J) = BLANK
70 PRC(MAX) = YAX
GOTO 90
80 PRC(J) = STAR
90 CONTINUE
XI = (100.0 - (1-1)*2.0)/FLOAT(LI)
IK = R1 * 10
II = (100.0 - (1-1)*2)/LI
III = 100/LI
IF(MOD(IK,III)) 110,100,110
100 WRITE(13,900)II,(PRC(J),J=1,MAX)
GOTO 120
110 WRITE(13,901) (PRC(J),J=1,MAX)
120 CONTINUE
900 FORMAT(1H ,T10,14,2H 1,100A1)
901 FORMAT(1H ,T15,1H1,100A1)
PRINT X-AXIS
WRITE(13,902) (X(J),J=10,MAX,10)
902 FORMAT(1H ,T17,10F10.0//)
WRITE(13,903)
903 FORMAT(1H///)
C
RETURN
END
APPENDIX D
SENSITIVITY ANALYSES FIGURES

The Figures contained in this Appendix are supplementary to those listed in the sensitivity analyses presented in chapter three.

Figures D.1 to D.30 in groups of three (i.e. D.1-D.3, D.4 - D.6, etc.) are similar to Figures 3.12, 3.13, and 3.14 and depict the percentage error of the cumulant method for case studies 2A to 11B.

Figures D.31 to D.40 are similar to 3.15 and show the percentage differences of the cumulant and derate methods from the piecewise linear method for plant energy production, for case studies 2A and 2B to 11A and 11B.

Finally, Figures D.41 to D.50 are similar to Figure 3.20 and show the effect of including energy forecast uncertainty in the probabilistic simulation of the electric generating systems for cases 2B, 3B, ..., and 11B.
Figure D.1  Percentage Differences of Original Load Probability Curves for Cases 2A and 2B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.2  Percentage Differences of Final Equivalent Load Probability Curves in Case 2A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.3  Percentage Differences of Final Equivalent Load Probability Curves in Case 2B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.4  Percentage Differences of Original Load Probability Curves for Cases 3A and 3B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.5  Percentage Differences of Final Equivalent Load Probability Curves in Case 3A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.6 Percentage Differences of Final Equivalent Load Probability Curves in Case 3B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.7  Percentage Differences of Original Load Probability Curves for Cases 4A and 4B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.8  Percentage Differences of Final Equivalent Load Probability Curves in Case 4A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.9 Percentage Differences of Final Equivalent Load Probability Curves in Case 4B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.10  Percentage Differences of Original Load Probability Curves for Cases 5A and 5B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.11 Percentage Differences of Final Equivalent Load Probability Curves in Case 5A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.12  Percentage Differences of Final Equivalent Load Probability Curves in Case 5B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.13 Percentage Differences of Original Load Probability Curves for Cases 6A and 6B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.14 Percentage Differences of Final Equivalent Load Probability Curves in Case 6A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.15 Percentage Differences of Final Equivalent Load Probability Curves in Case 6B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.16 Percentage Differences of Original Load Probability Curves for Cases 7A and 7B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.17 Percentage Differences of Final Equivalent Load Probability Curves in Case 7A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.18 Percentage Differences of Final Equivalent Load Probability Curves in Case 7B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.19  Percentage Differences of Original Load Probability Curves for Cases 8A and 8B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.20  Percentage Differences of Final Equivalent Load Probability Curves in Case 8A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.21 Percentage Differences of Final Equivalent Load Probability Curves in Case 8B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.22 Percentage Differences of Original Load Probability Curves for Cases 9A and 9B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.23 Percentage Differences of Final Equivalent Load Probability Curves in Case 9A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.24 Percentage Differences of Final Equivalent Load Probability Curves in Case 9B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method.
Figure D.25 Percentage Differences of Original Load Probability Curves for Cases 10A and 10B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.26  Percentage Differences of Final Equivalent Load Probability Curves in Case 10A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.27 Percentage Differences of Final Equivalent Load Probability Curves in Case 10B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.28 Percentage Differences of Original Load Probability Curves for Cases 11A and 11B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.29  Percentage Differences of Final Equivalent Load Probability Curves in Case 11A, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.30 Percentage Differences of Final Equivalent Load Probability Curves in Case 11B, Between Each of the 4-, 5-, and 8-Cumulant Series Approximations and the Reference Piecewise Linear Method
Figure D.31 Percentage Differences in Plant Energy Production for Cases 2A and 2B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.32 Percentage Differences in Plant Energy Production for Cases 3A and 3B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.33 Percentage Differences in Plant Energy Production for Cases 4A and 4B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.34  Percentage Differences in Plant Energy Production for Cases 5A and 5B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.35 Percentage Differences in Plant Energy Production for Cases 6A and 6B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.36 Percentage Differences in Plant Energy Production for Cases 7A and 7B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.37 Percentage Differences in Plant Energy Production for Cases 8A and 8B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.38 Percentage Differences in Plant Energy Production for Cases 9A and 9B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.39  Percentage Differences in Plant Energy Production for Cases 10A and 10B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.40 Percentage Differences in Plant Energy Production for Cases 11A and 11B, Between the Reference Piecewise Linear Method, and Each of the 4-, 5-, and 8-Cumulant Approximations. The Differences in Plant Energy Production Between the Piecewise Linear and the Derate Methods are Also Depicted.
Figure D.41 Percentage Difference in Plant Energy Production for Case 2B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty.
Figure D.42 Percentage Difference in Plant Energy Production for Case 3B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
Figure D.43 Percentage Difference in Plant Energy Production for Case 4B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
Figure D.44  Percentage Difference in Plant Energy Production for Case 5B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
Figure D.45 Percentage Difference in Plant Energy Production for Case 6B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
Figure D.46 Percentage Difference in Plant Energy Production for Case 7B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
Figure D.47 Percentage Difference in Plant Energy Production for Case 8B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
Figure D.48  Percentage Difference in Plant Energy Production for Case 9B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
Figure D.49  Percentage Difference in Plant Energy Production for Case 10B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
Figure D.50 Percentage Difference in Plant Energy Production for Case 11B, Between the Reference Piecewise Linear Method and the 5-Cumulant Series Approximation With and Without Uncertainty
This Appendix contains the Fortran source of the test program used in Chapter 4. This program was used for the numerical evaluation of the stage iterative dynamic programming algorithm.
SAMPLE OPTIMIZATION PROBLEM FOR NUMERICAL APPLICATION
OF THE "STAGE ITERATIVE DYNAMIC PROGRAMMING® ALGORITHM

