INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Constrained optimal neighborhoods and kernel estimators as improvements to applications of kriging

Moody, Marla Marie, Ph.D.

The University of Arizona, 1993
CONSTRAINED OPTIMAL NEIGHBORHOODS AND
KERNEL ESTIMATORS AS IMPROVEMENTS
TO APPLICATIONS OF KRIGING

by

Marla Marie Moody

A Dissertation submitted to the Faculty of the
COMMITTEE ON APPLIED MATHEMATICS (GRADUATE)
In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY
In the Graduate College
THE UNIVERSITY OF ARIZONA

1993
As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Marla Marie Moody entitled Constrained Optimal Neighborhoods and Kernel Estimators as Improvements to the Applications of Kriging and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Donald E. Myers

George Lamb

Arthur Warrick

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copy of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Dissertation Director (Donald E. Myers)
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Maria Marie Meeley
ACKNOWLEDGMENTS

Many people have contributed their efforts to this work and I thank them all. Several individuals stand out among these people. Foremost is my major professor Dr. Donald E. Myers whose constancy and support were an inspiration during the difficult times of this project. The members of my committee who assisted in the overview of this dissertation are Dr. George Lamb and Dr. Arthur Warrick. Many fellow students attended presentations and contributed to the discussion of issues which revealed new insights, especially: Hannah Rasmussen Rhodes, Jean Merilan, and Andrew Long. Special thanks go to Lois Gorski for all the friendly assistance she gave me while I was at the U. of A. Dr. Mario Rossi sent me the Robena coal mine thickness data on a diskette. The Earth and Environmental Division at Los Alamos National Laboratory supported this work. Several people there were very helpful, especially: Greg Cole, Pat Max, Bob Greene, and Marcia Jones who helped me get into UNIX after many years of DOS; Doug Walthers who digitized the horizon depth data; but, the most abiding was Barbara Hahn who managed to accomplish the difficult task of pulling all of the different pieces together into a coherent document.
DEDICATION

This dissertation is dedicated to my family for the love and support they have given me throughout my life: to my father Alfred for dreaming that I would complete my education; to the memory of my mother Cleo who sacrificed so that I could; to the memories of my grandparents who adored me; and to my many loving aunts and uncles who helped raise me and my son; to my son Eric who helps me keep my perspective; to all my cousins who keep the sense of family strong; and to Arlette who made it all worthwhile.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF ILLUSTRATIONS</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>18</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>19</td>
</tr>
<tr>
<td>CHAPTER 1. BACKGROUND</td>
<td>20</td>
</tr>
<tr>
<td>1.1 Dissertation Objectives</td>
<td>20</td>
</tr>
<tr>
<td>1.2 Literature Review of Kriging Estimators</td>
<td>22</td>
</tr>
<tr>
<td>1.3 Literature Review of Kernel Estimators</td>
<td>27</td>
</tr>
<tr>
<td>1.4 Neighborhoods in Kriging</td>
<td>30</td>
</tr>
<tr>
<td>CHAPTER 2. KRIGING AND KERNEL ESTIMATION</td>
<td>32</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>32</td>
</tr>
<tr>
<td>2.1.1 Kriging Estimators</td>
<td>35</td>
</tr>
<tr>
<td>2.1.2 Kernel Estimators</td>
<td>42</td>
</tr>
<tr>
<td>2.2 Kernel Estimation as an Auxiliary to Kriging</td>
<td>48</td>
</tr>
<tr>
<td>2.2.1 Experimental Variogram Calculation</td>
<td>49</td>
</tr>
<tr>
<td>2.2.2 Parameter Interpolation</td>
<td>51</td>
</tr>
<tr>
<td>2.3 Kernel Estimation as an Alternative to Kriging</td>
<td>51</td>
</tr>
<tr>
<td>2.3.1 Kernel Estimation</td>
<td>52</td>
</tr>
<tr>
<td>2.3.2 Confidence Intervals</td>
<td>55</td>
</tr>
<tr>
<td>CHAPTER 3. CONSTRAINED OPTIMAL NEIGHBORHOODS IN KRIGING</td>
<td>57</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>57</td>
</tr>
<tr>
<td>3.2 Neighborhood Definition and Parameters</td>
<td>59</td>
</tr>
<tr>
<td>3.3 Neighborhood Optimization</td>
<td>60</td>
</tr>
<tr>
<td>3.4 Interpolation of Neighborhood Parameters</td>
<td>67</td>
</tr>
<tr>
<td>3.5 Neighborhood Selection Protocol</td>
<td>68</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS — continued

<table>
<thead>
<tr>
<th>CHAPTER 4. VERIFICATION WITH SPECIFIC DATA SETS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>70</td>
</tr>
<tr>
<td>4.2 Study 1: Neighborhood Selection Using Standard Experimental Variograms for Reference and Test Data Sets</td>
<td>72</td>
</tr>
<tr>
<td>4.2.1 Candidate Neighborhoods for Wolfcamp Aquifer and Cadmium</td>
<td>74</td>
</tr>
<tr>
<td>4.2.2 Wolfcamp Aquifer Variograms and Cross Validation</td>
<td>81</td>
</tr>
<tr>
<td>4.2.3 Cadmium Variograms and Cross Validation</td>
<td>96</td>
</tr>
<tr>
<td>4.3 Study 2: Neighborhood Selection Using Kernel Estimated Experimental Variograms for Reference and Test Data Sets</td>
<td>107</td>
</tr>
<tr>
<td>4.3.1 Wolfcamp Aquifer Variograms and Cross Validation</td>
<td>109</td>
</tr>
<tr>
<td>4.3.2 Cadmium Variograms and Cross Validation</td>
<td>112</td>
</tr>
<tr>
<td>4.4 Study 3: Kernel Estimation of Reference and Test Data Sets</td>
<td>117</td>
</tr>
<tr>
<td>4.4.1 Wolfcamp Aquifer Analyses</td>
<td>121</td>
</tr>
<tr>
<td>4.4.2 Cadmium Analyses</td>
<td>124</td>
</tr>
<tr>
<td>4.5 Study 4: Comparing Kriging with Constrained Optimal Neighborhoods and Kernel Estimated Experimental Variograms to Universal Kriging</td>
<td>130</td>
</tr>
<tr>
<td>4.5.1 Robena Thickness Analyses</td>
<td>130</td>
</tr>
<tr>
<td>4.5.2 Robena Ash Analyses</td>
<td>134</td>
</tr>
<tr>
<td>4.5.3 Horizon Depth Analyses</td>
<td>144</td>
</tr>
<tr>
<td>4.5.4 Earthquake Acceleration Analyses</td>
<td>154</td>
</tr>
<tr>
<td>4.6 Study 5: Confidence Limits for Kernel Estimation of Horizon Depth Data Set</td>
<td>158</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 5. CONCLUSIONS</th>
<th>163</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>163</td>
</tr>
<tr>
<td>5.2 Constrained Optimal Neighborhoods</td>
<td>166</td>
</tr>
<tr>
<td>5.3 Kernel Estimated Experimental Variograms</td>
<td>165</td>
</tr>
<tr>
<td>5.4 Kernel Estimation as an Alternative to Kriging</td>
<td>166</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS — Continued

<table>
<thead>
<tr>
<th>CHAPTER 6. FUTURE WORK</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Improved Parameters for Kernel Estimators</td>
<td>168</td>
</tr>
<tr>
<td>6.2 Automation</td>
<td>168</td>
</tr>
<tr>
<td>6.3 Combined Kriging and Kernel Estimators</td>
<td>168</td>
</tr>
<tr>
<td>6.3.1 Profiles in Space</td>
<td>169</td>
</tr>
<tr>
<td>6.3.2 Profiles in Time</td>
<td>169</td>
</tr>
<tr>
<td>6.4 Extension to Cokriging</td>
<td>170</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>APPENDIX A - DATA SETS, HISTOGRAMS, AND LOCATION MAPS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 Wolfcamp Aquifer Data Set</td>
<td>171</td>
</tr>
<tr>
<td>A.2 Cadmium Data Set</td>
<td>176</td>
</tr>
<tr>
<td>A.3 Robena Data Set</td>
<td>180</td>
</tr>
<tr>
<td>A.4 Horizon Depth Data Set</td>
<td>189</td>
</tr>
<tr>
<td>A.5 Earthquake Acceleration Data Set</td>
<td>194</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>APPENDIX B - PROGRAMS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1 Neighborhood Optimization Routines</td>
<td>199</td>
</tr>
<tr>
<td>B.2 Variogram Routines</td>
<td>235</td>
</tr>
<tr>
<td>B.3 Kriging Routines</td>
<td>275</td>
</tr>
<tr>
<td>B.4 Auxiliary Routines</td>
<td>323</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REFERENCES</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>342</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.</td>
<td>Neighborhood selection and definition from data points in two dimensions</td>
<td>64</td>
</tr>
<tr>
<td>3.2.</td>
<td>Neighborhood parameters from three data points in two dimensions</td>
<td>65</td>
</tr>
<tr>
<td>4.1.</td>
<td>Various standard neighborhood experimental variograms for circles of 50, 75, and 100 distance units; solid, long dash and short dash respectively; Wolfcamp aquifer</td>
<td>83</td>
</tr>
<tr>
<td>4.2.</td>
<td>Various standard neighborhood experimental variograms for ellipses with major axis to minor axis ratio 100/50 in distance units for 90, 135, and 180 degrees orientation of major axis; solid, long dash and short dash respectively; Wolfcamp aquifer</td>
<td>84</td>
</tr>
<tr>
<td>4.3.</td>
<td>Experimental variograms for various fixed N neighborhoods and global neighborhood for 15, 20, 25, 30, and 82 neighbors; dot, dot-dash, dash, solid and dash respectively; Wolfcamp aquifer</td>
<td>85</td>
</tr>
<tr>
<td>4.4.</td>
<td>Experimental variograms for various limited variance neighborhoods and global neighborhood for 5%, 10%, 15%, 20%, and 100% of the data variance; dot, dot-dash, dash, solid, and dash respectively; Wolfcamp aquifer</td>
<td>86</td>
</tr>
<tr>
<td>4.5.</td>
<td>Experimental variograms for various data set combinations with limited variance neighborhoods: full data set and 7.5% neighborhoods, full data set and 10% neighborhoods, deleted data sets and 7.5% neighborhoods, deleted data set and 7.5% limited variance neighborhoods; dot-dash, solid, dot, and dash respectively; Wolfcamp aquifer</td>
<td>87</td>
</tr>
<tr>
<td>4.6.</td>
<td>Experimental variograms for various maximum number of points: 55, 60, 70, and 82 nearest neighbors used in search; dot, dot-dash, dash, solid respectively; Wolfcamp aquifer</td>
<td>88</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.7</td>
<td>Final experimental variogram compared to other variograms: final variogram, linear residual variogram, quadratic residual variogram, and global variogram; solid, lower dash, upper dash, dot, respectively; Wolfcamp aquifer</td>
<td>91</td>
</tr>
<tr>
<td>4.8</td>
<td>Final experimental variogram of 15% limited variance neighborhoods with different models for 5, 6, 7, and 8 fitted points counted from the origin; lower dash, upper dash, solid, dot-Dash respectively; Wolfcamp aquifer</td>
<td>93</td>
</tr>
<tr>
<td>4.9</td>
<td>Estimated values from simple kriging with variogram model of 8000 nugget, 28000 sill, 78.75 range and 15% limited variance neighborhood means versus data values; Wolfcamp aquifer</td>
<td>95</td>
</tr>
<tr>
<td>4.10</td>
<td>Experimental variograms for various fixed N neighborhoods and global neighborhood for 15, 30, 45, and 59 neighbors; dot, dot-dash, dash, and solid respectively; Cadmium data</td>
<td>97</td>
</tr>
<tr>
<td>4.11</td>
<td>Experimental variograms for various limited variance neighborhoods and global neighborhood for 25%, 50%, 75%, and 100% of the data variance; dot, dot-dash, dash, and solid respectively; Cadmium data</td>
<td>98</td>
</tr>
<tr>
<td>4.12</td>
<td>Experimental variograms for various data set combinations with limited variance neighborhoods: full data set and 50% neighborhoods, full data set and 75% neighborhoods, deleted data sets and 50% neighborhoods, deleted data set and 75% limited variance neighborhoods; dot, dot-dash, dash, and solid respectively; Cadmium data</td>
<td>99</td>
</tr>
<tr>
<td>4.13</td>
<td>Experimental variograms for various maximum number of points: 30, 45, 50, and 55 nearest neighbors used in search; dot, dot-dash, dash, and solid respectively; Cadmium data</td>
<td>100</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS — Continued

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.14.</td>
<td>Final experimental variogram of 90% limited variance neighborhoods with different models for 10, 11, 12, and 13 fitted points counted from the origin; dash, dot-dash, solid, and dot respectively; Cadmium data</td>
<td>101</td>
</tr>
<tr>
<td>4.15.</td>
<td>Estimated values from simple kriging with variogram model of 0.0 nugget, 10.9 sill, 61.0 range and 90% limited variance neighborhood means versus data values; Cadmium data</td>
<td>102</td>
</tr>
<tr>
<td>4.16.</td>
<td>Various kernel estimated experimental variograms for 50% neighborhood; global lambdas: 0.4 dot, 0.2 dot-dash, 0.1 short dash, 0.05 long dash; Wolfcamp aquifer</td>
<td>110</td>
</tr>
<tr>
<td>4.17.</td>
<td>Various kernel estimated experimental variograms for 15% neighborhood; global lambdas: 0.4 dot, 0.2 dot-dash, 0.1 short dash, 0.05 long dash; Wolfcamp aquifer</td>
<td>111</td>
</tr>
<tr>
<td>4.18.</td>
<td>Standard and kernel estimated experimental variograms 15% neighborhood: standard dot-dash, kernel estimated dash; Wolfcamp aquifer</td>
<td>113</td>
</tr>
<tr>
<td>4.19.</td>
<td>Undersmoothed experimental variogram 15% neighborhood; Wolfcamp aquifer</td>
<td>114</td>
</tr>
<tr>
<td>4.20.</td>
<td>Comparison of undersmoothed and smoothed experimental variograms; Wolfcamp aquifer</td>
<td>115</td>
</tr>
<tr>
<td>4.21.</td>
<td>Various kernel estimated experimental variograms for 75% neighborhood; global lambdas: 0.4 dot, 0.2 dot-dash, 0.1 short dash, 0.025 long dash; Cadmium data</td>
<td>116</td>
</tr>
<tr>
<td>4.22.</td>
<td>Standard and kernel estimated experimental variograms for 75% neighborhood; standard dot-dash, kernel estimated dash with global lambdas dash; Cadmium data</td>
<td>118</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>4.23.</td>
<td>Standard and kernel estimated experimental variograms for 75% neighborhood: standard dot-dash, kernel estimated with composite lambdas dash; Cadmium data</td>
<td>119</td>
</tr>
<tr>
<td>4.24.</td>
<td>Comparison of undersmoothed global and smoothed composite experimental variograms; Cadmium data</td>
<td>120</td>
</tr>
<tr>
<td>4.25.</td>
<td>Nearest neighbor estimates versus data values; Wolfcamp aquifer</td>
<td>122</td>
</tr>
<tr>
<td>4.26.</td>
<td>Isotropic two dimensional exponential kernel estimates versus data values; Wolfcamp aquifer</td>
<td>123</td>
</tr>
<tr>
<td>4.27.</td>
<td>Nearest neighbor estimates versus data values at 1.0 cut off; Cadmium data</td>
<td>125</td>
</tr>
<tr>
<td>4.28.</td>
<td>Nearest neighbor estimate versus data values at 2.0 cut off; Cadmium data</td>
<td>126</td>
</tr>
<tr>
<td>4.29.</td>
<td>Isotropic two dimensional exponential kernel estimates versus data values; Cadmium data</td>
<td>127</td>
</tr>
<tr>
<td>4.30.</td>
<td>Final experimental variogram and model for 90% limited variance neighborhood at 2.0 cut off; Cadmium data</td>
<td>128</td>
</tr>
<tr>
<td>4.31.</td>
<td>Estimated values from simple kriging for variogram model of 0.0 nugget, 10.4 sill, 68.0 range and 90% limited variance neighborhood means at 2.0 cut off versus data values; Cadmium data</td>
<td>129</td>
</tr>
<tr>
<td>4.32.</td>
<td>Estimated values from simple kriging for variogram model of 0.0 nugget, 10.4 sill, 68.0 range, and fixed N = 5 neighborhood means at 2.0 cut off versus data values; Cadmium data</td>
<td>131</td>
</tr>
<tr>
<td>4.33.</td>
<td>Estimated surface for kernel estimate; Wolfcamp aquifer</td>
<td>132</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>4.34.</td>
<td>Estimated surface from simple kriging for variogram model of 0.0 nugget, 10.4 sill, 68.0 range and 90% limited variance neighborhood means at 2.0 cutoff; Cadmium data</td>
<td>133</td>
</tr>
<tr>
<td>4.35.</td>
<td>Estimated surface from simple kriging for variogram model of 0.0 nugget, 10.4 sill, 68.0 range and fixed N = 5 neighborhood means at 2.0 cut off; Cadmium data</td>
<td>134</td>
</tr>
<tr>
<td>4.36.</td>
<td>Various kernel estimated experimental variograms for 50% limited variance neighborhoods; global lambdas: 0.01 dot, 0.005 dot-dash, 0.001 short dash, 0.005 long dash; Robena thickness</td>
<td>136</td>
</tr>
<tr>
<td>4.37.</td>
<td>Kernel estimated experimental variograms for various limited variance neighborhoods with global lambda 0.001: 45% dot, 40% dot-dash, 35% short dash, 30% long dash; Robena thickness</td>
<td>137</td>
</tr>
<tr>
<td>4.38.</td>
<td>Final experimental variogram and model for entire data set; Robena thickness</td>
<td>138</td>
</tr>
<tr>
<td>4.39.</td>
<td>Estimated values from simple kriging with variogram model of 0.0 nugget, 0.141 sill, 4950 range and 35% limited variance neighborhood means with composite smoothing parameter using the data working set in cross validation versus data values; Robena thickness</td>
<td>140</td>
</tr>
<tr>
<td>4.40.</td>
<td>Final experimental variogram and model for working data set; Robena thickness</td>
<td>141</td>
</tr>
<tr>
<td>4.41.</td>
<td>Estimated values from simple kriging with variogram model of 0.0 nugget, 0.141 sill, 4950 range and 35% limited variance neighborhood means with composite smoothing parameter using the data working set to predict reference set versus data values; Robena thickness</td>
<td>142</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>4.42.</td>
<td>Estimated surface from simple kriging with variogram model of 0.0 nugget, 0.141 sill, 4950 range and 35% neighborhood means; Robena thickness</td>
<td></td>
</tr>
<tr>
<td>4.43.</td>
<td>Kernel estimated experimental variograms for various neighborhoods with global lambda 0.001: 50% dot, 45% dot-dash, 40% dash; Robena ash</td>
<td></td>
</tr>
<tr>
<td>4.44.</td>
<td>Final experimental variogram and model for working data set with 47.5% limited variance neighborhoods; Robena ash</td>
<td></td>
</tr>
<tr>
<td>4.45.</td>
<td>Estimated values from simple kriging with variogram model of 0.0 nugget, 0.969 sill, 4700 range and 47.5% limited variance neighborhood means using the working data set to validate reference data set versus data values; Robena ash</td>
<td></td>
</tr>
<tr>
<td>4.46.</td>
<td>Estimated surface from simple kriging with variogram model of 0.0 nugget, 0.969 sill, 4700 range and 47.5% limited variance neighborhood means; Robena ash</td>
<td></td>
</tr>
<tr>
<td>4.47.</td>
<td>Kernel estimated experimental variograms for various neighborhoods with global lambda 0.001: 50% dot, 30% dot-dash, 20% short dash, 10% long dash; Horizon depth</td>
<td></td>
</tr>
<tr>
<td>4.48.</td>
<td>Final experimental variogram and model for data working set with 15% limited variance neighborhoods; Horizon depth</td>
<td></td>
</tr>
<tr>
<td>4.49.</td>
<td>Estimated values from simple kriging with variogram model of 0.00539 nugget, 0.0366 sill, 5500 range and 15% limited variance neighborhood means for the data working set versus data values; Horizon depth</td>
<td></td>
</tr>
<tr>
<td>4.50.</td>
<td>Estimated south half surface from simple kriging with variogram model of 0.00539 nugget, 0.0366 sill, 5500 range and using 15% limited variance neighborhood means; Horizon depth</td>
<td></td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS — Continued

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.51. Nearest neighbor estimates versus data values; Horizon depth</td>
<td>155</td>
</tr>
<tr>
<td>4.52. Isotropic two dimensional exponential kernel estimates versus data values; Horizon depth</td>
<td>156</td>
</tr>
<tr>
<td>4.53. Kernel estimated experimental variograms for various neighborhoods with global lambda 1.0: 90% Dot, 75% dot-dash, 50% dash; Earthquake acceleration</td>
<td>157</td>
</tr>
<tr>
<td>4.54. Estimated surface from simple kriging with variogram model of 745 nugget, 614 sill, 8.7 range and 75% limited variance neighborhood means versus data values; Earthquake acceleration</td>
<td>159</td>
</tr>
<tr>
<td>4.55. Root mean square error of reference set validation versus effective neighborhood radius using kernel estimator with confidence limit; Horizon depth</td>
<td>160</td>
</tr>
<tr>
<td>4.56. Profile of kernel estimates and limits plus nearest neighbor estimates versus distance from point 47 to point 19; solid, dot-dash, and dot respectively; Horizon depth</td>
<td>161</td>
</tr>
<tr>
<td>4.57. Profile of kriged and kernel estimates from point 47 to point 19; solid and dash respectively; Horizon depth</td>
<td>162</td>
</tr>
<tr>
<td>A.1.1. Histogram for Wolfcamp aquifer data</td>
<td>174</td>
</tr>
<tr>
<td>A.1.2. Location map for Wolfcamp aquifer data</td>
<td>175</td>
</tr>
<tr>
<td>A.2.1. Histogram for Cadmium data</td>
<td>178</td>
</tr>
<tr>
<td>A.2.2. Location map for Cadmium data</td>
<td>179</td>
</tr>
<tr>
<td>A.3.1. Histogram for Robena thickness data</td>
<td>186</td>
</tr>
<tr>
<td>A.3.2. Histogram for Robena ash data</td>
<td>187</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>A.3.3.</td>
<td>Location map for Robena data</td>
</tr>
<tr>
<td>A.4.1.</td>
<td>Histogram for Horizon depth data</td>
</tr>
<tr>
<td>A.4.2.</td>
<td>Location map for Horizon depth working data</td>
</tr>
<tr>
<td>A.4.3.</td>
<td>Location map for Horizon depth reference data</td>
</tr>
<tr>
<td>A.5.1.</td>
<td>Histogram for Earthquake acceleration data</td>
</tr>
<tr>
<td>A.5.2.</td>
<td>Location map for Earthquake acceleration data</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1. Cadmium neighborhood study for fixed N neighborhoods and the optimum neighborhood</td>
<td>75</td>
</tr>
<tr>
<td>4.2. Wolfcamp aquifer neighborhood study for fixed N neighborhoods and the optimum neighborhood</td>
<td>75</td>
</tr>
<tr>
<td>4.3. Cadmium neighborhood study for different limiting neighborhood variances</td>
<td>76</td>
</tr>
<tr>
<td>4.4. Wolfcamp aquifer neighborhood study for different limiting neighborhood variances</td>
<td>76</td>
</tr>
<tr>
<td>4.5. Cadmium neighborhood comparison between deleted and original data sets for candidate experimental variograms</td>
<td>77</td>
</tr>
<tr>
<td>4.6. Wolfcamp aquifer neighborhood comparison between deleted and original data sets for candidate experimental variograms</td>
<td>77</td>
</tr>
<tr>
<td>4.7. Cadmium 75% limited variance neighborhood study for different maximum number of points</td>
<td>78</td>
</tr>
<tr>
<td>4.8. Wolfcamp Aquifer 10% limited variance neighborhood study for different maximum number of points</td>
<td>78</td>
</tr>
<tr>
<td>4.9. Cross validation comparison of simple kriging with neighborhood means and ordinary kriging, and weight of the mean review for 7 variogram points fit to a model for the Wolfcamp aquifer</td>
<td>94</td>
</tr>
<tr>
<td>4.10. Cross validation comparison of simple kriging with neighborhood means and ordinary kriging, and weight of the mean review for 15% limited variance neighborhoods for the Wolfcamp aquifer</td>
<td>94</td>
</tr>
</tbody>
</table>
### LIST OF TABLES — Continued

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Variogram model parameters calculated with programs</td>
<td>167</td>
</tr>
<tr>
<td>5.2</td>
<td>Variogram model parameters from other authors</td>
<td>167</td>
</tr>
</tbody>
</table>
ABSTRACT

The motivation for this dissertation is to develop innovations in spatial, environmental data analyses, using kriging and kernel estimation, that form a basis for an eventual automation of the calculations. Special consideration should be given to the different requirements for environmental data as compared to the mining data generally used in the evaluation of kriging applications. It is common to use standard search neighborhoods in the applications of kriging. It is one object of this dissertation to develop variable search neighborhoods and to extend the use of these search neighborhoods to experimental variogram calculations. Other objectives include incorporating one dimensional kernel estimation into variogram calculation; and augmenting kriging with two and three dimensional kernel estimators. These three different areas require the development of programs to accomplish the following:

1. Generate elliptical neighborhoods with variable parameters in two dimensions and ellipsoidal neighborhoods with variable parameters in three dimensions; and calculate experimental variograms using these neighborhoods to limit the number of data pairs used and thereby reduce the effects of drift.

2. Calculate experimental variograms with a one dimensional kernel to separate the bin width from the number of points which is not possible with the standard experimental variogram.

3. Use two or three dimensional kernel estimators to provide an alternate to kriging.
CHAPTER 1
BACKGROUND

1.1 Dissertation Objectives

Using limited amounts of spatial data in the geosciences to generate a completed surface for further analysis has shifted in the last 20 years from parametric trend surface analysis to nonparametric kriging which is based on the theory of random functions. During this time many different kriging estimators have been developed for a wide variety of assumptions. In addition other estimators have begun to be used. With the large number of different estimators to choose from for spatial data, and the explosion in environmental data, geo-environmental scientists are facing a dilemma in evaluating the volumes of data being generated. The goal of this dissertation is to reduce these computational burdens through data preprocessing and the assistance of kernel estimators, which will allow the simpler kriging estimators to be used locally for almost all cases.

The objectives of this work are to develop a coherent package of kriging and auxiliary programs (listed in Appendix B) and data handling procedures (or protocols) that: reduce the effects of drift, are amenable to automation, and allow for the computation of confidence intervals as well as estimates. In this approach the data are used as much as possible to drive the analyses. The final system should be able to allow for a progression of analyses from auxiliary programs through kernel approximations to standard kriging using simple kriging with a known mean, ordinary
kriging with an unknown mean, or in special cases, a universal kriging program to
account for drift. An eventual extension to multivariate kriging (cokriging) is also
anticipated.

This dissertation contains only the fundamental parts that are considered
necessary for this approach to be implemented. In order for this approach to be
possible, it is necessary to improve and automate neighborhood definitions and utilize
an auxiliary estimator - kernel estimation - for fast preprocessing of the data and as
an auxiliary interpolator. When drift is considered to be present, it may be beneficial
to develop a data based evaluation of different methods of estimation since no
kriging estimator provides confidence intervals when drift is present.

Specific objectives are given in the following list:

Use constrained optimal neighborhoods to:

- Reduce effects of drift
- Krige locally varying neighborhoods
- Reduce size of pair comparison file.

Use kernel functions to:

- Calculate a new type of experimental variogram that allows for more plotted
  points to be calculated without reducing the number of pairs per plotted
  point.

Extend kernel estimation to 2-D or 3-D to:

- Interpolate neighborhood or other parameters during data processing
• Use as alternative to kriging for rapid analysis or to provide approximate confidence intervals, especially for data sets exhibiting the effects of drift.

1.2 Literature Review of Kriging Estimators

Geostatistics has evolved in the last few decades from the mining oriented method of ore evaluation to the generic spatial evaluation tool used today. Several kriging estimators have been derived to satisfy different mathematical conditions and assumptions. The most elementary ones, described in detail in section 2.1.1, are simple kriging, ordinary kriging and universal kriging. These univariate forms have been extended to the multivariate case called cokriging, with universal cokriging being the most general form used for program development. Only these elementary kriging estimators are considered since this work provides aids that enhance the utility of these elementary estimators. Myers [1989] presents background information concerning the assumptions used in the derivation of these kriging estimators.

All these kriging estimators require knowledge of either the covariance, if it exists, or the variogram to quantify the spatial structure for the phenomenon in order to create the system of equations unique to each kriging estimator. (The variogram is preferred since it is more general and does not require estimating the mean.) The solution of these equations at each interpolation point is based on the data and the variogram model.

In general the variogram is estimated by an experimental variogram calculated from the pairs within the spatial data set. Basic references for variograms and
models include those by Delhomme [1978], Journel and Huijbregts [1978], Myers [1991] and Barnes [1991]. Starks and Fang [1982] describe the effect of drift on the experimental variogram. Additional material concerning the allowable models for variograms and covariances that keep the variance nonnegative is in a paper by Christakos [1984]. These models and experimental calculations are described in detail in Section 2.1.1.

Aside from the increased level of mathematics involved in universal kriging, there is a real problem separating the deterministic drift component from the variogram. Some authors such as Armstrong [1984] still contest its utility and support a more advanced kriging estimator developed by Matheron [1973] and Delfiner [1976], to replace universal kriging, called intrinsic random functions of order k (IRF-k). Other authors such as Journel [1986] support the general use of ordinary kriging and recommend universal kriging only for extrapolation. Unfortunately there are few books on the subject of kriging and the one cited most: Mining Geostatistics by Journel and Huijbregts [1978] contains considerable mining terminology and is primarily concerned with problems unique to mining. Cressie [1990] published a history of the development of kriging in which he traces the origins back to the 1930's and early 1940's when Weiner, Kolmogorov and Wold were studying estimation of stationary time series. While Krige extended the application to the mining industry in the early 1950's he did not utilize the method to its fullest extent. In the early 1960's Matheron put it all together in the form known today and named it after Krige
for his mining application. Much of Matheron’s work, in French, remained unknown in this country throughout the 1960’s with one exception where Matheron [1963] published a paper on the principles of geostatistics. This work did not spark interest in the subject and it was not until Watson [1971, 1972] published explanations of Matheron’s work under the subject of trend surface analysis that kriging began to be used by earth scientists in lieu of the least squares trend surface analysis common at that time. Torelli and Tomasi [1977] compared kriging to trend surface analysis but referred to it as interpolation. But Kriging along with its name became accepted in the 1970’s and applications and new developments flourished. Finally Marcotte and David [1988] showed that trend surface analysis is a special case of IRF-k kriging except at the data points where the kriging estimator is exact.

Several authors have considered the best kriging estimator to use for a particular application or even propose another type of estimator for specific purposes. Some of this debate can be attributed to the different perspectives of mining engineering and earth sciences. Recent emphasis on environmental problems has created a new orientation for geostatistics that is not the same as that in the mining industry. In the mining industry, average values and error reduction are of greater interest than extreme values. In the environmental and many other earth sciences, extreme values are of paramount importance. Dubrule [1984] compared the spline generalized covariance to the variogram model version of kriging and considered the spline to be appropriate for general geological mapping. Boufassa
and Armstrong [1989] compared the use of kriging estimators and concluded that ordinary kriging should be used instead of simple kriging for ore deposits. Weber and Englund [1992] compared different types of spatial interpolators and also found that ordinary kriging performed slightly better than simple kriging. While comparisons are not a major emphasis of this work, the kriging results of several of these authors are used to compare the results of this work in Chapter 4.

One of the practical problems associated with kriging is that of choosing neighborhoods. Since kriging requires a system of linear equations be solved for every interpolation point, it is common practice to restrict the number of points used in some manner. There are differing opinions concerning this practice, especially in the mining industry where reduction in error is the most important issue. However in other applications where other issues such as preservation of the extremes are more critical, the use of neighborhoods to limit the system of equations is accepted more readily. Rivoirard [1984, 1987] is one of the few authors to consider neighborhoods in general. Most other authors such as Dubrule [1983] and Davis and Grivet [1984] assume that the largest neighborhood possible, the global or unique neighborhood, is the best. In Chapter 3 the general neighborhood concept will be refined and extended to variogram calculations. Other practical problems such as the solutions of linear equations, not considered in this work are discussed by David [1976].

As theoretical developments continued, applications expanded rapidly in many fields, especially in the hydrological sciences and many attempted modifications to
avoid universal kriging. In the late 1970's, Gambolati and Volpi [1979] used kriging to supplement trend surface analysis of water levels in Venice by kriging the residuals from a least squares trend to adjust the trend for residual spatial correlation. Neuman and Jacobson [1984] noted that using these two methods together required conflicting assumptions. They developed an iterative regression procedure to estimate the drift and then the variogram that had consistent assumptions but still did not require universal kriging; however, Cressie [1987] showed that residual kriging produces biased estimates. Aboufarassi and Marino [1983] used universal kriging to predict groundwater levels in Morroco. Groundwater levels in the Wolfcamp aquifer of Texas will be evaluated using the techniques developed in Chapter 4 that are designed to reduce the need for universal kriging when drift is present.

While there are numerous applications, the only other ones considered herein are geological and environmental. Carr and Glass [1985] and Carr and Roberts [1989] used kriging to study earthquake data. Some of Carr's [1990] earthquake data will be reevaluated in Chapter 4.

Public domain programs to implement kriging have been developed by the USGS and the EPA. BLUEPACK is a commercial program based on the more advanced form of kriging: Intrinsic Random Functions of order k, not considered in this work. The GEOEAS program developed by Englund and Sparks [1990] for the EPA has replaced the STATPACK program developed by the USGS. A modified version of GEOEAS was used in this work. Individuals have published special
Kriging programs are also available in the literature. Most have been published in the journal Computers and Geosciences. For example, Carr [1990] published a universal kriging program there; Carr, Myers and Glass [1985] published a cokriging program in the same journal.

1.3 Literature Review of Kernel Estimators

Kernel estimation did not start with the name kernel estimation; however, it became popular as a nameless non-parametric estimator for density functions after Rosenblatt [1956] presented a class of weight functions applicable to density estimation with desirable properties that carry over to the estimate; and Whittle [1958] described an optimum weighting function for smoothing a probability density function. Parzen [1962] extended their work to include an estimate of the mode of the probability density function. Watson and Leadbetter [1963] discussed the properties of these estimators on the basis of their mean integrated squared errors. Watson [1964] extended the concept to general smooth regression analysis and noted that the weight functions could be considered to be Dirac delta functions. Woodroofe [1970] introduced a modified set of kernel functions to estimate densities. Rosenblatt [1971] extended the idea to the spectral estimation of stationary processes. Priestly and Chao [1972] derived an approximate mean and variance for the density estimates and Benedetti [1977] proposed an optimal choice of a weighting function which she called a kernel. In the same year Breiman et al. [1977] used the name kernel in the title of their paper "Variable Kernel Estimates of Multivariate
Densities* and Gasser and Muller [1979] used the name in a paper on curve estimation. The name has been used ever since for this class of estimators. Recent work has centered on finding optimal bandwidths for the kernel estimator.

While kernel estimates are used as smoothers rather than interpolators, the choice of a smoothing parameter or its reciprocal, called a bandwidth for one dimensional situations, determines the amount of smoothing. The one dimensional kernel can be used to improve the calculation of experimental variograms by the introduction of the smoothing parameter to reduce erratic behavior. For small values of these parameters in higher dimensions, kernel estimates can be used as an approximate functional form for the weights calculated in kriging. These ideas are discussed in sections 2.2.1 and 2.2.2 respectively.

Cross validation as a method of parameter selection is used in many different methods of analysis, and a form of cross validation or generalized cross validation is used as a method of determining optimal bandwidths in kernel estimation. Roughness penalties are an alternate parameter selection criteria proposed by Good and Gaskins [1971] and extended to the maximum penalized likelihood method by Silverman [1982]. Klonais [1984] described a class of maximum penalized likelihood estimators.

Muller [1987] presented an evaluation of kernel estimates in non-parametric regression for 1-D longitudinal data. He considered polynomial kernels over a finite interval to be the most flexible of the nonparametric estimators that he considered.
He used cross validation to determine globally optimal bandwidths. For this work simple kernels of infinite extent have been chosen for demonstration purposes. These include the exponential and gaussian kernels. The exponential is used for 2-D kernel estimation and the gaussian is used for 1-D experimental variogram calculation.

Some authors have proposed various methods of bandwidth variation. Among them: Abramson [1982] proposed using a bandwidth variation proportional to the inverse square root of the function being smoothed; Jones et al. [1991] proposed a simple root n bandwidth selector. For this work the cross validated bandwidth has been selected as the method of general bandwidth selection since it is the most widely used, practical method of bandwidth selection in spite of the lack of mathematical demonstrations concerning asymptotic properties such as consistency. Other authors including Gasser, et al. [1991] have proposed automatic selection of bandwidth parameters.

For special kernel estimation of the experimental variogram, the variation of the curve can be severe for certain models such as the spherical model. At this stage of development, visual evaluation is used to evaluate and accept a global smoothing parameter and a piecewise linear variation if the global parameter is not adequate. Time for finding better ways to evaluate the smoothing parameter was not available, although the approaches mentioned by Rice [1984] appeared interesting.
Only a few authors have compared kernel estimation to kriging. Yakowitz and Szidarovszky [1985] derived a data driven estimator of the expected squared error in their comparison of kriging and non-parametric estimation. Warrick et al. [1988] and Warrick et al. [1990] numerically compared kriging to several other types of estimators.

Parallel work with spline estimates for regression led to the unification of some ideas between splines and kernels. Silverman [1984, 1985] developed a specific kernel approximation to cubic spline estimation. This special kernel is shown to be a variable kernel. Breiman et al. [1977] and Terrell and Scott [1990] describe variable kernels of multivariate densities.

1.4 Neighborhoods in Kriging

The use of a neighborhood to limit the size of the kriging matrix is common. Present forms of neighborhood kriging only use data points near the estimation point to formulate the kriging system of equations. Generally some maximum number of nearest points or a maximum radius is specified to economize on the computational cost of solving the resulting system of equations. In GEOEAS neighborhoods can be subdivided into quadrants and so many points may be taken from one quadrant or a minimum number of quadrants with data may be required to limit the extent of extrapolation. For linear drift, the variogram can be calculated only in the direction perpendicular to the drift. This work will demonstrate that the neighborhood concept can be extended in two ways. First, kriging neighborhoods can be made more
adaptable by allowing for variable parameters. In this way the neighborhood will not only limit the size of the kriging matrix but also reduce the effect of drift. Second, neighborhoods can be used in the calculation of the variogram to reduce the variability of the calculations and allow for local definition of the parameters. Using neighborhoods for variogram calculations as well as kriging, will reduce the size of any pair comparison file, used in some programs such as GEOEAS, and allow for larger data sets to be processed. Different neighborhoods can be used for variogram calculation and kriging depending on the order of the drift present, and accuracy in cross validation.

Optimization calculations for neighborhood parameters have been selected based on reliability rather than computational efficiency; this aspect has been deemed to be more important in this application where eventual automation will be important for novice users who may be required to utilize kriging. A special approach has been developed to capitalize on the discrete nature of the data sets and eliminate the problems of multi-optimization.