DEFINITION OF COMMON VARIABLES

COMMON ONE
IC : CONSTRAINT MATRIX: IF IC(J,K) .LE. 0, THAN STATE XI(J,K) IS NOT ACCEPTABLE.
IU : CONTROL VECTOR
IUR : STATE CONTROL VALUES FOR ALL STAGES
JSOL : STATE INDEX FOR OPTIMAL SOLUTION
JSOLD : STATE INDEX FOR OLD SOLUTION
OBJ : OBJECTIVE FUNCTION VALUES FOR ALL STATES AND STAGES
SOL : OBJECTIVE FUNCTION VALUES FOR ALL STAGES OF THE OPTIMAL SOLUTION
SOLD : SAME AS SOL BUT FOR OLD SOLUTION

COMMON TWO
JEXP : STATE EXPONENT IN OBJ DEFINITION
IEXP : CONTROL EXPONENT IN OBJ DEFINITION
OBJEXP: OBJ UNIVERSAL EXPONENT

DEFINITION OF LOCAL VARIABLES
BCF : FINAL BOUNDARY CONDITION
BCO : ORIGIN BOUNDARY CONDITION
EXPONS: OBJ EXPONENTS MATRIX; (N,J), N=1,2,3 FOR 1, J & OBJ EXPONENTS; J=1,7 FOR IE CASES.
I : CONTROL INDEX
IE : INDEX FOR THE OBJ EXPONENTS LOOP.
IMAX : MAXIMUM I
IT : STATE - STAGE MATRIX PRINT FLAG
ITMAX : ITERATION LIMIT
J : STATE INDEX
JF : J OF FINAL STATE
JGUESS: INITIAL GUESS VECTOR.
JL : INITIAL B.C. INPUT VALUE
JLDIFF: DIFFERENCE BETWEEN THE OLD AND NEW SOLUTIONS
JMAX : MAXIMUM J
JO : J OF ORIGIN STATE
J1 : ORIGIN STATE IN K1 STAGE FOR EACH DP STEP
J4 : TARGET STATE IN K4 STAGE FOR EACH DP STEP
K : STAGE INDEX
KEND : KF - 1
KF : TARGET STAGE
KMAX : MAXIMUM K
KO : ORIGIN STAGE
KSTART: ORIGIN STAGE FOR THAT STAGE IN WHICH THE OLD AND NEW SOLUTIONS ARE DIFFERENT FOR THE FIRST TIME
KSTEP : DP ITERATION STEP
KSTP : INDEX FOR THE KSTEP LOOP.
K1 : ORIGIN STAGE FOR EACH DP STEP
K4 : TARGET STAGE FOR EACH DP STEP
MAXIE : MAXIMUM NUMBER OF OBJ DEFINITIONS(CASES)
MAXKST : MAXIMUM NUMBER OF KSTEP'S

COMMON /ONE/ IC(21,21), IU(9), IUR(21,21), JSOL(21),
+ OBJ(21,21), SOL(21)
COMMON /TWO/ IEXP, JEXP, OBJEXP
DIMENSION EXPONS(3,7), JGUESS(21,10), JSOLD(21), SOLD(21)
+ KAR(21)

INITIALIZE OBJ, KAR, AND IUR ARRAYS
DO 20 K = 1,21
DO 10 J = 1,21
10 IUR(J,K) = 99
SOLD(K) = 100000.0

COMMON /ONE/ IC(21,21), IU(9), IUR(21,21), JSOL(21),
+ OBJ(21,21), SOL(21)
COMMON /TWO/ IEXP, JEXP, OBJEXP
DIMENSION EXPONS(3,7), JGUESS(21,10), JSOLD(21), SOLD(21)
+ KAR(21)

INITIALIZE OBJ, KAR, AND IUR ARRAYS
DO 20 K = 1,21
DO 10 J = 1,21
10 IUR(J,K) = 99
SOLD(K) = 100000.0
70. KAR(K)=K
71. 20 CONTINUE
72. C READ INPUT DATA FROM DATA SET "LARSON_INPUT.DATA"
73. C 1. MATRIX DIMENSIONS, STAGE STEP, ITERATION LIMIT
74. READ(10,900) JMAX, KMAX, KSTEP, ITMAX
75. 900 FORMAT(1H,2112)
76. C 2. B.C.
77. READ(10,900) JO, KO, JF, KF
78. READ(10,901) OBJ(JO,KO), BCF
79. JSOL(KO) = JO
80. SOL(KO) = 0.0
81. 901 FORMAT(3(1X,F7.0))
82. C 3. CONSTRAINTS
83. DO 40 J = 1, JMAX
84. 40 READ(10,900) (IC(J,K), K = 1, KMAX)
85. C 4. CONTROL VECTOR
86. READ(10,900) IMAX, (IU(I), I = 1, 9)
87. C 5. INITIAL GUESS
88. DO 45 IG = 1, 10
89. 45 READ(10,900) IGUESS(K, IG), IG = 1, KMAX)
90. C 6. OBJECTIVE FUNCTION EXPONENTS
91. DO 50 IE = 1, 7
92. 50 READ(10,901) (EXPS(I), I = 1, 3)
93. C DEFINE MAXIE, MAXKST, AND MAXIG INTERACTIVELY.
94. WRITE(6,897)
95. 897 FORMAT(I4, 697)
96. READ(5,900) MAXIE, MAXKST, MAXIG, ITOPFL
97. C PRINT KEY INPUT PARAMETERS
98. WRITE(6,917)
99. 917 FORMAT(35H IMAX,JMAX,ITMAX,JO,KO,JF,KF,KSTEP)
100. WRITE(6,900) IMAX, JMAX, ITMAX, JO, KO, JF, KF, KSTEP
101. DO 47 IG = 1, 10
102. 47 WRITE(6,900) (IGUESS(K, IG), IG = 1, KMAX)
103. C CHANGE INPUT DATA
104. 49 WRITE(6,902)
105. 902 FORMAT(40H IF YOU WANT TO CHANGE ANY OF THE INPUT/
106. +46H DATA, ENTER THE NEW VALUES; OTHERWISE TYPE -5)
107. WRITE(6,903)
108. 903 FORMAT(13H IMAX, ITMAX)
109. READ(5,90) JMAX, ITMAX
110. IF (JMAX .LT. 0) GO TO 85
111. JMAX = JMAX
112. ITMAX = ITMAX
113. WRITE(6,904)
114. 904 FORMAT(35H OBJ EXPONENTS? (IEXP, JEXP, OBJEXP))
115. READ(5,90) IEXP, JEXP, OBJEXP
116. IF (IEXP .LT. -4) GO TO 55
117. IEXP = IEXP
118. JEXP = JEXP
119. OBJEXP = OBJEXP
120. C 55 WRITE(6,905)
121. 905 FORMAT(24H CONSTRAINTS? (J,K,ICV))
122. READ(5,90) J, K, ICV
123. IF (J .LT. 0) GO TO 70
124. IF (K .LT. 0) GO TO 70
125. IC(J,K) = ICV
126. GO TO 60
127. 70 WRITE(6,906)
128. C 906 FORMAT(30H CONTROL VECTOR? (I,IU,IMAX))
129. READ(5,90) I, IU, IMAX
130. IF (I .LT. 0) GO TO 80
131. IU(I) = IU
140. IMAX = IMAXT
141. GO TO 70
142. 80 CONTINUE
143. C
144. WRITE(6,912)
145. 912 FORMAT(44H BOUNDARY CONDITIONS? (JO,KO,BCO,JF,KF,BCF) )
146. READ(5,* ) JOT, KOT, BCOT, JFT, KFT, BCFT
147. IF (JOT .LT. 0) GO TO 85
148. JO = JOT
149. KO = KOT
150. JF = JFT
151. KF = KFT
152. OBJ(JO,KO) = BCOT
153. BCF = BCFT
154. JSOL(KO) = JO
155. SOL(KO) = BCOT
156. JSOL(KF) = JF
157. 85 CONTINUE
158. C PRINT KEY INPUT PARAMETERS
159. WRITE(11,917)
160. WRITE(11,900) IMAX, JMAX, ITMAX, JO, KO, JF, KF, KSTEP
161. NORMDP = KF - 2
162. C
163. C BEGIN OBJ DEFINITION LOOP.
164. DO 300 IE = 1,MAXIE
165. TIMOBJ = 0.0
166. C
167. C DEFINE OBJ EXPONENTS.
168. IEXP = EXPONS(1,IE)
169. JEXP = EXPONS(2,IE)
170. OBJEXP = EXPONS(3,IE)