Constraints are required to expand these optimal neighborhoods to a practical size since optimal neighborhoods are generally too small (3 or 4 points) to be effective. These constraints are discussed in Chapter 3.
CHAPTER 2.
KRIGING AND KERNEL ESTIMATION

2.1 Introduction

Kernel estimation has not been used in the earth sciences as a standard method; with kriging being established as the best linear unbiased estimator there has been no emphasis on kernel estimation. However, it is a major point of this dissertation that kernel estimation has a contribution to make for prediction in the earth sciences - especially in the environmental division where large amounts of data are being generated. Kernel estimation can be used as an independent or auxiliary estimator in kriging; or it can also be used in the calculation of the experimental variogram.

Both kriging and kernel estimation are linear methods of spatial prediction that use the data set as the basis of the prediction rather than an a priori function that is fit to the data in some way. Both have the same form:

\[ Z = \sum_{i=0}^{n} (W_i Z_i); \quad (Z_0 = 1.0) \]  

They differ in the manner in which the weights \( W_i \) are chosen. (\( W_0 \) is zero for kernel estimation and some kriging estimators). Kriging is cast in a probabilistic framework and kernel estimation is cast is a deterministic framework.

Kriging is traditionally the name given to the set of unbiased estimators derived by minimization of the estimation variance:
Min: \[ \text{Var}(Z - \hat{Z}) = E[(Z - E(\hat{Z}))^2] \] (2.2)

with or without constraints; for example, in simple kriging there is no constraint and in ordinary kriging there is a constraint.

Classical optimization theory is used to calculate the weights. The objective function is the variance plus a Lagrange multiplier times any constraint. The partial derivatives of the objective function with respect to the unknown parameters are set to zero to derive the kriging equations. When the solution is substituted into the general equation it is seen to relate this special case of minimum variance simply to the covariance in the case of simple kriging and to the variogram in the case of ordinary kriging.

For conditions where the appropriate stationarity assumptions appear to be satisfied the linear estimators are simple kriging, ordinary kriging and ordinary multivariate kriging or cokriging. When these stationarity assumptions appear not to be satisfied (drift is present) there is primarily universal kriging or IRF-k kriging and universal cokriging. This dissertation will only deal with linear estimators and how they can be utilized to the fullest, by data preprocessing and the inclusion of kernel estimation.

Since kernel estimation is cast in a deterministic setting the weights are not amenable to the same analysis as in kriging. These weights are calculated from a kernel weight function which is restricted to a class of functions with desirable properties. While there are several different approaches, the standard conditions are
that the weight function be a positive, symmetric, maximally centered delta function with a bandwidth parameter and the integral over all space be equal to one. The Gaussian density function with the variance as the bandwidth parameter is an example of a popular kernel.

As the bandwidth becomes arbitrarily small, the kernel estimator becomes a nearest neighbor interpolator. Under these conditions the technique produces a tesselation of the data space, where each tile has the value of its nearest neighbor. As the bandwidth is increased, kernel estimators become smoothers; and in the limit as the bandwidth becomes arbitrarily large, the estimate is the mean of the data values. While kriging weights can be negative, the delta functions used with kernel estimators produce only positive weights, Stakgold [1979].

In kernel estimation, since the kernels are always positive, requiring the weights to sum to 1.0 results in estimates that are always bounded by the minimum and maximum data values. By using the sum of the weights equal to a number slightly larger than 1.0 in kernel estimation, the range of the estimates could be extended to recapture the extremes.

An analog to drift is not considered in classical kernel estimation since the estimator can be made to follow a trend in interpolation with an appropriate choice of the bandwidth; however, kernel estimators do not follow a trend in extrapolation, rather they plateau at the boundary data values. The nature of the function naturally provides for a local neighborhood as a consequence of the bandwidth. Kriging is not
naturally limited in extent the way that kernel estimators are, so that a standard search neighborhood is generally used to limit the size of the solution matrix, and a maximum pair distance is used to limit the size of the pair comparison file.

2.1.1 Kriging Estimators

There are several different linear kriging estimators. The primary differences in the development of these kriging systems is the result of different assumptions concerning stationarity. Depending on the stationarity assumptions used, different systems of equations are derived to produce the various kriging estimators. Unfortunately there is no way to verify these assumptions because the data represents only one realization of the random function.

The strictest form of stationarity called strong or strict stationarity requires that the joint distribution of the random function be translation invariant. Nonlinear kriging estimation requires this assumption, but the linear systems considered herein do not require this severe a restriction.

Second order stationarity requires that the covariance exist or that the variogram have a sill. An even weaker assumption, that called the intrinsic hypothesis, requires that the first order differences in the random function be second order stationary. Only the intrinsic hypothesis is required for the linear systems. The absence of drift in the system is required for either of these hypothesis to be valid. When the data produce an experimental variogram with a limited growth or a slow
growth or at least a slow growth with a power less than 2.0, it is generally taken as an indication that at least the intrinsic hypothesis is valid.

If the spatial variable consists of two components that can be specified as a deterministic and a random part then the deterministic component is the drift. In this case, the residuals are assumed to obey at least the intrinsic hypothesis. Universal kriging requires this assumption. Universal kriging also requires that the form of the drift be known a priori so that the proper residuals may be obtained to calculate the experimental variogram. Often geohydrologic data with a regional gradient appears to satisfy this requirement.

If the drift is not severe, then some form of stationarity may be valid for local areas and this quasistationarity is often assumed to be valid for neighborhood or local kriging to be used with the desired form of kriging. Ideally, local second order stationarity would generate a covariance for a limited search neighborhood.

Depending on the stationarity assumptions used, different approaches can be taken to derive the alternate systems of equations that constitute the following different kriging estimators. The following five pages of development are derived from a synthesis of Journel [1989], Myers [1982], and Cressie, [1988]:

Simple kriging (SK) is the name applied to the linear estimator obtained from the \( n \) data points \( Z_i \) when the mean is a known constant \( m \). The weights are obtained when the estimation variance is minimized without constraints.
For unbiased estimation, the expected value of the estimate and the expected value of the function have to be the same. A constant term $W_0$ which is called a shift operator is required to obtain the correct result for simple kriging showing that simple kriging is a residual form of kriging, where the expected value is a known constant not included in the formulation of the kriging equations. With $[Z] = m$:

$$E[Z] = E[W_0 + \sum_{i=1}^{n} W_i Z_i] = W_0 + m \sum_{i=1}^{n} W_i$$  \hspace{1cm} (2.3)

Therefore:

$$W_0 + m \sum_{i=1}^{n} W_i = m; \quad W_0 = \left(1 - \sum_{i=1}^{n} W_i\right)$$  \hspace{1cm} (2.4)

is called the weight of the mean. As a result $W_0$ is the known mean minus the sum of the weights times the mean which allows for the expression of the simple kriging estimator as a residual estimator:

$$\hat{Z}_0 - m = \sum_{i=1}^{n} [W_i (Z_i - m)]$$  \hspace{1cm} (2.5)

The estimation variance is obtained from the error of estimation by substituting the above expression into the equation for the variance:
The resulting expression in terms of the covariance is thus:

\[ \sigma^2 = E \left[ Z_o(X) - \left( \sum_{i=1}^{n} (W_i(Z_i - m)) \right)^2 \right] \]  \hspace{1cm} (2.6)

minimizing this variance via classical unconstrained optimization provides the simple kriging equations. A linear system of equations is obtained via classical optimization of the variance which is a quadratic objective function. (This system of equations should be used for interpolation only when the mean is known; however, it is used later in this work with estimated means to help identify design neighborhoods.) The linear system of equations is:

\[
\begin{bmatrix}
C_{11} & C_{12} & \cdots & C_{1n} \\
C_{21} & C_{22} & \cdots & C_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
C_{n1} & \cdots & \cdots & C_{nn}
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_n
\end{bmatrix}
=
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_n
\end{bmatrix}
\begin{bmatrix}
C_{o1} \\
C_{o2} \\
\vdots \\
C_{on}
\end{bmatrix}
\]  \hspace{1cm} (2.8)

Substituting this solution to this system back into the estimation variance gives the simple kriging variance in terms of the covariance:

\[ \sigma^2_{sk} = C_{oo} - \sum_{i=1}^{n} [W_i C_{io}] \]  \hspace{1cm} (2.9)

(Note that if the sum of the weights were one, the variance would be equal to the sum of the weights times the variogram. This would be approximately true for some simple kriging systems when the sum of the weights happened to be close to 1.0).
Ordinary kriging is the name applied to the same linear estimator when the variance is minimized while requiring that the estimator be unbiased with an unknown mean value m:

$$E[Z_o] = m$$  \hspace{1cm} (2.10)

This means that the average error between the estimate and the true value is 0.0. In order to have an unbiased estimator, the expected value of the estimate has to equal the expected value of the function (again using the shift parameter $W_0$ for demonstration):

$$E[Z_o] = W_0 + \sum_{i=1}^{n} (W_i Z_i) = W_0 + m \sum_{i=1}^{n} (W_i)$$  \hspace{1cm} (2.11)

$$E[Z_o] = m$$

Therefore,

$$m = W_0 + m \sum_{i=1}^{n} (W_i); \quad W_0 = m \left(1 - \sum_{i=1}^{n} (W_i)\right)$$  \hspace{1cm} (2.12)

as in simple kriging, but since m is unknown, the only way the equation can be satisfied in general is for $W_0$ to be zero and the sum of the weights to be 1.0. So the form of the ordinary kriging estimator is expressed in terms of the data directly:

$$\hat{Z}_o = \sum_{i=1}^{n} W_i Z_i$$  \hspace{1cm} (2.13)

The estimation variance is obtained from the equation for the variance the same as in simple kriging with the same general result:
The constraint equation adds another term to the objective function and yields the Lagrange multiplier $\mu$ as part of the solution:

$$
\begin{bmatrix}
C_{11} & C_{12} & \ldots & C_{1n} & 1 & W_1 \\
C_{21} & C_{22} & \ldots & C_{2n} & 1 & W_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
C_{n1} & \ldots & \ldots & C_{nn} & 1 & W_n \\
1 & \ldots & \ldots & 1 & 0 & \mu
\end{bmatrix} \begin{bmatrix}
C_{01} \\
C_{02} \\
\vdots \\
C_{0n} \\
1
\end{bmatrix} = \begin{bmatrix}
i \\
i \\
i \\
i \\
i
\end{bmatrix}
$$

Substituting the solution to this system back into the estimation variance gives the simple form for the kriging variance:

$$
\sigma_k^2 = C_{oo} - \sum_{i=1}^{n} W_i C_{io} + \mu
$$

(2.16)

Since the sum of the weights is one, $C_{oo}$ can be brought inside the summation and the kriging variance can be expressed in terms of the variogram $\gamma$:

$$
\sigma_k^2 = \sum W_i \gamma_{io} + \mu
$$

(2.17)

The kriging variance lies between the nugget and the sill plus the Lagrange multiplier $\mu$, depending on location within the data field.

Universal kriging is an extension of ordinary kriging for systems with drift expressed as a linear combination of known basis functions. Matheron [1973] showed
that the estimator is unbiased whenever the following set of universality conditions are satisfied:

\[ \sum W_{fj}(x) = f(x), \text{ for } j = 0, 1, \ldots, k \]  

(2.18)

These (k+1) universality conditions, one for each basis function, are combined with additional Lagrange multipliers and added to the objective function. Taking partial derivatives of the new objective function with respect to the weights yields the universal kriging equations. Both the weights and the coefficients for the basic functions are determined in the solution of this kriging estimator. Unfortunately the indeterminacy of the variogram makes this system suspect per Armstrong [1984].

All kriging systems require a variogram or a set of variograms to build the defining set of equations. These variograms define the spatial correlation of the regionalized variable. Since this correlation is generally unknown a priori, the variogram is modeled using an experimental variogram that is calculated from the data pairs. (For each data pair \( Z_i - Z_j \) is plotted versus the distance between the pair.) This model is then used to construct the kriging equations. The structure of the underlying regionalized function affects the variogram in many ways. The spatial dimension, anisotropy, drift, and geometry of data locations affect the variogram.

The data sets used in this work are all in two spatial dimensions and anisotropy is not a problem. Several data sets exhibit the effects of drift. Calculating experimental variograms for these data sets is a primary objective of this work.
Frequently cross validation is used in kriging to verify that the model chosen accurately reproduces the data set. In cross validation, each data point is deleted in turn and estimated from the remaining n-1 data points. The mean squared error is given by:

$$MSE_{xy} = \frac{1}{n} \sum_{i=1}^{n} \left[ Z_i - \sum_{j=1}^{n} W_j Z_j \right]^2$$

(2.19)

Where $W_j$ is the weighting function.

2.1.2 Kernel Estimators

One-dimensional kernel estimation has been extensively used to generate nonparametric probability density functions Rosenblatt [1956], Whittle [1957], Parzen [1962], and Watson and Leadbetter [1963], and to display longitudinal medical data Muller [1988]. It has not been advocated for predicting two or three dimensional spatial data. Kriging or other analyses such as trend surfaces or spline functions are used for spatial data. One objective of this dissertation is to establish the value of kernel estimation for spatial prediction. The form of the estimator is very simple. It is the same linear estimator used in kriging where the weights are given by a kernel function:

$$Z_o = \sum_{i=1}^{n} W_i Z_i = \sum_{i=1}^{n} F(r_i,b) Z_i$$

(2.20)

where $F(r_i,b)$ is a kernel function of distance between interpolation point and the i data point and a bandwidth parameter b. The general expression for $F(r_i,b)$ is:
Classical kernel estimation requires the a priori selection of a kernel function to determine the weights in the linear estimator. These kernels are chosen from positive functions that have been converted to delta functions by the introduction of a bandwidth normalizer. The bandwidth which occurs naturally as a result of conversion to a delta function, is the parameter which scales the function in such a way that the area remains finite while the maximum point becomes unbounded in the limit as the parameter goes to zero. The potential for kernel functions is endless and there is generally no real reason to choose any given kernel over any other, except after the fact by saying that it works better for a given criterion on a particular data set.

The properties normally associated with the definition of a kernel function are attributed to Rosenblatt [1956]. The function $W_n$ is chosen to be a delta sequence so that as $n$ goes to infinity the weight is concentrated at the origin:

$$\int W_n(x)dx=1$$  \hspace{1cm} (2.22)

These functions are strictly positive:

$$W_n(x) \geq 0$$  \hspace{1cm} (2.23)

These functions are symmetric:
\[ \int x W_n(x) dx = 0 \quad (2.24) \]

These functions are also square integrable

\[ \int W_n^2(x) dx \leq \infty \quad (2.25) \]

With these properties these functions can be converted to delta sequences, Stakgold [1979]. The delta function representation allows for asymptotic analysis of convergence and error evaluation. By a coordinate conversion, a probability density function can be converted to a delta sequence. The normal density function is a delta sequence in the variance parameter. It is the most common example of a function that generates a delta function in the limit as the variance goes to zero. Yakowitz and Szidarovszky [1985] limited their work to the univariate normal pdf which can be used for isotropic 2-D and 3-D. But extending the technique to two dimensions in general requires a bivariate normal density function. Radially symmetric densities require only one or two parameters for optimization and are preferred for isotropic data sets. There are many families of probability densities to choose from for kernels. The most important point is that they are delta functions in the limit as the bandwidth parameter goes to zero. Kernels used in this work are symmetric and integrable functions; they include the Gaussian for 1-D experimental variogram calculation and exponential for 2-D interpolation.
Once the kernel function is selected, the choice of the bandwidth parameter or other parameters may be made on an a priori basis, or as in this work by cross validation. The selection of this bandwidth parameter (or its reciprocal, the smoothing parameter) is much more important than the selection of a particular kernel per a study by Marron [1988]. Several authors have studied the problems associated with automatic bandwidth selection, including Hardle, Hall and Marron [1988] and Chiu [1990, 1991]. Too small a bandwidth produces extraneous variation, and too large a bandwidth eliminates real variation. Some authors including Good and Gaskins [1971], Silverman [1982], and Klonais [1984] describe the use of a roughness penalty functional to determine goodness of fit. Others including Hardle and Marron [1985] describe the use of some form of cross validation to determine goodness of fit. In this work the smoothing parameter is chosen to minimize the mean squared error in cross validation. This approach was chosen because it could be used in both the experimental variogram calculation and direct 2-D or 3-D kernel estimation.

Variable bandwidth functions allow for the local adaptation of the kernel to different spatial densities of points. The function proposed by Breiman et al. [1977] appears to be suitable for potential extension of this work. It is suggested that the bandwidth should be a linear function of the mth nearest neighbor where m is not too small. The bandwidth is set to make the function practically 0.0 at the nth nearest neighbor where n is greater than m. Then the value of n is selected via some
goodness of fit criteria. This results in a discrete, single dimensional optimization. Future work would be required to evaluate this application. Unfortunately, Terrell and Scott [1982] do not recommend using this approach; they recommend a generalization of the square root law of Abramson [1982].

There are other methods of prediction that resemble kernel estimators, except that the weight functions are not kernels. Interpolating moving least squares (IMLS) are commonly used methods of interpolation. IMLS interpolators include the zero order inverse distance weighting (IDW) but offer other higher order functions besides IDW. The appeal that these interpolators have is their simplicity. However, these interpolators are only made exact by inducing a singularity at the data points, and these singularities produce poor surfaces according to Lancaster and Salkauskus [1986]. In many cases the analogous kernel estimator could be used to produce similar results. The kernel function is a delta sequence so the integral remains finite as the function increases indefinitely at the data point and is amenable to analyses.

IDW has been used to interpolate regional geochemical data, Kane et al. [1982] using the following - almost kernel- function:

\[
W_i = \begin{cases} 
0, & \text{if } r_i > C \\
 r_i^{-p}, & \text{if } r_i \leq C
\end{cases}
\] (2.26)

with \( r_i \) the distance from the point of interest to the data point being weighted. The parameter \( p \) was chosen via numerical minimization of the normalized least square function:
\[ S(p) = \frac{\sum_{i=1}^{n'} (Z_i - \bar{Z}(p))^2}{\sum_{i=1}^{n'} (Z_i - \bar{Z})^2} \]  

(2.27)

where \( \bar{Z} \) is the sample mean of validation sample and \( n' \) is the set of validation samples set aside from the \( n \) data samples. Integral values of \( p \) were chosen because the confidence intervals were large. (For isotropic kernel estimation the error expression is a single parameter function of the kernel.)

The most critical feature of this work was the construction of confidence intervals for the results. Approximate - in the sense that they are nonlinear - confidence intervals were given by:

\[ S(p)[1 + (n' - 1)^{-1} t_{n' - 1}(1 - \alpha/2)] \]  

(2.28)

where \( t_{n' - 1}(1 - \alpha/2) \) in the \( (1 - \alpha/2) \) percentile of a t-distribution with \( n' - 1 \) degrees of freedom. The same analysis applies to simple parameter kernel functions.

For arbitrarily small values of the bandwidth, kernel estimation reduces to nearest neighbor estimation with a mean squared error that is generally smaller than the data variance. As the bandwidth is increased, this error generally decreases to a minimum value and then increases again as the bandwidth becomes large enough so that the estimate is the mean value of the data and the mean squared error in cross validation becomes the variance of the data. While the nearest neighbor estimate is generally smaller than the data variance, for data sets where the nearest
neighbor estimate is larger than the data variance, the mean value produces the minimum cross validation mean square error estimate.

There is no analog to the kriging variance. The mean squared error of the estimation for the nearest neighbor limit can be visualized as increasing as a function of distance from the data points. As the bandwidth is increased to a moderate value, the error at each data point increases and the mean squared error of estimation as a function of distance from the data points decreases. At the large bandwidth limit the error of estimation is the constant data variance. Further work would be needed to attempt to quantify these mean squared errors or obtain satisfactory bounds for all values of bandwidth.

2.2 Kernel Estimation as an Auxiliary to Kriging

Kernel estimation can be used as an auxiliary function to calculate experimental variograms. A 1-D kernel function can be used to calculate experimental variograms that display more structure than can be obtained in the standard experimental variogram calculation, especially if the number of available pairs is restricted by neighborhoods. In addition, a 2-D or 3-D kernel can be used to interpolate neighborhood parameters from data points to interpolation points.

2.2.1 Experimental Variogram Calculation

In a standard experimental variogram calculation, data pairs are separated into direction windows and distances classes which are called bins. Then the averages of the half squared differences for the pairs in each bin are calculated and plotted
versus either the mean distance within the bin or the average bin distance. The resulting experimental variogram can be quite erratic depending on the window angles, the number of lags used for the distance classes and the total number of pairs available (there are \(N(N+1)/2\) potential pairs in each data set of size \(N\)). Using a kernel function to estimate the experimental variogram can give a smoother curve with fewer pairs because the smoothing parameter is independent of the number of points calculated. In standard calculations the bins are generally of constant size and have firm boundaries; each pair is counted in only one of the bins. In kernel interpolation each half squared difference of a data pair can be counted at a different rate at more than one point (or in more than one bin which produces fuzzy bin boundaries). While Goulard and Voltz [1992] describe the standard variogram and cross variogram as a special case of the kernel estimator in equation 2.20 with a rectangular kernel, unless a kernel with a bandwidth parameter is used, the smoothing properties required for the part of this work on experimental variograms are not available. (In this work neither standard window classes nor anisotropic kernels were required for experimental variogram calculations because the neighborhood calculations which are described in Chapter 3, basically perform this function when they are used in conjunction with experimental variogram calculations.

As the bandwidth is reduced from an arbitrarily large number, the kernel estimated experimental variogram will vary from a pure nugget effect to an exact interpolator of the variogram pair data. Neither of these two extremes are
desirable. What is desired is a smooth curve that identifies all the important features of the variogram, the nugget, the sill (if it exists) and the range. This determination is made visually.

The Gaussian function was chosen as the kernel function in variogram estimation because it can be made to approximate the standard bin with the appropriate choice of bandwidth. For the Gaussian form the bandwidth is the variance parameter.

There are two main considerations in kernel estimation of experimental variograms (aside from the choice of the kernel function). The first is the functional form of the bandwidth parameter, and the other is the criteria for selecting the best curve. A constant bandwidth would be preferred, but if a variable parameter is required, a piecewise linear function is chosen for the smoothing parameter variation. (The smoothing parameter was linearized rather than the bandwidth.) A variable bandwidth is analogous to variable size bins in the standard variogram calculation.

The first criteria used to select the experimental variogram as a function of bandwidth was the minimization of the successive difference in the experimental variogram values squared, while holding the nugget value at zero and the sill value at the maximum value. This approach is described in Rice [1985] as representing a smoothing criteria where successive differences are used as an approximation to the variance. (The nugget and sill were independently determined.) While this criteria works well for small numbers of calculated points it loses its stability and the method loses its power for a large numbers of points. Since a large number of points in the
experimental variogram is desired, a simple protocol described in Section 3.5 was developed using cross validation. The parameter or parameter function that minimizes the mean squared error in cross validation is accepted as the design parameter.

2.2.2 Parameter Interpolation.

In preprocessing the data, kernel estimators can be used to interpolate neighborhood parameters. For this study arbitrarily small values of bandwidths were used which result in nearest neighbor parameters being used. The general case would require further work.

For cases in 3-D where well-boring data is sampled frequently, but wells are spaced far apart, a combination of kriging and kernel estimation could give better results than either alone. Important data horizons such as maxima or minima could be kriged or cokriged and parameters for polynomials fitting the well-boring data between these horizons could be interpolated with kernel functions. Future work would be required for their development.

2.3 Kernel Estimation as an Alternative to Kriging

Kernel estimation is generally considered to be a data smoothing technique with little emphasis on variance reduction. The versatility of kernel estimation allows it to be used to reduce variance as an approximation to kriging depending on how the kernel and bandwidths are chosen. It is the purpose of this section to explore the value of kernel estimation as an alternative predictor.
2.3.1 Kernel Estimation

In general, kernel estimation is recommended to be used instead of kriging where calculation speed is emphasized over accuracy or when confidence intervals are needed for data sets exhibiting drift. The problem with universal kriging, noted by several authors, is that indeterminacy in evaluating the drift reduces the practical value of the best linear unbiased estimator. With kernel estimators on the other hand, drift is handled naturally with the appropriate choice of the bandwidth parameter. The major difference is extrapolation: the kernel estimator does not follow a drift function as the universal kriging estimator does. Rather it follows the last data point (last in the sense of nearest). So a kernel estimator plateaus in extrapolation and a universal kriging estimator continues to follow. The higher dimensional kernel functions are generally chosen from pdfs that are delta sequences in the variance parameter. The bivariate normal is the most common 2-D pdf and is used for anisotropic kernel function. The uniform density is used for the isotropic case.

In general, for kernel estimation the choice of the bandwidth is more critical than that of kernel function. Actually the bandwidth should be a function of the data density. Variable bandwidth functions allow for the local adaptation of the estimate. The function proposed by Breiman et al. [1977] appears to be suitable for further work. It suggests that the bandwidth should be a linear function of the mth nearest neighbor where m is not a small number. It could be used to set the kernel to zero.
at the mth nearest neighbor and the value of m selected from some goodness of fit
criteria.

The optimum parameters for each data set are obtained by the cross
validation of equation 2.19. The minimum squared error as a function of the parameters(s) provides the basis for selecting the optimum parameters for the given data set. Since the error is not an analytic expression the computation is intensive. However there is no system of linear equations to solve and the entire data set can be processed rapidly.

The primary approach in this work is to let the data generate the kernel function from the covariance of the data. Once the experimental variogram is calculated, a covariance function would be obtained from the best fitting variogram model. While these functions are more restricted than the classical kernel functions, the fact that fewer a priori choices have to be made is an advantage. By just using neighborhood kriging, it is hoped that a local covariance function can be found for each data set.

Within the set of classical kernel functions is a subset of functions that are variogram models. These particular variogram models can be converted to covariance models that can be used as kernels in kernel estimation. The advantage of using this class of functions is that the selection of the model is data driven. The selection of the best model from this restricted class of functions is the same as the
model that would be used in the kriging system of equations. Only the bandwidth needs to be evaluated.

For large values of the smoothing parameter, the mean squared error approaches the variance of data set; for small values of the parameter the mean squared error approaches the mean squared error of the nearest neighbor estimator. The optimum smoothing parameter is generally seen to be at the bottom of a moderately smooth depression in this curve for moderate values of the parameter. If the data set is too small to provide a good spatial correlation, there may not be a minimum in the error and the nearest neighbor mean squared error may exceed the data variance. A graph of the general type of curve is shown in Study 5, Section 4.5.

If trend is present in the data, the kernel method will follow the trend in interpolation but will not follow it in extrapolation as happens in universal kriging. In order to extrapolate with the kernel method, some form of independent regression would be required. There are several approaches. One is to fit a low order trend to the data, fit the parameter(s) to the residuals, and estimate the residuals. This is opposite to the method of UK where the variogram is fit with the residuals but the kriging is done with the original data.

Since the calculation is numerically intensive the optimization techniques need to be efficient. For isotropic data sets with single parameter kernels, there is not too much of a problem, but with multiple parameter kernels, the computation definitely requires a balance between searching for regions with local minima and determining
the values of these local minima. In this work, calculations are visually monitored to generate acceptable parameters. Further work is necessary to adapt optimization routines into the programs.

The spatial pattern of the data points is much more critical in kernel estimation than it is in kriging, in kriging, data clustering is accounted for automatically, while in kernel estimation some form of variable bandwidth adjustment is required to effectively estimate clustered data embedded in a region of sparse data. As contrasted to kriging, regularly gridded data is most desirable for global estimation of the kernel bandwidth.

Kernel estimation as described in the previous sections appears to be most analogous to ordinary kriging because of the normalization of the weights. The basic interpolation equations are the same, but the optimization method differs. The biggest difference is the interdata location interaction of kriging which is absent in kernel estimation. The results in most cases appear to be very similar. The main advantage is that a variogram is not required. The main disadvantages are that the technique is not as versatile with respect to error analysis or to special purpose applications such as indicator kriging.

2.3.2 Confidence Intervals

Kernel estimation is considered to be a non parametric technique used for smoothing data to an aesthetic judgment. However, it can be used to approximate several different other methods depending on the choice of the kernel function and
bandwidth. When a small bandwidth is used, it approximates trend surface analysis. The residuals can be seen to not exhibit spatial correlation - one of the criteria for statements about confidence regions for parameters. When these conditions are met, classical nonlinear regression techniques can be used to approximate confidence regions for kernel estimates similar to those obtained for IDW in Kane et al. [1982]. The only difference is that the class of weight functions is restricted to the class of kernel functions. So by using partitioned validation on a data set or neighborhood set, the kernel estimate can be considered as a nonlinear least squares estimator and approximate confidence intervals given per Section 2.1.2. The choice of the partitioning number \( n' \) is important; it is used in the confidence calculation. Several values of \( n' \) should be evaluated to minimize the confidence regions.
CHAPTER 3.

CONSTRAINED OPTIMAL NEIGHBORHOODS IN KRIGING

3.1 Introduction

The use of constrained optimal neighborhoods in kriging has not been investigated thoroughly nor has it been applied to kriging as a standard option available in any of the public domain programs. The number of points in a neighborhood used in kriging, both for experimental variogram calculation and interpolation is at present an arbitrary choice imposed by the structure of the program. In the case of the experimental variogram calculation, all pairs with separation distances below a specified interpair distance are included in the calculation. In the case of interpolation, the size of the kriging matrix can be limited to a maximum value in a few standard ways such as a set number of nearest neighbors.

In this work, finding the best neighborhood sizes for experimental variogram calculation is the primary goal. The main advantage of using constrained optimal neighborhoods in variogram calculation is that the effects of drift can be minimized almost all of the time. The effect on the experimental variogram when using these neighborhoods to limit the number of pairs is to reduce large values of squared differences. This is similar to what a trimmed mean would do. Another advantage of constrained optimal neighborhoods is that the pair comparison file, which normally increases quadratically with the size of the data set, can be reduced in size. By limiting the potential pairs to those in the optimal neighborhood the pair comparison
file can be made to grow linearly with the size of the data set. Since the data is preprocessed, the pair comparison file has the same format only it is shorter with a limited number of pairs.

In general, applying the neighborhoods used for experimental variogram calculation to kriging is recommended instead of using standard kriging neighborhoods. Certainly in the case of drift, limiting the kriging neighborhood to the same local neighborhood used for the variogram calculation makes sense. In the case of poor cross validation performance, kriging neighborhoods may be varied in an attempt to avoid the trend to the mean and loss of accuracy noted by Boufassa and Armstrong [1989] and Spease and Carr [1988]. It is possible that the additional dimension provided by an independent determination of constrained optimal neighborhoods will allow cross validation to be used to improve the accuracy of interpolation with minimal loss of precision and without serious edge effects. Some preliminary conclusions are available as a result of the case studies discussed in Chapter 4.

Some, such as Dubrule [1983] and Davis and Grivet [1984] recommend using the global neighborhood for kriging. Their research was based on improving the efficiency of the matrix calculations. Others such as O'Dowd [1991] demonstrated that larger kriging systems are less well conditioned. It appears that one way to evaluate this conflict in opinion is by the use of constrained optimal neighborhoods via cross validation.
The value of cross validation in kriging has been questioned by some researchers, particularly Solow [1990] who cautioned against relying on the results to verify adequacy of the kriging model. Actually he was cautioning against the associated criteria on reduced errors used in conjunction with cross validation. With the addition of neighborhood criteria within the kriging framework, an additional dimension is provided to assess accuracy as well as minimum variance and unbiasedness through the use of cross validation.

Neighborhoods are not important in kernel estimation since the shape of the kernel function provides a natural boundary in the process of weight determination. Even for kernels without unlimited spatial extent, an effective boundary is established when the assigned weights as a function of distance become negligible.

3.2 Neighborhood Definition and Parameters

Present neighborhood kriging practice limits the number of points used to generate the kriging equations at the interpolation point to a given number or to those points contained within a standard neighborhood. Most standard neighborhoods are circles or spheres with given radii. Some standard 3-D neighborhoods are ellipsoids with limited orientation options or elliptical cylinders.

In the GEOEAS program a search for the prescribed number of points within a standard volume is completed prior to building the reduced n' x n' matrix. If the right number, and possibly distribution, of points is found, the matrix is built and the system solved; if not, the point is flagged as unsolved. It is possible to improve the
versatility of these neighborhoods by allowing for local variation of parameters and allowing for complete freedom in the orientation of the neighborhood in space. In 2-D there are three parameters, the major and minor axis and the angle that the major axis makes with the x coordinate axis. In 3-D there are three axes and a full set of direction cosines for the neighborhood axes in space.

3.3 Neighborhood Optimization

There are many problems in defining optimal neighborhoods using conventional optimization techniques. The first is that neighborhoods beyond the isotropic ones used in standard neighborhood definitions, have several parameters, especially in 3-D and multidimensional optimization is tedious. In addition to the usual problems with multidimensional optimization, with the discrete data sets there are no unique parameters for a single neighborhood to enclose a given subset of data points.

After many attempts to find a reliable multidimensional optimization routine for determining unique ellipse and ellipsoid parameters that could be reproduced, a different approach based on the discrete nature of the problem was developed to provide a set of neighborhood parameters that could be reproduced by different users. This involved a series of data sortings to generate the desired neighborhood associations for the best neighborhood evaluation. The data points themselves are used in turn to define the ellipse or ellipsoid parameters. For isotropic systems, this requires calculating the radius of the potential neighborhoods as the distance from
the interpolation point to the data point. For \( n \) data points this requires \( n \) estimations. For anisotropic systems, more than one point is required to define a neighborhood. For 2-D systems, three points are required to define a unique neighborhood. If one point is in the center of the neighborhood, only two other points are required to define the neighborhood. For 3-D systems, four points, or three besides the central point are required to define an ellipsoid. Finally, the parameters that those points determine are determined at the end of the process.

Since the defining data points are located on the limiting ellipse or ellipsoid, they provide a unique set of parameters for each local neighborhood. Since the data points are discrete, the process is entirely reproducible if all the data points are included in the sorting process. If the data set is too large for the sorting processes to be accomplished in a timely manner, the maximum number of points included in the process is a parameter of the process.

The optimization of neighborhood parameters may be based on any of several criteria. Initially a cross validation with the mean or the mean squared value of the neighborhoods tested against the deleted, central value or its square was tried but it did not provide good neighborhoods for all data configurations. A series of calculations based on the variance of the points contained in each neighborhood performed much better. The centered point may be deleted or retained in this calculation. If deleted, it assists in evaluating potential outliers in neighborhood studies. If retained, it aids in finding the best neighborhood for variogram calculation.
Unconstrained optimal neighborhoods resulted in small neighborhoods - too small for experimental variogram calculation so constrained optimal neighborhoods were used to extend the size of the neighborhood while providing for limited variance in the selected data; this results in more stable andr experimental variograms. Because of the discrete nature of the calculations, these constraints can be incorporated into the calculation directly. So the first constrained optimization method examined, required the neighborhoods to have a given number of points and then found the neighborhood with the minimum variance:

$$\min \frac{1}{n} \sum_{i=1}^{n'} (Z_i - \bar{Z})^2$$  \hspace{1cm} (3.1)

subject to \( n' \) a given constant where \( n' \) is the number of points in the neighborhood and \( \bar{Z} \) is the neighborhood mean. These neighborhoods are called Fixed N neighborhoods.

The second constrained optimization method examined maximizes the number of points in the neighborhood for a fixed limiting neighborhood variance. \( \max n' \), subject to:

$$\frac{1}{n} \sum_{i=1}^{n} (Z_i - \bar{Z})^2 < L$$  \hspace{1cm} (3.2)

where \( L \) is a limiting value of the neighborhood variance. This last method gives the best experimental variograms and is the best compromise between fixing the number of reference points and limiting the variance. In particular it extends the maximum distance available for variogram lag calculations by allowing for large numbers of
points in favorable areas. These neighborhoods are called limited variance neighborhoods.

The design of the optimal neighborhood algorithm for both constraining methods is based on using a discrete number of data points to generate a series of data location sortings with increasing numbers of points within ellipses or ellipsoids. The ellipse or ellipsoid with the minimum neighborhood variance will be retained in an array of ellipse parameters; or a set of these ellipse parameters will be obtained for each data point.

The algorithm used sorts the data points by increasing distance from the data point selected as the neighborhood center and starts the evaluation process with the closest set of points needed to define a neighborhood. New points are added in order of distance to the central point as shown in Figure 3.1. As each next point out is added, it defines an isotropic subset with the minimum radius that contains the m nearest neighbors. Then the smaller anisotropic subsets contained within this nearest neighbor subset are evaluated for comparison. As each point is added, there are m-1 new anisotropic subsets to evaluate. This new point is used with all the other m-1 points individually in turn to define a set of ellipses. All the points that fall inside this each of these ellipses are used to calculate the neighborhood variance. The m point on the circle defines the major axis of the ellipses and each of other m-1 points in turn defines the minor axis as shown in Fig. 3.2. By this method of selection approximately 1/2 n^2 subsets are evaluated as opposed to the full power set of 2^n.
Figure 3.1. Neighborhood selection and definition from data points in two dimensions.
Figure 3.2. Neighborhood parameters from three data points in two dimensions.
The subset satisfying the specific criteria for the selected method is retained as the optimal subset.

An additional check on the adequacy of the neighborhood selection process is to consider the similarity in simple kriging and ordinary kriging. According to Rivoirard [1984] when these two kriging methods give the same results, the neighborhood is adequate. Supposedly the smallest neighborhood that gives this equality between simple and ordinary kriging is the smallest allowable neighborhood. This can be easily checked in cross validation. When the sum of the weights in simple kriging is greater than about 0.8, the neighborhood is generally adequate. Rivoirard [1987] showed that in general the weight of the mean increases as the size of the neighborhood is increased in a gridded fashion from a few points to many points.

If drift is present the optimal neighborhood will be elongated perpendicular to the drift with the minimum axis along the drift direction. For drift directions with significant curvature, other patterns besides ellipses may be required to generate optimally shaped neighborhoods. For example, a domed surface might respond best to concentric half ring shaped neighborhoods centered on the peak of the dome.

Global minima and maxima might require special consideration, because their single nearest neighbor or few nearest neighbors would give the best neighborhood. Starting parameters and stopping criteria are of prime importance for any optimization method chosen. Starting parameters are not a problem in this
approach. By starting with the closest points and adding them in order of increasing parameter values, an inherent order is placed on the system. Stopping is not as simple. For most of the small data sets used in this study, all of the potential neighborhoods were evaluated so stopping criteria were not needed. For the larger data sets, a limit of 82 nearest points was used to include all the points in the test data set. Parameter studies showed that the effect of this parameter is small for a value of the parameter as large as 82.

3.4 Interpolation of Neighborhood Parameters

Once constrained optimal neighborhood parameters have been found for all the data points, then two or three dimensional kernel estimation can be used to interpolate these parameters to the desired estimation points; or, nearest neighbor parameters can be used since nearest neighbors is just a special case of kernel estimation with an arbitrarily small smoothing parameter. (Having a separate algorithm for nearest neighbors provides a check for the more general subroutine in the limit of arbitrarily large smoothing parameter - or small bandwidth). In the general interpolation case, the kernel function used to apportion weights to the interpolation point can be based on the covariance as determined from the variogram with the aid of these neighborhoods. Of course this covariance model still needs to be evaluated for an optimal smoothing parameter in spatial interpolation.
3.5 Neighborhood Selection Protocol

Neighborhood selection is based on a series of calculations in which the variance of the points contained within the neighborhood is used in some manner to determine the best neighborhood. It may be used as the objective function or as a limiting constraint. Three major optimization calculations are available in the neighborhood optimization program: (1) determination of the unconstrained optimum (minimum neighborhood variance), (2) determination of the optimum for a set number of neighbors and (3) determination of the maximum number of points within a fixed limiting neighborhood variance.