171. C
172. WRITE(6,899) IEXP,JEXP,OBJEXP
173. WRITE(11,899) IEXP,JEXP,OBJEXP
174. 899 FORMAT(/10(2H 9),2X,11H OBJ EXP=,2I3,F5.1,1X,10(2H 9))
175. C
176. C BEGIN "INITIAL GUESS LOOP".
177. DO 250 IG = 1,MAXIG
178. TIMGES = 0.0
179. WRITE(6,918) IG, (JGUESS(K,IG),K=KO,KF)
180. 918 FORMAT(1H /15X*2I4,2X*16H INITIAL GUESS*/9H *2113/F5.1,1X,10(2H 9))
181. WRITE(11,918) IG, (JGUESS(K,IG),K=KO,KF)
182. C
183. C BEGIN KSTEP LOOP.
184. DO 180 KSTP = 1,MAXKST
185. TIMSTP = 0.0
186. CALL SCK1
187. IF(KSTP.GT.NORMDP) GO TO 180
188. IF (ITDPPFL.NE.1.AND.KSTP.NE.NORMDP) GO TO 180
189. C
190. C SELECT ONLY KSTEPS IN MULTIPLES OF 5 WHEN KSTP > 5.
191. KM = MOD(KSTP,5)
192. IF (KSTP.GT.5.AND.KM.NE.0.AND.KSTP.NE.NORMDP) GO TO 180
193. C
194. C DEFINE KSTEP
195. KSTEP = KSTP
196. C
197. WRITE(6,895) KSTEP
198. WRITE(11,895) KSTEP
199. 895 FORMAT(1H /5J2H 9),2X,8H KSTEP=,13,1X,5(2H 9))
200. C
201. C REDEFINE INITIAL GUESS.
202. DO 60 K = 1,KMAX
203. 60 CONTINUE
204. C
205. C BEGIN ITERATIVE OPTIMIZATION
206. C
207. C
208. C
209. C KEND = KF - 1
DO 200 IT = 1, ITMAX
DO 120 KC = KO, KEND
K1 = KC
J1 = JSOL(K1)
K4 = KC + KSTEP + KDIFF
IF (K4 .GT. KF) GO TO 130
J4 = JSOL(K4)
CALL DPI(J1,K1,J4,K4,JMAX,ITMAX)
120 CONTINUE
C
C ADD B.C. FINAL
C
130 SOL(KF) = SOL(KF) + BCF
C
C COMPARE THE NEW AND OLD SOLUTIONS
DO 140 K=2, KF
LDIFF = JDIFF(K) - JDIFF(K)
IF (IAABS(LDIFF) .LT. 1) GO TO 140
KSTART = K - KSTEP
IF (KSTART .LE. KO) KSTART = KO
GO TO 150
140 CONTINUE
C
140 CONTINUE
C
GO TO 210
C
C REPLACE OLD WITH NEW SOLUTION
C
150 DO 160 K = KSTART, KF
SOL(K) = SOL(K)
160 JSOLD(K) = JDIFF(K)
C
PRINT ITERATION NUMBER
WRITE(11, 914) IT
914 FORMAT(1H , I3, 11H ITERATIONS)
C
PRINT SUBOPTIMAL SOLUTION
C
WRITE(11, 915)
915 FORMAT(21H SUBOPTIMAL SOLUTION)
WRITE(11, 891) (JDIFF(K), K = KO, KF)
WRITE(11, 892) (SOL(K), K = KO, KF)
C
200 CONTINUE
C
DROPPING THROUGH THE "200" LOOP MEANS NO CONVERGENCE
WRITE(11, 916)
WRITE(6, 916)
916 FORMAT(42H PROGRAM STOPS BECAUSE OF ITERATION LIMIT )
GOTO 180
C
C REACHING "210" IMPLIES CONVERGENCE
WRITE(11, 914) IT
WRITE(6, 914) IT
C
PRINT RESULTS
C
TIMSTP = RCLK1(0.001)
TIMGES = TIMGES + TIMSTP
WRITE(6, 880) KSTEP, TIMSTP
WRITE(11, 880) (KSTEP, TIMSTP)
880 FORMAT(1H, 12H STAGE STEP=, I3, 7H TIME=, F12.6, + 13H MILLISECONDS)
C
WRITE(11, 907)
907 FORMAT(18H OPTIMAL SOLUTION)
WRITE(6, 889) SOL(KF)
WRITE(11, 890) (KAR(K), K = KO, KF)
WRITE(11, 891) (JDIFF(K), K = KO, KF)
WRITE(11, 892) (SOL(K), K = KO, KF)
C
180 CONTINUE
C
TIMOBJ = TIMOBJ + TIMGES
WRITE(*,881) IE,TIMOBJ
WRITE(*,882) IE,TIMOBJ
882 FORMAT(1H *10H OBJ NUM. = ,I3,7H TIME=,F12.6,
+ 13H MILLISECONDS)
250 CONTINUE
WRITE(*,883) TIMALL
WRITE(*,883) TIMALL
883 FORMAT(1H *12H TOTAL TIME=,F12.6)
298 STOP
END