After choosing the criteria for determining best neighborhoods, the data is sorted according to the procedure described in Section 3.3. Using this method of selecting subsets, each subset is arranged so that all the adjacent neighbors are contained within an ellipse or ellipsoid centered about the data point of interest. Each subset has a point a and a b point associated with it. These points are retained and after the best subset is selected for each data point, the ellipse or ellipsoid parameters are determined from the retained a and b points and saved with each data point in a special file.

This file of subsets, one for each data point, constitutes a set of candidate working neighborhoods. Several sets of candidate neighborhoods are generated based on neighborhood and data statistics. However, neighborhood statistics are not sufficient to determine the best neighborhood alone, so these candidate
neighborhoods need to have experimental variograms generated for comparison. The set of candidate neighborhoods with the best "looking" experimental variograms are selected as the working neighborhoods. (Developing a way to quantify this criteria would be necessary to automate the protocol.)

Further analyses using ordinary and simple kriging with the working neighborhood means, instead of the global mean, determines the design variogram and model. When the ordinary and simple kriging results are very similar, the parameters that produce the minimum mean squared error in cross validation are chosen as the final experimental variogram and model. Sometimes results for working neighborhoods will require the addition of neighborhoods not originally contained within the candidate neighborhoods. While ordinary and simple kriging may give approximately the same estimates, they will not give the same kriging variance. Then ordinary kriging should be used for interpolation to incorporate the correction to the kriging variance for estimating the mean.
CHAPTER 4
VERIFICATION WITH SPECIFIC DATA SETS

4.1 Introduction

Several data sets used in this dissertation are listed in Appendix A along with histograms and maps of the data locations. Two data sets were evaluated thoroughly to establish the data handling procedure or protocol for programs listed in Appendix B. The Wolfcamp aquifer potentiometric data from Harper and Furr [1986], with an obvious trend over the sampled area, was used as the reference set to establish the protocol. The second, Cadmium from the GEOEAS example data set, without an obvious trend, was used as the test data set to verify the protocol or to refine it if need be. Four additional data sets were evaluated to demonstrate different aspects of kriging with the protocol. The Robena Mine thickness with a minor linear trend was used by Journel and Rossi [1989]. The Robena Mine ash data for the same sample locations with more noise that obscures the minor trend was used by Cressie [1986]. A geological horizon data set for a synclinal structure with an obvious second order trend was used by Dubrule [1984]. A difficult earthquake data set with neighboring highs and lows was used by Carr [1990].

The Wolfcamp aquifer data were obtained from 85 wells in the Palo Duro basin of northern Texas and eastern New Mexico. These wells are located in a region of approximately 250 by 200 miles. In this region there are three areas of dense well locations, and a few areas devoid of wells, with the rest of the area
containing few well locations. This pattern is good for experimental variogram
calculation because of the three clusters of close wells and is adequate for kriging in
all but the large northwestern area that is devoid of wells.

In the Cadmium data set the sample locations appear to be mostly a series of
alternating, regular and irregular transects in the north south direction with one
regular transect in the east west direction through the middle of the other transects.
The total area is small, of approximately 200 by 250 feet There are few close points
so the nugget is difficult to identify.

Journel and Rossi [1989] used thickness data from a regression study of sulfur
and ash in the Robena Mine coal seams by Gomez and Hazen [1970]. This data set
is located on a regularly spaced grid, but with an irregular outline. The grid spacing
is 2500 by 2500 feet, and the samples cover about 50 square miles within an 8 by 12
mile region. There were 208 samples split into two sets of 100 and 108 samples.
Cressie [1986] used the ash data from the same 208 locations but did not split the
data set. These data sets are not ideal for experimental variogram calculations since
there are no pairs less than 2500 feet apart; the nugget and range are both difficult
to identify for the variables in these data sets.

Dubrule [1984] used the depth to a geologic horizon in a basin. The data
locations are irregularly spaced over an area 15500 meters by 30500 meters with a
few small areas devoid of samples. The data set was split into a working data set of
82 data points and a reference data set of 30 points. The data fit a second order trend so variogram modeling is difficult even though there are several close pairs.

Carr [1990] used ground motion acceleration from the 1983 Coalinga, California earthquake. He normalized the coordinates to minimize numerical instability in the program calculations, so true areas are not available for the data set. The data set is composed of three irregular transects shaped like a short F. Each leg of the transects contains 5 to 9 points. In addition to the transect sample values, there are about a dozen extra sample points located on the backside of the short F for a total of 44 samples. This is an unusual data pattern. The short F with no validating points inside the short F arms is a difficult pattern to evaluate especially when using universal kriging although it is adequate for experimental variogram calculation. Carr [1990] normalized the coordinate data, but the linear extent for an earthquake like this one is on the order of 30 miles (between 10 and 100 miles).

4.2 Study 1: Neighborhood Selection Using Standard Experimental Variograms for Reference and Test Data Sets

By limiting the pairs retained in the pair comparison file to those that are contained within at least one of the candidate neighborhoods, the effects of drift are reduced. Then by varying the parameters of the candidate neighborhood, the effects can be further minimized through subsequent variogram modeling and cross validation.
The object of case study one is to develop a data handling procedure, or protocol using constrained optimal neighborhoods as described in Chapter 3 to produce a good experimental variogram for data sets exhibiting a trend.

An evaluation of data statistics generally indicates the expected ranges of neighborhood parameters that will produce a good set of candidate neighborhoods. Then for each of these candidate neighborhoods experimental variograms are calculated from neighborhood data points only. Working neighborhoods are chosen from the candidate neighborhoods based on the quality of the experimental variograms. Finally, ordinary kriging and simple kriging with the neighborhood means are used to determine the design neighborhood and design experimental variogram. These design values are obtained from the minimum mean squared error of estimation in cross validation for the best fit of a simple model to the experimental variograms of the working neighborhoods that give approximately the same results in OK and SK with neighborhood means.

The Wolfcamp aquifer data set and the Cadmium from the GEOEAS Example.dat were used to establish the neighborhood selection protocol described in Section 3.5. The Wolfcamp aquifer data set, with a known regional trend is used as the test data set to establish the protocol for selecting a good experimental variogram from neighborhood subsets. Cadmium is used as the reference data set. The protocol applied to Cadmium data should also produce a good experimental variogram. The data locations are irregularly spaced and the data appear to be
respectively, somewhat uniformly and normally distributed. The Cadmium experimental variogram, in Englund and Sparks [1990], using all the possible pairs has a finite sill which indicates that there is no significant regional trend.

4.2.1 Candidate Neighborhoods for Wolfcamp Aquifer and Cadmium

The three neighborhood optimization techniques described in Chapter 3 were evaluated using the test and reference data sets. The results are shown in Tables 4.1 through 4.8. Each data point has an associated neighborhood containing few to many data points. Each such associated neighborhood has a mean data value and variance. Statistics for these sets of neighborhood mean values and variances are listed versus neighborhood parameters in the tables. The number of points to be used in kriging, both for experimental variogram calculation and interpolation is at present an arbitrary choice imposed by the structure of the program. In the case of the standard experimental variogram calculation, all pairs whose separation distances are below a maximum interpair distance are included in the calculation. In the case of interpolation, the size of the kriging matrix can be limited in several standard ways. Some researchers such as Davis and Grivet [1984] recommend using the global neighborhood of all data points for kriging. They based their research on improving the efficiency of the matrix calculations. Others such as O'Dowd [1991] demonstrate that larger kriging systems are less well conditioned. In this work, finding the best size of neighborhoods for experimental variogram calculation is the primary goal. In general using the same local neighborhood for kriging as for variogram calculation
Table 4.1. Cadmium neighborhood study for fixed N neighborhoods and the optimum neighborhood.

<table>
<thead>
<tr>
<th>No. Nbrs:</th>
<th>59</th>
<th>45</th>
<th>30</th>
<th>15</th>
<th>10</th>
<th>optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Mean:</td>
<td>8.02</td>
<td>9.18</td>
<td>10.26</td>
<td>11.49</td>
<td>11.80</td>
<td>11.67</td>
</tr>
<tr>
<td>Min. Mean:</td>
<td>7.74</td>
<td>7.87</td>
<td>7.80</td>
<td>7.17</td>
<td>5.66</td>
<td>0.70</td>
</tr>
<tr>
<td>Mean Mean:</td>
<td>7.88</td>
<td>8.58</td>
<td>8.89</td>
<td>9.13</td>
<td>9.17</td>
<td>9.29</td>
</tr>
<tr>
<td>Var. Mean:</td>
<td>0.004</td>
<td>0.10</td>
<td>0.30</td>
<td>0.87</td>
<td>2.08</td>
<td>5.82</td>
</tr>
<tr>
<td>Mean Var:</td>
<td>15.3</td>
<td>11.6</td>
<td>8.6</td>
<td>5.6</td>
<td>4.0</td>
<td>0.18</td>
</tr>
<tr>
<td>No. Pairs:</td>
<td>1770</td>
<td>1304</td>
<td>858</td>
<td>389</td>
<td>269</td>
<td>77</td>
</tr>
</tbody>
</table>

Table 4.2. Wolfcamp aquifer neighborhood study for fixed N neighborhoods and the optimum neighborhood.

<table>
<thead>
<tr>
<th>No. Data Points:</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Mean:</td>
<td>2002.</td>
</tr>
<tr>
<td>Data Variance:</td>
<td>369339.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. Nbrs:</th>
<th>84</th>
<th>65</th>
<th>45</th>
<th>30</th>
<th>15</th>
<th>optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Var. Mean:</td>
<td>54.</td>
<td>28869.</td>
<td>101459.</td>
<td>187890.</td>
<td>239743.</td>
<td>260030.</td>
</tr>
<tr>
<td>Mean Var:</td>
<td>369288.</td>
<td>226453.</td>
<td>133488.</td>
<td>68168.</td>
<td>25520.</td>
<td>809.</td>
</tr>
<tr>
<td>No. Pairs:</td>
<td>3570</td>
<td>3100</td>
<td>2159</td>
<td>1435</td>
<td>736</td>
<td>139</td>
</tr>
</tbody>
</table>
Table 4.3. Cadmium neighborhood study for different limiting neighborhood variances.

<table>
<thead>
<tr>
<th>% Nbhd Variance</th>
<th>75</th>
<th>50</th>
<th>25</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Mean:</td>
<td>8.06</td>
<td>0.2</td>
<td>12.5</td>
<td>11.6</td>
</tr>
<tr>
<td>Min. Mean:</td>
<td>7.74</td>
<td>8.08</td>
<td>6.14</td>
<td>5.14</td>
</tr>
<tr>
<td>Mean Mean:</td>
<td>7.96</td>
<td>8.67</td>
<td>9.20</td>
<td>8.87</td>
</tr>
<tr>
<td>Var. Mean:</td>
<td>0.005</td>
<td>0.20</td>
<td>1.22</td>
<td>2.15</td>
</tr>
<tr>
<td>Mean Var:</td>
<td>14.9</td>
<td>11.0</td>
<td>7.07</td>
<td>3.13</td>
</tr>
<tr>
<td>Max No. Pts:</td>
<td>59</td>
<td>53</td>
<td>37</td>
<td>27</td>
</tr>
<tr>
<td>No. Pairs:</td>
<td>1752</td>
<td>1226</td>
<td>723</td>
<td>356</td>
</tr>
</tbody>
</table>

Table 4.4. Wolfcamp aquifer neighborhood study for different limiting neighborhood variances.

<table>
<thead>
<tr>
<th>% Nbhd Variance</th>
<th>20</th>
<th>15</th>
<th>10</th>
<th>7.5</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Mean:</td>
<td>2738</td>
<td>2738</td>
<td>2820.</td>
<td>2793.</td>
<td>2737.</td>
</tr>
<tr>
<td>Min. Mean:</td>
<td>1517.</td>
<td>1382.</td>
<td>1317.</td>
<td>1195.</td>
<td>1151.</td>
</tr>
<tr>
<td>Var. Mean:</td>
<td>194395.</td>
<td>189430.</td>
<td>241368.</td>
<td>249376.</td>
<td>247922.</td>
</tr>
<tr>
<td>Mean Var:</td>
<td>67738.</td>
<td>48455.</td>
<td>31697.</td>
<td>23546.</td>
<td>15584.</td>
</tr>
<tr>
<td>Max No. Pts:</td>
<td>46</td>
<td>39</td>
<td>33</td>
<td>29</td>
<td>24</td>
</tr>
<tr>
<td>No. Pairs:</td>
<td>1499</td>
<td>1221</td>
<td>968</td>
<td>789</td>
<td>640</td>
</tr>
</tbody>
</table>
Table 4.5. Cadmium neighborhood comparison between deleted and original data sets for candidate experimental variograms.

<table>
<thead>
<tr>
<th>% Nbhd Variance:</th>
<th>75</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Set</strong></td>
<td><strong>Original</strong></td>
<td><strong>Deleted</strong></td>
</tr>
<tr>
<td>Max. Mean:</td>
<td>10.2</td>
<td>9.94</td>
</tr>
<tr>
<td>Min. Mean:</td>
<td>8.08</td>
<td>7.69</td>
</tr>
<tr>
<td>Mean Mean:</td>
<td>8.67</td>
<td>8.81</td>
</tr>
<tr>
<td>Var. Mean:</td>
<td>0.20</td>
<td>0.21</td>
</tr>
<tr>
<td>Mean Var:</td>
<td>11.0</td>
<td>9.39</td>
</tr>
<tr>
<td>Max No. Pts:</td>
<td>53</td>
<td>51</td>
</tr>
<tr>
<td>No. Pairs</td>
<td>1226</td>
<td>1158</td>
</tr>
</tbody>
</table>

Table 4.6. Wolfcamp aquifer neighborhood comparison between deleted and original data sets for candidate experimental variograms.

<table>
<thead>
<tr>
<th>% Nbhd Variance:</th>
<th>10</th>
<th>7.5</th>
<th>7.5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Set</strong></td>
<td><strong>Original</strong></td>
<td><strong>Deleted</strong></td>
<td><strong>Original</strong></td>
</tr>
<tr>
<td>Max. Mean:</td>
<td>2820.</td>
<td>2760.</td>
<td>2793.</td>
</tr>
<tr>
<td>Min. Mean:</td>
<td>1317.</td>
<td>1195.</td>
<td>1195.</td>
</tr>
<tr>
<td>Var. Mean:</td>
<td>241368.</td>
<td>253486.</td>
<td>249376.</td>
</tr>
<tr>
<td>Mean Var:</td>
<td>31697.</td>
<td>26249.</td>
<td>23546.</td>
</tr>
<tr>
<td>Max No. Pts:</td>
<td>33</td>
<td>31</td>
<td>29</td>
</tr>
<tr>
<td>No. Pairs</td>
<td>968</td>
<td>857</td>
<td>789</td>
</tr>
</tbody>
</table>
Table 4.7. Cadmium 75% limited variance neighborhood study for different maximum number of points.

<table>
<thead>
<tr>
<th>No. Max Points:</th>
<th>56</th>
<th>50</th>
<th>45</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Mean:</td>
<td>9.94</td>
<td>9.84</td>
<td>10.0</td>
<td>11.04</td>
</tr>
<tr>
<td>Min. Mean:</td>
<td>7.69</td>
<td>7.69</td>
<td>7.50</td>
<td>7.91</td>
</tr>
<tr>
<td>Mean Mean:</td>
<td>8.81</td>
<td>8.85</td>
<td>8.90</td>
<td>8.99</td>
</tr>
<tr>
<td>Var. Mean:</td>
<td>0.21</td>
<td>0.18</td>
<td>0.24</td>
<td>0.36</td>
</tr>
<tr>
<td>Max No. Pts:</td>
<td>51</td>
<td>50</td>
<td>44</td>
<td>40</td>
</tr>
<tr>
<td>No. Pairs:</td>
<td>1158</td>
<td>1103</td>
<td>1017</td>
<td>883</td>
</tr>
</tbody>
</table>

Table 4.8. Wolfcamp aquifer 10% limited variance neighborhood study for different maximum number of points.

<table>
<thead>
<tr>
<th>No. Max. Points:</th>
<th>85</th>
<th>70</th>
<th>60</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Mean:</td>
<td>2760</td>
<td>2760</td>
<td>2760</td>
<td>2760</td>
</tr>
<tr>
<td>Min. Mean:</td>
<td>1195</td>
<td>1195</td>
<td>1195</td>
<td>1195</td>
</tr>
<tr>
<td>Mean Mean:</td>
<td>1973</td>
<td>1974</td>
<td>1954</td>
<td>1952</td>
</tr>
<tr>
<td>Var. Mean:</td>
<td>253486</td>
<td>260890</td>
<td>268705</td>
<td>270092</td>
</tr>
<tr>
<td>Mean Var:</td>
<td>26249</td>
<td>26400</td>
<td>26045</td>
<td>24909</td>
</tr>
<tr>
<td>Max No. Pts:</td>
<td>31</td>
<td>30</td>
<td>28</td>
<td>27</td>
</tr>
<tr>
<td>No. Pairs:</td>
<td>857</td>
<td>852</td>
<td>828</td>
<td>783</td>
</tr>
</tbody>
</table>
is recommended. In some cases of extra large neighborhoods giving poor cross validation performance even smaller kriging neighborhoods may be required to avoid the trend to the mean and loss of accuracy in cross validation noted by Armstrong [1984] and Spease and Carr [1988].

The objective function utilized in this initial neighborhood study to determine neighborhood parameters is the variance of the neighborhood with the central point deleted; preliminary work indicated that deleting the central data point is particularly useful for initial neighborhood studies by assisting in evaluating how well data points fit with their neighbors. While the central data point could have been included in the analysis, the neighborhoods would not be independent of the central data point and would not be as useful in data editing. (Later evaluations will find the undeleted central point more useful.) In the first set of neighborhood analyses, unconstrained optimization for the neighborhood with minimum variance was applied to both data sets. The results of these single runs are included in Tables 4.1 and 4.2 with the second set of fixed N analyses. The unconstrained optimal neighborhoods are too small for practical use, generally with 3 or 4 neighbors, but they provide a limiting case for the second analyses.

In the second set of neighborhood analyses, fixed N constrained optimization is used with the minimum neighborhood variance objective function. The results of the first neighborhood analyses for both data sets are shown in Tables 4.1 and 4.2. Both Cadmium and the Wolfcamp aquifer fixed N neighborhoods show the same
general trends as the number of neighborhood points is reduced from the full set to the optimal or minimum set of ellipse or ellipsoid determining points. (The minimum number of neighbors allowable in the evaluation of the 3 ellipse parameters for 2-D is two.)

A third set of neighborhood analyses evolved from an attempt to improve the experimental variogram by reducing the neighborhood variance while providing the largest number of pairs in the constrained optimal neighborhoods. The results, of this new method of choosing the data subset with the maximum number of points within a set limiting neighborhood variance, are shown in Tables 4.3 and 4.4. The neighborhood statistics are very similar to those for the fixed N method but the experimental variograms based on these neighborhoods differ significantly for the Wolfcamp aquifer.

The neighborhood studies show that both data sets contain some data points with a relatively high mean squared error of estimation. For Cadmium, which is a mineral evaluation data set, the presence of a few zero data points increases the variance of the data while for the Wolfcamp aquifer, the presence of a few large numbers increases the variance of the data significantly. Data values that were far removed from their neighborhood means were deleted. Three points were deleted from each data set. In the Cadmium data, low values were deleted, two of which were zero. In the Wolfcamp aquifer data set, three high values were deleted that were along the southwestern edge of the data field. Eliminating three data points in
both data sets results in another set of neighborhood evaluations shown in Tables 4.5 and 4.6, which are compared to the original neighborhood statistics for two different neighborhood variance limits.

A final set of neighborhood analyses, shown in Tables 4.7 and 4.8, demonstrate that for both the Wolfcamp aquifer and Cadmium data sets, the first series of analyses are not affected by the number of points used in the neighborhood search.

Despite the major differences in Cadmium and the Wolfcamp aquifer data, the neighborhood statistics do not reflect these basic differences except in the behavior of the variance of the neighborhood mean. The variance of the neighborhood mean increases slowly for Cadmium as the number of pairs increases, but it increases rapidly for the Wolfcamp aquifer. The neighborhood statistics combined with the data statistics and the number of pairs generated in the pair comparison file generation indicate that relatively large neighborhoods for cadmium and small neighborhoods for the Wolfcamp aquifer are appropriate. Experimental variograms for a range of neighborhoods are still required to discern a good set of working neighborhoods.

4.2.2 Wolfcamp Aquifer Variograms and Cross Validation

Experimental variograms for the candidate neighborhoods of the test Wolfcamp aquifer data set are evaluated first, then the protocol is applied to the reference cadmium data set for verification.
The first set of figures are intended to demonstrate the need for the development of the programs used in this work. Standard experimental variograms for standard neighborhoods were calculated for the Wolfcamp aquifer data set. The results of experimental variogram analyses calculated from data points of standard, fixed size neighborhoods are shown in Figures 4.1 and 4.2. For these simple neighborhoods, the experimental variograms are based only on pairs within at least one of the neighborhoods. These standard neighborhoods were applied to the Wolfcamp aquifer data set, with the experimental variogram calculated from the reduced set of potential pairs. As can be seen in Figure 4.1, radial neighborhoods do not reduce the effects of drift in the Wolfcamp aquifer data. Only standard, elliptical neighborhoods with a major axis of 100, a minor axis of 50, and an orientation of 135 degrees begin to improve the calculation of the experimental variogram as shown in Figure 4.2. By making the neighborhoods unique to each data point, further improvement in the variogram calculation is expected. While early work centered directly on multidimensional optimization of the ellipse parameters, problems with tolerances, stopping criteria and nonuniqueness led to the development of a discrete approach for the selection of neighbors, described in Section 3.3. This selection of neighbors improves the calculation of the experimental variogram for the Wolfcamp aquifer as shown in the sequence of figures from Figure 4.3 through Figure 4.6. Figure 4.3 is a standard experimental variogram determined from the pairs of Wolfcamp aquifer data with their respective neighbors contained in the best elliptical
Figure 4.1. Various standard neighborhood experimental variograms for circles of 50, 75, and 100 distance units; solid, longdash and short dash respectively; Wolfcamp aquifer.
Figure 4.2. Various standard neighborhood experimental variograms for ellipses with major axis to minor axis ratio 100/50 in distance units for 90, 135, and 180 degrees orientation of major axis; solid, long dash and short dash respectively; Wolfcamp aquifer.
Figure 4.3. Experimental variograms for various fixed N neighborhoods and global neighborhood for 15, 20, 25, 30, and 82 neighbors; dot, dot-dash, dash, solid and dash respectively; Wolfcamp aquifer.
Figure 4.4. Experimental variograms for various limited variance neighborhoods and global neighborhood for 5%, 10%, 15%, 20%, and 100% of the data variance; dot, dot-dash, dash, solid, and dash respectively; Wolfcamp aquifer.
Figure 4.5. Experimental variograms for various data set combinations with limited variance neighborhoods: full data set and 7.5% neighborhoods, full data set and 10% neighborhoods, deleted data sets and 7.5% neighborhoods, deleted data set and 7.5% limited variance neighborhoods; dot-dash, solid, dot, and dash respectively; Wolfcamp aquifer.
Figure 4.6. Experimental variograms for various maximum number of points: 55, 60, 70, and 82 nearest neighbors used in search; dot, dot-dash, dash, solid respectively; Wolfcamp aquifer.
neighborhood as determined by minimum neighborhood variance for a series of set numbers of neighbors. Neighborhood and data statistics indicated that the data had a regional trend, so only from about 20% to 40% (about 15 to 30) of the adjacent data points were used as neighbors to generate the variograms. While the variograms start the same at the origin, they soon vary with the variograms based on the smallest number pairs varying the most as the lag distance increases. Here the set of fixed N neighbors with the minimum neighborhood variance is selected for each data point. For 15 and 20 neighbors, the experimental variograms display a sill to a distance of about 70 lag units before the effects of drift begin to dominate the variogram.

Further improvement is obtained by using the maximum number of points in the elliptical neighborhood that results in a neighborhood variance less than a specified fraction of the data variance, as shown in Figure 4.4. Here the 5%, 10%, 15%, and 20% limited variance neighborhoods, defined in Section 3, display very similar experimental variogram to a distance of about 70 lag units. The variograms in Figure 4.4 are better than those in Figure 4.2, which can be seen by defining an area with a plateau in the variogram that can be used for a local variogram sill. This particular type of neighborhood is used to generate neighborhoods for calculating experimental variograms because it allows for the most points for a given limit to the neighborhood can contain additional variation as shown in Figure 4.6. As the data sets increase in size, the time required for sorting increases rapidly. The 85 point Wolfcamp aquifer data set represents the largest data set that was entirely searched.
The working neighborhoods of 10%, 15%, and 20% limited variance for the edited Wolfcamp aquifer data were evaluated through cross validation using ordinary kriging and simple kriging with neighborhood means. Each neighborhood set was used to create a 15 point experimental variogram and 7 points from that variogram were fit to obtain a working model for cross validation with OK and SK. The 15% variance neighborhood gives the best cross validation result for both OK and SK with a good weight of the mean being low. The 15% variance neighborhoods in general contain a fairly large number of data points (20-30), more than is routinely used in standard kriging neighborhood restrictions.

Residual data sets from the Wolfcamp aquifer compiled by Harper and Furr [1986], were used to generate experimental variograms calculated from all possible pairs and to compare them to experimental variograms calculated from data pairs limited to those contained within constrained optimal neighborhoods. The comparison is shown in Figure 4.7. Here, the variogram for both the linear and quadratic residuals are compared to that of the 15% limited variance neighborhood which is the design neighborhood. As can be seen in the figure, up to a distance of about 70 lag units, the constrained optimal neighborhood variogram lies between those of the linear and quadratic residuals. The favorable comparison of the maximum working experimental variogram with those for both linear and quadratic residuals confirms the utility of the approach.
Figure 4.7. Final experimental variogram compared to other Variograms: final variogram, linear residual variogram, quadratic residual variogram, and global variogram; solid, lower dash, upper dash, dot, respectively; Wolfcamp aquifer.
After choosing the 15% variance neighborhood as the design neighborhood, additional analyses were required to refine the set of model parameters. A simple fitting routine for nugget, sill, and range of a spherical, exponential, or Gaussian model was used for this purpose. Various numbers of points from the experimental variogram were fitted to a model. The best result of fitting points of the 15 point experimental variogram is highlighted as a solid line in Figure 4.8 where all these models are shown along with the experimental variogram. Even with the last half of the experimental variogram being unstable, the first half allowed for good results based on both OK and SK with the neighborhood means. Tables 4.9 and 4.10 compare the results of the limited variance and fixed N analyses.

By using cross validation and fitting different number of variogram points, the best variogram model can be seen to be independent of the points at large lags which are affected by drift. The model fit to only 7 points yields the best cross validation result for the design experimental variograms shown in Figure 4.8 as a solid line. The graph of estimated values versus data for this model is shown in Figure 4.9 where the good agreement for almost all data points is apparent; there is a uniform spread of points along the slope that equals one line. The neighborhood optimization program and protocol produce very good results for the Wolfcamp aquifer data with a known drift expected to be between linear and quadratic.
Figure 4.8. Final experimental variogram of 15% limited variance neighborhoods with different models for 5, 6, 7, and 8 fitted points counted from the origin; lower dash, upper dash, solid, dot-dash respectively; Wolfcamp aquifer.
Table 4.9. Cross validation comparison of simple kriging with neighborhood means and ordinary kriging, and weight of the mean review for seven variogram points fit to a model for the Wolfcamp aquifer.

<table>
<thead>
<tr>
<th>Vgm Pts &amp; Nbhd</th>
<th>Xvalid MSE</th>
<th>Variogram Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OK</td>
<td>SK</td>
</tr>
<tr>
<td>20 10%:</td>
<td>170.9</td>
<td>172.2</td>
</tr>
<tr>
<td>20 15%:</td>
<td>155.7</td>
<td>155.6</td>
</tr>
<tr>
<td>20 20%:</td>
<td>166.4</td>
<td>165.7</td>
</tr>
<tr>
<td>20/N=15:</td>
<td>166.7</td>
<td>166.6</td>
</tr>
<tr>
<td>20/N=10:</td>
<td>164.7</td>
<td>164.1</td>
</tr>
</tbody>
</table>

** good: almost all sums of wts are above 0.7 with most above 0.9
fair almost all sums of wts are above 0.5 with most above 0.7
poor almost all sums of wts are above 0.3 with most above 0.5

Table 4.10. Cross validation comparison of simple kriging with neighborhood means and ordinary kriging, and weight of the mean review with 15% limited variance neighborhoods for the Wolfcamp aquifer.

<table>
<thead>
<tr>
<th>No. Fit/No. Lags</th>
<th>Xvalid MSE</th>
<th>Variogram Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OK</td>
<td>SK</td>
</tr>
<tr>
<td>5/15:</td>
<td>157.4</td>
<td>157</td>
</tr>
<tr>
<td>6/15:</td>
<td>156.5</td>
<td>156.6</td>
</tr>
<tr>
<td>7/15:</td>
<td>155.7</td>
<td>155.6</td>
</tr>
<tr>
<td>8/15:</td>
<td>156.5</td>
<td>156.4</td>
</tr>
</tbody>
</table>

** good: almost all sums of wts are above 0.7 with most above 0.9
fair almost all sums of wts are above 0.5 with most above 0.7
poor almost all sums of wts are above 0.3 with most above 0.5
Figure 4.9. Estimated values from simple kriging from variogram model of 8000 nugget, 28000 sill, 78.75 range and 15% limited variance neighborhood means versus data values; Wolfcamp aquifer.
For comparison it is noted that Harper and Furr [1986] calculated anisotropic variograms that demonstrate the trend in the data is towards the northeast overall (in New Mexico it is to the east). Variograms calculated in a northwest direction displayed a finite sill while those in the major directions between east and north did not. Both a linear and quadratic trend surface model were used to obtain residuals. The variogram of the linear trend model residuals indicated a finite sill of about 40,000 ft sq, a nugget of about 10,000 ft sq and a range of between 50 and 75 miles. A contoured surface was generated with universal kriging and a spherical model fit to the residual variogram. This surface drops to the east in New Mexico steeper than it drops to the northeast in Texas. The few most westerly wells in New Mexico appear to be distinctly higher than the rest of the New Mexico wells producing the apparently steep eastward dip. If three of these samples were deleted, the regional trend, as shown in Harper and Furr [1986], would continue from Texas into New Mexico.

4.2.3 Cadmium Variograms and Cross Validation

The second data set evaluated with these procedures is the Cadmium data set. This data set produces more consistent experimental variograms for different neighborhood parameters than the Wolfcamp aquifer data but does not perform as well in cross validation, as is shown in the sequence of figures from Figures 4.10 through 4.15.
Figure 4.10. Experimental variograms for various fixed N neighborhoods and global neighborhood for 15, 30, 45, and 59 neighbors; dot, dot-dash, dash, and solid respectively; Cadmium data.
Figure 4.11. Experimental variograms for various limited variance neighborhoods and global neighborhood for 25%, 50%, 75%, and 100% of the data variance; dot, dot-dash, dash, and solid respectively; Cadmium data.
Figure 4.12. Experimental variograms for various data set combinations with limited variance neighborhoods: full data set and 50% neighborhoods, full data set and 75% neighborhoods, deleted data sets and 50% neighborhoods, deleted data set and 75% limited variance neighborhoods; dot, dot-dash, dash, and solid respectively; Cadmium data.
Figure 4.13. Experimental variograms for various maximum number of points: 30, 45, 50, and 55 nearest neighbors used in search; dot, dot-dash, dash, and solid respectively; Cadmium data.
Figure 4.14. Final experimental variogram of 90% limited variance neighborhoods with different models for 10, 11, 12, and 13 fitted points counted from the origin; dash, dot-dash, solid, and dot respectively; Cadmium data.
Figure 4.15. Estimated values from simple kriging with variogram model of 0.0 nugget, 10.9 sill, 61.0 range, and 90% limited variance neighborhood means versus data values; Cadmium data.
Figure 4.10 is a standard experimental variogram determined from the pairs of Cadmium data with their respective neighbors contained in the best elliptical neighborhood as determined by minimum neighborhood variance for a set number of neighbors. Neighborhood statistics indicated that the data lacked a regional trend, so from 25% to 100% (15 to 59) of the adjacent data points were used as neighbors to generate the variograms. While the variograms start the same at the origin, they soon diverge from the others with the variograms based on the smallest number of pairs diverging the most.

Figure 4.11 is a standard experimental variogram determined from the pairs of Cadmium data with their respective neighbors contained in the best elliptical neighborhood as determined by the maximum number of points for a limited neighborhood variance. Based on the neighborhood statistics from 25% to 100% (15 to 59) of the adjacent data points were used as neighbors to generate the variograms. While the toes of the variograms start the same, they soon begin to vary with the variograms based on the smallest number of pairs varying the most. The variograms in Figure 4.11 do not appear to be significantly different from those in Figure 4.10.

The Cadmium data set appears to have a simple finite sill that is demonstrated for all sets of nearest neighbors as shown in Figure 4.10 and for all limited variance neighborhoods shown in Figure 4.11. Deleting the zero points from the data set reduces the variance of the variograms, but does not change the variograms
significantly, as shown in Figure 4.12 which compares deleted and undeleted data sets. There is little difference in these variograms.

Figure 4.13 is a series of experimental variograms generated for cadmium from the same 75% variance neighborhood, but with reduced searches. The total number of adjacent neighbors included in the search for best subset is reduced from 100% to 75% with only a small reduction in total pairs. However, the variograms show significant effect at large lags.

The 90% limited variance neighborhood produced the minimum mean squared error of estimation in cross validation for a spherical model of 12 out of 20 variogram points as shown in Figure 4.14 with a zero nugget, a sill of 10.9, and a range of 61. This model produced the estimated values versus data shown in Figure 4.15. Surprisingly, these results are not as good as those for the data set with a trend; this graph shows considerable smoothing in the cross validation with low highs and high lows. It is a common occurrence in the references from which data sets were evaluated for inclusions in this work. This result will be examined further in Section 4.3.

The GEOEAS manual shows one working variogram model to be a spherical model with a sill of 11, a nugget of 5 and a range of 100; another is an exponential model with a sill of 13.5, a nugget of 4.5, and a range of 160. Because the Cadmium data locations are spaced fairly evenly, the nugget is hard to find with as few points as used in the demonstration. Later in the manual, the influence of the number of
calculated points on the standard variogram is demonstrated. The nugget value is particularly sensitive to the number of points, requiring more calculated points than the sill for a good definition, and the range is intermediate in its sensitivity to the number of points. The examples shown there with the most points show a zero nugget which is the result obtained in this study.

There is an irregularity in the directional variograms shown in the GEOEAS manual; the sill obtained is a function of the direction. This is not a geometric anisotropy, since geometric anisotropy refers to a distance irregularity that can be removed with an affine transformation. This irregularity is due to the saddle-like surface of the data with the two high on an east-west line and the two lows on a north-south line; along the two directions to the northeast and northwest from the center point the data are relatively level.

A comparison of the results for the two data sets indicate the following points: The first standard variogram analyses for set N data subsets shown in Figures 4.3 and 4.10 indicate that 45 to 59 neighbors generate good variograms for Cadmium, but that over 20 points in the Wolfcamp aquifer data set allows for drift effects to dominate the variogram. Figures 4.3 and 4.11 indicate that for a limited variance technique a 50% or 75% neighborhood produces a good experimental variogram for Cadmium while a 7.5% or 10% neighborhood produces a good experimental variogram for the Wolfcamp aquifer.
This first study demonstrates that the experimental variograms for the reference data set without a regional drift are not sensitive to the method of variogram generation and data editing, while those for the test data set with a regional drift are very sensitive to the method of neighborhood generation and data editing. In spite of this sensitivity, the method described for obtaining the most points in a neighborhood with a limited neighborhood variance appears to provide a good experimental variogram for both data sets. On the other hand, kriging in cross validation appears to be sensitive to something which will be examined in Study 3, after the kernel calculations.

The method of neighborhood selection, based on the criteria developed in this work, can be compared to the practice of selecting neighbors from a standard volume such as a circle or an ellipse with set parameters. These standard neighborhoods should work well for Cadmium, but not so well for the Wolfcamp aquifer as is shown in Figures 4.1 and 4.2. Circular neighborhoods do not work at all in eliminating the affect of drift, while elliptical ones do reduce the effect when aligned perpendicular to the direction of the trend.

As the number of neighborhood points are reduced, the variance of the neighborhood means increases considerably and the stability of the variogram is reduced. The number of pairs decreases below the point of a stable experimental variogram at about 30 points for Cadmium and 25 points for the Wolfcamp aquifer. In general, about 800 well spaced pairs provide a good variogram; however, as few
as 400 pairs may provide an adequate variogram with a kernel smoother improving the stability as shown in the next study.

4.3 Study 2: Neighborhood Selection Using Kernel Estimated Experimental Variograms for Reference and Test Data Sets

The object of this study is to improve the calculation of the experimental variogram with a kernel function and to compare the results to the standard calculation method. In the standard method, the pairs of points are partitioned into bins like a histogram and the average variogram value calculated for each bin. The bin width is determined by the number of variogram points to be calculated and the maximum bin distance. Kernel estimation is a method of curve generation that does not partition data into bins as per a histogram calculation, but, weights all the data according to a kernel function that rapidly decreases from the point of interest. In this manner kernel estimation of the experimental variogram creates an effective bin width that improves the calculation by separating this effective bin width from the number of points to be calculated. Each variogram point calculation weights all the pair half squared differences according to the nature and parameters of the kernel function, with many weights being close to zero. It is as though the bins of the standard variogram were of variable width and had fuzzy boundaries. The main effect of using a kernel function to estimate the experimental variogram is that with all the pairs being used in each calculation, as many points as are necessary can be calculated without reducing the accuracy of the graph.
Various choices of kernel function and smoothing parameters can be made a priori and then the one confirmed as the best for the purpose and data can be selected. A Gaussian kernel was selected since it creates an effective bin with a weight distribution that can be made to be close to the uniform weight of the standard bin. This Gaussian function requires an argument of three times the natural argument to provide an effective bin width comparable to the standard bin. For this argument the Gaussian is reduced to 5% of its original value at the edge of the standard bin.

The easiest variograms to calculate with a kernel are the ones closest to linear; and the most difficult ones are those with rapid changes over a short distance as in spherical variograms. For the easy ones, a global bandwidth parameter is generally sufficient to generate a good experimental variogram. For the difficult ones, a variable bandwidth parameter is needed to allow for different effective bin widths at different locations. In regions of rapid change, a small effective bin width is needed; however, in regions of sparse data a large effective bin width is needed. At the origin, these criteria can be competing with each other which may be troublesome. Frequently, the ratio of the large bandwidth required at the tail of the variogram to the small bandwidth required at the origin of the variogram is excessive (over about 50). Finding functions to provide a good distribution of bandwidths over this range of values is not easy. No one distribution function would be good for all data sets. The course taken herein is to use global parameters for a first calculation of the
experimental variogram. If the results show wide variation over a reasonable range of global parameters, then a simple piecewise linear function between the two end points and one mid point is used to create a range of bandwidths over the total pair distance.

Sometimes under smoothing is required to demonstrate a point, and sometimes over smoothing is required. These conditions are not as difficult to specify, as compared to finding the perfectly smoothed variogram with a piecewise linear function for the smoothing parameter. An under smoothed global parameter can be used to generate many points that bracket the unavailable perfectly smooth variogram. In this manner, the indicated by the changes in the point scatter from one area to another.

4.3.1 Wolfcamp Aquifer Variograms and Cross Validation

Again the Wolfcamp aquifer will be evaluated first. A judicious choice of a limited variance neighborhood produces a nearly linear experimental variogram for demonstration purposes only. Figure 4.16 demonstrates the effectiveness of a global parameter for calculating experimental variograms when there are no regions of rapid change. This variogram is for the Wolfcamp aquifer data using a 50% limited variance neighborhood, and with different values of the global smoothing parameter lambda. Global values of 0.2 and 0.1 produce almost identical curves and those of 0.4 and 0.05 made similar curves. Figure 4.17 shows that as the neighborhood variance limit is reduced to 15%, producing a typical variogram. The variogram
Figure 4.16. Various kernel estimated experimental variograms for 50% neighborhood; global lambdas: 0.4 dot, 0.2 dot-dash, 0.1 short dash, 0.05 long dash; Wolfcamp aquifer.
Figure 4.17. Various kernel estimated experimental variograms for 15% neighborhood; global lambdas: 0.4 dot, 0.2 dot-dash, 0.1 short dash, 0.05 long dash; Wolfcamp aquifer.
graphs vary for these same values of lambda even though these are the same values of lambda used in the first figure. In this situation the higher values of lambda are effective near the origin and the lower values of lambda are effective near the tail.

Figure 4.18 compares a standard experimental variogram to the kernel smoothed variogram in Figure 4.17 that has the same effective bin width as the standard experimental variogram. The resulting curves are almost identical.

Figure 4.19 shows the advantage of the kernel estimator. Many points are generated with an under smoothing value of lambda that is slightly higher than the average value would be. It provides a good indication of the variation in the experimental variogram. In this case the experimental variogram may be seen to be more variable in the area of the sill rather than near the origin. In Figure 4.20 the under smoothed graph is compared to a smoothed graph. A good kernel estimated variogram reduces variance without introducing significant bias. Only it is not specified in a calculation and is presently found visually by trial and error.

4.3.2 Cadmium Variograms and Cross Validation

The standard experimental variogram has almost all its variation near the origin. Figure 4.21 shows the kernel estimator for the Cadmium data set for this difficult case where the values near the origin are rapidly varying and irregular. The data is for the 75% limited variance neighborhood of Cadmium and demonstrates the ineffectiveness of a global parameter in their situation. This figure also demonstrates the effect a global or initial parameter has on the apparent nugget. The 0.2
Figure 4.18. Standard and kernel estimated experimental variograms 15% neighborhood: standard dot-dash, kernel estimated dash; Wolfcamp aquifer.
Figure 4.19. Undersmoothed experimental variogram 15% neighborhood; Wolfcamp aquifer.
Figure 4.20. Comparison of undersmoothed and smoothed experimental variograms; Wolfcamp aquifer.
Figure 4.21. Various kernel estimated experimental variograms for 75% neighborhood; global lambdas: 0.4 dot, 0.2 dot-dash, 0.1 short dash, 0.025 long dash; Cadmium data.
smoothing parameter value is chosen as a global lambda in Figure 4.22 where the standard and kernel smoothed variograms are compared. The differences near the origin are not acceptable and a variable bandwidth is calculated with a composite set of smoothing parameters that are used to improve the match near the origin as shown in Figure 4.23. A simple piecewise linear function improves the kernel estimator over the standard in regions of sparse data and is comparable near the origin.

Figure 4.24 compares the best graph for the variable lambda of Figure 4.23 with a global under smoothed value of lambda equal to 0.4 to demonstrate the variance of the variogram and show that the average graph has indeed reduced the variance while remaining relatively unbiased as judged visually.

Neighborhood selection using a kernel smoothed experimental variograms differs from that of Study 1 in Section 4.2 with standard variograms only in the manner in which the experimental variogram is calculated. Once the candidate neighborhoods have been selected per Study 1, the experimental variograms are calculated with a kernel smoothing function instead of the standard histogram method.

4.4 Study 3: Kernel Estimation of Reference and Test Data Sets

In addition to improving the experimental variogram calculation with a one dimensional kernel, a two or three dimensional kernel estimator can be used to calculate the response surface of the variable of interest. There are several
Figure 4.22. Standard and kernel estimated experimental variograms for 75% neighborhood; standard dot-dash, kernel estimated dash with global lambdas dash; Cadmium data.
Figure 4.23. Standard and kernel estimated experimental variograms for 75% neighborhood: standard dot-dash, kernel estimated with composite lambdas dash; Cadmium data.
Figure 4.24. Comparison of undersmoothed global and smoothed composite experimental variograms; Cadmium data.
advantages to this approach: first, the nearest neighbor limit provides an additional tool for data editing; second, it can be used to calculate approximate confidence intervals; and third it is computationally fast, and fourth it can be readily automated.

Kernel estimation with an arbitrarily small smoothing parameter produces a nearest neighbor estimate and one with an arbitrarily small smoothing parameter produces the data mean. While it would be easy to define the best kernel estimate similarly to kriging, as the one that minimizes the variance and is unbiased, it would be difficult to produce the curve with this exact property. Instead, it is presently defined to be that which produces the minimum mean square cross validation error. In general this mean square error versus bandwidth parameter curve decreases from the nearest neighbor mean square error to a minimum and then increases to the data variance when the estimate becomes the mean.

4.4.1 Wolfcamp Aquifer Analyses

By comparing estimated values versus the data in cross validation, it can be seen that best kernel estimates are generally better than nearest neighbors which are almost always better than the data mean. But not all kriging estimators turn out to be better than the best kernel estimators in cross validation. The Wolfcamp aquifer results from Study 1 appear to be comparable to the best kernel estimator. Figure 4.25 is the nearest neighbor limit and Figure 4.26 is the best kernel estimator. Figures 4.25 and 4.26 show that the best kernel estimator is better than the nearest neighbor, but comparing Figure 4.26 to Figure 4.15 of Study 1 shows that best kernel
Figure 4.25. Nearest neighbor estimates versus data values; Wolfcamp aquifer.
Figure 4.26. Isotropic two dimensional exponential kernel estimates versus data values; Wolfcamp aquifer.
estimator is about the same as kriging in cross validation. The kriging neighborhoods were the same as the variogram neighborhoods, both were 15% limited variance neighborhoods. This information is of value when confidence intervals are required for a surface with drift.

4.4.2 Cadmium Analyses

Figure 4.27 shows the results of the nearest neighbor analysis of the Cadmium data set. It is apparent that not enough low values were deleted from the data subset for variogram analysis. Four additional low value points shown in Figure 4.27 are very different from their nearest neighbors. By deleting these points and using the reduced data set, better nearest neighbor results are obtained as shown in Figure 4.28. Here a cut off of 2.0 gives a much better nearest neighbor estimate versus data graph. The best isotropic kernel estimates obtained with a radius of influence of 10% of the maximum distance are only slightly better as shown in Figure 4.29.

A second evaluation, for the reduced data set, produced the design experimental variogram and model shown in Figure 4.30. For this new analysis the model based on fitting 24 out of 40 points is an exponential with a zero nugget, a sill of 10.4 and a range of 68. While the parameters are very similar to the first analysis, the model changed from a spherical to an exponential. The cross validation estimates versus data are shown in Figure 4.31. These results are better than those obtained in study 1 but still offset from the slope of 1.0 shown as a solid line in the graph. As a test, the kriging neighborhood was reduced to $N = 5$ as shown in
Figure 4.27. Nearest neighbor estimates versus data values at 1.0 cut off; Cadmium data.
Figure 4.28. Nearest neighbor estimates versus data values at 2.0 cut off; Cadmium data.
Figure 4.29. Isotropic two dimensional kernel estimates versus data values; Cadmium data.
Figure 4.30. Final experimental variogram and model for 90% limited variance neighborhood at 2.0 cut off; Cadmium data.
Figure 4.31. Estimated values from simple kriging for variogram model of 0.0 nugget, 10.4 sill, 68.0 range and 90% limited variance neighborhood means at 2.0 cut off versus data values; Cadmium data.
Figure 4.32. This is a set of much smaller neighborhoods, than the 90% limited variance neighborhoods produced in the first analyses. The new cross validation results show that the best neighborhood for variogram evaluation is not necessarily the best one for kriging. This result will be discussed in Chapter 5.

Estimated surfaces for both the Wolfcamp data set and both of the cadmium data sets are shown in Figures 4.33, 4.34 and 4.35. These surfaces illustrate the relative complexities of the Cadmium data set compared to the Wolfcamp aquifer data set and to provide a comparison of the difference in the two kriged cadmium surfaces using large and small neighborhoods. The kernel estimates for the Wolfcamp aquifer are shown in Figure 4.33. The different surfaces for cadmium use the Study 3 design parameters of a 90% limited variance neighborhood and fixed N neighborhoods respectively.

4.5 Study 4: Comparing Kriging With Constrained Optimal Neighborhoods and Kernel Estimated Experimental Variograms to Universal Kriging

Several Articles on the use of universal kriging included the data sets or have referenced the data set in another publication. Four such data sets were found in the literature. These sets are evaluated using the combined protocols developed in Studies 1 and 2. If special problems arose, then the results of Study 3 were used to augment the evaluation.

4.5.1 Robena Thickness Analyses

The first data set is that of the Robena Mine thickness used in a regression analysis by Gomez and Hazen [1970] and also used by Journel and Rossi [1989] to
Figure 4.32. Estimated values from simple kriging for variogram model of 0.0 nugget, 10.4 sill, 68.0 range and fixed N = 5 neighborhood means at 2.0 cut off versus data values; Cadmium data.
Figure 4.33. Estimated surface for kernel estimate; Wolfcamp aquifer.
Figure 4.34. Estimated surface from simple kriging for variogram model of 0.0 nugget, 10.4 sill, 68.0 range and 90% limited variance neighborhood means at 2.0 cut off; Cadmium data.
Figure 4.35. Estimated surface from simple kriging for variogram model of 0.0 nugget, 10.4 sill, 68.0 range and fixed $N = 5$ neighborhood means at 2.0 cutoff; Cadmium data.
compare ordinary and universal kriging. They did not calculate the linear trend or the variogram, but used those from an unpublished 1982 Stanford thesis by B. E. Buxton. The model has a nugget of 0.03, and two spherical components each with a sill of 0.06. One has a range of 5000 feet and the other a range of 15,000 feet. They split the 208 point data set into a 108 point working data set and a 100 point reference data set. They used the working set to predict the values at the reference set locations using both ordinary and universal kriging. Since they used a model based on the entire data set, they did not have to consider the effect of the split on the variogram.

Using the 208 point data set from Journel and Rossi, experimental variograms for various values of smoothing parameters were calculated for a 50% limited variance neighborhood and are shown in Figure 4.36. The neighborhood analyses appear to be consistent with the small variance to mean ratio; relatively large neighborhoods are generated for the candidate neighborhoods. Here the need for a composite or undersmoothed kernel variogram is apparent. An undersmoothed kernel with a smoothing parameter of .001 was used to study from 25% to 50% limited variance neighborhoods and the graphs of those from 30% to 45% are shown in Figure 4.37. The 35% limited variance neighborhood produced the best model in data cross validation. This design experimental variogram and model for the 35% limited variance neighborhood with composite kernel bandwidths are shown in Figure 4.38. The estimated values versus the data values, obtained from the cross
Figure 4.36. Various kernel estimated experimental variograms for 50% limited variance neighborhoods; global lambdas: 0.01 dot, 0.005 dot-dash, 0.001 short dash, 0.005 long dash; Robena thickness.
Figure 4.37. Kernel estimated experimental variograms for various limited variance neighborhoods with global lambda 0.001: 45% dot, 40% dot-dash, 35% short dash, 30% long dash; Robena thickness.
Figure 4.38. Final experimental variogram model for entire data set; Robena Thickness.
validation, are shown in Figure 4.39. There are a few points that could be candidates for elimination, but in general the cloud of points is close to the 45° line.

Another evaluation of the data, using the working data set to predict the reference set in validation, produced a very similar model for the experimental variogram as shown in Figure 4.40. In this case many points are not close to the 45° line and are much more clustered as shown in Figure 4.41.

The surface calculated from the model of the entire data set is shown in Figure 4.42. The zero points outside the data field are included to illustrate the plateau dominance over the data trend. The trend is a small percentage of the mean value.

4.5.2 Robena Ash Analyses

The second data set is also from the Robena mine. It is the ash data taken from the same samples as the thickness data. Cressie [1986] used the entire 208 point data to demonstrate the use of median polish in kriging for gridded data sets with a trend. With this approach, he obtained a pure nugget model variogram of 0.75. Also Starks and Sparks [1987], using the same data set with various extremes in data values removed, also obtained a pure nugget variogram of 0.9 for standard neighborhoods of 8 nearest neighbors.

Preliminary data evaluation indicated that, although the ash data was more variable than the thickness data, they were correlated. So the same composite kernel parameters as for the Robena thickness were used to generate a set of experimental
Figure 4.39. Estimated values from simple kriging with varogram model of 0.0 nugget, 0.141 sill, 4950 range and 35% limited variance neighborhood means with composite smoothing parameter using the data working set in cross validation versus data values; Robena thickness.
Figure 4.40. Final experimental variogram and model for working data set; Robena thickness.
Figure 4.41. Estimated values from simple kriging with variogram model of 0.0 nugget, 0.141 sill, 4950 range and 35% limited variance neighborhood means with composite smoothing parameter using the data working set to predict reference set versus data values; Robena thickness.
Figure 4.42. Estimated surface from simple kriging with variogram model of 0.0 nugget, 0.141 sill, 4950 range and 35% neighborhood means; Robena thickness.
variograms for different neighborhoods of ash data. The results for the ash data which are shown in Figure 4.43 does mimic the thickness data in the undulation of the sill. The experimental variogram for the 47.5% limited variance neighborhood that produced the best model in cross validation is shown in Figure 4.44 with the model. This is definitely not a pure nugget variogram; in fact, the model nugget is zero with a sill of 0.97 and a range of 4700. The estimated values versus the data for cross validation, shown in Figure 4.45, demonstrate that there is much more scatter than for the thickness data and this data set could probably use editing to improve the results. The surface, shown in Figure 4.46, demonstrates that it is indeed similar to the thickness surface, but that one area of high values is out of place for the trend to be as dominant as it is for the thickness data. The important result is obtaining a full set of model parameters.

4.5.3 Horizon Depth Analyses

The third data set is a data set that Dubrule [1984] used to compare kriging with splines. His data of the depth to a geologic horizon was presented graphically and required digitizing to include in this study.

Dubrule used a depth to a geological horizon marker data set. He used 82 points for the working set and 30 points for the reference data set. He used the working set to generate the variogram and predict the values at the locations of the reference set. Structural analyses with BLUEPACK produced a trend of order two. Structural analyses with constrained optimal neighborhoods showed that the second
Figure 4.43. Kernel estimated experimental variograms for various neighborhoods with global lambda 0.001: 50% dot, 45% dot-dash, 40% dash; Robena ash.
Figure 4.44. Final experimental variogram and model for working data set with 47.5% limited variance neighborhoods; Robena ash.
Figure 4.45. Estimated values from simple kriging with variogram model of 0.0 nugget, 0.969 sill, 4700 range and 47.5% limited variance neighborhood means using the working data set to validate reference data set versus data values; Robena ash.
Figure 4.46. Estimated surface from simple kriging with variogram model of 0.0 nugget, 0.969 sill, 4700 range and 47.5% limited variance neighborhood means; Robena Ash.
order trend dominates the variogram for very low values of the limiting variance as shown in Figure 4.47. The experimental variograms do not exhibit a common plateau as in the earlier studies. As a result, these curves only indicate a range of possible variogram parameters; these are because the nugget is less than 0.01 and a range should be located somewhere between a distance of 4000 to 8000 lag units. The sill should be between 0.02 and 0.1. A design, 15% limited variance neighborhood, is obtained by cross validation of the data with the smaller (10-20%) neighborhoods.

The design experimental variogram and model with a nugget of 0.005, a sill of 0.037, and a range of 5500 are shown in Figure 4.48. The estimated values versus the data, shown in Figure 4.49 indicate a good fit.

Validation of the reference data set with the same set of neighborhoods used for the data structural analysis showed good results for all reference points inside the convex hull of the working data set. One point outside this hull which dominated the results was eliminated from the reference data set. The results are compared to Dubrule's results in Table 5.11. While there is slightly more divergence for a few points, the results are encouraging especially since the drift order is two. This particular data set probably requires universal kriging for extrapolation outside the region containing data; however, constrained optimal neighborhoods should still be used for variogram evaluation. The partial surface shown in Figure 4.50 demonstrates that the response surface for OK outside the convex hull of the data set levels off as opposed to universal kriging where the surface follows the drift.
Figure 4.47. Kernel estimated experimental variograms for various neighborhoods with global lambda 0.001: 50% dot, 30% dot-dash, 20% short dash, 10% long dash; Horizon depth.
Figure 4.48. Final experimental variogram and model for data working set with 15% limited variance neighborhoods; Horizon depth.
Figure 4.49. Estimated values from simple kriging with variogram model of 0.00539 nugget, 0.0366 sill, 5500 range and 15% limited variance neighborhood means for the data working set versus data values; Horizon depth.
Figure 4.50. Estimated south half surface from simple kriging with variogram model of 0.00539 nugget, 0.0366 sill, 5500 range and 15% neighborhood means; Horizon depth.
The nearest neighbor estimates versus the data are shown in Figure 4.51 and the best kernel estimate versus the data with a radius of influence of 8.75% of the maximum distance is shown in Figure 4.52. The best kernel estimates are better than nearest neighbor and comparable to the kriging estimates obtained with the design parameters.

4.5.4 Earthquake Acceleration Analyses

The fourth data set evaluated is a small data set of earthquake accelerations that Carr [1990] used to demonstrate a universal kriging program but did not indicate the variogram determination method. He used a nugget of 600, a sill of 2340, and a range of 100 for all drift orders, 0 to 3. This data set is a difficult one; the locations are very irregular, and the distribution is bimodal. He used cross validation to test drift orders of 0, 1, 2, and 3. The ordinary kriging results showed the least relief in the response surface. Kriged values varied from about 80 to 140 gals. The linear drift model showed more relief going from 60 to 140 gals. The quadratic drift model produced the most relief going from negative to 160 gals. The quadratic drift model increased the peak values and located it in an area of missing data. Results with constrained optimal neighborhoods indicate that the data set has a relatively high nugget compared to the sill as shown in Figure 4.53. The high nugget allows for a global smoothing parameter to be used for the variogram. By evaluating the edited dataset and comparing it to Carr's results, this study shows that the choice of a drift order can adversely affect the results for data sets with inadequate coverage. The surface
Figure 4.51. Nearest neighbor estimates versus data values; Horizon depth.
Figure 4.52. Isotropic two dimensional exponential kernel estimates versus data values; Horizon depth.
Figure 4.53. Kernel estimated experimental variograms for various neighborhoods with global lambda 1.0: 90% dot, 75% dot-dash, 50% dash; Earthquake acceleration.
obtained from this analysis and shown in Figure 4.54 is much different than that obtained by Carr using a universal kriging program with second order drift. By requiring the analyst to choose a sample drift model, the results are subject to severe extrapolations as in polynomial interpolation. When using constrained optimal neighborhoods this choice does not have to be made and the analysis is more data driven. Here the data do not indicate a peak, where the second order drift model locates one in an area devoid of data locations.

4.6 Study 5: Confidence Limits for Kernel Estimation of Horizontal Depth Data Set

Confidence intervals can not be obtained from kriging estimators with drift, but they can be obtained from kernel estimators. Dubrule [1984] set aside 30 points for a reference data set. By using the best kernel estimator for the data set a confidence interval on the parameters can be obtained. Figure 4.55 shows the root mean square error of the reference data validation using the working data set. An upper limit for a confidence interval is shown. This limit corresponds to the upper and lower parameter bounds required. In Figure 4.56, the best kernel estimate with the bounding curves are shown for a traverse from point 47 to point 19. The kriged values for the design parameters are compared to the kernel estimate for the same traverse in Figure 4.57. The curve estimated by kriging with design parameter values does lie within the bounds of the kernel estimates.
Figure 4.54. Estimated surface from simple kriging with variogram model of 745 nugget, 614 sill, 8.7 range and 75% limited variance neighborhood means versus data values; Earthquake acceleration.
Figure 4.55. Root mean square error of reference set validation versus effective neighborhood radius using kernel estimator with confidence limit; Horizon depth.
Figure 4.56. Profile of kernel estimates and limits plus nearest neighbor estimates versus distance from point 47 to point 19; solid, dot-dash, dot respectively; Horizon depth.
Figure 4.57. Profile of kriged and kernel estimates from point 47 to point 19; solid and dash respectively; Horizon depth.
CHAPTER 5
CONCLUSIONS

5.1 Introduction

The general objectives of this work have been accomplished. Preprocessing the data improved the applications of kriging as the studies of Chapter 4 indicate. The three main areas of application, constrained optimal neighborhoods, 1-D kernel smoothed experimental variograms, and the 2-D or 3-D kernel estimators will be summarized in order.

5.2 Constrained Optimal Neighborhoods

Constrained optimal neighborhood kriging has been shown in Studies 1 and 4 to avoid most of the effects of drift on geostatistical data analyses. In most cases of data analyses, with or without drift, neighborhood definition by the use of simple kriging via cross validation produces good results.

For most constrained optimal neighborhoods, the weight of the mean is usually small enough that simple and ordinary kriging yield the same estimates; however, ordinary kriging should be used in interpolation to provide the appropriate kriging variance. Simple kriging with a calculated mean for each neighborhood is used to test the adequacy of the size of the neighborhood, which occurs when simple kriging and ordinary kriging give approximately the same results (when the weight of the mean is small).
When smaller neighborhoods are used because of trend, the same neighborhoods can be used for experimental variogram calculation and kriging. For data sets not exhibiting a trend, and where large numbers of points were used to calculate the experimental variogram, smaller neighborhoods give better results in kriging cross validation. Others have noted this loss of accuracy in the tails of the data distribution. For example, Boufassa and Armstrong [1989] noted that simple kriging was biased towards the known mean; areas of low density sampling were biased towards the mean whether the data values were high or low. Spease and Carr [1988] proposed using biased kriging in order to recover the tails of the distribution. This loss of accuracy in cross validation has been observed in the present work with some of the data sets. Oddly enough it is the data sets, apparently exhibiting stationarity for which large neighborhoods were used to provide a better variogram definition, that exhibited this effect. Data sets requiring small neighborhoods did not produce a loss of accuracy in the tails of the data distribution and did not show a noticeable smoothing of the resulting interpolated surface.

Problems with universal kriging have been documented by several others. Armstrong [1984] noted that if a drift free direction is not easily identified there are additional difficulties in the variogram estimation. In that case ... "it is extremely difficult to deduce either the degree of the drift or the type of the underlying variogram." Delfiner [1976] proposed using IRF-k kriging to avoid these problems, and Carr and Roberts [1989] proposed using ordinary kriging for earthquake data.
From the results of this study, whenever universal kriging is appropriate, as in Study 4c, these problems can be reduced, if not eliminated by using the procedures developed in this work.

Another benefit of using constrained optimal neighborhoods to calculate the experimental variogram is that the size of the pair comparison file is reduced which allows analyses of larger data sets. Since only standard neighborhoods can be specified for kriging at the present time, local neighborhoods generated via the preprocessing program allow the pair comparison file to be made to grow linearly with the size of the data set instead of quadratically.

5.3 Kernel Estimated Experimental Variograms

The results of Study 2 show that 1-D kernel estimation improves the calculation of the experimental variogram by separating the choice of the bandwidth from the choice of the number of experimental points to be calculated. In some cases the smoothing parameter can be specified by a constant; but, in others a functional variation of the smoothing parameter is required to adequately estimate the variogram in areas where rapid change is occurring in the variogram. A simple piecewise linear function was used for the smoothing parameter. (The smoothing parameter appeared to linearize better than the bandwidth.) While some preliminary work has been done examining different criteria for automatic bandwidth parameter selection, including which functions to use, and when, no definitive conclusions have been made in this study.
Tables 5.1 and 5.2 compare results obtained in this work to those given in various references.

The most important result in the comparison is that constrained optimal neighborhoods allow for local definition of a variogram sill for the horizon depth data of Study 4 with its order 2 drift and generalized covariance function. The most striking results are the differences in model parameters, especially for ash where a full set of parameters were obtained with a zero nugget rather than the pure nugget result of the other authors, and the increased number of zero nuggets in general.

5.4 Kernel Estimation as an Alternative to Kriging

The results of Studies 3 and 5 show that 2-D kernel estimation can be used as an alternative estimator where confidence limits are needed or where the speed of computation exceed the need for precision. In most of the cases the kernel estimates were generally comparable to the kriging estimates as shown in the estimates versus data value graphs which indicates that a good kernel approximation to kriging can probably be found most of the time.

Cross validation of the data was used to yield isotropic smoothing parameters. This could probably be improved upon by using anisotropic kernels with more than one smoothing parameter. No true 3-D data sets were available for study, but should be particularly suitable for isotropic kernel estimation because of the general sparsity of data in this space.
Table 5.1. Variogram model parameters calculated with programs.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Nbhd</th>
<th>Vgrm</th>
<th>No. Fit</th>
<th>Model</th>
<th>Nugget</th>
<th>Sill</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wolfcamp</td>
<td>15%</td>
<td>Standard</td>
<td>7/15</td>
<td>Exp</td>
<td>8000</td>
<td>28000</td>
<td>78.75</td>
</tr>
<tr>
<td>Cadmium</td>
<td>50%</td>
<td>Standard</td>
<td>12/20</td>
<td>Sph</td>
<td>0.0</td>
<td>10.9</td>
<td>61.0</td>
</tr>
<tr>
<td>Cadmium</td>
<td>90%</td>
<td>Kernel</td>
<td>24/40</td>
<td>Exp</td>
<td>0.0</td>
<td>10.4</td>
<td>68.0</td>
</tr>
<tr>
<td>Thickness</td>
<td>30%</td>
<td>Kernel</td>
<td>12/20</td>
<td>Sph</td>
<td>0.0</td>
<td>0.141</td>
<td>4950</td>
</tr>
<tr>
<td>Ash</td>
<td>47.5%</td>
<td>Kernel</td>
<td>12/20</td>
<td>Sph</td>
<td>0.0</td>
<td>0.969</td>
<td>4700</td>
</tr>
<tr>
<td>Hor. Depth</td>
<td>15%</td>
<td>Kernel</td>
<td>9/15</td>
<td>Gauss</td>
<td>0.00539</td>
<td>0.0366</td>
<td>5500</td>
</tr>
<tr>
<td>E.Q. Accel.</td>
<td>60%</td>
<td>Kernel</td>
<td>11/</td>
<td>Exp</td>
<td>745</td>
<td>614</td>
<td>8.7</td>
</tr>
</tbody>
</table>

Table 5.2. Variogram model parameters from other authors.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Author(s)</th>
<th>Model</th>
<th>Nugget</th>
<th>Sill</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wolfcamp</td>
<td>Harper and Furr</td>
<td>Sph</td>
<td>11,000</td>
<td>34,000</td>
<td>60</td>
</tr>
<tr>
<td>Cadmium</td>
<td>Englund and Sparks</td>
<td>Exp</td>
<td>4.5</td>
<td>13.5</td>
<td>160</td>
</tr>
<tr>
<td>Thickness</td>
<td>Journel and Rossi</td>
<td>Nested Sph</td>
<td>0.03</td>
<td>0.06</td>
<td>5000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.06</td>
<td>15000</td>
</tr>
<tr>
<td>Ash</td>
<td>Cressie</td>
<td>Nugget</td>
<td>0.75</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ash</td>
<td>Starks and Sparks</td>
<td>Nugget</td>
<td>0.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hor. Depth</td>
<td>Dubrule</td>
<td>Gen. Cov.*</td>
<td>0.0</td>
<td>-3.118/h/</td>
<td>-</td>
</tr>
<tr>
<td>E.Q. Accel.</td>
<td>Carr</td>
<td>Sph</td>
<td>600</td>
<td>2340</td>
<td>10</td>
</tr>
</tbody>
</table>

*General cov. functional form is given
CHAPTER 6

FUTURE WORK

6.1 Improved Parameters for Kernel Estimator

Further work is required to evaluate the best functional forms to use for variable smoothing parameters and when to use a variable smoothing parameter.

6.2 Automation

Kernel estimation, as an auxiliary to kriging, has been shown to enhance variogram calculations, especially for a novice user of geostatistics. From the results obtained in this work, it appears that it would be feasible to automate experimental variogram calculations using kernel estimation. This would allow for variogram calculations with a minimum of choices by a novice user. Hopefully this would reduce the differences in results described by Englund [1990]. In addition the parameters for 2-D or 3-D kernel functions could probably also be determined automatically. The pilot method described by Muller [1986] may be suitable for the automation of the determination of both the experimental variogram and model. Possibly other constraints, as those on data values described by Barnes [1992], could be incorporated into this work.

6.3 Combined Kriging and Kernel Estimators

There are special circumstances where the advantages of both kriging and kernel estimation can be combined. Data sets are rarely truly three dimensional. In general they are contained in lower dimensional spaces. Examples include a layered
aquifer, or a series of core holes that are spaced far apart in comparison to their depth.

6.3.1 Profiles in Space

Core hole data that are from widely spaced bore holes would require an anisotropic variogram with the vertical variogram scale being very small compared to the horizontal variogram scale. Instead of attempting to krig this type of data in three dimensions directly, an alternative method is to first cokrige important data horizons and the distances between them to produce maps of important two dimensional features, and then to interpolate profile parameters for each boring with a kernel estimator to desired points. Basically this would be a method of interpolating profiles directly and changes in profiles could be estimated along a particular transect.

6.3.2 Profiles in Time

Monitoring data through time can also be evaluated similarly to that of profiles in 3-D. The spatial horizons of interest such as maxima and minima and their times of occurrence could be kriged or cokriged and profile parameters from the monitoring stations interpolated by general kernel estimators. Profiles at any desired point could then be generated based on the kriged values and the interpolated profile parameters at the desired point. Some work has been done to extend kriging to a spatiotemporal setting and many problems have been identified by authors including Rouhani and Myers [1990]. Combining kriging and kernel estimation may reduce some of the problems.
6.4 Extension to Cokriging

An extension of this work to cokriging is recommended. There are many areas of additional study, among them: 1) use neighborhoods and correlation matrices to subdivide data sets with a large number of variables into cokriged subsets; 2) find criteria to determine the best composite neighborhoods; 3) develop kernel smoothed experimental cross variograms and determine the effect on the work of Goulard and Voltz [1992]. Adding a bandwidth would add another dimension to their optimization problem; 4) determine the effect of neighborhoods on cokriging nonstationary data per Stein et al. [1991]; and 5) evaluate the potential for kernel approximation of the intervariable weights in cokriging with generalized kernels such as \( \sin(r)/r \) that may take on negative values for values of \( r \) greater than \( \pi \).
APPENDIX A

DATA SETS, HISTOGRAMS, AND LOCATION MAPS

A.1 Wolfcamp aquifer data set from Harper and Furr [1986]

Note the modified GEOEAS header adding the number of independent variables after the total number of variables.

Wolfcamp Aquifer Potentiometric Heads with Residuals

<table>
<thead>
<tr>
<th>easting (ft.)</th>
<th>northing (ft.)</th>
<th>head (ft.)</th>
<th>lin res (ft.)</th>
<th>Quad res (ft.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>42.783</td>
<td>127.623</td>
<td>1464</td>
<td>-74.64</td>
<td>-146.20</td>
</tr>
<tr>
<td>-27.397</td>
<td>90.787</td>
<td>2553</td>
<td>319.96</td>
<td>296.04</td>
</tr>
<tr>
<td>-1.163</td>
<td>84.896</td>
<td>2158</td>
<td>67.02</td>
<td>20.47</td>
</tr>
<tr>
<td>-18.618</td>
<td>76.452</td>
<td>2455</td>
<td>195.59</td>
<td>155.46</td>
</tr>
<tr>
<td>96.465</td>
<td>64.581</td>
<td>1756</td>
<td>202.70</td>
<td>136.96</td>
</tr>
<tr>
<td>108.562</td>
<td>82.923</td>
<td>1702</td>
<td>339.85</td>
<td>210.41</td>
</tr>
<tr>
<td>88.364</td>
<td>56.453</td>
<td>1805</td>
<td>148.34</td>
<td>112.66</td>
</tr>
<tr>
<td>90.042</td>
<td>39.258</td>
<td>1797</td>
<td>48.62</td>
<td>81.57</td>
</tr>
<tr>
<td>93.173</td>
<td>33.059</td>
<td>1714</td>
<td>-50.52</td>
<td>12.44</td>
</tr>
<tr>
<td>97.611</td>
<td>56.279</td>
<td>1466</td>
<td>-129.40</td>
<td>-165.76</td>
</tr>
<tr>
<td>90.629</td>
<td>35.082</td>
<td>1729</td>
<td>-40.54</td>
<td>11.67</td>
</tr>
<tr>
<td>92.553</td>
<td>41.752</td>
<td>1638</td>
<td>-78.34</td>
<td>-55.78</td>
</tr>
<tr>
<td>99.490</td>
<td>59.158</td>
<td>1736</td>
<td>170.40</td>
<td>122.97</td>
</tr>
<tr>
<td>-24.067</td>
<td>184.766</td>
<td>1476</td>
<td>171.67</td>
<td>130.55</td>
</tr>
<tr>
<td>-26.063</td>
<td>114.075</td>
<td>2200</td>
<td>115.43</td>
<td>132.52</td>
</tr>
<tr>
<td>56.278</td>
<td>26.848</td>
<td>1999</td>
<td>-51.67</td>
<td>16.53</td>
</tr>
<tr>
<td>73.039</td>
<td>18.881</td>
<td>1680</td>
<td>-303.12</td>
<td>-186.85</td>
</tr>
<tr>
<td>80.267</td>
<td>12.616</td>
<td>1806</td>
<td>-167.74</td>
<td>-9.71</td>
</tr>
<tr>
<td>80.230</td>
<td>14.618</td>
<td>1682</td>
<td>-280.02</td>
<td>-132.87</td>
</tr>
<tr>
<td>68.838</td>
<td>107.774</td>
<td>1306</td>
<td>-175.30</td>
<td>-298.09</td>
</tr>
<tr>
<td>76.399</td>
<td>95.994</td>
<td>1722</td>
<td>221.32</td>
<td>98.79</td>
</tr>
<tr>
<td>64.461</td>
<td>110.396</td>
<td>1437</td>
<td>-58.06</td>
<td>-175.25</td>
</tr>
<tr>
<td>43.397</td>
<td>53.615</td>
<td>1828</td>
<td>-149.34</td>
<td>-174.68</td>
</tr>
<tr>
<td>39.078</td>
<td>61.998</td>
<td>2118</td>
<td>161.67</td>
<td>118.06</td>
</tr>
<tr>
<td>12.804</td>
<td>45.548</td>
<td>1725</td>
<td>167.76</td>
<td>176.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>54.259</td>
<td>147.820</td>
<td>1606.</td>
<td>265.97</td>
<td>200.39</td>
</tr>
<tr>
<td>6.132</td>
<td>48.328</td>
<td>2648.</td>
<td>387.57</td>
<td>366.35</td>
</tr>
<tr>
<td>-3.805</td>
<td>40.404</td>
<td>2560.</td>
<td>185.01</td>
<td>173.72</td>
</tr>
<tr>
<td>-2.231</td>
<td>29.911</td>
<td>2544.</td>
<td>116.50</td>
<td>128.79</td>
</tr>
<tr>
<td>-2.362</td>
<td>33.820</td>
<td>2386.</td>
<td>-18.86</td>
<td>-15.78</td>
</tr>
<tr>
<td>-2.189</td>
<td>33.682</td>
<td>2400.</td>
<td>-4.52</td>
<td>-1.00</td>
</tr>
<tr>
<td>63.224</td>
<td>79.499</td>
<td>1757.</td>
<td>68.57</td>
<td>-17.17</td>
</tr>
<tr>
<td>-10.779</td>
<td>175.113</td>
<td>1402.</td>
<td>-213.99</td>
<td>-5.97</td>
</tr>
<tr>
<td>-18.989</td>
<td>171.917</td>
<td>1364.</td>
<td>-326.27</td>
<td>-109.53</td>
</tr>
<tr>
<td>-38.579</td>
<td>158.527</td>
<td>1735.</td>
<td>-167.83</td>
<td>35.56</td>
</tr>
<tr>
<td>83.145</td>
<td>159.116</td>
<td>1376.</td>
<td>298.38</td>
<td>175.30</td>
</tr>
<tr>
<td>-21.802</td>
<td>15.026</td>
<td>2729.</td>
<td>80.46</td>
<td>109.52</td>
</tr>
<tr>
<td>-23.565</td>
<td>9.414</td>
<td>2766.</td>
<td>71.76</td>
<td>113.90</td>
</tr>
<tr>
<td>-20.113</td>
<td>22.093</td>
<td>2736.</td>
<td>141.33</td>
<td>154.87</td>
</tr>
<tr>
<td>-16.627</td>
<td>17.256</td>
<td>2432.</td>
<td>-168.08</td>
<td>-138.53</td>
</tr>
<tr>
<td>29.907</td>
<td>175.129</td>
<td>1024.</td>
<td>-317.27</td>
<td>-235.28</td>
</tr>
<tr>
<td>100.916</td>
<td>22.978</td>
<td>1611.</td>
<td>-161.58</td>
<td>-41.90</td>
</tr>
<tr>
<td>101.295</td>
<td>22.964</td>
<td>1548.</td>
<td>-221.98</td>
<td>-102.06</td>
</tr>
<tr>
<td>103.266</td>
<td>20.342</td>
<td>1591.</td>
<td>-181.16</td>
<td>-45.22</td>
</tr>
<tr>
<td>-14.311</td>
<td>31.265</td>
<td>2540.</td>
<td>39.29</td>
<td>38.51</td>
</tr>
<tr>
<td>-18.134</td>
<td>30.181</td>
<td>2352.</td>
<td>-181.07</td>
<td>-182.89</td>
</tr>
<tr>
<td>-18.122</td>
<td>29.532</td>
<td>2528.</td>
<td>-8.69</td>
<td>-9.27</td>
</tr>
<tr>
<td>-9.888</td>
<td>38.145</td>
<td>2575.</td>
<td>145.36</td>
<td>134.64</td>
</tr>
<tr>
<td>-12.163</td>
<td>39.111</td>
<td>2468.</td>
<td>28.80</td>
<td>14.98</td>
</tr>
<tr>
<td>11.658</td>
<td>18.733</td>
<td>2646.</td>
<td>245.48</td>
<td>303.15</td>
</tr>
<tr>
<td>61.691</td>
<td>32.494</td>
<td>1739.</td>
<td>-241.37</td>
<td>-192.52</td>
</tr>
<tr>
<td>69.579</td>
<td>33.808</td>
<td>1674.</td>
<td>-245.16</td>
<td>-197.40</td>
</tr>
<tr>
<td>66.722</td>
<td>33.933</td>
<td>1868.</td>
<td>-69.82</td>
<td>-23.98</td>
</tr>
<tr>
<td>-36.654</td>
<td>150.915</td>
<td>1865.</td>
<td>-70.32</td>
<td>92.74</td>
</tr>
<tr>
<td>-19.551</td>
<td>137.784</td>
<td>1777.</td>
<td>-121.37</td>
<td>-46.92</td>
</tr>
<tr>
<td>-21.298</td>
<td>131.825</td>
<td>1579.</td>
<td>-367.04</td>
<td>-308.69</td>
</tr>
<tr>
<td>-22.362</td>
<td>137.137</td>
<td>1771.</td>
<td>-150.41</td>
<td>-72.64</td>
</tr>
<tr>
<td>21.147</td>
<td>139.262</td>
<td>1408.</td>
<td>-206.88</td>
<td>-213.46</td>
</tr>
<tr>
<td>7.685</td>
<td>126.838</td>
<td>1527.</td>
<td>-253.28</td>
<td>-260.89</td>
</tr>
<tr>
<td>-8.332</td>
<td>107.777</td>
<td>2003.</td>
<td>0.67</td>
<td>-17.75</td>
</tr>
<tr>
<td>56.707</td>
<td>171.264</td>
<td>1386.</td>
<td>202.59</td>
<td>187.15</td>
</tr>
<tr>
<td>59.001</td>
<td>164.549</td>
<td>1089.</td>
<td>-118.83</td>
<td>-159.81</td>
</tr>
<tr>
<td>68.969</td>
<td>177.248</td>
<td>1384.</td>
<td>319.31</td>
<td>282.12</td>
</tr>
<tr>
<td>70.902</td>
<td>161.381</td>
<td>1030.</td>
<td>-116.44</td>
<td>-199.76</td>
</tr>
<tr>
<td>73.002</td>
<td>162.990</td>
<td>1092.</td>
<td>-30.60</td>
<td>-116.63</td>
</tr>
<tr>
<td>59.662</td>
<td>170.105</td>
<td>1161.</td>
<td>-9.29</td>
<td>-37.26</td>
</tr>
<tr>
<td>61.874</td>
<td>174.302</td>
<td>1415.</td>
<td>284.97</td>
<td>262.36</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>63.708</td>
<td>173.915</td>
<td>1231.</td>
<td>111.00</td>
<td>81.32</td>
</tr>
<tr>
<td>5.627</td>
<td>79.087</td>
<td>2300.</td>
<td>220.22</td>
<td>169.49</td>
</tr>
<tr>
<td>18.247</td>
<td>77.392</td>
<td>2238.</td>
<td>233.27</td>
<td>176.80</td>
</tr>
<tr>
<td>85.688</td>
<td>139.817</td>
<td>1038.</td>
<td>-137.83</td>
<td>-290.00</td>
</tr>
<tr>
<td>105.076</td>
<td>132.032</td>
<td>1332.</td>
<td>240.34</td>
<td>42.19</td>
</tr>
</tbody>
</table>
| -101.643  |  10.651 |  3510. |  295.96 |  216.40 | *
| -145.237  |  28.023 |  3490. |   85.73 |  -51.73 | *
|  -73.993  |  87.973 |  2594. |   29.83 |    30.25 |
|  -94.482  |  86.626 |  2650. |  -60.61 |  -53.89 |
|  -88.850  |  76.710 |  2533. | -199.16 | -219.43 |
| -120.259  |  80.765 |  3571. |  651.24 |  648.22 | *
|  -86.025  |  54.363 |  2811. |  -35.68 |  -91.97 |
|  -72.791  |  43.092 |  2728. |  -96.87 | -152.59 |
|  -100.174 |  42.899 |  3136. |  124.94 |    49.56 |
|  -78.835  |  40.821 |  2553. | -326.25 | -386.62 |
|  -83.691  |  46.505 |  2798. |  -80.19 | -141.71 |
|  -95.617  |  35.822 |  2691. | -331.44 | -407.32 |
|  -87.555  |  29.393 |  2946. |  -60.73 | -129.24 |
Figure A.1.1. Histogram for Wolfcamp aquifer data.
Figure A.1.2. Location map for Wolfcamp aquifer data.
A.2 GEOEAS Example1 Data Set from Englund and Sparks [1990]

Note the modified GEOEAS header adding the number of independent variables after the total number of variables.