C
C
C SUBROUTINE DP(J1,K1,J4,K4,JMAX, IMAX)
C FORWARD DYNAMIC PROGRAMMING
C
C DEFINITION OF LOCAL VARIABLES
C I = CONTROL INDEX
C J = STATE INDEX
C JB = BACKWARD TRACE OF STATE INDEX
C JNEW = NEW STATE FOR EVALUATION IN STAGE K+1
C JOP = STATE INDEX FOR OPTIMAL SOLUTION
C K = STAGE INDEX
C KB = BACKWARD TRACE OF STAGE INDEX
C K2 = K1 + 1
C K3 = K4 - 1
C OBJNEW = OBJ FOR STATE X(JNEW,K+1)
C OP = DUMMY VARIABLE
C
C COMMON /ONE/ IC(21,21), IU(9), IUR(21,21), JSOL(21),
+ OBJ(21,21), SOL(21)
C COMMON /TWO/ IEXP,JEXP,OBJEXP
C K2 = K1 + 1
C K3 = K4 - 1
C INITIALIZE RELEVANT OBJ VALUES
C DO 7 K = K2, K4
C DO 5 J = 1,JMAX
C OBJ(J,K) = 1000000.0
C 7 CONTINUE
C
C STAGE LOOP
C DO 100 K = K1,K3
C STATE LOOP
C DO 90 J = 1,JMAX
C IF (K.EQ.K1.AND.J.NE.J1) GO TO 90
C CHECK IF ORIGIN STATE (J,K) IS A VALID ONE
C IF (IC(J,K)) 90,90,10
C 90 CONTINUE
C IF (K.EQ.K3.AND.JNEW.NE.J4) GO TO 80
C CONTROL LOOP
C 10 DO 80 I = 1,JMAX
C DEFINE THE STATE TO BE CONSIDERED IN STAGE K+1
C JNEW = J + IU(I)
C CHECK IF "JNEW" IS A VALID STATE
C IF (JNEW.GT.0 .AND. JNEW.LE.JMAX .AND. IC(JNEW,K+1) .GT. 0)
C +GO TO 20
C GO TO 80
C 20 CONTINUE
C CALCULATE THE OBJ VALUE FOR STATE X(JNEW,K+1)
OBJNEW=((J-1)**JEXP*(IU(I)**IEXP)**OBJEXP*OBJ(J,K)
C REPLACE OLD OBJ VALUE WITH THE NEW ONE IF OBJNEW IS SMALLER
IF (OBJNEW .GT. OBJ(JNEW,K+1)) GO TO 80
OBJ(JNEW,K+1) = OBJNEW
C UPDATE THE TRACE VECTOR
IUR(JNEW,K+1) = IU(I)
80 CONTINUE
90 CONTINUE
100 CONTINUE
SOL(K4) = OBJ(J4,K4)
JSOL(K4) = J4
C TRACE THE OPTIMAL PATH
DO 130 K = K2,K3
KB = K3-K+K2
J = JSOL(KB+1)
JB = J-IUR(J,KB+1)
JSOL(KB) = JB
SOL(KB) = OBJ(JB,KB)
130 CONTINUE
C RETURN
RETURN
END
REFERENCES