Example1.dat (* deleted after neighborhood analyses)
5 2 (** deleted after nearest neighbor analyses)

<table>
<thead>
<tr>
<th>Easting feet</th>
<th>Northing feet</th>
<th>Arsenic ppm</th>
<th>Cadmium ppm</th>
<th>Lead ppm</th>
</tr>
</thead>
<tbody>
<tr>
<td>288.0</td>
<td>311.0</td>
<td>.85</td>
<td>11.5</td>
<td>18.25 'Sample 1'</td>
</tr>
<tr>
<td>285.6</td>
<td>288.0</td>
<td>.63</td>
<td>8.5</td>
<td>30.25 'Sample 2'</td>
</tr>
<tr>
<td>276.0</td>
<td>269.0</td>
<td>1.02</td>
<td>7.0</td>
<td>20.00 'Sample 3'</td>
</tr>
<tr>
<td>285.6</td>
<td>249.0</td>
<td>1.02</td>
<td>10.7</td>
<td>19.25 'Sample 4'</td>
</tr>
<tr>
<td>276.0</td>
<td>231.0</td>
<td>1.01</td>
<td>11.2</td>
<td>151.50 'Sample 5'</td>
</tr>
<tr>
<td>276.0</td>
<td>206.0</td>
<td>1.47</td>
<td>11.6</td>
<td>37.50 'Sample 6'</td>
</tr>
<tr>
<td>285.6</td>
<td>180.0</td>
<td>.72</td>
<td>7.2</td>
<td>80.00 'Sample 7'</td>
</tr>
<tr>
<td>288.0</td>
<td>164.0</td>
<td>.30</td>
<td>5.7</td>
<td>46.00 'Sample 8'</td>
</tr>
<tr>
<td>292.8</td>
<td>137.0</td>
<td>.36</td>
<td>5.2</td>
<td>10.00 'Sample 9'</td>
</tr>
<tr>
<td>278.4</td>
<td>119.0</td>
<td>.70</td>
<td>7.2</td>
<td>13.00 'Sample 10'</td>
</tr>
<tr>
<td>360.0</td>
<td>315.0</td>
<td>.05</td>
<td>3.9</td>
<td>21.25 'Sample 11'</td>
</tr>
<tr>
<td>355.2</td>
<td>291.0</td>
<td>.71</td>
<td>9.5</td>
<td>16.75 'Sample 12'</td>
</tr>
<tr>
<td>367.2</td>
<td>270.0</td>
<td>1.32</td>
<td>8.9</td>
<td>55.00 'Sample 13'</td>
</tr>
<tr>
<td>367.2</td>
<td>250.0</td>
<td>1.33</td>
<td>11.5</td>
<td>122.20 'Sample 14'</td>
</tr>
<tr>
<td>352.8</td>
<td>226.0</td>
<td>.08</td>
<td>10.7</td>
<td>127.70 'Sample 15'</td>
</tr>
<tr>
<td>350.4</td>
<td>203.0</td>
<td>1.54</td>
<td>8.3</td>
<td>25.75 'Sample 16'</td>
</tr>
<tr>
<td>369.6</td>
<td>180.0</td>
<td>.71</td>
<td>6.1</td>
<td>21.50 'Sample 17'</td>
</tr>
<tr>
<td>369.6</td>
<td>165.0</td>
<td>.00</td>
<td>6.7</td>
<td>4.00 'Sample 18'</td>
</tr>
<tr>
<td>357.6</td>
<td>139.0</td>
<td>.24</td>
<td>6.2</td>
<td>4.25 'Sample 19'</td>
</tr>
<tr>
<td>355.2</td>
<td>118.0</td>
<td>.00</td>
<td>.0</td>
<td>9.50 'Sample 20'</td>
</tr>
<tr>
<td>434.4</td>
<td>310.0</td>
<td>.53</td>
<td>5.5</td>
<td>24.00 'Sample 21'</td>
</tr>
<tr>
<td>451.2</td>
<td>295.0</td>
<td>.37</td>
<td>4.0</td>
<td>9.50 'Sample 22'</td>
</tr>
<tr>
<td>448.8</td>
<td>268.0</td>
<td>.62</td>
<td>7.0</td>
<td>3.50 'Sample 23'</td>
</tr>
<tr>
<td>430.0</td>
<td>250.0</td>
<td>.12</td>
<td>5.3</td>
<td>16.25 'Sample 24'</td>
</tr>
<tr>
<td>441.6</td>
<td>228.0</td>
<td>1.06</td>
<td>11.6</td>
<td>18.00 'Sample 25'</td>
</tr>
<tr>
<td>441.6</td>
<td>204.0</td>
<td>1.58</td>
<td>9.0</td>
<td>56.50 'Sample 26'</td>
</tr>
<tr>
<td>444.0</td>
<td>180.0</td>
<td>1.57</td>
<td>14.5</td>
<td>118.00 'Sample 27'</td>
</tr>
<tr>
<td>441.6</td>
<td>160.0</td>
<td>1.09</td>
<td>12.1</td>
<td>31.00 'Sample 28'</td>
</tr>
<tr>
<td>430.0</td>
<td>140.0</td>
<td>.00</td>
<td>.9</td>
<td>12.25 'Sample 29'</td>
</tr>
<tr>
<td>444.0</td>
<td>119.0</td>
<td>.00</td>
<td>.0</td>
<td>1.00 'Sample 30'</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>254.4</td>
<td>170.0</td>
<td>.26</td>
<td>3.2</td>
<td>19.75 'Sample 31'</td>
</tr>
<tr>
<td>254.4</td>
<td>128.0</td>
<td>.00</td>
<td>1.2</td>
<td>4.50 'Sample 32'</td>
</tr>
<tr>
<td>254.4</td>
<td>299.0</td>
<td>1.E31</td>
<td>1.7</td>
<td>14.50 'Sample 33' **</td>
</tr>
<tr>
<td>333.6</td>
<td>301.0</td>
<td>1.E31</td>
<td>1.2</td>
<td>25.50 'Sample 34' **</td>
</tr>
<tr>
<td>333.6</td>
<td>271.0</td>
<td>.95</td>
<td>7.6</td>
<td>36.25 'Sample 35'</td>
</tr>
<tr>
<td>333.6</td>
<td>194.0</td>
<td>.99</td>
<td>11.6</td>
<td>37.50 'Sample 36'</td>
</tr>
<tr>
<td>333.6</td>
<td>163.0</td>
<td>1.30</td>
<td>8.7</td>
<td>36.00 'Sample 37'</td>
</tr>
<tr>
<td>412.8</td>
<td>285.0</td>
<td>.51</td>
<td>5.8</td>
<td>32.25 'Sample 38'</td>
</tr>
<tr>
<td>254.4</td>
<td>257.0</td>
<td>.20</td>
<td>3.8</td>
<td>16.50 'Sample 39'</td>
</tr>
<tr>
<td>412.8</td>
<td>170.0</td>
<td>1.12</td>
<td>10.4</td>
<td>48.50 'Sample 40'</td>
</tr>
<tr>
<td>412.8</td>
<td>150.0</td>
<td>1.07</td>
<td>10.0</td>
<td>49.75 'Sample 41'</td>
</tr>
<tr>
<td>490.0</td>
<td>280.0</td>
<td>.61</td>
<td>7.1</td>
<td>14.25 'Sample 42'</td>
</tr>
<tr>
<td>490.0</td>
<td>249.0</td>
<td>.42</td>
<td>4.4</td>
<td>23.50 'Sample 43'</td>
</tr>
<tr>
<td>490.0</td>
<td>315.0</td>
<td>1.19</td>
<td>10.4</td>
<td>302.50 'Sample 44' **</td>
</tr>
<tr>
<td>490.0</td>
<td>150.0</td>
<td>.00</td>
<td>1.6</td>
<td>42.50 'Sample 45'</td>
</tr>
<tr>
<td>444.0</td>
<td>190.0</td>
<td>1.60</td>
<td>15.0</td>
<td>56.50 'Sample 46'</td>
</tr>
<tr>
<td>436.8</td>
<td>240.0</td>
<td>.60</td>
<td>3.4</td>
<td>12.25 'Sample 47'</td>
</tr>
<tr>
<td>360.0</td>
<td>195.0</td>
<td>1.07</td>
<td>6.8</td>
<td>33.25 'Sample 48'</td>
</tr>
<tr>
<td>345.6</td>
<td>210.0</td>
<td>5.61</td>
<td>10.8</td>
<td>59.00 'Sample 49'</td>
</tr>
<tr>
<td>254.4</td>
<td>216.0</td>
<td>.09</td>
<td>14.9</td>
<td>146.70 'Sample 50'</td>
</tr>
<tr>
<td>280.8</td>
<td>216.0</td>
<td>1.42</td>
<td>9.9</td>
<td>268.00 'Sample 51'</td>
</tr>
<tr>
<td>307.2</td>
<td>216.0</td>
<td>1.31</td>
<td>11.6</td>
<td>98.00 'Sample 52'</td>
</tr>
<tr>
<td>333.6</td>
<td>216.0</td>
<td>1.21</td>
<td>6.5</td>
<td>44.00 'Sample 53'</td>
</tr>
<tr>
<td>360.0</td>
<td>216.0</td>
<td>1.67</td>
<td>10.1</td>
<td>94.25 'Sample 54'</td>
</tr>
<tr>
<td>386.4</td>
<td>216.0</td>
<td>1.59</td>
<td>11.8</td>
<td>68.00 'Sample 55'</td>
</tr>
<tr>
<td>412.8</td>
<td>216.0</td>
<td>2.88</td>
<td>11.0</td>
<td>60.75 'Sample 56'</td>
</tr>
<tr>
<td>439.2</td>
<td>216.0</td>
<td>1.95</td>
<td>16.7</td>
<td>70.00 'Sample 57'</td>
</tr>
<tr>
<td>465.6</td>
<td>216.0</td>
<td>.93</td>
<td>11.6</td>
<td>25.00 'Sample 58'</td>
</tr>
<tr>
<td>490.0</td>
<td>216.0</td>
<td>.75</td>
<td>6.9</td>
<td>33.00 'Sample 59'</td>
</tr>
<tr>
<td>345.6</td>
<td>216.0</td>
<td>1.45</td>
<td>9.9</td>
<td>40.75 'Sample 60'</td>
</tr>
</tbody>
</table>
Figure A.2.1 Histogram for Cadmium data.
Figure A.2.2 Location map for Cadmium data.
A.3 Robena Mine Data Set From Gomez and Hazen [1970], Journel and Rossi [1989], and Cressie [1986].

Note the modified GEOEAS header adding the number of independent variables after the total number of variables.

Robena Mine Thickness and Ash Concentration

<table>
<thead>
<tr>
<th></th>
<th>x ft.</th>
<th>y ft.</th>
<th>z ft.</th>
<th>ash %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>65790</td>
<td>83415</td>
<td>7.50</td>
<td>11.31 *</td>
</tr>
<tr>
<td></td>
<td>65046</td>
<td>80891</td>
<td>6.42</td>
<td>11.51</td>
</tr>
<tr>
<td></td>
<td>64102</td>
<td>77702</td>
<td>7.50</td>
<td>10.01</td>
</tr>
<tr>
<td></td>
<td>66551</td>
<td>85992</td>
<td>7.92</td>
<td>9.92</td>
</tr>
<tr>
<td></td>
<td>64749</td>
<td>69276</td>
<td>7.50</td>
<td>10.21</td>
</tr>
<tr>
<td></td>
<td>67371</td>
<td>88578</td>
<td>7.95</td>
<td>10.73 *</td>
</tr>
<tr>
<td></td>
<td>68040</td>
<td>91041</td>
<td>7.62</td>
<td>10.82 *</td>
</tr>
<tr>
<td></td>
<td>65478</td>
<td>91814</td>
<td>7.92</td>
<td>9.92</td>
</tr>
<tr>
<td></td>
<td>66245</td>
<td>94343</td>
<td>8.08</td>
<td>11.17</td>
</tr>
<tr>
<td></td>
<td>68688</td>
<td>93648</td>
<td>7.56</td>
<td>10.14</td>
</tr>
<tr>
<td></td>
<td>64083</td>
<td>60576</td>
<td>7.08</td>
<td>10.59</td>
</tr>
<tr>
<td></td>
<td>71870</td>
<td>94776</td>
<td>7.75</td>
<td>9.97</td>
</tr>
<tr>
<td></td>
<td>71203</td>
<td>92610</td>
<td>7.58</td>
<td>9.93</td>
</tr>
<tr>
<td></td>
<td>70607</td>
<td>90580</td>
<td>7.50</td>
<td>11.65</td>
</tr>
<tr>
<td></td>
<td>70040</td>
<td>88545</td>
<td>7.40</td>
<td>9.46</td>
</tr>
<tr>
<td></td>
<td>69561</td>
<td>86635</td>
<td>7.19</td>
<td>12.50</td>
</tr>
<tr>
<td></td>
<td>68848</td>
<td>84585</td>
<td>7.42</td>
<td>11.05 *</td>
</tr>
<tr>
<td></td>
<td>68239</td>
<td>82514</td>
<td>7.21</td>
<td>9.41</td>
</tr>
<tr>
<td></td>
<td>67663</td>
<td>80505</td>
<td>7.29</td>
<td>9.91</td>
</tr>
<tr>
<td></td>
<td>66976</td>
<td>78646</td>
<td>7.42</td>
<td>10.82</td>
</tr>
<tr>
<td></td>
<td>66500</td>
<td>76518</td>
<td>7.12</td>
<td>8.23</td>
</tr>
<tr>
<td></td>
<td>65357</td>
<td>72497</td>
<td>7.54</td>
<td>10.41</td>
</tr>
<tr>
<td></td>
<td>64625</td>
<td>70449</td>
<td>7.46</td>
<td>9.76</td>
</tr>
<tr>
<td></td>
<td>64147</td>
<td>68412</td>
<td>7.43</td>
<td>10.93</td>
</tr>
<tr>
<td></td>
<td>66078</td>
<td>65724</td>
<td>6.73</td>
<td>9.64</td>
</tr>
<tr>
<td></td>
<td>66642</td>
<td>67630</td>
<td>6.92</td>
<td>10.94 *</td>
</tr>
<tr>
<td></td>
<td>67230</td>
<td>69922</td>
<td>7.00</td>
<td>11.10</td>
</tr>
<tr>
<td></td>
<td>67782</td>
<td>71737</td>
<td>6.90</td>
<td>10.82</td>
</tr>
<tr>
<td></td>
<td>68372</td>
<td>73660</td>
<td>6.96</td>
<td>11.11 *</td>
</tr>
<tr>
<td></td>
<td>68894</td>
<td>75751</td>
<td>7.50</td>
<td>11.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-----</td>
</tr>
<tr>
<td>69584</td>
<td>77827</td>
<td>7.42</td>
<td>11.75</td>
<td>*</td>
</tr>
<tr>
<td>70754</td>
<td>81907</td>
<td>7.36</td>
<td>9.37</td>
<td>*</td>
</tr>
<tr>
<td>71393</td>
<td>83813</td>
<td>7.41</td>
<td>10.11</td>
<td></td>
</tr>
<tr>
<td>71935</td>
<td>85817</td>
<td>7.54</td>
<td>9.63</td>
<td>*</td>
</tr>
<tr>
<td>72457</td>
<td>88142</td>
<td>7.21</td>
<td>9.35</td>
<td></td>
</tr>
<tr>
<td>72945</td>
<td>89955</td>
<td>7.33</td>
<td>8.96</td>
<td>*</td>
</tr>
<tr>
<td>73679</td>
<td>91803</td>
<td>7.37</td>
<td>10.27</td>
<td></td>
</tr>
<tr>
<td>74287</td>
<td>93957</td>
<td>7.29</td>
<td>9.70</td>
<td>*</td>
</tr>
<tr>
<td>75488</td>
<td>97951</td>
<td>8.23</td>
<td>10.74</td>
<td></td>
</tr>
<tr>
<td>76139</td>
<td>99961</td>
<td>8.23</td>
<td>9.79</td>
<td>*</td>
</tr>
<tr>
<td>79199</td>
<td>101234</td>
<td>7.25</td>
<td>10.39</td>
<td></td>
</tr>
<tr>
<td>77730</td>
<td>97216</td>
<td>9.05</td>
<td>12.80</td>
<td>*</td>
</tr>
<tr>
<td>77371</td>
<td>95146</td>
<td>7.19</td>
<td>9.89</td>
<td></td>
</tr>
<tr>
<td>76811</td>
<td>93164</td>
<td>7.08</td>
<td>9.84</td>
<td></td>
</tr>
<tr>
<td>76222</td>
<td>91154</td>
<td>7.75</td>
<td>10.21</td>
<td>*</td>
</tr>
<tr>
<td>75566</td>
<td>89180</td>
<td>7.27</td>
<td>9.88</td>
<td></td>
</tr>
<tr>
<td>74965</td>
<td>87072</td>
<td>7.62</td>
<td>9.78</td>
<td>*</td>
</tr>
<tr>
<td>74444</td>
<td>85385</td>
<td>7.12</td>
<td>10.82</td>
<td></td>
</tr>
<tr>
<td>72910</td>
<td>82615</td>
<td>7.87</td>
<td>11.46</td>
<td>*</td>
</tr>
<tr>
<td>72779</td>
<td>79046</td>
<td>7.12</td>
<td>10.55</td>
<td>*</td>
</tr>
<tr>
<td>72081</td>
<td>77045</td>
<td>7.56</td>
<td>9.78</td>
<td></td>
</tr>
<tr>
<td>71470</td>
<td>75007</td>
<td>7.56</td>
<td>10.28</td>
<td>*</td>
</tr>
<tr>
<td>70808</td>
<td>73050</td>
<td>7.75</td>
<td>10.96</td>
<td></td>
</tr>
<tr>
<td>70303</td>
<td>71050</td>
<td>6.58</td>
<td>17.61</td>
<td>*</td>
</tr>
<tr>
<td>69682</td>
<td>68959</td>
<td>7.21</td>
<td>10.80</td>
<td></td>
</tr>
<tr>
<td>69142</td>
<td>67058</td>
<td>6.50</td>
<td>9.53</td>
<td>*</td>
</tr>
<tr>
<td>68014</td>
<td>62952</td>
<td>6.54</td>
<td>8.75</td>
<td>*</td>
</tr>
<tr>
<td>67344</td>
<td>60926</td>
<td>7.00</td>
<td>10.43</td>
<td></td>
</tr>
<tr>
<td>69944</td>
<td>60180</td>
<td>6.38</td>
<td>9.32</td>
<td></td>
</tr>
<tr>
<td>70450</td>
<td>62198</td>
<td>6.69</td>
<td>8.96</td>
<td></td>
</tr>
<tr>
<td>71628</td>
<td>66215</td>
<td>6.27</td>
<td>10.61</td>
<td></td>
</tr>
<tr>
<td>72229</td>
<td>68245</td>
<td>6.21</td>
<td>8.86</td>
<td>*</td>
</tr>
<tr>
<td>72765</td>
<td>70360</td>
<td>6.79</td>
<td>10.87</td>
<td></td>
</tr>
<tr>
<td>73368</td>
<td>72313</td>
<td>7.21</td>
<td>10.83</td>
<td>*</td>
</tr>
<tr>
<td>74240</td>
<td>74419</td>
<td>7.25</td>
<td>13.07</td>
<td></td>
</tr>
<tr>
<td>75091</td>
<td>78326</td>
<td>7.15</td>
<td>11.61</td>
<td></td>
</tr>
<tr>
<td>75782</td>
<td>80247</td>
<td>7.08</td>
<td>9.93</td>
<td>*</td>
</tr>
<tr>
<td>76360</td>
<td>82469</td>
<td>7.08</td>
<td>10.41</td>
<td></td>
</tr>
<tr>
<td>77022</td>
<td>84283</td>
<td>7.08</td>
<td>10.12</td>
<td>*</td>
</tr>
<tr>
<td>77486</td>
<td>86380</td>
<td>7.17</td>
<td>10.38</td>
<td></td>
</tr>
<tr>
<td>78151</td>
<td>88474</td>
<td>7.42</td>
<td>8.90</td>
<td>*</td>
</tr>
<tr>
<td>78697</td>
<td>90380</td>
<td>7.08</td>
<td>11.09</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>79302</td>
<td>92403</td>
<td>7.25</td>
<td>10.29*</td>
<td></td>
</tr>
<tr>
<td>79875</td>
<td>94384</td>
<td>7.13</td>
<td>10.34*</td>
<td></td>
</tr>
<tr>
<td>80456</td>
<td>96463</td>
<td>7.91</td>
<td>10.03</td>
<td></td>
</tr>
<tr>
<td>81014</td>
<td>98374</td>
<td>8.21</td>
<td>10.70*</td>
<td></td>
</tr>
<tr>
<td>81592</td>
<td>100366</td>
<td>7.29</td>
<td>10.65</td>
<td></td>
</tr>
<tr>
<td>84782</td>
<td>101686</td>
<td>7.79</td>
<td>11.62</td>
<td></td>
</tr>
<tr>
<td>84186</td>
<td>99762</td>
<td>7.25</td>
<td>10.36*</td>
<td></td>
</tr>
<tr>
<td>83475</td>
<td>97926</td>
<td>7.17</td>
<td>11.21</td>
<td></td>
</tr>
<tr>
<td>82969</td>
<td>95668</td>
<td>6.96</td>
<td>9.36*</td>
<td></td>
</tr>
<tr>
<td>82308</td>
<td>93726</td>
<td>7.50</td>
<td>8.20*</td>
<td></td>
</tr>
<tr>
<td>80639</td>
<td>87653</td>
<td>7.46</td>
<td>10.18</td>
<td></td>
</tr>
<tr>
<td>80052</td>
<td>85653</td>
<td>7.46</td>
<td>9.79*</td>
<td></td>
</tr>
<tr>
<td>79294</td>
<td>83579</td>
<td>6.96</td>
<td>9.40</td>
<td></td>
</tr>
<tr>
<td>78932</td>
<td>81666</td>
<td>6.69</td>
<td>8.45*</td>
<td></td>
</tr>
<tr>
<td>78259</td>
<td>79637</td>
<td>5.83</td>
<td>10.70</td>
<td></td>
</tr>
<tr>
<td>77644</td>
<td>77601</td>
<td>6.92</td>
<td>9.16*</td>
<td></td>
</tr>
<tr>
<td>76992</td>
<td>75549</td>
<td>6.97</td>
<td>9.79</td>
<td></td>
</tr>
<tr>
<td>75854</td>
<td>71702</td>
<td>7.14</td>
<td>10.09</td>
<td></td>
</tr>
<tr>
<td>74245</td>
<td>67333</td>
<td>6.50</td>
<td>9.48</td>
<td></td>
</tr>
<tr>
<td>74070</td>
<td>65486</td>
<td>6.41</td>
<td>10.27*</td>
<td></td>
</tr>
<tr>
<td>73555</td>
<td>63839</td>
<td>6.96</td>
<td>12.65</td>
<td></td>
</tr>
<tr>
<td>75362</td>
<td>60598</td>
<td>6.96</td>
<td>8.14</td>
<td></td>
</tr>
<tr>
<td>75873</td>
<td>62377</td>
<td>6.87</td>
<td>9.63*</td>
<td></td>
</tr>
<tr>
<td>76355</td>
<td>65200</td>
<td>6.29</td>
<td>9.59</td>
<td></td>
</tr>
<tr>
<td>77221</td>
<td>66714</td>
<td>6.29</td>
<td>9.22*</td>
<td></td>
</tr>
<tr>
<td>77785</td>
<td>68714</td>
<td>6.83</td>
<td>13.06</td>
<td></td>
</tr>
<tr>
<td>78350</td>
<td>70807</td>
<td>6.95</td>
<td>8.69*</td>
<td></td>
</tr>
<tr>
<td>79007</td>
<td>72819</td>
<td>6.89</td>
<td>11.58</td>
<td></td>
</tr>
<tr>
<td>79603</td>
<td>74877</td>
<td>7.08</td>
<td>10.19*</td>
<td></td>
</tr>
<tr>
<td>80139</td>
<td>76870</td>
<td>6.92</td>
<td>10.04</td>
<td></td>
</tr>
<tr>
<td>80643</td>
<td>78763</td>
<td>6.75</td>
<td>9.27*</td>
<td></td>
</tr>
<tr>
<td>81318</td>
<td>80970</td>
<td>6.74</td>
<td>8.90</td>
<td></td>
</tr>
<tr>
<td>81684</td>
<td>83158</td>
<td>6.75</td>
<td>9.48*</td>
<td></td>
</tr>
<tr>
<td>82537</td>
<td>84979</td>
<td>7.29</td>
<td>8.91</td>
<td></td>
</tr>
<tr>
<td>83093</td>
<td>86881</td>
<td>7.08</td>
<td>9.34*</td>
<td></td>
</tr>
<tr>
<td>83591</td>
<td>88879</td>
<td>6.89</td>
<td>8.82</td>
<td></td>
</tr>
<tr>
<td>84274</td>
<td>90977</td>
<td>7.18</td>
<td>10.01*</td>
<td></td>
</tr>
<tr>
<td>84864</td>
<td>92992</td>
<td>6.98</td>
<td>9.82</td>
<td></td>
</tr>
<tr>
<td>85682</td>
<td>94904</td>
<td>6.83</td>
<td>8.57</td>
<td></td>
</tr>
<tr>
<td>86067</td>
<td>96883</td>
<td>7.08</td>
<td>8.98*</td>
<td></td>
</tr>
<tr>
<td>86642</td>
<td>98931</td>
<td>6.98</td>
<td>9.58</td>
<td></td>
</tr>
<tr>
<td>87276</td>
<td>101055</td>
<td>7.17</td>
<td>10.91*</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>89722</td>
<td>100321</td>
<td>7.71</td>
<td>8.76</td>
<td></td>
</tr>
<tr>
<td>88541</td>
<td>96291</td>
<td>7.14</td>
<td>9.27</td>
<td></td>
</tr>
<tr>
<td>88036</td>
<td>94287</td>
<td>7.17</td>
<td>9.01</td>
<td></td>
</tr>
<tr>
<td>87309</td>
<td>92267</td>
<td>7.25</td>
<td>10.06</td>
<td></td>
</tr>
<tr>
<td>86200</td>
<td>88156</td>
<td>7.08</td>
<td>10.18</td>
<td></td>
</tr>
<tr>
<td>85628</td>
<td>86174</td>
<td>7.50</td>
<td>10.56</td>
<td></td>
</tr>
<tr>
<td>84701</td>
<td>84651</td>
<td>7.31</td>
<td>9.22</td>
<td></td>
</tr>
<tr>
<td>84343</td>
<td>82174</td>
<td>6.96</td>
<td>10.99</td>
<td></td>
</tr>
<tr>
<td>83815</td>
<td>80169</td>
<td>6.75</td>
<td>8.07</td>
<td></td>
</tr>
<tr>
<td>83220</td>
<td>78165</td>
<td>6.87</td>
<td>9.28</td>
<td></td>
</tr>
<tr>
<td>82640</td>
<td>76109</td>
<td>7.06</td>
<td>11.19</td>
<td></td>
</tr>
<tr>
<td>81946</td>
<td>74051</td>
<td>6.75</td>
<td>9.15</td>
<td></td>
</tr>
<tr>
<td>81544</td>
<td>72178</td>
<td>6.65</td>
<td>9.46</td>
<td></td>
</tr>
<tr>
<td>80827</td>
<td>70080</td>
<td>6.69</td>
<td>11.17</td>
<td></td>
</tr>
<tr>
<td>80271</td>
<td>68078</td>
<td>6.29</td>
<td>11.41</td>
<td></td>
</tr>
<tr>
<td>79681</td>
<td>66063</td>
<td>6.14</td>
<td>9.61</td>
<td></td>
</tr>
<tr>
<td>81600</td>
<td>63311</td>
<td>6.38</td>
<td>7.81</td>
<td></td>
</tr>
<tr>
<td>82203</td>
<td>65343</td>
<td>6.50</td>
<td>8.20</td>
<td></td>
</tr>
<tr>
<td>82728</td>
<td>67380</td>
<td>6.25</td>
<td>9.96</td>
<td></td>
</tr>
<tr>
<td>83318</td>
<td>69486</td>
<td>6.17</td>
<td>9.39</td>
<td></td>
</tr>
<tr>
<td>83958</td>
<td>71322</td>
<td>5.91</td>
<td>8.54</td>
<td></td>
</tr>
<tr>
<td>85119</td>
<td>75422</td>
<td>6.06</td>
<td>8.10</td>
<td></td>
</tr>
<tr>
<td>85704</td>
<td>77340</td>
<td>6.04</td>
<td>10.13</td>
<td></td>
</tr>
<tr>
<td>86901</td>
<td>81453</td>
<td>6.54</td>
<td>9.92</td>
<td></td>
</tr>
<tr>
<td>87509</td>
<td>83405</td>
<td>7.08</td>
<td>11.43</td>
<td></td>
</tr>
<tr>
<td>88082</td>
<td>85484</td>
<td>7.08</td>
<td>9.06</td>
<td></td>
</tr>
<tr>
<td>88673</td>
<td>87499</td>
<td>7.00</td>
<td>9.34</td>
<td></td>
</tr>
<tr>
<td>89854</td>
<td>91529</td>
<td>6.71</td>
<td>8.58</td>
<td></td>
</tr>
<tr>
<td>90445</td>
<td>93545</td>
<td>7.34</td>
<td>9.04</td>
<td></td>
</tr>
<tr>
<td>91264</td>
<td>95514</td>
<td>7.29</td>
<td>8.19</td>
<td></td>
</tr>
<tr>
<td>91651</td>
<td>97580</td>
<td>7.00</td>
<td>8.92</td>
<td></td>
</tr>
<tr>
<td>92122</td>
<td>99613</td>
<td>6.75</td>
<td>8.89</td>
<td></td>
</tr>
<tr>
<td>95386</td>
<td>100748</td>
<td>7.48</td>
<td>9.00</td>
<td></td>
</tr>
<tr>
<td>94719</td>
<td>98763</td>
<td>7.14</td>
<td>9.10</td>
<td></td>
</tr>
<tr>
<td>94162</td>
<td>96694</td>
<td>5.83</td>
<td>7.80</td>
<td></td>
</tr>
<tr>
<td>93544</td>
<td>94828</td>
<td>6.96</td>
<td>7.88</td>
<td></td>
</tr>
<tr>
<td>92981</td>
<td>92774</td>
<td>6.71</td>
<td>7.28</td>
<td></td>
</tr>
<tr>
<td>92365</td>
<td>90811</td>
<td>7.25</td>
<td>8.89</td>
<td></td>
</tr>
<tr>
<td>91704</td>
<td>88783</td>
<td>7.46</td>
<td>9.25</td>
<td></td>
</tr>
<tr>
<td>91168</td>
<td>86768</td>
<td>6.34</td>
<td>8.61</td>
<td></td>
</tr>
<tr>
<td>89330</td>
<td>80700</td>
<td>6.29</td>
<td>7.85</td>
<td></td>
</tr>
<tr>
<td>88748</td>
<td>78723</td>
<td>5.75</td>
<td>7.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>88247</td>
<td>76618</td>
<td>5.96</td>
<td>8.61</td>
<td>*</td>
</tr>
<tr>
<td>87624</td>
<td>74676</td>
<td>6.19</td>
<td>11.30</td>
<td>*</td>
</tr>
<tr>
<td>87054</td>
<td>72725</td>
<td>6.67</td>
<td>9.20</td>
<td></td>
</tr>
<tr>
<td>86174</td>
<td>70515</td>
<td>6.43</td>
<td>10.87</td>
<td>*</td>
</tr>
<tr>
<td>91266</td>
<td>77975</td>
<td>5.92</td>
<td>7.90</td>
<td>*</td>
</tr>
<tr>
<td>94645</td>
<td>87779</td>
<td>5.87</td>
<td>7.83</td>
<td>*</td>
</tr>
<tr>
<td>94826</td>
<td>90073</td>
<td>6.70</td>
<td>8.64</td>
<td>*</td>
</tr>
<tr>
<td>95942</td>
<td>94201</td>
<td>6.12</td>
<td>7.61</td>
<td>*</td>
</tr>
<tr>
<td>96663</td>
<td>96191</td>
<td>5.96</td>
<td>7.84</td>
<td></td>
</tr>
<tr>
<td>00357</td>
<td>99431</td>
<td>6.41</td>
<td>8.91</td>
<td>*</td>
</tr>
<tr>
<td>99078</td>
<td>95317</td>
<td>6.25</td>
<td>9.03</td>
<td>*</td>
</tr>
<tr>
<td>98530</td>
<td>93463</td>
<td>5.92</td>
<td>8.20</td>
<td></td>
</tr>
<tr>
<td>97930</td>
<td>91351</td>
<td>6.08</td>
<td>9.69</td>
<td>*</td>
</tr>
<tr>
<td>97339</td>
<td>89335</td>
<td>6.37</td>
<td>7.04</td>
<td></td>
</tr>
<tr>
<td>96806</td>
<td>87186</td>
<td>6.29</td>
<td>9.14</td>
<td></td>
</tr>
<tr>
<td>99826</td>
<td>88605</td>
<td>6.33</td>
<td>8.81</td>
<td>*</td>
</tr>
<tr>
<td>00451</td>
<td>90611</td>
<td>6.25</td>
<td>9.96</td>
<td></td>
</tr>
<tr>
<td>00974</td>
<td>92652</td>
<td>6.04</td>
<td>8.77</td>
<td>*</td>
</tr>
<tr>
<td>01633</td>
<td>94722</td>
<td>5.46</td>
<td>8.60</td>
<td></td>
</tr>
<tr>
<td>02849</td>
<td>98837</td>
<td>5.06</td>
<td>9.99</td>
<td></td>
</tr>
<tr>
<td>02933</td>
<td>89865</td>
<td>5.96</td>
<td>9.91</td>
<td>*</td>
</tr>
<tr>
<td>02312</td>
<td>87879</td>
<td>5.71</td>
<td>7.95</td>
<td></td>
</tr>
<tr>
<td>01633</td>
<td>85776</td>
<td>5.75</td>
<td>7.63</td>
<td>*</td>
</tr>
<tr>
<td>04191</td>
<td>85084</td>
<td>6.37</td>
<td>9.07</td>
<td></td>
</tr>
<tr>
<td>64749</td>
<td>63044</td>
<td>7.13</td>
<td>9.29</td>
<td>*</td>
</tr>
<tr>
<td>65821</td>
<td>74489</td>
<td>7.40</td>
<td>10.39</td>
<td>*</td>
</tr>
<tr>
<td>70073</td>
<td>79773</td>
<td>7.44</td>
<td>10.17</td>
<td></td>
</tr>
<tr>
<td>74883</td>
<td>95833</td>
<td>7.08</td>
<td>11.21</td>
<td>*</td>
</tr>
<tr>
<td>78478</td>
<td>99153</td>
<td>7.83</td>
<td>9.06</td>
<td></td>
</tr>
<tr>
<td>73269</td>
<td>80976</td>
<td>7.29</td>
<td>11.21</td>
<td></td>
</tr>
<tr>
<td>68551</td>
<td>64921</td>
<td>7.17</td>
<td>9.52</td>
<td></td>
</tr>
<tr>
<td>70975</td>
<td>64244</td>
<td>6.50</td>
<td>10.06</td>
<td>*</td>
</tr>
<tr>
<td>74621</td>
<td>76389</td>
<td>6.91</td>
<td>11.00</td>
<td>*</td>
</tr>
<tr>
<td>81727</td>
<td>91645</td>
<td>7.25</td>
<td>9.84</td>
<td></td>
</tr>
<tr>
<td>81188</td>
<td>89692</td>
<td>7.52</td>
<td>10.63</td>
<td>*</td>
</tr>
<tr>
<td>76431</td>
<td>73549</td>
<td>7.00</td>
<td>10.47</td>
<td>*</td>
</tr>
<tr>
<td>72979</td>
<td>61504</td>
<td>6.67</td>
<td>8.27</td>
<td>*</td>
</tr>
<tr>
<td>89107</td>
<td>98316</td>
<td>7.08</td>
<td>10.66</td>
<td>*</td>
</tr>
<tr>
<td>87020</td>
<td>90314</td>
<td>7.04</td>
<td>9.01</td>
<td></td>
</tr>
<tr>
<td>79090</td>
<td>64048</td>
<td>6.58</td>
<td>9.82</td>
<td>*</td>
</tr>
<tr>
<td>84533</td>
<td>73514</td>
<td>5.69</td>
<td>8.15</td>
<td></td>
</tr>
<tr>
<td>86323</td>
<td>79396</td>
<td>6.34</td>
<td>7.96</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>89264</td>
<td>89514</td>
<td>6.92</td>
<td>7.68</td>
<td></td>
</tr>
<tr>
<td>92729</td>
<td>101628</td>
<td>6.73</td>
<td>8.59</td>
<td></td>
</tr>
<tr>
<td>85852</td>
<td>68631</td>
<td>6.00</td>
<td>9.56</td>
<td></td>
</tr>
<tr>
<td>85263</td>
<td>66565</td>
<td>6.33</td>
<td>9.15</td>
<td></td>
</tr>
<tr>
<td>90842</td>
<td>75483</td>
<td>5.92</td>
<td>8.78</td>
<td></td>
</tr>
<tr>
<td>91886</td>
<td>79918</td>
<td>6.00</td>
<td>8.21</td>
<td></td>
</tr>
<tr>
<td>95386</td>
<td>92200</td>
<td>8.62</td>
<td>9.58</td>
<td></td>
</tr>
<tr>
<td>97227</td>
<td>98024</td>
<td>6.38</td>
<td>7.62</td>
<td></td>
</tr>
<tr>
<td>97809</td>
<td>100167</td>
<td>7.46</td>
<td>11.86</td>
<td></td>
</tr>
<tr>
<td>99897</td>
<td>97314</td>
<td>6.29</td>
<td>9.65</td>
<td></td>
</tr>
</tbody>
</table>
Figure A.3.1. Histogram for Robena thickness data.
Figure A.3.2. Histogram for Robena ash data.
Figure A.3.3. Location map for Robena data.
A.4 Horizon Depth Data Set Digitized from Dubrule [1984]

Note the modified GEOEAS header adding the number of independent variables after the total number of variables.

Horizon Depth Working and Reference Data Sets
3 2

<table>
<thead>
<tr>
<th>x (ft.)</th>
<th>y (ft.)</th>
<th>z (ft. divided by 1000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7201.89</td>
<td>34088.92</td>
<td>16.202 1</td>
</tr>
<tr>
<td>6718.01</td>
<td>32702.69</td>
<td>15.948 2</td>
</tr>
<tr>
<td>8499.08</td>
<td>31321.25</td>
<td>15.808 3</td>
</tr>
<tr>
<td>5872.85</td>
<td>30115.62</td>
<td>15.824 4</td>
</tr>
<tr>
<td>13935.22</td>
<td>29144.25</td>
<td>15.932 5</td>
</tr>
<tr>
<td>5016.10</td>
<td>27684.23</td>
<td>15.700 6</td>
</tr>
<tr>
<td>7220.65</td>
<td>25754.48</td>
<td>15.338 7</td>
</tr>
<tr>
<td>11938.32</td>
<td>25986.00</td>
<td>15.546 8</td>
</tr>
<tr>
<td>15446.22</td>
<td>25635.18</td>
<td>16.006 9</td>
</tr>
<tr>
<td>18022.59</td>
<td>25808.24</td>
<td>16.276 10</td>
</tr>
<tr>
<td>1106.07</td>
<td>24581.03</td>
<td>16.212 11</td>
</tr>
<tr>
<td>4338.73</td>
<td>24909.70</td>
<td>15.588 12</td>
</tr>
<tr>
<td>6568.04</td>
<td>24765.79</td>
<td>15.408 13</td>
</tr>
<tr>
<td>8876.18</td>
<td>24854.12</td>
<td>15.304 14</td>
</tr>
<tr>
<td>11072.92</td>
<td>24688.30</td>
<td>15.350 15</td>
</tr>
<tr>
<td>4381.68</td>
<td>23406.74</td>
<td>15.514 16</td>
</tr>
<tr>
<td>5925.67</td>
<td>23925.67</td>
<td>15.358 17</td>
</tr>
<tr>
<td>8689.55</td>
<td>23556.40</td>
<td>15.266 18</td>
</tr>
<tr>
<td>12412.38</td>
<td>23851.46</td>
<td>15.454 19</td>
</tr>
<tr>
<td>13888.90</td>
<td>23210.94</td>
<td>15.666 20</td>
</tr>
<tr>
<td>846.59</td>
<td>20881.44</td>
<td>16.548 21</td>
</tr>
<tr>
<td>2329.09</td>
<td>21349.33</td>
<td>15.972 22</td>
</tr>
<tr>
<td>3851.13</td>
<td>21904.36</td>
<td>15.546 23</td>
</tr>
<tr>
<td>5609.46</td>
<td>22380.96</td>
<td>15.294 24</td>
</tr>
<tr>
<td>8755.68</td>
<td>22557.04</td>
<td>15.134 25</td>
</tr>
<tr>
<td>8869.37</td>
<td>22293.23</td>
<td>15.130 26</td>
</tr>
<tr>
<td>11378.51</td>
<td>22074.74</td>
<td>15.348 27</td>
</tr>
<tr>
<td>13339.36</td>
<td>21078.20</td>
<td>15.626 28</td>
</tr>
<tr>
<td>2599.65</td>
<td>20086.64</td>
<td>16.074 29</td>
</tr>
<tr>
<td>5359.34</td>
<td>20666.36</td>
<td>15.248 30</td>
</tr>
<tr>
<td>7902.17</td>
<td>21034.83</td>
<td>15.132 31</td>
</tr>
<tr>
<td>9874.21</td>
<td>20791.76</td>
<td>15.102 32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>11013.01</td>
<td>21004.27</td>
<td>15.23033</td>
</tr>
<tr>
<td>11599.30</td>
<td>20576.35</td>
<td>15.28434</td>
</tr>
<tr>
<td>2158.89</td>
<td>17979.84</td>
<td>16.44835</td>
</tr>
<tr>
<td>4615.17</td>
<td>18221.08</td>
<td>15.72436</td>
</tr>
<tr>
<td>8669.03</td>
<td>19173.50</td>
<td>15.05737</td>
</tr>
<tr>
<td>8908.86</td>
<td>19113.20</td>
<td>15.06638</td>
</tr>
<tr>
<td>13291.94</td>
<td>19491.4</td>
<td>15.54039</td>
</tr>
<tr>
<td>14108.43</td>
<td>17898.43</td>
<td>15.69840</td>
</tr>
<tr>
<td>7567.92</td>
<td>17827.46</td>
<td>15.15641</td>
</tr>
<tr>
<td>9500.41</td>
<td>17497.25</td>
<td>15.08442</td>
</tr>
<tr>
<td>6601.95</td>
<td>16290.17</td>
<td>15.57243</td>
</tr>
<tr>
<td>10003.88</td>
<td>16098.16</td>
<td>15.41244</td>
</tr>
<tr>
<td>10590.32</td>
<td>16456.24</td>
<td>15.25045</td>
</tr>
<tr>
<td>13518.12</td>
<td>16776.05</td>
<td>15.54846</td>
</tr>
<tr>
<td>5391.57</td>
<td>15026.86</td>
<td>15.69647</td>
</tr>
<tr>
<td>8977.85</td>
<td>15009.69</td>
<td>15.33248</td>
</tr>
<tr>
<td>10732.13</td>
<td>14761.83</td>
<td>15.60449</td>
</tr>
<tr>
<td>12225.59</td>
<td>15211.67</td>
<td>15.48450</td>
</tr>
<tr>
<td>2586.42</td>
<td>14059.33</td>
<td>16.62251</td>
</tr>
<tr>
<td>6516.49</td>
<td>14275.81</td>
<td>15.56852</td>
</tr>
<tr>
<td>11403.72</td>
<td>13921.44</td>
<td>15.68453</td>
</tr>
<tr>
<td>5869.70</td>
<td>13613.14</td>
<td>15.62654</td>
</tr>
<tr>
<td>3008.79</td>
<td>12942.34</td>
<td>16.55055</td>
</tr>
<tr>
<td>4704.83</td>
<td>12730.40</td>
<td>15.99056</td>
</tr>
<tr>
<td>10028.44</td>
<td>13012.22</td>
<td>15.58057</td>
</tr>
<tr>
<td>3391.25</td>
<td>11821.50</td>
<td>16.48458</td>
</tr>
<tr>
<td>11459.04</td>
<td>12085.31</td>
<td>15.79059</td>
</tr>
<tr>
<td>16946.32</td>
<td>13114.17</td>
<td>16.49260</td>
</tr>
<tr>
<td>17656.28</td>
<td>12625.33</td>
<td>16.68261</td>
</tr>
<tr>
<td>16503.25</td>
<td>12347.55</td>
<td>16.50262</td>
</tr>
<tr>
<td>16208.50</td>
<td>11694.00</td>
<td>16.65463</td>
</tr>
<tr>
<td>15783.28</td>
<td>10996.29</td>
<td>16.57464</td>
</tr>
<tr>
<td>15021.30</td>
<td>9297.08</td>
<td>16.67465</td>
</tr>
<tr>
<td>14654.97</td>
<td>8414.95</td>
<td>16.59866</td>
</tr>
<tr>
<td>8702.07</td>
<td>10893.47</td>
<td>15.67067</td>
</tr>
<tr>
<td>8037.91</td>
<td>10053.23</td>
<td>15.74268</td>
</tr>
<tr>
<td>9224.92</td>
<td>10037.80</td>
<td>15.77069</td>
</tr>
<tr>
<td>10509.29</td>
<td>10164.16</td>
<td>15.96070</td>
</tr>
<tr>
<td>9521.12</td>
<td>9543.14</td>
<td>15.84271</td>
</tr>
<tr>
<td>7047.69</td>
<td>7436.41</td>
<td>16.07272</td>
</tr>
<tr>
<td>6065.53</td>
<td>6275.73</td>
<td>16.34673</td>
</tr>
<tr>
<td>7080.85</td>
<td>4864.87</td>
<td>16.32274</td>
</tr>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>10747.08</td>
<td>4826.39</td>
<td>16.180 75</td>
</tr>
<tr>
<td>13597.33</td>
<td>4627.82</td>
<td>16.538 76</td>
</tr>
<tr>
<td>13125.68</td>
<td>2118.80</td>
<td>16.504 77</td>
</tr>
<tr>
<td>9404.13</td>
<td>1533.97</td>
<td>16.368 78</td>
</tr>
<tr>
<td>11321.24</td>
<td>580.67</td>
<td>16.518 79</td>
</tr>
<tr>
<td>8175.28</td>
<td>12640.17</td>
<td>15.522 80</td>
</tr>
<tr>
<td>7367.28</td>
<td>18931.05</td>
<td>15.022 81</td>
</tr>
<tr>
<td>2160.89</td>
<td>15473.62</td>
<td>16.612 82</td>
</tr>
</tbody>
</table>

30 Point Reference Set:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>4810</td>
<td>8976</td>
<td>16.412</td>
</tr>
<tr>
<td>10220</td>
<td>8199</td>
<td>15.886</td>
</tr>
<tr>
<td>14148</td>
<td>6400</td>
<td>16.560</td>
</tr>
<tr>
<td>8810</td>
<td>12983</td>
<td>15.552</td>
</tr>
<tr>
<td>3980</td>
<td>15852</td>
<td>15.986</td>
</tr>
<tr>
<td>14534</td>
<td>15596</td>
<td>15.810</td>
</tr>
<tr>
<td>2352</td>
<td>16694</td>
<td>16.532</td>
</tr>
<tr>
<td>5965</td>
<td>18614</td>
<td>15.260</td>
</tr>
<tr>
<td>5223</td>
<td>17204</td>
<td>15.590</td>
</tr>
<tr>
<td>10213</td>
<td>22982</td>
<td>15.320</td>
</tr>
<tr>
<td>6001</td>
<td>27999</td>
<td>15.532</td>
</tr>
<tr>
<td>12104</td>
<td>27828</td>
<td>15.694</td>
</tr>
<tr>
<td>2970</td>
<td>28291</td>
<td>16.028</td>
</tr>
<tr>
<td>8627</td>
<td>27816</td>
<td>15.442</td>
</tr>
<tr>
<td>5987</td>
<td>20999</td>
<td>15.126</td>
</tr>
<tr>
<td>12760</td>
<td>18673</td>
<td>15.446</td>
</tr>
<tr>
<td>15276</td>
<td>16790</td>
<td>15.944</td>
</tr>
<tr>
<td>9010</td>
<td>16268</td>
<td>15.336</td>
</tr>
<tr>
<td>8503</td>
<td>8090</td>
<td>15.836</td>
</tr>
<tr>
<td>13808</td>
<td>17919</td>
<td>15.606</td>
</tr>
<tr>
<td>11463</td>
<td>16932</td>
<td>15.280</td>
</tr>
<tr>
<td>9038</td>
<td>13868</td>
<td>15.514</td>
</tr>
<tr>
<td>7390</td>
<td>22788</td>
<td>15.148</td>
</tr>
<tr>
<td>15338</td>
<td>10056</td>
<td>16.632</td>
</tr>
<tr>
<td>4999</td>
<td>25997</td>
<td>15.608</td>
</tr>
<tr>
<td>10763</td>
<td>18879</td>
<td>15.256</td>
</tr>
<tr>
<td>1550</td>
<td>19207</td>
<td>16.570</td>
</tr>
<tr>
<td>6236</td>
<td>16147</td>
<td>15.508</td>
</tr>
<tr>
<td>6945</td>
<td>12046</td>
<td>15.586</td>
</tr>
<tr>
<td>8394</td>
<td>3222</td>
<td>16.266</td>
</tr>
</tbody>
</table>
Figure A.4.1. Histogram for horizon depth data.
Figure A.4.2. Location map for horizon depth working data.
Figure A.4.3. Location map for horizon depth reference data.
A.5 Earthquake Acceleration Data Set from Carr [1900]

Note the modified GEOEAS header adding the number of independent variables after the total number of variables.

<table>
<thead>
<tr>
<th>E.Q. data (*) deleted after neighborhood analyses</th>
<th>3 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>x ft.</td>
<td></td>
</tr>
<tr>
<td>y dt.</td>
<td></td>
</tr>
<tr>
<td>z gals</td>
<td></td>
</tr>
<tr>
<td>48.3</td>
<td>108.4</td>
</tr>
<tr>
<td>46.5</td>
<td>107.1</td>
</tr>
<tr>
<td>45.6</td>
<td>103.8</td>
</tr>
<tr>
<td>45.0</td>
<td>102.8</td>
</tr>
<tr>
<td>42.3</td>
<td>102.3</td>
</tr>
<tr>
<td>40.8</td>
<td>100.4</td>
</tr>
<tr>
<td>39.9</td>
<td>97.8</td>
</tr>
<tr>
<td>36.2</td>
<td>95.4</td>
</tr>
<tr>
<td>60.4</td>
<td>97.3</td>
</tr>
<tr>
<td>59.6</td>
<td>93.9</td>
</tr>
<tr>
<td>55.0</td>
<td>89.9</td>
</tr>
<tr>
<td>53.2</td>
<td>88.0</td>
</tr>
<tr>
<td>50.1</td>
<td>87.3</td>
</tr>
<tr>
<td>47.3</td>
<td>85.6</td>
</tr>
<tr>
<td>45.0</td>
<td>82.1</td>
</tr>
<tr>
<td>64.9</td>
<td>95.1</td>
</tr>
<tr>
<td>66.4</td>
<td>92.6</td>
</tr>
<tr>
<td>64.9</td>
<td>89.9</td>
</tr>
<tr>
<td>63.4</td>
<td>87.6</td>
</tr>
<tr>
<td>68.0</td>
<td>85.0</td>
</tr>
<tr>
<td>66.5</td>
<td>83.2</td>
</tr>
<tr>
<td>65.6</td>
<td>81.9</td>
</tr>
<tr>
<td>64.2</td>
<td>80.8</td>
</tr>
<tr>
<td>63.5</td>
<td>80.0</td>
</tr>
<tr>
<td>62.4</td>
<td>79.3</td>
</tr>
<tr>
<td>61.7</td>
<td>78.8</td>
</tr>
<tr>
<td>60.5</td>
<td>77.5</td>
</tr>
<tr>
<td>59.7</td>
<td>76.0</td>
</tr>
<tr>
<td>57.9</td>
<td>74.5</td>
</tr>
<tr>
<td>54.4</td>
<td>71.0</td>
</tr>
<tr>
<td>49.5</td>
<td>102.5</td>
</tr>
<tr>
<td></td>
<td>102.5</td>
</tr>
<tr>
<td>---</td>
<td>-------</td>
</tr>
<tr>
<td>49.2</td>
<td>100.4</td>
</tr>
<tr>
<td>51.6</td>
<td>100.1</td>
</tr>
<tr>
<td>54.3</td>
<td>99.5</td>
</tr>
<tr>
<td>52.1</td>
<td>98.2</td>
</tr>
<tr>
<td>49.8</td>
<td>97.3</td>
</tr>
<tr>
<td>56.3</td>
<td>97.3</td>
</tr>
<tr>
<td>54.0</td>
<td>96.5</td>
</tr>
<tr>
<td>52.4</td>
<td>95.4</td>
</tr>
<tr>
<td>55.1</td>
<td>92.6</td>
</tr>
<tr>
<td>59.2</td>
<td>88.9</td>
</tr>
<tr>
<td>61.2</td>
<td>87.1</td>
</tr>
<tr>
<td>63.2</td>
<td>84.3</td>
</tr>
</tbody>
</table>
Figure A.5.1. Histogram for earthquake acceleration data.
Figure A.5.2. Location map for earthquake acceleration data.
APPENDIX B

PROGRAMS

The routines in this appendix are a combination of ones that were written for this work plus some that were modified from GEOEAS code by the author and Hannah Rasmussen Rhodes as well as some unmodified GEOEAS utility routines. These routines are the prototype study routines and as such contain options that will be eliminated in the final version or options not yet developed for the final version.

These are the UNIX versions of the code written or converted to UNIX at Los Alamos National Laboratory on an HP workstation with a Fortran 5.1 compiler at the Facility for Information Management Analysis and Display (FIMAD). The graphics routines used there with this version are unique to that installation and are not included with the code. Those prototype graphics codes were written in fortran using GKS graphics utility routines by Carla Branda in the summer of 1991 at Los alamos and are available from FIMAD. The final, DOS version of these routines, which use GRAFLIB graphics utility subroutines, are available from the author through the Applied Mathematics department at the University of Arizona.

B.1 Neighborhood Optimization Routines

PROGRAM Constrained_Optimal_Neighborhoods

C Writes constrained optimal neighborhoods for all data points into Vario.nbhd file. Can minimize nbhd variance absolutely, or can specify the number of C number of points or limit the variance and maximize C the points for this limited variance. Uses central C point in calculation for undeleted Nbhd.