22. S. Nakamura, editor, Electric Generating System Expansion Analysis, Conference Proceeding, The Mechanical Engineering Department of The Ohio State University, Columbus, Ohio, March 5-6, 1981.


36. J. Stremel, J. Dickson, "The Cumulant Method for Production Costing: Evaluations and Suggestions for the Practitioner", Proceedings of the Electric Generating System Expansion Analysis Conference, Mechanical Engineering Department, The Ohio State University, Columbus, Ohio, March 5-6, 1981.


43. F.E. Davis, "Closed Form Financial and End Effects Modification to WASP", Conference Proceedings on Electric Generating System Expansion Analysis, Mechanical Engineering Department, The Ohio State University, Columbus, Ohio March 5-6, 1981.

44. T.J. LaVene, "End Effects and Fixed Change Rates", Conference Proceedings of the 2nd WASP Conference, August 18-19, 1977, The Ohio State University, Department of Mechanical Engineering, Columbus, Ohio.


48. S. Nakamura (Editor) et al., "Electric Utility Analysis Package", Final Report submitted to the Public Utilities Commission of Ohio by The Ohio State University, Columbus, Ohio, October, 1977.


64. W. Bayless, R.T. Jenkins, "More Than You Really Need to Know About the Cumulant Method to do Simulation and Expansion Studies", Proceedings of the Electric Generation System Expansion Analysis Conference, Mechanical Engineering Department, The Ohio State University, Columbus, Ohio, March 5-6, 1981.

65. K.F. Schenck, University of Ottawa, personal communication.

66. J. Amlin, Battelle columbus laboratories, personal communication.


71. T. Morin, Purdue University, Personal Communication.


77. C. Mount-Campbell, The Ohio State University, Personal Communication.