C Written by M.M.Moody, Sept, 1992

character*10 Varnam(10), units(10)
character*14 Data_File, Nbhd_File, Out_File
real x0,y0,z0,dx,dy,de,value(10),square(10),
+ variance(10), varlim(4),arg,datarec(300),missing
integer i,j,k,m,minptrs,maxptrs,toggle,maxdelta,
+ mvalue,mdata,nprmin,nprmax,istudy,count(10),
+ krigvar(10),xvar,yvar,evar,unt
C The small include file opt.inc is listed after 'opt.inc':

```c
#include 'opt.inc'
```

```c
real data(300,10), xdata(300), ydata(300), edata(300)
integer kdvar, iest, iso, datadim, inbhd, ndata, ierr, nkvar
+
C
common data, xdata, ydata, edata, kdvar, iest, iso, datadim,
C +
inbhd, ndata, ierr, nkvar, idel

write(*,*)' data in data file = Vario.dat'
    iest=1
C set for other estimators: krig only with iest = 1
maxvar=10
maxdata=299
write(*,*)' enter: 1 deleted center Nbhd, '
write(*,*)' 2 undeleted center Nbhd, '
read(*,*)idel
write(*,*)' enter: 1 for nbhd study, '
write(*,*)' 2 for kriging input '
read(*,*)istudy
write(*,*)' enter calculation process for err '
write(*,*)' 1 for mean, '
write(*,*)' 2 for mean of the squares, '
write(*,*)' 3 for nbhd variance '
read(*,*)ierr
maxdelta=85
Data_File='Vario.dat'
Nbhd_File='Vario.nhd'
Out_File='Vario.out'
xvar=1
yvar=2
evar=3
Missing=1.e31
xcmin=1.e31
ycmin=1.e31
ecmin=1.e31
xcmax=-1.e31
ycmax=-1.e31
ecmax=-1.e31
open(unit=10,File=Data_File,status='unknown',err=999)
open(unit=12,File=Nbhd_File,status='unknown',err=999)
open(unit=11,File=Out_File,status='unknown',err=999)
Ndata=0
Nrecord=0
Maxdata=299
unt=10C unt is reading file unit
call read_data(unt)
close(unt)
do 40 i=1,110
value(i)=0.0
square(i)=0.0
count(i)=0
40 continue
write(*,*)'Nvar,datadim,ndata;,Nvar,datadim,ndata
do 43 j=1,NVar-datadim
do 42 i=1,ndata
if(data(i,j).lt.Missing)then
  value(j)=value(j)+data(i,j)
square(j)=square(j)+data(i,j)**2
  count(j)=count(j)+1
endif
42 continue
43 continue
write(*,*)'j,count(j),mean(j),square(j),variance(j):'
do 44 j=1,NVar-datadim
  value(j)=value(j)/float(count(j))
square(j)=square(j)/float(count(j))
  variance(j)=square(j)-value(j)**2
write(*,*)'j,count(j),value(j),square(j),variance(j)
44 continue
write(*,*)'number of data points:ndata
write(*,*)'Enter range of points to process 1 to N'
read(*,*)'nprmin,nprmax
write(*,*)'Enter type of nbhd'
write(*,*)'(only 1,3&4 for krig input):'
write(*,*)'1 only optimal (variable N),'
write(*,*)'2 add fixed N (sub-opt)'
write(*,*)'3 only fixed N (sub-opt),'
write(*,*)'4 max n for limited var (sub-opt).'
write(*,*)'5 all.'
read(*,*)'inbhd
if(inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5)then
  write(*,*)'enter number of points for fixed N nbhd'
  write(*,*)'nbhd shape and orientation is optimized'
read(*,*)npts
else
  npts=0
endif
write(*,*)'Total Nvar,datadim:',Nvar,datadim
nkvar=1
C program set up for future muti var runs - nkvar = 1 now
write(*,*)'input depend. var column number:'
do 50 j=1,nkvar
  read(*,*)Krigvar(j)
50 continue
if(inbhd.eq.4.or.inbhd.eq.5)then
  write(*,*)'enter limiting variance:'
  write(*,*)'as percent of data variance (1-100):'
do 55 j=1,nkvar
  kdvar=krigvar(j)
  write(*,*)'variable, data variance:',kdvar,
  variance(kdvar)
  read(*,*)arg
  varlim(j)=arg*variance(kdvar)/100.
55 continue
endif
ndata=ndata-1
write(*,*)'max neighborhood difference is:',maxdelta
70 write(*,*)'enter max no. of data pts to be examined'
read(*,*)maxptrs
80 write(*,*)'enter min no. of data pts to be examined'
read(*,*)minptrs
C calc max/minptrs from nbrs depending on dimension
if(inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5)then
  if(maxptrs.lt.npts)then
    write(*,*)'set maxpts (.lt. npts) =',npts
    maxptrs=npts
  endif
  if(minptrs.gt.npts)then
    write(*,*)'minpts too large, set to npts'
    minptrs=npts
  endif
endif
if(datadim.eq.1)then
  if(maxptrs.gt.mdata)maxptrs=mdata
elseif(datadim.eq.2)then
  maxptrs=maxptrs-1
  minptrs=minptrs-1
  if(maxptrs.ge.mdata)maxptrs=mdata-1
elseif(datadim.eq.3)then
  maxptrs=maxptrs-2
  minptrs=minptrs-2
  if(maxptrs.ge.(mdata-1))maxptrs=mdata-2
endif
if(maxptrs.lt.minptrs)then
  write(*, *)'too little gap between max and min nbrs'
  write(*, *)minptrs,maxptrs:',minptrs,maxptrs
  go to 70
endif
if((maxptrs-minptrs).gt.maxdelta)then
  write(*, *)'too large a gap between max and min nbrs'
  write(*, *)minptrs,maxptrs:',minptrs,maxptrs
  write(*, *)'max difference = ',maxdelta
  go to 70
endif
mvalue=0
if(inbhd.eq.a)then
  if(idel.eq.1)then
    write(12, *)ndata,nkvar,inbhd,iso,iest,ierr,
+     minptrs,maxptrs,varlim(1)
  else
    write(12, *)ndata, nkvar, inbhd, iso, iest,
+      ierr, minptrs, maxptrs, varlim(1)
  endif
else
  if(idel.eq.1)then
    write(12, *)ndata, nkvar, inbhd, iso, iest, ierr,
+     minptrs, maxptrs, npts
  else
    write(12, *)ndata, nkvar, inbhd, iso, iest,
+      ierr, minptrs, maxptrs, npts
  endif
endif
endif
call sqoptimize_multi_nbhd(varlim,x0,y0,z0,minptrs,
+ maxptrs,N_var,npts,nprmin,nprmax,istudy,krigvar)
write(*,*')'enter 1 for new options, 2 to stop'
read(*,*')m
if(m.eq.1)go to 60
990 FORMAT(A80)
995 FORMAT(2AlO)
999 continue
end

SUBROUTINE sqoptimize_multi_nbhd(varlim,x0,y0,z0,
+ minptrs,maxptrs,Nvar,npts,nprmin,nprmax,
+ istudy,krigvar)

C select nbhds: sort data subsets by ellipse parameters
C optimize nbhd through xvalidation' determine pointers
C and select working points from data matrix, xval gives
C no of xvalid pt; this pt goes to x0,y0,z0, central points
C about which selection occurs; if 2d edata and z0 are 0;
C output to Vario.nhd. Written by M.M.Moody, Sept,1992

real x0,y0,z0,vO,aarray(300),mina(4),minb(4),minc(4),
+ varlim(4),barray(300),carray(300),arg,value,
+ x1,y1,z1,x2,y2,z2,minkvar(4),minjerr,menja(4),
+ minjb(4),minjc(4),tiny,minjqval(4),jaaxis(4,3),
+ jbaxis(4,3),jcaxis(4,3),x3,y3,z3,minval(4),err,
+ minerr,minkval(4),minaaxis(4,3),minbaxis(4,3),
+ mincaxis(4,3),minkerr,minka(4),minkb(4),testee,
+ testor,minsqerr,minjsqerr,minksqerr,sqerr,
+ minvar(4),minjvar(4),minkc(4),kaxis(4,3),
+ kaxis(4,3),kcaxis(4,3),minsqval,minjsqval,
+ minksqval,variance,sqvalue,missing,maxmean(4),
+ minmean(4),meanvalue(4),varmean(4),sqmean(4),
+ a0(300),b0(300),c0(300),aaxis(300,4),
+ baxis(300,4),caxis(300,4),meanvalue(4),
+ meankvalue(4),sqjmean(4),sqkmean(4),meanjvar(4),
+ meankvar(4),varjmean(4),varkmean(4),
+ minjmean(4),minkmean(4),maxjmean(4),maxkmean(4),
+ meanvar(4),relerr(300),reljerr(300),relkerr(300)
integer ptrs(4),aptr(300),bptr(300),mini(4),
+ minval(4),minptr(4),cptr(300),apts,bpts,cpts,
+ ind,mdata,flag,finflag(4),minj(4),mink(4),
C find ptr(1) from distance to origin
n=100
apt=0
bpt=0
cpt=0
mdata=ndata-1
if( datadim.eq.1 )then
  if( maxptrs.gt.n ) maxptrs = n
elseif( datadim.ge.2 )then
  if( maxptrs.ge.n ) maxptrs = n-1
elseif( datadim.ge.3 )then
  if( maxptrs.ge.(n-1) ) maxptrs = n-2
endif
if( maxptrs.lt.minptrs ) minptrs = maxptrs
C do  5  j=1,4
sum(j)=0
maxknval(j)=0
meanvalue(j)=0.0
meanjvalue(j)=0.0
meankvalue(j)=0.0
sqmean(j)=0.0
sqjmean(j)=0.0
sqkmean(j)=0.0
meanvar(j)=0.0
meanjvar(j)=0.0
meankvar(j)=0.0
varmean(j)=0.0
varjmean(j)=0.0
varkmean(j)=0.0
minmean(j)=1.e31
minjmean(j)=1.e31
minkmean(j)=1.e31
maxmean(j)=0.0
maxjmean(j)=0.0
maxkmean(j)=0.0
count(j)=0
countj(j)=0
countk(j)=0
do 5 i=1,ndata
results(j,i)=0
5 continue
missing=1.e31

C cycle through selected data points
do 2000 m=nprmin,nprmax
do 1000 k=1,nkvar
mvar=k
kdvar=Krigvar(k)
if(data(m,kdvar).lt.missing)then
  value=data(m,kdvar)
sqvalue=value*value
else
  sqvalue=0.0
endif
do 15 j=1,4
do 10 i=1,3
ptra(i)=0
ptrb(i)=0
ptrc(i)=0
aaxis(j,i)=0.0
baxis(j,i)=0.0
caxis(j,i)=0.0
jaaxis(j,i)=0.0
jbaxis(j,i)=0.0
jcaxis(j,i)=0.0
kaaxis(j,i)=0.0
kbaxis(j,i)=0.0
kcaxis(j,i)=0.0
10 continue
mina(j)=0.0
minb(j)=0.0
minc(j) = 0.0
minja(j) = 0.0
minjb(j) = 0.0
minjc(j) = 0.0
minka(j) = 0.0
minkb(j) = 0.0
minkc(j) = 0.0
minnval(j) = 0
minknval(j) = 0
minflag(j) = 0
mink(j) = 0
minj(j) = 0
mini(j) = 0
minjk(j) = 0
minij(j) = 0
minji(j) = 0
minkk(j) = 0
minkj(j) = 0
minki(j) = 0
minval(j) = 1.e31
minjval(4) = 1.e31
minkval(4) = 1.e31
minvar(j) = 1.e31
minjvar(j) = 1.e31
minkvar(j) = 1.e31
15 continue
mvalue = 0
minerr = 1.e31
minjerr = 1.e31
minkerr = 1.e31
minsqerr = 1.e31
minjsqerr = 1.e31
minksqerr = 1.e31
minsqval = 1.e31
minjsqval = 1.e31
minksqval = 1.e31
x0 = xdata(m)
y0 = ydata(m)
z0 = edata(m)
v0 = data(m,kdvar)
if(v0.ge.missing) then
    write(*,*) 'missing data:', m, kdvar
minflag(k) = 10
write(12,*) m, minflag(k)
go to 1000
endif
write(11,*) 'Center Point:', m, x0, y0, z0, v0
xval = m
ind = 0
do 20 j = 1, ndata
if(j, eq, xval) go to 20
ind = ind + 1
arg = (xdata(j) - x0)**2 + (ydata(j) - y0)**2 + (edata(j) - z0)**2
aarray(ind) = sqrt(arg)
aptr(ind) = j
20 continue
call vqsort(aarray, aptr, mdata)
do 30 j = 1, maxptrs + datadim - 1
30 continue
ptrs(1) = xval
if(iso, eq, 1) then
C do isotropic calculation, independent of dimension
  do 40 j = minptrs + datadim - 1, maxptrs + datadim - 1
    ptrs(2) = j
    bpts = j
    call estimate_multi_values(ptrs, aptr, bptr, cptr, bpts, +          value, sqvalue, variance, nvalue(k))
    mvalue = mvalue + 1
    arg = value - data(xval, kdvar)
    err = abs(arg)
    arg = sqvalue - (data(xval, kdvar))**2
    sqerr = abs(arg)
    if(ierr, eq, 1) then
      testee = err
    elseif(ierr, eq, 2) then
      testee = sqerr
    else
      testee = variance
    endif
    if(inbhd, eq, 1 or inbhd, eq, 2 or inbhd, eq, 5) then
      if(ierr, eq, 1) then
        testor = minerr
      elseif(ierr, eq, 2) then
        testor = minsqerr
      endif
else
    testor=minvar(k)
endif
if(testee.lt.testor)then
    apt=ptrs(2)
    mink(k)=j
    mini(k)=0
    mini(k)=0
    ptra(1)=ptrs(2)
    minerr=err
    minsqerr=sqerr
    minval(k)=value
    relerr(k)=(minval(k)-v0)/(abs(minval(k))+abs(v0))
+ (abs(minval(k)))+abs(v0))
    minsval=sqval
    minvar(k)=variance
    minnval(k)=nvalue(k)
    mina(k)=array(j)
    minb(k)=mina(k)
    minc(k)=mina(k)
    if(datadim.eq.2)minc(k)=0.0
    minptr(1)=ptrs(1)
endif
if((inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5)
+ .and.nvalue(k).eq.npts)then
    if(ierr.eq.1)then
        testor=minjerr
    elseif(ierr.eq.2)then
        testor=minjsqerr
    else
        testor=minjvar(k)
    endif
if(testee.lt.testor)then
    apt=ptrs(2)
    minjk(k)=j
    mini(k)=0
    mini(k)=0
    ptra(2)=ptrs(2)
    minjerr=err
    minjsqerr=sqerr
    minjval(k)=value
\[ \text{reljerr}(k) = \frac{\text{minjval}(k) - v_0}{\sqrt{\text{abs(minjval}(k)) + \text{abs}(v_0)}} \]

\[ \text{minjsqval} = \text{sqvalue} \]

\[ \text{minjvar}(k) = \text{variance} \]

\[ \text{minjnval}(k) = \text{nvalue}(k) \]

\[ \text{minja}(k) = \text{aarray}(j) \]

\[ \text{minjb}(k) = \text{minja}(k) \]

\[ \text{minjc}(k) = \text{minja}(k) \]

\[ \text{if}(\text{datadim}.eq.2) \text{minjc}(k) = 0.0 \]

\[ \text{minjp}t(1) = \text{ptrs}(1) \]

\[ \text{endif} \]

\[ \text{endif} \]

\[ \text{if}(\text{inbhd}.eq.4 \text{or} \text{inbhd}.eq.5) \text{then} \]

\[ \text{if}(\text{ierr}.eq.1) \text{then} \]

\[ \text{testor} = \text{minkerr} \]

\[ \text{elseif}(\text{ierr}.eq.2) \text{then} \]

\[ \text{testor} = \text{minsqerr} \]

\[ \text{else} \]

\[ \text{testor} = \text{minkvar}(k) \]

\[ \text{endif} \]

\[ \text{if}(\text{variance}.lt.\text{varlim}(k)) \text{then} \]

\[ \text{if}(\text{nvalue}(k).gt.\text{minknval}(k).\text{or.} \]

\[ + (\text{nvalue}(k).eq.\text{minknval}(k).\text{and.} \]

\[ + \text{testee}.lt.\text{testor}) \text{then} \]

\[ \text{apt} = \text{ptrs}(2) \]

\[ \text{minkk}(k) = j \]

\[ \text{minjk}(k) = 0 \]

\[ \text{minki}(k) = 0 \]

\[ \text{pt}ra(3) = \text{ptrs}(2) \]

\[ \text{minkerr} = \text{err} \]

\[ \text{minsqerr} = \text{sqerr} \]

\[ \text{minkval}(k) = \text{value} \]

\[ \text{relkerr}(k) = \frac{\text{minkval}(k) - v_0}{\sqrt{\text{abs(minkval}(k)) + \text{abs}(v_0)}} \]

\[ \text{minjsqval} = \text{sqvalue} \]

\[ \text{minkvar}(k) = \text{variance} \]

\[ \text{minknval}(k) = \text{nvalue}(k) \]

\[ \text{if}(\text{nvalue}(k).gt.\text{maxknval}(k)) \]

\[ + \text{maxknval}(k) = \text{nvalue}(k) \]

\[ \text{minka}(k) = \text{aarray}(j) \]

\[ \text{minkb}(k) = \text{minka}(k) \]

\[ \text{minkc}(k) = \text{minka}(k) \]
if(datadim.eq.2)minkc(k)=0.0
minkptr(1)=ptrs(1)
endif
endif
endif
40 continue
C
else
C  do anisotropic calculation
C
do 500 k1=minptrs+datadim-1,maxptrs+datadim-1
apts=k1
bpts=0
a0(k)=aarray(k1)
C  write(*,*)'k1,a0:',k1,a0(k)
x1=xdata(aptr(k1))
y1=ydata(aptr(k1))
z1=edata(aptr(k1))
C  find ptrs(2) from aptr in steps up to n
ptrs(2)=aptr(k1)
C
if(datadim.gt.1)then
C  calculate barray then sort
  if(datadim.eq.2.and.bpts.gt.maxptrs)go to 200
  do 45 k2=I,kl-l
    x2=xdata(aptr(k2))
y2=ydata(aptr(k2))
z2=edata(aptr(k2))
C    call ellipse3d(x0,y0,z0,x1,y1,z1,x2,y2,z2,
      + a0(k),b0(k),alpha,bflag)
C  for bflag = 1 degenerate: b0=0
    bpts=bpts+1
    bptr(bpts)=bpts
    barray(bpts)=b0(k)
45 continue
  call vqsort(barray,bptr,bpts)
if(datadim.eq.2)then
  do 50 j=1,bpts
    i=bptr(j)
50  continue
else
j = aptr(aptx)
i = bptr(bptx)
write(11,*)'aarray:',aptx,aarray(aptx),
+ data(j,kdvar)
do 55 j=1,bptx
i = bptr(j)
55 continue
endif
if(datadim.eq.2)then
  C check if zero b array, sort by aarray
  if(barray(bptx).eq.0.0)then
    do 57 j = 1,bptx
    barray(j) = aarray(j)
bptr(j) = aptr(j)
57 continue
    do 58 j = 1,bptx
    i = bptr(j)
58 continue
endif
do 70 i=minptx,bptx
if(i.gt.maxptx)go to 70
ptrs(3) = aptr(bptr(i))
call estimate_multi_values(ptrs,aptr,bptr,cptr,
+ i,value,sqvalue,variance,nvalue(k))
mvalue = mvalue + 1
arg = value-data(xval,kdvar)
err = abs(arg)
arg = sqvalue-(data(xval,kdvar))**2
sqerr = abs(arg)
if(ierr.eq.1)then
testee = err
elseif(ierr.eq.2)then
testee = sqerr
else
testee = variance
endif
if(inbhd.eq.1.or.inbhd.eq.2.or.inbhd.eq.5)then
  if(ierr.eq.1)then
    testor = minerr
  elseif(ierr.eq.2)then
    testor = minsqerr
  else
endif
testor=minvar(k)
endif
if(testee.lt.testor)then
   apt=ptrs(2)
bpt=ptrs(3)
mink(k)=k1
minj(k)=i
mini(k)=0
ptra(1)=ptrs(2)
ptrb(1)=ptrs(3)
minerr=err
minsqerr=sqerr
minval(k)=value
relerr(k)=(minval(k)-v0)/
+ (abs(minval(k))+abs(v0))
minsqval=sqvalue
minvar(k)=variance
minnval(k)=nvalue(k)
mina(k)=aarray(k1)
minb(k)=barray(i)
minc(k)=0.0
do 60 l=1,3
   minptr(l)=ptrs(l)
continue
60
x2=xdata(ptrs(3))
y2=ydata(ptrs(3))
z2=edata(ptrs(3))
call ellipse3d(x0,y0,z0,x1,y1,z1,x2,y2,z2,
+ a0(k),b0(k), alpha,bflag)
C for bflag = 1 degenerate: b0=0
C arg=cos(theta)
arg=(x1-x0)/sqrt((x1-x0)**2+(y1-y0)**2)
theta=acos(arg)
minaaxis(k,1)=arg
minbaxis(k,2)=arg
C arg=sin(theta)
arg=(y1-y0)/sqrt((x1-x0)**2+(y1-y0)**2)
minaaxis(k,2)=arg
minbaxis(k,1)=-arg
endif
endif
if((inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5) .and.nvalue(k).eq.npts)then
  if(ierr.eq.1)then
    testor=minjerr
  elseif(ierr .eq.2)then
    testor=minjsqerr
  else
    testor=minjvar(k)
  endif
  if(testee.lt.testor)then
    apt=ptrs(2)
    bpt=ptrs(3)
    minjk(k)=k1
    minjj(k)=i
    minji(k)=0
    ptra(2)=ptrs(2)
    ptrb(2)=ptrs(3)
    minjerr=err
    minjsqerr=sqerr
    minjval(k)=value
    reljerr(k)=(minjval(k)-v0)/(abs(minjval(k))+abs(v0))
    minjsqval=sqvalue
    minjvar(k)=variance
    minjnval(k)=nvalue(k)
    minja(k)=aarray(k1)
    minjb(k)=barray(i)
    minjc(k)=0.0
  do 65 1=1,3
    minjptr(1)=ptrs(l)
  continue
  x2=xdata(ptrs(3))
  y2=ydata(ptrs(3))
  z2=edata(ptrs(3))
  call ellipse3d(x0,y0,z0,x1,y1,z1,x2,y2,z2,
    a0(k),b0(k),alpha,bflag)
  C for bflag = 1 degenerate: b0=0
  C arg=cos(theta)
  arg=(x1-x0)/sqrt((x1-x0)**2+(y1-y0)**2)
  theta=acos(arg)
  jaaxis(k,1)=arg
  jbaxis(k,2)=arg
C

\[ \text{arg} = \sin(\theta) \]
\[ \text{arg} = \frac{(y_1-y_0)}{\sqrt{(x_1-x_0)^2 + (y_1-y_0)^2}} \]
\[ \text{jaaxis}(k,2) = \text{arg} \]
\[ \text{jbaxis}(k,1) = -\text{arg} \]

endif
endif
if(inbhd.eq.4.or.inbhd.eq.5)then
if(variance.lt.varlim(k))then
if(ierr.eq.1)then
  testor=minkerr
elseif(ierr.eq.2)then
  testor=minksqerr
else
  testor=minkvar(k)
endif
if(nvalue(k).gt.minknval(k).or.
  (nvalue(k).eq.minknval(k).and. 
  testee.lt. testor)
then
  apt=ptrs(2)
bpt=ptrs(3)
minkk(k)=k1
minkj(k)=i
minki(k)=0
ptr(3)=ptrs(2)
ptrb(3)=ptrs(3)
minkerr=err
minksqerr=sqerr
minkval(k)=value
relkerr(k)=(minkval(k)-v0)/
  (abs(minkval(k))+abs(v0))
minksqval=sqvalue
minkvar(k)=variance
minknval(k)=nvalue(k)
if(nvalue(k).gt.maxknval(k))
  maxknval(k)=nvalue(k)
minka(k)=aarray(k1)
minkb(k)=barray(i)
minkc(k)=0.0
minkptr(1)=ptrs(1)
x2=xdata(ptrs(3))
y2=ydata(ptrs(3))
z2=edata(ptrs(3))

+ (nvalue(k).eq.minknval(k).and. 
  testee.lt. testor)
then
  apt=ptrs(2)
bpt=ptrs(3)
minkk(k)=k1
minkj(k)=i
minki(k)=0
ptr(3)=ptrs(2)
ptrb(3)=ptrs(3)
minkerr=err
minksqerr=sqerr
minkval(k)=value
relkerr(k)=(minkval(k)-v0)/
  (abs(minkval(k))+abs(v0))
minksqval=sqvalue
minkvar(k)=variance
minknval(k)=nvalue(k)
if(nvalue(k).gt.maxknval(k))
  maxknval(k)=nvalue(k)
minka(k)=aarray(k1)
minkb(k)=barray(i)
minkc(k)=0.0
minkptr(1)=ptrs(1)
x2=xdata(ptrs(3))
y2=ydata(ptrs(3))
z2=edata(ptrs(3))

call ellipse3d(xO,yO,zO,x1,y1,z1,x2,y2,z2,  
a0(k),b0(k),alpha,bflag)
C for bflag = 1 degenerate: b0=0
C  arg=cos(theta)
arg=(x1-xO)/sqrt((x1-xO)**2+(y1-yO)**2)
theta=acos(arg)
kaaxis(k,1)=arg
kbaxis(k,2)=arg
C  arg=sin(theta)
arg=(y1-yO)/sqrt((x1-xO)**2+(y1-yO)**2)
kaaxis(k,2)=arg
kbaxis(k,1)=-arg
endif
endif
endif
70 continue
endif
if(data_dim.gt.2)then
  do 100 j=minptrs+1,bpts
    k2=bptr(j)
    b0(k)=barray(j)
    ptrs(3)=aptr(k2)
    tiny=0.001*a0(k)
    x2=xdata(aptr(k1))
    y2=ydata(aptr(k1))
    z2=edata(aptr(k1))
    cpts=0
    do 80 k3=1,j-1
      if(cpts.eq.maxpt) go to 100
      k2=bptr(k3)
      x3=xdata(aptr(k2))
      y3=ydata(aptr(k2))
      z3=edata(aptr(k2))
      call ellipsoid(xO,yO,zO,x1,y1,z1,x2,y2,z2,  
        + x3,y3,z3,a0(k),b0(k),c0(k),aaxis(k,1),
        + baxis(k,1),caxis(k,1),m,flag)
      if(flag.eq.1)then
        b0(k)=tiny
        c0(k)=tiny
      elseif(flag.eq.2.or.flag.eq.3)then
        c0(k)=tiny
      endif
cpts = cpts + 1
clspt(k3) = cpts
arr(cpts) = ci0(k)
continue

call vqsort(arr, clspt, cpts)
k2 = bptr(j)
l = aptr(k2)
write(11, *)'barray: j, k2, i, barray(j),
data(l, kdvar)
+ write(11, *)'sorted carray'
do 85 k2 = 1, cpts
  i = bptr(clspt(k2))
  write(11, *) clspt(k2), i, arr(k2),
data(aptr(i), kdvar)
+ continue

C check if zero c array, sort by barray
if (arr(cpts).eq.0.0) then
do 87 k2 = 1, cpts
  arr(k2) = barray(k2)
  clspt(k2) = bptr(k2)
continue
do 88 k2 = 1, cpts
  i = bptr(k2)
continue
endif
do 95 k2 = minptrs, cpts
if (k2 .gt. maxptrs) go to 95
  i = bptr(clspt(k2))
C find ptrs(4) from clspt in steps up to k2 for 3D
ptrs(4) = aptr(i)
call estimate_multi_values(ptrs, aptr, bptr, clspt,
+ bpts, value, sqvalue, variance, nvalue(k))
mvalue = mvalue + 1
arg = value - data(xval, kdvar)
err = abs(arg)
arg = sqvalue - (data(xval, kdvar))**2
sqerr = abs(arg)
if (ierr .eq. 1) then
  testee = err
elseif (ierr .eq. 2) then
  testee = sqerr
else

217
testee = variance
endif
if (inbhd.eq.1.or.inbhd.eq.2.or.inbhd.eq.5) then
if (testee.gt.testor) then
    apt = ptrs(2)
    bpt = ptrs(3)
    cpt = ptrs(4)
    mink(k) = k1
    mini(k) = j
    mini(k) = k2
    ptrb(1) = ptrs(2)
    ptrb(1) = ptrs(3)
    ptrb(1) = ptrs(4)
    minerr = err
    minsqerr = sqerr
    minval(k) = value
    relerr(k) = (minval(k)-vO)/
        (abs(minval(k)) + abs(vO))
    minsqval = sqvalue
    minvar(k) = variance
    minnval(k) = nvalue(k)
    mina(k) = aarray(k1)
    minb(k) = barray(j)
    minc(k) = carray(k2)
    x3 = xdata(ptrs(4))
    y3 = ydata(ptrs(4))
    z3 = edata(ptrs(4))
    call ellipsoid(xO,yO,zO,x1,y1,z1,x2,y2,z2,
        + x3,y3,z3,a0(k),b0(k),c0(k),
        + aaxis(k,1),baxis(k,1),caxis(k,1),m,flag)
    if (flag.eq.1) then
        b0(k) = tiny
        c0(k) = tiny
    elseif (flag.eq.2.or.flag.eq.3) then
        c0(k) = tiny
    endif
    do 90 l = 1,3
        minaaxis(k,l) = aaxis(k,l)
        minbaxis(k,l) = baxis(k,l)
        mincaxis(k,l) = caxis(k,l)
90          continue
endif
endif
if((inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5) + .and.nvalue(k).eq.npts)then
  if(testee.lt.testor)then
    apt=ptrs(2)
    bpt=ptrs(3)
    cpt=ptrs(4)
    minjk(k)=k1
    minjj(k)=j
    minji(k)=k2
    ptra(2)=ptrs(2)
    ptrb(2)=ptrs(3)
    ptrc(2)=ptrs(4)
    minjerr=err
    minjsqerr=sqerr
    minjval(k)=value
    reljerr(k)=(minjval(k)-vO)/ + (abs(minjval(k))+abs(vO))
    minjsqval=sqvalue
    minjvar(k)=variance
    minja(k)=aarray(k1)
    minjb(k)=barray(j)
    minjc(k)=carray(k2)
    minjptr(1)=ptrs(1)
    x3=xdata(ptrs(4))
    y3=ydata(ptrs(4))
    z3=edata(ptrs(4))
    call ellipsoid(xO,yO,zO,x1,y1,z1,x2,y2,z2, + x3,y3,z3,a0(k),b0(k),c0(k), + aaxis(k,1),baxis(k,1),caxis(k,1),m,flag)
    if(flag.eq.1)then
      b0(k)=tiny
      c0(k)=tiny
    elseif(flag.eq.2.or.flag.eq.3)then
      c0(k)=tiny
    endif
  do 92 l=1,3
    jaaxis(k,l)=aaxis(k,l)
    jbaxis(k,l)=baxis(k,l)
    jcaxis(k,l)=caxis(k,l)
continue
endif
endif
if(inbhd.eq.4.or.inbhd.eq.5)then
if(variance.lt.varlim(k))then
if(nvalue(k).gt.minknval(k).or.  
  (nvalue(k).eq.minknval(k).and.  
  testee.lt.testor))then
apt=ptrs(2)

bpt=ptrs(3)
cpr=ptrs(4)
minkk(k)=k1
minkj(k)=i
minki(k)=k2
ptra(3)=ptrs(2)
ptrb(3)=ptrs(3)
ptrc(3)=ptrs(4)
minkerr=err
minksqerr=sqerr
minkval(k)=value
relkerr(k)=(minkval(k)-vO)/  
  (abs(minkval(k))+abs(vO))

minkvar(k)=variance
minknval(k)=nvalue(k)
if(nvalue(k).gt.maxknval(k))
  maxknval(k)=nvalue(k)

minka(k)=aarray(k1)
minkb(k)=barray(i)
minkc(k)=carray(k2)
minkptr(1)=ptrs(1)
x3=xdata(ptrs(4))
y3=ydata(ptrs(4))
z3=edata(ptrs(4))
call ellipsoid(x0,y0,z0,x1,y1,z1,x2,y2,z2,  
  x3,y3,z3,a0(k),b0(k),c0(k),  
  aaxis(k,1),baxis(k,1),caxis(k,1),m,flag)
if(flag.eq.1)then
b0(k)=tiny
c0(k)=tiny
elseif(flag.eq.2.or.flag.eq.3)then
c0(k) = tiny
endif
do 94 l=1,3
kaaxis(k,l) = aaxis(k,l)
kbaxis(k,l) = baxis(k,l)
kcaxis(k,l) = caxis(k,l)
94 continue
endif
endif
endif
95 continue
100 continue
endif
200 continue
else
write(*,*), can not do anisotropic calc for 1-D
stop
endif
end isotropic/anisotropic calc
500 continue
endif
if(datadim.eq.2)then
if((minaaxis(k,1).lt.0.0.and.minaaxis(k,2).lt.0.0) + or.(minaaxis(k,1).gt.0.0.and.minaaxis(k,2).lt.0.0))
+ then
minaaxis(k,1) = -1.0*minaaxis(k,1)
minaaxis(k,2) = -1.0*minaaxis(k,2)
derif
if((minbaxis(k,1).lt.0.0.and.minbaxis(k,2).lt.0.0) + or.(minbaxis(k,1).gt.0.0.and.minbaxis(k,2).lt.0.0))
+ then
minbaxis(k,1) = -1.0*minbaxis(k,1)
minbaxis(k,2) = -1.0*minbaxis(k,2)
derif
if((inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5)then
if((kaaxis(k,1).lt.0.0.and.kaaxis(k,2).lt.0.0) + or.(kaaxis(k,1).gt.0.0.and.kaaxis(k,2).lt.0.0))
+ then
kaaxis(k,1) = -1.0*kaaxis(k,1)
kaaxis(k,2) = -1.0*kaaxis(k,2)
derif
if((kbaxis(k,1).lt.0.0.and.kbaxis(k,2).lt.0.0)
+ .or.(kbaxis(k,1).gt.0.0.and.kbaxis(k,2).lt.0.0))
+ then
  kbaxis(k,1)=-1.0*kbaxis(k,1)
  kbaxis(k,2)=-1.0*kbaxis(k,2)
endif
endif
if(inbhd.eq.4.or.inbhd.eq.5)then
  if((kaaxis(k,1).gt.0.0.and.kaaxis(k,2).lt.0.0)+
  .or.(kaaxis(k,1).gt.0.0.and.kaaxis(k,2).lt.0.0))
+ then
    kaaxis(k,1)=-1.0*kaaxis(k,1)
    kaaxis(k,2)=-1.0*kaaxis(k,2)
endif
  if((kbaxis(k,1).lt.0.0.and.kbaxis(k,2).lt.0.0)+
  .or.(kbaxis(k,1).lt.0.0.and.kbaxis(k,2).lt.0.0))
+ then
    kbaxis(k,1)=-1.0*kbaxis(k,1)
    kbaxis(k,2)=-1.0*kbaxis(k,2)
endif
elseif(datadim.eq.3)then
  if(minaaxis(k,3).lt.0.0)then
    do 510 j=1,3
      minaaxis(k,j) =-1.0*minaaxis(k,j)
      continue
  510  enddo
endif
  if(minbaxis(k,3).lt.0.0)then
    do 520 j=1,3
      minbaxis(k,j) =-1.0*minbaxis(k,j)
      continue
  520  enddo
endif
  if(mincaxis(k,3).lt.0.0)then
    do 530 j=1,3
      mincaxis(k,j) =-1.0*mincaxis(k,j)
      continue
  530  enddo
endif
C all directions are in positive z hemisphere
C note that ellipsoids no longer follow right hand rule
if(inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5)then
  if(kaaxis(k,3).lt.0.0)then
    do 540 j=1,3
      kaaxis(k,j)=-1.0*kaaxis(k,j)
  540  enddo
endif
540 continue
endif
if(kbaxis(k,3).lt.0.0)then
  do 550 j=1,3
    kbaxis(k,j)=-1.0*kbaxis(k,j)
  continue
endif
if(kcaxis(k,3).lt.0.0)then
  do 560 j=1,3
    kcaxis(k,j)=-1.0*kcaxis(k,j)
  continue
endif
if(inbhd.eq.1.or.inbhd.eq.2.or.inbhd.eq.5)then
  if(kaaxis(k,3).lt.0.0)then
    do 570 j=1,3
      kaaxis(k,j)=-1.0*kaaxis(k,j)
    continue
  endif
  if(kbaxis(k,3).lt.0.0)then
    do 580 j=1,3
      kbaxis(k,j)=-1.0*kbaxis(k,j)
    continue
  endif
  if(kcaxis(k,3).lt.0.0)then
    do 590 j=1,3
      kcaxis(k,j)=-1.0*kcaxis(k,j)
    continue
  endif
endif
endif
C print study results
if(istudy.eq.1)then
C note odd offset
write(11,*),minflag(k)
if(inbhd.eq.1.or.inbhd.eq.2.or.inbhd.eq.5)then
  write(11,*),minnval(k),minval(k),minvar(k)
  write(11,*),mink(k),minj(k),mini(k)
  write(11,*),ptra(l),ptrb(l),ptrc(l)
  write(11,*),minerr,minsqerr
  write(11,*),(minaaxis(k,i),i=1,3)
write(11,*) (minbaxis(k,i), i=1,3)
write(11,*) (mincaxis(k,i), i=1,3)
endif
if(inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5) then
write(11,*) npts, minjval(k), minjvar(k)
write(11,*) minjk(k), minji(k)
write(11,*) ptra(2), ptrb(2), ptrc(2)
write(11,*) minjerr, minjsqerr
write(11,*) minja(k), minjb(k), minjc(k)
write(11,*) (jaaxis(k,i), i=1,3)
write(11,*) (jbaxis(k,i), i=1,3)
write(11,*) (jcaxis(k,i), i=1,3)
endif
if(inbhd.eq.4.or.inbhd.eq.5) then
write(11,*) minknval(k), minkval(k), minkvar(k)
write(11,*) minkk, minkj, minki, ptra(3), ptrb(3), ptrc(3)
write(11,*) minkerr, minksqerr
write(11,*) minka(k), minkb(k), minkc(k)
write(11,*) (kaaxis(k,i), i=1,3)
write(11,*) (kbaxis(k,i), i=1,3)
write(11,*) (kcaxis(k,i), i=1,3)
endif
C note odd offset on istudy
endif
1000 continue
C end species cycle, print final results
do 1500 j=1,nkvar
write(12,*) m, minflag(j), apt, bpt, cpt
if(inbhd.eq.1.or.inbhd.eq.2.or.inbhd.eq.5) then
write(12,*) minnval(j), minval(j), minvar(j), relerr(j)
write(12,*) mina(j), minb(j), minc(j)
write(12,*) (minaaxis(j,i), i=1,3)
write(12,*) (minbaxis(j,i), i=1,3)
write(12,*) (mincaxis(j,i), i=1,3)
if(minval(j).gt.maxmean(j)) maxmean(j)=minval(j)
if(minval(j).lt.minmean(j)) minmean(j)=minval(j)
meanvalue(j) = meanvalue(j) + minval(j)
sqmean(j) = sqmean(j) + minval(j)**2
meanvar(j) = meanvar(j) + var(j)
count(j) = count(j) + 1
endif
if(inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5) then
write(12,*)npts,minjval(j),minjvar(j),reljerr(j)
write(12,*)minja(j),minjb(j),minjc(j)
write(12,*),'direction cosines x,y,z axis:'
write(12,*)jaaxis(j,i),i=1,3)
write(12,*)jbaxis(j,i),i=1,3)
write(12,*)jcaxis(j,i),i=1,3)
if(minjval(j).gt.maxjmean(j))
  maxjmean(j)=minjval(j)
if(minjval(j).lt.minjmean(j))
  minjmean(j)=minjval(j)
meanjvalue(j)=meanjvalue(j)+minjval(j)
sqjmean(j)=sqjmean(j)+minjval(j)**2
meanjvar(j)=meanjvar(j)+minjvar(j)
countj(j)=countj(j)+1
endif
if(inbhd.eq.4.or.inbhd.eq.5)then
  if(minknval(j).ne.0)then
    results(minknval(j))=results(minknval(j))+1
    sum(j)=sum(j)+1
    if(minkval(j).gt.maxkmean(j))
      maxkmean(j)=minkval(j)
    if(minkval(j).lt.minkmean(j))
      minkmean(j)=minkval(j)
    meankvalue(j)=meankvalue(j)+minkval(j)
sqkmean(j)=sqkmean(j)+minkval(j)**2
    meankvar(j)=meankvar(j)+minkvar(j)
countk(j)=countk(j)+1
  else
    minkval(j)=missing
    minkvar(j)=missing
  endif
write(12,*)minknval(j),minkval(j),minkvar(j),
  relkerr(j)
write(12,*),'direction cosines x,y,z axis:'
write(12,*)kaaxis(j,i),i=1,3)
write(12,*)kbaxis(j,i),i=1,3)
write(12,*)kcaxis(j,i),i=1,3)
write(11,*)minkval(j),minkvar(j),meankvar(j)
endif
1500 continue
C end cycle
2000 continue
   do 2200 j=1,nkvar
      if(inbhd.eq.1.or.inbhd.eq.2.or.inbhd.eq.5)then
         meanvalue(j)=meanvalue(j)/float(count(j))
         sqmean(j)=sqmean(j)/float(count(j))
         meanvar(j)=meanvar(j)/float(count(j))
         varmean(j)=sqmean(j)-meanvalue(j)**2
      endif
      if(inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5)then
         meanjvalue(j)=meanjvalue(j)/float(countj(j))
         sqjmean(j)=sqjmean(j)/float(countj(j))
         meanjvar(j)=meanjvar(j)/float(countj(j))
         varjmean(j)=sqjmean(j)-meanjvalue(j)**2
      endif
      if(inbhd.eq.4.or.inbhd.eq.5)then
         meankvalue(j)=meankvalue(j)/float(countk(j))
         sqkmean(j)=sqkmean(j)/float(countk(j))
         meankvar(j)=meankvar(j)/float(countk(j))
         varkmean(j)=sqkmean(j)-meankvalue(j)**2
      endif
   2200 continue
      if(inbhd.eq.1.or.inbhd.eq.2.or.inbhd.eq.5)then
         write(12,*)'maxmean,minmean,meanvalue,sqmean,'
         + varmean,meanvar:'
         write(12,*)(maxmean(j),j=1,nkvar)
         write(12,*)(minmean(j),j=1,nkvar)
         write(12,*)(meanvalue(j),j=1,nkvar)
         write(12,*)(sqmean(j),j=1,nkvar)
         write(12,*)(varmean(j),j=1,nkvar)
         write(12,*)('max nbhd mean:',(maxmean(j),j=1,nkvar)
         write(12,*)('min nbhd mean:',(minmean(j),j=1,nkvar)
         write(12,*)('mean nbhd mean:',
         + (meanvalue(j),j=1,nkvar)
         write(12,*)('mean nbhd sqmean:',
         + (sqmean(j),j=1,nkvar)
         write(12,*)('var nbhd mean:',(varmean(j),j=1,nkvar)
         write(12,*)('mean nbhd var:',(meanvar(j),j=1,nkvar)
      endif
  2200 continue
      if(inbhd.eq.2.or.inbhd.eq.3.or.inbhd.eq.5)then
         write(12,*)'maxmean,minmean,meanvalue,sqmean,'
do 3000 j=nprmin,nprmax
   write(12,*),(results(mvar,j), mvar = 1,nkvar)
3000  continue
endif
return
end

SUBROUTINE estimate_multi_values(ptrs,aptr,bptr,cptr, +
   nptrs,value,sqvalue,variance,nvalue)
C calculate mean and variance of nbhd data values
C Written by M.M.Moody Sept, 1992

real value, sqvalue,variance,missing
integer ptrs(4),aptr(200),bptr(200),cptr(200)
include 'opt.inc'
C use average as mean for iest=1
missing=1.e31
if(iest.eq.1)then
   if(iso.eq.1)then
      if(idel.eq.1)then
         value=0.0
         sqvalue=0.0
         nvalue=0
      else
         value=data(ptrs(1),kdvar)
         sqvalue=value*value
         nvalue=1
      endif
   endif
   do 10 i = 1,nptrs
      if(data(aptr(i),kdvar).eq.missing)then
         write(ll, *)'missing value aptr array
                     npts'
         go to 10
      endif
      value =value + data(aptr(i),kdvar)
      sqvalue =sqvalue + (data(aptr(i),kdvar))**2
      nvalue=nvalue+ 1
10   continue
else
   if(datadim.eq.1)then
      write(*,*)'fatal error datadim in estimate s.r.'
stop
endif
if(idel.eq.1)then
  value=data(ptrs(2),kdvar)
  sqvalue=value*value
  nvalue=1
else
  value=data(ptrs(1),kdvar)
  sqvalue=value*value
  value=value+data(ptrs(2),kdvar)
  sqvalue=sqvalue+(data(ptrs(2),kdvar))**2
  nvalue=2
endif
if(datadim.eq.2)then
  do 20 i=1,nptrs
    k=bptr(i)
    if(data(aptr(k),kdvar).eq.missing)then
      write(11,*)'missing value bptr array < npts'
      go to 20
    endif
    value=value+data(aptr(k),kdvar)
    sqvalue=sqvalue+(data(aptr(k),kdvar))**2
    nvalue=nvalue+1
  20 continue
else
  value=value+data(ptrs(3),kdvar)
  sqvalue=sqvalue+(data(ptrs(3),kdvar))**2
  nvalue=nvalue+1
  do 30 i=1,nptrs
    k=cptr(i)
    if(data(aptr(k),kdvar).eq.missing)then
      write(11,*)'missing value cptr array < npts'
      go to 30
    endif
    value=value+data(aptr(k),kdvar)
    sqvalue=sqvalue+(data(aptr(k),kdvar))**2
    nvalue=nvalue+1
  30 continue
endif
value=value/float(nvalue)
sqvalue=sqvalue/float(nvalue)
SUBROUTINE ellipse3d(x0,y0,z0,x1,y1,z1,x2,y2,z2,
+ a,b,alpha)

C find ellipse parameters from 3 points
Written by M.M.Moody Sept, 1992

real x0,y0,z0,x1,y1,z1,x2,y2,z2,a,b,arg,radius,chord,
+ alpha,pi
integer un11
un11=11
pi=4.0*atan(1.0)

C find 2 sides of triangle to a of three ellipse pts
arg=(x2-x0)**2+(y2-y0)**2+(z2-z0)**2
radius=sqrt(arg)
arg=(x2-x1)**2+(y2-y1)**2+(z2-z1)**2
chord=sqrt(arg)

C law of cosines gives alpha = ellipse angle
arg=(a**2+radius**2-chord**2)/(2.0*a*radius)
if(arg.gt.1.0.or.arg.lt.-1.0)then
  write(un11,*)' cosine argument exceeded: ',arg
endif
if(arg.le.-1.0)then
  alpha=pi
  b=0.0
elseif(arg.ge.1.0)then
  alpha=0.0
  b=0.0
else
  C use radial version of ellipse equation to find b
  alpha=acos(arg)
  arg=(radius*a*sin(alpha))**2/(a**2-(radius*arg)**2)
  b=sqrt(arg)
endif

C alpha in radians
return
end
SUBROUTINE ellipsoid(xO,yO,zO,x1,y1,zl,x2,y2,z2,x3,y3,z3,a,b,c,aaxis,baxis,caxis,m,flag)

C completes ellipsoid parameters from 4 points defining ellipsoid and a and b; x0 is center, x1 is a pt, x2 is b pt and x3 is c pt.
C Written by M.M.Moody Sept 1992

real xO,yO,zO,x1,y1,zl,x2,y2,z2,x3,y3,z3,a,b,c,
+ pi,arg,tiny5,xp,yp,zp,vabs,dot,bsq,
+ aaxis(3), baxis(3), caxis(3), apt(3), bpt(3),
+ cpt(3), tst(3)
integer un11, un12, flag, m
save pi,arg,tiny5,xp,yp,zp
C pt vectors are local coords translated to origin of ellipse given by x0,y0,z0; axis vectors are directions of new axis in local coords, alpha is ellipse angle
flag=0
un11=11
un12=12
pi=4.0*atan(1.0)
arg=(x1-xO)**2+(y1-yO)**2+(zl-z0)**2
+ (x2-xO)**2+(y2-yO)**2+(z2-z0)**2
+ (x3-xO)**2+(y3-yO)**2+(z3-z0)**2
tiny5 =0.00001*sqrt(arg)
bsq=b*b
apt(1)=x1-x0
apt(2)=y1-yO
apt(3)=z1-z0
arg=apt(1)**2+apt(2)**2+apt(3)**2
bpt(1)=x2-x0
bpt(2)=y2-yO
bpt(3)=z2-z0
C test for line degeneracy
call cross(apt,bpt,tst)
arg=vabs(tst)
if(arg.lt.tiny5)then
    write(un11,*)'degenerate ellipsoid:
+ b point almost on a axis'
    flag=1
    b=0
    c=0
endif

cpt(1)=x3-x0

cpt(2)=y3-y0

cpt(3)=z3-z0

C test for line degeneracy

call cross(apt,cpt,tst)

arg=vabs(tst)

if(arg.lt.tiny5)then

write(un11,*)'degenerate ellipsoid:
+c point almost on a axis'

c=0

flag=2

endif

C test for plane degeneracy

arg=dot(tst,bpt)

arg=abs(arg)

if(arg.le.tiny5)then

write(un11,*)'degenerate ellipsoid,
+c pt almost in ab plane'

c=0

flag=3

endif

C find direction cosines of new x axis = a axis

aaxis(1)=apt(1)/a

aaxis(2)=apt(2)/a

aaxis(3)=apt(3)/a

C remove degeneracy in dir cos

if(flag.eq.1)then

bpt(1)=0.0

bpt(2)=0.0

bpt(3)=0.0

if(aaxis(1).lt.0.8)then

bpt(1)=1

else

bpt(2)=1

endif

endif

C caxis: cross new a axis and point defining b axis

call cross(aaxis,bpt,caxis)

C baxis: cross new c axis and new a axis

call cross(caxis,aaxis,baxis)

C direction cosines of new y axis = b axis
arg = sqrt(baxis(1)**2 + baxis(2)**2 + baxis(3)**2)
baxis(1) = baxis(1)/arg
baxis(2) = baxis(2)/arg
baxis(3) = baxis(3)/arg

C direction cosines of new z axis = c axis
arg = sqrt(caxis(1)**2 + caxis(2)**2 + caxis(3)**2)
caxis(1) = caxis(1)/arg
caxis(2) = caxis(2)/arg
caxis(3) = caxis(3)/arg
if(flag.ne.0) then
  return
endif

C rotate coordinates of cpt to new axes
xp = (cpt(1))*aaxis(1) + (cpt(2))*aaxis(2) + (cpt(3))*aaxis(3)
yp = (cpt(1))*baxis(1) + (cpt(2))*baxis(2) + (cpt(3))*baxis(3)
zp = (cpt(1))*caxis(1) + (cpt(2))*caxis(2) + (cpt(3))*caxis(3)
arg = 1.0 - ((xp/a)**2 - (yp/b)**2)
if(arg.lt.0.0) then
  write(un11,*),'negative zsq/csq:',arg
endif
if(arg.lt.tiny5) then
  write(un11,*),inaccurate calc. set arg to tiny'
  arg = zp**2/tiny5
if(arg.gt.bsq) then
  write(un11,*),'c gt b; set c to b; (zp/c)**2:',arg
c = b
  return
endif
c = sqrt(arg)
else
  arg = zp**2/arg
  if(arg.gt.bsq) then
    write(un11,*),'c gt b; set c to b; (zp/c)**2:',arg
c = b
  endif
c = sqrt(arg)
endif
arg = abs((b-c)/(b+c))
if (arg.le.tiny5) then
write(un11,*)'b practically c,'
+     set to ellipsoid of revolution'
b=(b+c)/2.0
    c=b
endif
return
end
B.2 Variogram Routines

C vario.inc - global variables for combined programs
C - prevar and vario with added kernel
C - smoothed variograms

C Description of Global variables and program limits

C MAXDATA   - maximum samples allowed
C MAXVAR    - maximum variables allowed
C MAXPAIR   - maximum pairs allowed
C MAXPAIRLAG - maximum pairs in a lag
C MAXCUTOFF - maximum number of lag cutoffs
C MAXLAG    - maximum number of lag intervals
C MAXSTRUC  - max. no. of nested variogram structures
C GAM       - estimator type (used with GammaType)
C MAXEQ     - Maximum number of equations
C MAXNEIGHBOR- Maximum number of neighbors
C MAXPTS    - maximum number of points for kriged grid
C UN10 - UN15- input output unit numbers

INTEGER MAXVAR, MAXDATA, MAXPAIR, MAXPAIRLAG
INTEGER MAXCUTOFF, MAXLAG, MAXSTRUC, MAXEQ
INTEGER MAXNEIGHBOR, MAXPTS
INTEGER GAMABS1, GAMABS2, GAMREL, GAMERGO
INTEGER UN10, UN11, UN12, UN13, UN14, UN15
PARAMETER (MAXVAR = 4, MAXDATA = 100)
PARAMETER (MAXPAIR = 2000, MAXPAIRLAG = 500)
PARAMETER (MAXCUTOFF = 40, MAXLAG = 40)
PARAMETER (MAXSTRUC = 4, MAXNEIGHBOR = 50)
PARAMETER (MAXEQ = MAXNEIGHBOR+6)
PARAMETER (MAXPTS = 1000)
PARAMETER (GAMABS2 = 0, GAMABS1 = 2)
PARAMETER (GAMREL = 1, GAMERGO = 3)
PARAMETER ( UN10 = 10, UN11 = 11, UN12 = 12)
PARAMETER ( UN13 = 13, UN14 = 14, UN15 = 15)

C VarNam    - array of variable names
C Units     - array of variable units
C Data_File - data file
C Pair_File - pair comparison file
C Nbhd_File - optimal parameters for neighborhoods file
C Sort_File - file of sorted arrays
C OutPut_File - Output file of run parameters and data
C Results_File - Output file of estimate values for plotting
C Title - array of titles for variogram plot
C Header - Header for output file and title for graphs

CHARACTER VarNam(MAXVAR)*10, Units(MAXVAR)*10,
+ Data_File*14, Pair_File*14, OutPut_File*14,
+ Results_File*14, Title(4)*60, Header*80,
+ Nbhd_File*14, Sort_File*14

COMMON /CHARS/ VarNam, Units, Data_File,
+ Pair_File, Nbhd_File, Sort_File, OutPut_File,
+ Results_File, Title, Header

C Ndata - the number of samples (rows in matrix)
C Nvar - the total number of variables (columns)
C Xvar - column number for x coordinate
C Yvar - column number for y coordinate
C Evar - column number for e coordinate
C kVar - column number for current data value
C kdVar - kvar-datadim: current data column no.
C Npair - the number of pairs in pair file
C TotalPairs - the total number of pairs retained
C Nlag - the number of lag cutoffs
C Nlagfit - the number of lags used in model fitting
C GammaType - code for the type of estimator
C Nkept - the no. of values used in variogram calc.
C NMiss - the number of missing values in data
C NumNeg - the number of sample values <=0.0
C Current_Lag - current lag number
C Mtype - array of variogram type codes for models
C Pairs - the no. of pairs in a lag by var.
C From - ptr to data for 1st sample in pair
C To - ptr to data for second sample in pair
C DiffPtr - array of pointers to squared differences
C Flpairs - floating value of real pairs
C Ppair - holds sorted order index (by distance)
C Ppair - the number of possible pairs
C Dimension - the dimension of the problem space
C VarDimen - the dimension of the variogram space
C DataDim - The dimension of the data set coordinates
C Layers - the number of layers in a pseudo 2D space
C Language - Language code 1 = english.
C NumEstimate- the number of estimates produced
C Neq - the number of equations to be solved
C Nrecord - the number of Ndata records kept
C KrigVar - array of dependent variables to be kriged
C NkVar - the number of variables to be kriged
C Mvar - index for the variable being kriged
C numptsx - the number of grid pts in x direction
C numptsy - the number of grid pts in y direction
C numptse - the number of grid pts in e direction
C Same - the no. of interp. pt.s equal to data pt.s
C Npoints - the number of interpolation points
C Num_Errors - the number of errors in system solution
C Num_NoFound- the no. of no neighbors found in solution
C chckpt - the data pt. for printout check, 0 = none

C Variogrm parameters:
C Imodel - the model type for each structure (up to 4)
C NZeroDist - number of zero distances in pcf
C Nstruct - the number of structures in variogram model
C Nsills - the number of sills in finite models

C Neighborhood Search Parameters:
C YValPtr - Pointer list to data (parallel to Sptr)
C Zptr - Pointer ro data set entry
C Min_Nbhd_points - min number of points per nbhd to solve
C Evalptr - Pointer list to data (parallel to Sptr)

C Modeling parameters:
C mfit - the number of terms to be fit
C nfit - the number of data points to be fit
C Imod1 - the type of function used for model 1
C Imod2 - the type of function used for model 2
C Imodn - the type of function used for model n

C Option codes:
C Ikrig - code for kriging system
C Layers - code for number of layers in 2-D system
C Ilocal - code for local or global neighborhoods
C Ioption - code for calculation options
C Iso - code for isotropic(1) or anisotropic(2)
C xval  - no. of xvalid point, else 0
C inbhd  - nbhd code
C icalc  - code for smoothing parameter calculation
C ivgm  - variogram code
C kfun  - kernel function array
C ifk  - current kernel function index

INTEGER Ndata, Nvar, Xvar, Yvar, Evar, kVar
INTEGER kdVar, Mvar, Npair, TotalPairs, Nlag
INTEGER Nlagfit, GammaType, icalc, Nkept, NMiss
INTEGER NumNeg, Current_Lag, Language, ivgm
INTEGER Ppair, PairPtr(MAXPAIR), Dimension
INTEGER VarDimen, DataDim, Mtype(MAXSTRUC)
INTEGER Pairs(MAXLAG, MAXVAR), inbhd, From(MAXPAIR)
INTEGER To(MAXPAIR), Flpairs(MAXLAG), NModel
INTEGER DiffPtr(MAXPAIRLAG), Imodel(MAXSTRUC)
INTEGER Nstruc, NZeroDist, NumEstimate, Ikrig,
INTEGER Layers, Ilocal, Ioption, Min_Nbhd_Points
INTEGER Nrecord, KrigVar(MAXVAR), NkVar, iso, xval
INTEGER Numptsx, Numpty, Numptse, Same, Npoints
INTEGER chkpt, Num_Errors, Num_NoFound, mfit
INTEGER Y Valptr(MAXDATA), Zptr(MAXDATA), nfit
INTEGER Evalptr(MAXDATA), kfun(MAXVAR), ifk
INTEGER imod1, imod2, imod3, imod4, imod5

COMMON /INTS/ Ndata, Nvar, Xvar, Yvar, Evar,
  + kVar, kdVar, Mvar, Npair, TotalPairs, Nlag,
  + Nlagfit, GammaType, Nkept, Nmiss, NumNeg, mfit,
  + Current_Lag, Neq, NModel, Pairs, From, To, nfit,
  + DiffPtr, Flpairs, DataDim, Ppair, PairPtr,
  + Dimension, Layers, Language, Nstruc, NZeroDist,
  + Imodel, NumEstimate, Ikrig, Ilocal, Ioption,
  + Min_Nbhd_Points, iso, xval, VarDimen, Nrecord,
  + KrigVar, NkVar, Numptsx, Numpty, Numptse, Mtype,
  + Same, Npoints, Nsills, Num_Errors, Num_NoFound,
  + Zptr, Y Valptr, Evalptr, inbhd, icalc, ivgm, kfun,
  + ifk, chkpt, imod1, imod2, imod3, imod4, imod5

C Xdata  - the array of X values
C Ydata  - the array of Y values
C Edata  - the array of E values
C Data  - the matrix of Data Values
C Resid - the matrix of residuals
C MaxDist - the maximum interpair distance used
C MinDist - the minimum interpair distance used
C VarMin - minimum data value
C VarMax - maximum data value to use
C XCMin - minimum x coordinate value
C XCMax - maximum x coordinate value
C YCMin - minimum y coordinate value
C YCMax - maximum y coordinate value
C ECMIn - minimum e coordinate value
C ECMax - maximum e coordinate value
C Distance - distance for each pair
C Vector - coordinates of vector formed by pair
C Azimuth - horizontal direction parameter
C Dip - vertical direction parameter
C Tolerance - tolerance for calculation
C BandWidth - maximum bandwidth parameter
C Missing - missing value
C LagMinDist - first value for lag cutoffs
C LagMaxDist - last value for lag cutoffs
C LagIncDist - lag Increment Value
C LagCutoffs - lag cutoffs values
C Tcutoff - individual lagcutoff value
C Fraction - value used [in prevar] to subset pairs
C StdDev - array of standard deviations
C Xdist - length over x axis for estimate
C Ydist - length over y axis for estimate
C Edist - length over e axis for estimate
C Delx - distance between estimated points
C Dely - distance between estimated points
C Dele - distance between estimated points
C DataRec - temporary data array for input

C for each lag:
C AvgDist - mean distance per lag
C LagData - values used in differences
C Estimate - the values retained for modelling
C Difference - array of differences squared in xvalid
C Zscore - (Difference/Variance) in xvalid
C Mfrom, Mto - mean of from pairs, to pairs
C Vmin, Vmax - min max for Variogram values
C Emin, Emax - min max for Estimated values
C Smin, Smax - min max for StdDev values
C Dmin, Dmax - min max for Differences
C Zmin, Zmax - min max for Zscores
C VQ1, VQ3 - 1st and 3rd Quartiles of Variogram data
C EQ1, EQ3 - 1st and 3rd Quartiles of Variogram data
C SQ1, SQ3 - 1st and 3rd Quartiles of Variogram data
C DQ1, DQ3 - 1st and 3rd Quartiles of Variogram data
C ZQ1, ZQ3 - 1st and 3rd Quartiles of Variogram data
C VMean, VMedian- increment mean, median of Variograms
C EMean, EMedian- increment mean, median of Estimates
C SMean, SMedian- increment mean, median of StdDev values
C DMean, DMedian- increment mean, median of Differences
C ZMean, ZMedian- increment mean, median of Zscore values
C Estd - Standard Deviation of estimated values
C Sstd - Standard Deviation of standard deviations
C Dstd - Standard Deviation of Differences
C Vstd - Standard Deviation of Data
C Zstd - Standard Deviation of Zscores
C Mean - mean of data values
C dVariance - Variance of input data
C nVariance - Variance of data in nbhds
C xVariance - error of estimate in xvalid
C LagMean - mean of all values in the lag
C LagVariance - variance of all values in the lag
C VarianceFrom - variance of FROM pairs
C VarianceTo - variance of TO pairs
C GamPower1 - madogram values
C GamPower2 - variogram values
C GamRelative - relative variogram values
C NonErgodic - nonErgodic variogram values
C Scale - scale of kriging nbhd to vgm nbhd

C the variogram and model:
C Sill - array of variogram sills for models
C Range1 - array of variogram ranges for models
C Range2 - array of variogram ranges for models
C Range3 - array of variogram ranges for models
C Varazimuth - Array of variogram ellipsoid azimuths
C Vardip - Array of variogram ellipsoid dips
C Nugget - variogram nugget effect
C Mglobal - Global mean for simple kriging
C Model - the model values for AvgLagDist
C GamMax - maximum value of variogram model
C Lambda - inverse of pseudo range for some models
C SPram - array of smoothing parameters
C gam - array of model values

C Neighborhood Search Variables and Estimation Variables
C 3-D parameters follow:
C B - vector for solving kriging system
C B2 - vector for solving kriging system
C W - array for solving kriging system
C Zptr - pointer array to data set entry
C Vk - working value of estimate
C Sk - working value of stddev
C xorigin - kriging grid x origin
C yorigin - kriging grid y origin
C eorigin - kriging grid e origin
C EVValues - List of ascending E coordinate values

C 2-D parameters follow:
C Yvalues - Array of values in Y ascending order
C SrMajor - a axis of ellipse
C SrMinor - b axis of ellipse
C SrAngle - ellipse angle in degrees?
C aaxis - direction cosines of a axis
C baxis - direction cosines of b axis
C caxis - direction cosines of c axis
C a0,b0,c0 - axes lengths
C theta - angle
C alpha - angle

C variogram parameters:
C lam0 - beginning value of lambda
C lamb - working value of lambda
C lamr - right side pegged lambda
C laml - leftt side pegged lambda
C lmbda - array of lambda values
C lams - another lambda
C maxlamb - max lambda
C minlamb - min lambda
C nug0 - initial and working value or nugget
C sil0 - initial and working value of sill
REAL*4     XCmin, XCmax, YCmin, YCmax, ECmin, ECmax,
+     theta, alpha, eps, asq, bsq, csq, xi, yi, zi, xp, yp, zp, rsq,
+     MinDist, MaxDist, Missing, VarMin, VarMax, Mean,
+     Azimuth, Dip, Tolerance, BandWidth, PA(3),
+     dVariance, LagMinDist, LagMaxDist, LagIncDist,
+     Fraction, xVariance, Xdata(MAXDATA),
+     Ydata(MAXDATA), Edata(MAXDATA), nVariance,
+     Data(MAXDATA, MAXVAR), Distance(MAXPAIR),
+     Vector(3, MAXPAIR), VMean(MAXLAG),
+     VMedian(MAXLAG), Resid(MAXDATA, MAXVAR),
+     AvgDist(MAXLAG), Zscore(MAXLAG),
+     Difference(MAXPAIRLAG), LagCutoff(MAXCUTOFF),
+     StdDev(MAXDATA), Mfrom(MAXLAG), Mto(MAXLAG),
+     Vmin(MAXLAG), Vmax(MAXLAG), Estimate(MAXLAG),
+     VQ1(MAXLAG), VQ3(MAXLAG), DataRec(MAXVAR),
+     LagMean(MAXLAG), LagVariance(MAXLAG), Scale,
+     VarianceFrom(MAXLAG), VarianceTo(MAXLAG),
+     GamPower1(MAXLAG), GamPower2(MAXLAG),
+     GamRelative(MAXLAG), NonErgodic(MAXLAG),
+     Model(MAXLAG), Sill(MAXSTRUC), Nugget, Mglobal,
+     Range1(MAXSTRUC), Range2(MAXSTRUC), SillO,
+     Range3(MAXSTRUC), Varazimuth(MAXSTRUC),
+     Vardip(MAXSTRUC), LMBDA(MAXDATA), nugo,
+     Delx, Dely, Dele, lamO, lamr, lamI, lamb, lams,
+     Xdist, Ydist, Edist, GamMax, b(maxeq), b2(maxeq),
+     W(maxeq, maxeq), Vk, Sk, maxlam, minlam, gam(MAXLAG),
+     Tcutoff, Lambda, EQ1, EQ3, SQ1, SQ3, DQ1, DQ3,
+     ZQ1, ZQ3, Emin, Emax, Smin, Smax, Dmin, Dmax, Zmin,
+     Zmax, Emean, Smean, Dmean, Zmean, Estd, Sstd, Dstd,
+     Vstd, Zstd, Emedian, Smedian, Dmedian, Zmedian,
+     xorigin, yorigin, eorigin, Yvalues(MAXDATA),
+     Smin, Smax, Dmin, Dmax, Zmin, Zmax, Emean, Smean, Dmean, Zmedian,
+     xorigin, yorigin, eorigin, Yvalues(MAXDATA),
+     SrMajor, SrMinor, SrAngle, SParam(MAXVAR),
+     aaxis(3), baxis(3), caxis(3), a0, b0, c0

COMMON /FLOATS/     XCmin, XCmax, YCmin, YCmax, ECmin, ECmax,
+     ECmax, MinDist, MaxDist, Missing, Fraction,
+     xVariance, VarMin, VarMax, Mean, dVariance,
+     DataRec, Azimuth, Dip, Tolerance, BandWidth,
+     LagMinDist, LagMaxDist, LagIncDist, Zscore,
+     Xdata, Ydata, Edata, Data, Distance, Vector,
+     Difference, LagCutoff, AvgDist, Estimate,
+     Mfrom, Mto, Vmin, Vmax, VMean, VMedian, PA,
PROGRAM Kernel_Smoothed_Variograms

C - variosma.f - Variogram analysis program
C - Geo-EAS PC version transported to VAX envir
C - and code modified to handle 3 dimensions
C - menu-driven user interface stripped off
C - NCAR graphics stripped off
C - tested with Example.dat from Geo-EAS
C - August 1990 by Hannah Rasmussen Rhodes.
C - Criteria added for exp. variogram selection
C - Added nugget and sill selection criteria
C - Added simple default model calc. in finite.f
C - Need nbhd file generated in opt.x then need
C - .pcf file generated in geostat.x before run
C - by M.M.Moody in 1992
C - code for graphics based on GKS not given

include 'vario.inc'
character*14 Plot_File, Mod_File

C + Lag_File, Gamma_File
     real xxx(82), YYY(82), zzz(82), DIF, DIFF(82),
     + xerror(82), mu, xxx0, yyy0, zzz0, r1, r2, r10, r20,
     + flpts, iresid, xmu, xlamb, fn, z1, z2, arg, sum,
     + xminerr, ratio0, ratio1, ratio2, band, bandmin,
     + bandmax, denom
     integer three, Plot_id, Retained(82), fpair, xk,
+ iopt,iset,ismth,minpairs,ifun,itype,maxlags,
+ minlags,iker,nlagret,lpair,icycle,i,j,k,
+ ncycle,kx,kend,imon,two,ipass,unt
  common XXX, YYY, ZZZ, XXX0, YYY0, ZZZ0, DIFF,
+ r1,r2,r10,r20,flpts
  save xerror,three, Plot_id, Retained, fpair, xk,
+ iopt,iset,ismth,minpairs,ifun,itype,maxlags,
+ minlags,iker,nlagret,lpair,icycle,i,j,k,
+ ncycle,imon,ipass,mu,iresid,xmu,ratio0,
+ ratio1,ratio2,dif,xlamb,fn,z1,z2,arg,sum,
+ xminerr,kx,ken

Plot_File = 'Vario.plt'
Pair_File = 'Vario.pcf'
Out_File = 'Vario.out'
Mod_File = 'Vario.mod'
Nbhd_File = 'Vario.nhd'

C Open Output files
  open(unit=11,File=Out_File,status = 'unknown',ERR=999)
  open(unit=12,File=MOD_File,status = 'unknown',ERR=999)
Azimuth = 0.0
Dip = 0.0
Tolerance = 90.0
gambarate=2
iresid = 1
Plot_id = 1
DO 2 I=1,24
  Retained(I)=0
 2 continue
C
  Restart Point
5 continue
  write(*,*),'Enter variable of interest:'
  read(*,*)kdvar
  IOPT = 1
  ISET = 1
  NCYCLE = 1
  write(*,*),'ENTER 1 FOR STANDARD ONLY,'
  write(*,*),' 2 FOR SMOOTHING ONLY,'
  write(*,*),' 3 FOR BOTH'
  read(*,*),ISMTH
  if(ISMTH.eq.1)then
    write(*,*),'ENTER MIN NO. PAIRS IN STANDARD LAG'
C note min pairs is important parameter in calculation
C it sets nlag for standard variogram only option
if(ISMTH.EQ.2 .OR. ISMTH.EQ.3)then
   write(*,*)'ENTER LAMBDA FUNCTION CALCULATION METHOD'
   write(*,*)' 1 G lobal'
   write(*,*)' 2 GLOBAL WITH Linear PERTURBATIONS'
   write(*,*)' 3 GLOBAL WITH Quadratic PERTURBATIONS'
   write(*,*)' 4 GLOBAL WITH Cubic PERTURBATIONS'
   write(*,*)' 5 POWER LAMBDA'
   write(*,*)' 6 GAUSS LAMBDA'
else
   read(*,*)IOPT
endif
C RECYCLE POINT FOR NEW VARIABLE/KERNEL
10 continue
   call VInitialize
C get data and pair comparison information
   open(unit=UN15,File=Pair_File,status=’unknown’,
      + ERR = 9999)
C specify unit
   unt=un15
   call Read_Data(unt)
call read_Pairs(unt)
close(unt)
   LAGMAXDIST=DISTANCE(NPAIR)
C set default lag spacing
   NOTE LAGMAXDIST CHANGE
if(ISMTH.EQ.1)then
if(NCYCLE.EQ.1) then
  write('*,*)' NUMBERS OF PAIRS AND SAMPLES:',
  NPAIR,NKEPT
if(NPAIR.EQ.0.OR.NKEPT.EQ.0) then
  write('*,*)' PROGRAM NEEDS NON ZERO VALUES'
  STOP
endif
NLAG=INT(DISTANCE(NPAIR)/DISTANCE(MINPAIRS))
write('*,*)' CALCULATE NLAG:',NLAG
MAXLAGS=80
MINLAGS=8
if(NLAG.GT.80) NLAG=80
if(NLAG.LT.8) NLAG=8
endif
endif
C input smoothing variogram selection criteria
12 NCYCLE=1
if(ISMTH.EQ.2.OR.ISMTH.EQ.3) then
  write('*,*)' INPUT KERNEL SMOOTHER: 1=EXP, 2=GAUSS'
  read('*,*) IKER
  write('*,*)' SMOOTHING PARAM'S: 1=INPUT, 2=CALCULATE'
  read('*,*) ICALC
  write('*,*)' enter number of points for variogram:'
  read(' *,*) nlag
  nlagret=nlag
  fpair=1
  lpair=npair
if(ICALC.EQ.1) then
  if(iopt.le.4) then
    write(' *,*)' INPUT MIN LAMBDA, MAX LAMBDA:'
    read(' *,*) MINLAM,MAXLAM
    write(' *,*)' INPUT MID-PT LAMBDA:'
    read(' *,*) LAMB
  else
    write(' *,*)' INPUT PARAMETER: MU'
    read(' *,*) mu
endif
elseif(ICALC.EQ.2) then
  Maxlam=100.0/Distance(Npair)
  Minlam=1.0/Distance(NPAIR)
endif
call calculate_nugget
endif
15 if(ISET.EQ.1.OR.ISET.EQ.3)then
   NCYCLE=1
C compute lags  RECYCLE POINT FOR NEW LAGS
   LagMinDist = 0.0
   write( '**' )'max pair distance:',Distance(Npair)
   write( '**' )'enter maxdistance for exp vgm:'
   read( '**' )LagMaxDist
   LagIncDist = LagMaxDist / FLOAT(nlag-0.01)
   write( '**' )
   write( '**' )'Lag Parameters:'
   write( '**' )'LagMinDist',LagMinDist
   write( '**' )'LagMaxDist',LagMaxDist
   write( '**' )'LagIncDist',LagIncDist
   write( un12,**)'LagMaxDist',LagMaxDist
   write( un12,**)'LagIncDist',LagIncDist
   if (Nlag.LE.1)then
      write( '**' )'too few lags'
      go to 9999
   endif
   Tcutoff = LagMinDist + LagIncDist
   DO 100 i = 1,nlag
      LagCutoff(I) = Tcutoff
      Tcutoff = Tcutoff + LagIncDist
100 continue
   call VRank(Lagcutoff,Nlag)
endif
C compute standard variogram to study lag spacing
call Compute_Vario
NLAGRET=NLAG
C REENTRY POINT TO EXAMINE LAGS
140 continue
   if(ISMTH.EQ.1.OR.ISMTH.EQ.3)then
      if(NCYCLE.EQ.1)then
9999
      endif
   endif
   if(ISMTH.EQ.1.OR.ISMTH.EQ.3)then
      write( '**' )'FOR STANDARD VARIOGRAM ENTER:'
      write( '**' )'1 = PLOT PAIRS, 2 = REDEFINE LAGS,'
      write( '**' )'3 = write PAIRS, 4 = PRINT PAIRS'
      write( '**' )'5 = continue'
      read( '**' )ICYCLE
      NCYCLE=NCYCLE+1
   endif
C PLOT lag numbers
   if(ICYCLE.EQ.1)then
DO 160 J=1,NLAG
   FLPAIRS(J) = FLOAT(PAIRS(J,kdvar))
160   continue
   write(*,*)'REMEMBER HOW MANY LAGS TO RETAIN'
   write(*,*)
   open(unit=UN10,File=Plot_File,status='unknown',
        ERR=9999)
   Plot_ID=1
   write(*,*) Plot_ID, nlag,three,
   + (avgdist(i),flpairs(i),flpairs(i), i=1,nlag)
   write(10,*) Plot_ID, nlag,three,
   + (avgdist(i),flpairs(i),flpairs(i), i=1,nlag)
   close(10)
   display pairs information
   call read_array(Plot_file)
   call vario_twod (Plot_id)
   write(*,*)' MAX NO. OF LAG CUTOFFS IS:', NLAG
   write(*,*)'ENTER NOW MANY LAGS TO RETAIN'
   read(*,*)NLAGRET
   GO TO 150
C REDEFINE lags
elseif(ICYCLE.EQ.2)then
   write(*,*)'CURRENT NUMBER OF LAG CUTOFFS IS:', NLAG
   write(*,*)'ENTER NEW NUMBER OF LAGS (8-24)',
   read(*,*)NLAG
   ISET=3
   GO TO 15
C look at specific lags - note looking at nlag, not nlagret
elseif(ICYCLE.EQ.3)then
   DO 165 J=1,NLAG
      CURRENT_LAG=J
      if (Pairs(Current_Lag,kdvar) .EQ. 0) then
         write(6,*) 'No pairs in this lag',CURRENT_Lag
      endif
      call VCompute_Lag_Results(Current_Lag)
165   continue
   GO TO 150
elseif(ICYCLE.EQ.4)then
C display variogram pairs results
   open(unit=UN10,File=Plot_File,status='unknown',
        ERR=9999)
Plot_ID = 1
    write(10,* ) Plot_ID, nlag, three,
        (avgdist(i), flpairs(i), fipairs(i), i=1,nlag)
close(10)
call read_array(Plot_file)
call vario_twod (Plot_id)
GO TO 150
endif
endif
else
  icycle = 5
endif
C Calculate smoothed variogram for standard lags retained
C but compute kernel function from total number of lags
ratio0 = sqrt(2.0)
ratio1 = sqrt(ratio0)
ratio2 = sqrt(ratio1)
C set up for first pass
ipass = 1
if (ICALC.eq.2) lamb = Minlam
C recycle pt after global lamb calc c
  if(ISMTH.EQ.2.OR.ISMTH.EQ.3)then
    set kernel spacing to standard for comparison
    DO 175 i=1,nlag
        xxx(i) = avgdist(i)
    175 continue
  else
    use cell centered spacing
    xxx(1) = 0.5* LagMaxDist/float(nlag)
    avgdist(1) = xxx(1)
    DO 177 i=2,nlag
        xxx(i) = xxx(i-1) + 2.0*xxx(1)
        avgdist(i) = xxx(i)
    177 continue
  endif
  xminerr = 1.e30
  if (ICALC.EQ.2)then
    pass through twice
    C cycle through lambda function up to 60 times
    C first pass then only once for now
kx=1
if(ipass.eq.1)then
  kend=60
  if(iopt.eq.5.or.iot.eq.6)then
    kend=30
    mu=1.0/16.0
    lamb=0.0
  else
    mu=0.0
  endif
else
  kend=15
endif
write(12,*)'k,lamb,mu,Xerror(k),xk,xminerr:'
DO 200 k=1,kend
  C set up lambda function: use mu or lamb in lambfn
  if(ipass.eq.1)then
    if(IOPT.eq.5)then
      mu=ratiol*mu
    elseif(IOPT.eq.6)then
      mu= float(32*k)/float(kend)
    endif
    lamb = rati02*lamb
  else
    Minlam=Minlam/rati02
    Maxlam=Maxlam*rati02
  endif
  if(lamb.ge.maxlam)go to 200
  Xerror(k)=0.0
  C calc smoothed variogram from lambda function of mu
  if(iker.eq.2)ARG=ARG*ARG
C or function of lamb
  laml=Maxlam
  lamr=Minlam
  call Lambfn(mu,nlagret,iopt,ifun,iker,ipass)
DO 185 I=1,NLAG
  SUM=0.0
  YYY(I)=0.0
DO 180 J=I,NPAIR
  ARG=LMBDA(I)*ABS(DISTANCE(J)-AVGDIST(I))
iif(iker.eq.2)ARG=ARG*ARG
if(ARG.LT.30.0)then
  Z1 = Data(From(J),kdvar)
  Z2 = Data(To(J),kdvar)
  if((Z1.EQ.1.E31.OR.Z2.EQ.1.E31))GO TO 180
  FN=EXP(-ARG)
  YYY(I)=YYY(I)+0.5*FN*(Z1-Z2)*(Z1-Z2)
  SUM=SUM+FN
endif
180  continue
if(SUM.LE.0.0)then
  C use previous est
  write(11,*)'PREVIOUS ESTIMATE USED - k,lag,';k,i
  if(i.eq.1)then
    YYY(I)=Nug0
  else
    YYY(I)=YYY(I-1)
  endif
  else
    YYY(I)=YYY(I)/SUM
  endif
185  continue
C calculate criteria error first set ends
if(ITYPE.eq.1)then
  xerror(k)=xerror(k)+(yyy(I)-nug0)**2
  xerror(k)=xerror(k)+(yyy(nlag)-sil0)**2
  xerror(k)=xerror(k)+(yyy(nlag)-yyy(nlag-1))**2
elseif(ITYPE.eq.2)then
  arg=nug0-2.0*yyy(1)+(yyy(2)+yyy(1))/2.0
  denom=(avgdist(2)+avgdist(1))/2.0+avgdist(1))/2.0
  xerror(k)=xerror(k)+abs(arg)/denom**2
  arg= yyy(nlag-1)-2.0*yyy(nlag)+sil0
  denom=(avgdist(nlag)-avgdist(nlag-1))
  xerror(k)=xerror(k)+abs(arg)/denom**2
elseif(ITYPE.eq.3)then
  xerror(k)=xerror(k)+(yyy(1)-nug0)**4
  xerror(k)=xerror(k)+(yyy(nlag)-sil0)**4
  xerror(k)=xerror(k)+(yyy(nlag)-yyy(nlag-1))**4
endif
C then calc interior
do 190 i=2,nlag-1
if(ITYPE.eq.1)then
xerror(k) = xerror(k) + (yyy(i) - yyy(i-1))^2
elseif(ITYPE.eq.2) then
  arg = yyy(i-1) - 2.0*yyy(i) + yyy(i+1)
  denom = (avgdist(i+1) - avgdist(i-1))/2.0
  xerror(k) = xerror(k) + abs(arg)/denom**2
elseif(ITYPE.eq.3) then
  xerror(k) = xerror(k) + (yyy(i) - yyy(i-1))^4
endif

continue
if(xerror(k).lt.xminerr) then
  xk = k
  xlam = lamb
  xmu = mu
  xminerr = xerror(k)
endif
write(12, *) k, lamb, mu, xerror(k), xk, xminerr

continue
lamb = xlam
mu = xmu
else
C set input parameters
  laml = Maxlam
  lamr = Minlam
endif
if(ipass.eq.1.and.ICALC.ne.1.and.iop.ne.1) then
  ipass = 0
  Minlam = lamb
  Maxlam = lamb
C pass back with lamb or mu = global best
  go to 170
endif
C recalculate best variogram or INPUT VARIOGRAM
  call LAMBF (mu, nlagret, iop, ifun, iker, ipass)
  DO 215 I = 1, NLAG
    SUM = 0.0
    YYY(I) = 0.0
  DO 210 J = 1, NPAIR
    ARG = LMBDA(I)*ABS(DISTANCE(J) - AVGDIST(I))
    if(IKER.EQ.2) ARG = ARG * ARG
    if(ARG.LT.30.0) then
      Z1 = Data(From(J), kdvar)
      Z2 = Data(To(J), kdvar)
if(Z1.EQ.1.0 OR Z2.EQ.1.0) GO TO 210
FN=EXP(-ARG)
YYY(I)=YYY(I)+0.5*FN*(Z1-Z2)*(Z1-Z2)
SUM=SUM+FN
endif

210 continue
if(SUM.LE.0.0) then
   use previous est
   write(11,*) 'PREVIOUS ESTIMATE USED - k,lag:', k, i
   if(i.eq.1) then
      YYY(I)=Nug0
   else
      YYY(I)=YYY(I-1)
   endif
   else
      YYY(I)=YYY(I)/SUM
   endif
else
   endif
215 continue
DO 230 I=1,Nlagret
if(ICALC.eq.1) then
   write(12,*) 'i,lmbda(i),yyy(i):', i, lmbda(i), yyy(i)
else
   write(12,*) 'xk,i,lmbda(i),yyy(i):', xk, i, lmbda(i),
+       yyy(i)
endif
230 continue
band = 2.0/lamb
write(*,*) 'average bandwidth (2/lambda):', band
write(12,*) 'average bandwidth (2/lambda):', band
if(iopt.gt.1) then
   bandmax=2.0/minlam
   bandmin=2.0/maxlam
   write(12,*) 'maximum bandwidth:', bandmax
   write(12,*) 'minimum bandwidth:', bandmin
endif
C Compare the two variograms
C by average absolute or squared differences
if(ISMTH.eq.3) then
   call VCOMPARE(ESTIMATE,YYY,DIF,NLAGRET)
   DIFF(NLAG)=DIF
   RETained(NLAG)=NLAGRET
endif
endif
if(ISMTH.eq.2.or.ISMTH.eq.3)then
if(iopt.ne.l)then
C plot lambda function
   open(unit=UN10,File=Plot_File,status='unknown',
   ERR=9999)
   call plotxy(avgdist(i),lmbda(i),lmbda(i),nlagret)
   Plot_ID=1
   write(10,*)(Plot_ID, nlagret,three,
   + (avgdist(i),lmbda(i),lmbda(i), i=1,nlagret)
   close(10)
   call read_array(Plot_file)
   call vario_twod(Plot_id)
endif
endif
C plot experimental variogram(s)
250 continue
   open(unit=UN10,File=Plot_File,status='unknown',
   ERR=9999)
   Plot_ID=1
if(ISMTH.EQ.1)then
   call plotxy(avgdist(i),estimate(i),estimate(i),
   + nlagret)
elseif(ISMTH.eq.2)then
   call plotxy(avgdist(i),YYY(i),YYY(i),nlagret)
elseif(ISMTH.eq.3)then
   call plotxy(avgdist(i),estimate(i),YYY(i),nlagret)
endif
close(10)
C Select options
   write(12,*)'best xk,xlamb,xmu:',xk,xlamb,xmu
220 write('*,*)' ENTER: 1=REPLOT EXP. VGM,'
    write('*,*)' 2 = PRINT EXP. VRGM(S))'
    write('*,*)' 3=RESTART, '
    write('*,*)' 4=NEW VARIABLE,'
    write('*,*)' 5=NEW KERNEL'
    write('*,*)' 6=REEXAMINE LAG PAIRS,'
    write('*,*)' 7=SUMMARIZE EXP. RUNS'
    write('*,*)' 8=MODEL KERNEL VGM,'
    write('*,*)' 9=END'
C can not return to exp vgm options after modeling
read(*,*)ICYCLE
if(ICYCLE.EQ.1)then
  write(*,*)'REMEMBER HOW MANY LAGS TO RETAIN'
  GO TO 250
elseif(ICYCLE.EQ.2)then
print to printer
  write(*,*)'can not print variograms at this time'
  GO TO 220
elseif(ICYCLE.EQ.3)then
  close(10)
  GO TO 5
elseif(ICYCLE.EQ.4)then
  ISET=1
  close(10)
  GO TO 10
elseif(ICYCLE.EQ.5)then
  ISET=2
  GO TO 12
elseif(ICYCLE.EQ.6)then
  NCYCLE=1
  GO TO 140
elseif(ICYCLE.EQ.7)then
  write(un11,*)'SUMMARY OF CURVE DIFFERENCES 
  write(un11,*)'FOR NUMBER RETAINED BY NUMBER OF LAGS'
  DO 300 I=1,MAXLAGS
  write(*,*)I,RETained(I),DIFF(I)
300  continue
  write(*,*)'ENTER ANY NUMBER TO continue'
read(*,*)ICYCLE
  call Lambfn(mu,nlagret,iopt,ifun,iker,ipass)
  write(un11,*)'SUMMARY OF LAMBDA FOR MAX NPTS 
  + FOR IFUN:',IFUN
  DO 350 I=1,MAXLAGS
  write(*,*)LMBDA(I)
350  continue
  GO TO 220
elseif(ICYCLE.EQ.8)then
  if(ISMTH.EQ.3)then
    write(*,*)'CAN NOT MODEL BOTH - MODEL KERNEL'
    write(*,*)'rerun with option 1 for standard'
  endif
write(*,*)' Enter no. of lags to retain in model'
read(*,*)nlagfit
write(*,*)'enter 1 for monotonic vgm, '
write(*,*)' 2 for sin fn vgms'
read(*,*)imon
if(ISMTH.EQ.1)then
   call CALCULATE_DEFAULT_VARIOGRAMS(Estimate, +
   avgdist,two,imon)
else
   call CALCULATE_DEFAULT_VARIOGRAMS(YYY,xxx,two, +
   imon)
endif
open (unit=UN10, File=Plot_file, status='unknown', +
   ERR=9999)
Plot_ID=1
write(10,'(I6)') Plot_ID, nlag,three, +
   (avgdist(i),Estimate(i),model(i), i = 1,nlag)
close(10)
C display default model variogram
   call read_array(Plot_file)
   call vario_twod (Plot_id)
endif
go to 1000
999 write(6,*)'error reading file'
go to 1000
9999 write(6,*) 'Error doing something'
1000 close(11)
close(12)
end

SUBROUTINE VInitialize

C GEOEAS S.R. - initialize VVario global values

include 'vario.inc'
integer i
   Ndata = 0
   Npair = 0
   Nkept = 0
   NumNeg = 0
   NMiss = 0
   Xvar = 1
Yvar = 2
Evar = 3
TotalPairs = 0
MinDist = 0.0
MaxDist = 0.0
Nugget = 0.0
DO 10 I = 1, MAXSTRUC
  Mtype(I) = 0
  Sill(I) = 0.0
  Range1(I) = 0.0
10 continue
Azimuth = 0.0
Dip = 0.0
Tolerance = 90.0
BandWidth = 1.E31
Missing = 1.E31
return
end

SUBROUTINE Vcompare(zz,yy,DIFXY,N)

C Compares standard and smoothed variograms
C by average squared error. M.M.Moody, Sept, 1992

integer i,n
real YY(82), ZZ(82), DIFXY
DIFXY=0.0
DO 10 I=1,N
DIFXY=DIFXY+(YY(I)-ZZ(I))**2
10 continue
DIFXY=DIFXY/FLOAT(N)
write('*,*)'AVVERAGE SQ DIFF FOR N POINTS:',N,DIFXY
return
end

SUBROUTINE Lambfn(mu,mpts,i,opt,ifun,iker,ipass)

C functional forms for smoothing parameter,
C Written by M.M.Moody, Sept, 1992

include 'vario.inc'
real A, C, arg
real XXX(82), YYY(82), ZZZ(82), XXX0, YYY0, ZZZ0,
+    DIFF(82), r1, r2, r10, r20, flpts(82), mu, delamb
integer mpts, iopt, ifun, iker, i, midpt, ipass
common XXX, YYY, ZZZ, XXX0, YYY0, ZZZ0, DIFF,
+    r1, r2, r10, r20, flpts
if(ICALC.eq.2.and.ipass.eq.1.or.
+    iopt.eq.1.and.ICALC.eq.1)then
  C    global
    DO 20 I=1,MPTS
    LMBDA(I)=LAMB
20    continue
else
  if(IOPT.EQ.2.or.IOPT.EQ.3.or.IOPT.EQ.4)then
    C    global perturbation
    MIDPT=MPTS/2
    LMBDA(MIDPT)=LAMB
    if(IFUN.EQ.1)then
      LMBDA(MPTS)=LAMR
      DELAMB=(LMBDA(MIDPT)-LMBDA(MPTS))/FLOAT(MPTS-MIDPT)
      DO 30 I=MIDPT+1,MPTS-1
      LMBDA(I)=LMBDA(I-1)-DELAMB
      if(LMBDA(I).LT.0.0)then
        write(*,*),NEGATIVE LAMBDA - SET TO ZERO'
        LMBDA(I)=0.0
      endif
    endif
30    continue
    LMBDA(1)=LAML
    DELAMB=(LMBDA(1)-LMBDA(MIDPT))/FLOAT(MIDPT-1)
    DO 40 I=2,MIDPT-1
    LMBDA(I)=LMBDA(I-1)-DELAMB
    if(LMBDA(I).LT.0.0)then
      write(*,*),NEGATIVE LAMBDA - SET TO ZERO'
      LMBDA(I)=0.0
    endif
40    continue
  endif
endif
if(IFUN.EQ.2)then
  LMBDA(MPTS)=LAMR
  A=LAMB
  C=(LAMR-LAMB)/(XXX(MPTS)-XXX(MIDPT))**2
  DO 50 I=MIDPT+1,MPTS-1
LMBDA(I) = A + C*(XXX(I)-XXX(MIDPT))**2
if(LMBDA(I).LT.0.0)then
    write(*,*)'NEGATIVE LAMBDA - SET TO ZERO'
    LMBDA(I) = 0.0
endif

50 continue

LMBDA(1) = LAML
A = LAMB
C = (LAML-LAMB)/(XXX(MIDPT)-XXX(1))**2
DO 60 I = 2, MIDPT-1
LMBDA(I) = A + C*(XXX(MIDPT)-XXX(I))**2
if(LMBDA(I).LT.0.0)then
    write(*,*)'NEGATIVE LAMBDA - SET TO ZERO'
    LMBDA(I) = 0.0
endif
60 continue

endif
if(IFUN.EQ.3)then
    LMBDA(MPTS) = LAMR
    A = LAMB
    C = (LAMR-LAMB)/(XXX(MPTS)-XXX(MIDPT))**3
    DO 70 I = MIDPT+1, MPTS-1
    LMBDA(I) = A + C*(XXX(I)-XXX(MIDPT))**3
    if(LMBDA(I).LT.0.0)then
        write(*,*)'NEGATIVE LAMBDA - SET TO ZERO'
        LMBDA(I) = 0.0
    endif
70 continue
endif

if(IOPT.EQ.5)then
    DO 90 I = 1, MPTS
        power
90 continue
endif
elseif(IOPT.EQ.5)then
    DO 90 I = 1, MPTS
        power
LMBDA(I) = LAML*(1.0-(FLOAT(I-1)/FLOAT(MPTS-1))**MU) + LAMR*((FLOAT(I-1)/FLOAT(MPTS-1))**MU)

if(LMBDA(I).LT.0.0) then
   write(*, *) 'NEGATIVE LAMBDA - SET TO ZERO'
   LMBDA(I) = 0.0
endif
90 continue
elseif(IOPT.EQ.6) then
   LMBDA(1) = LAML
   LMBDA(MPTS) = LAMR
   DO 120, I = 2, MPTS-1
   ARG = mu*XXX(I)/lagmaxdist
   ARG = ARG*ARG
   if(ARG.LT.30.0) then
      LMBDA(I) = LAML + (LAMR-LAML)*(1.0-EXP(-ARG))
   else
      LMBDA(I) = LAMR
   endif
   if(LMBDA(I).LT.0.0) then
      write(*, *) 'NEGATIVE LAMBDA - SET TO ZERO'
      LMBDA(I) = 0.0
   endif
120 continue
endif
endif
return
end

SUBROUTINE Compute_Vario

C - GEOEAS S.R. Computes the variogram results for all lags using sorted pcf file which does not need extra pointers back to data like geostat does

include 'vario.inc'
integer I, J, K, N, NumPair
real Sum, SumSq
character*10 Number
save I, K, J, N, NumPair,
+ Sum, SumSq,
+ Number
C calculate mean and variance for population to be Kriged

\[ J = K_{dvar} \]

Totalpairs = 0
N = 0
Sum = 0.0
SumSq = 0.0
Mean = 0.0
dVariance = 0.0

DO 10 I = 1, Nkept
C use raw data smoothed data or residuals by data set input
if (Data(I,J).LT. Missing) then
N = N + 1
Sum = Sum + Data(I,J)
SumSq = SumSq + (Data(I,J) * Data(I,J))
endif

10 continue
if (N .NE. 0) then
\[
\text{dVariance} = \frac{((\text{float}(N) * \text{SumSq}) - (\text{Sum} * \text{Sum}))}{\text{float}(N)}
\]
Mean = Sum / float(N)
end if
write(12,*)
write(12,*)'Variable',J
write(12,*)'Data Variance', dVariance
write(12,*)'Data Mean', Mean

C reduced limit from nlag +1 to nlag

DO 20 I = 1, NLag
if (I .GT. MaxLag) go to 20
call VCompute_Lag_Results(I,NumPair)
Pairs(I,J) = NumPair
TotalPairs = TotalPairs + NumPair
20 continue
write(12,*)
write(12,*)'Total Pairs', TotalPairs
return
end

SUBROUTINE VCompute_Lag_Results(I,NumPair)
C - GEOEAS S.R. Compute results for an individual lag.
C - A binary search is performed to determine the
C - first pair in the lag >= the lag minimum
C - distance. Then for each successive pair < the
C - lag cutoff and within the direction tolerance
C - statistics are accumulated which are used to
C - compute the spatial correlation estimators
C - including the Variogram, Madogram, Relative
C - Variogram, and Non-Ergodic Variogram.
C Numpair - number of pairs in lag
C Dmin - minimum distance for lag (all pairs > Dmin)
C Dmax - maximum distance for lag (all pairs <= Dmax)
include 'vario.inc'
integer Numpair,Ptr,I,K,Isign,one,Retain,Omni
real Zl, Z2, Incr, Incr2
real Rnp, Np2, Temp, Tiny, Tol, DTR, Proj
double precision SumFrom, SumTo, SumSqFrom,
+ SumSqTo, Sum, SumSq
save Ptr, K, Isign, one, Retain, Omni,
+ Zl, Z2, Incr, Incr2,
+ Rnp, Np2, Temp, Tiny, Tol, DTR, Proj,
+ SumFrom, SumTo, SumSqFrom, SumSqTo, Sum, SumSq
C code 1 = true, 0 = false all set to 1 now!
data Tiny /1.E-5/,
data DTR /0.017453292519943148261757904/
C omni set for isotropic variables
     Omni = 1
     if (ABS(Azimuth) .LE. Tiny .AND. ABS(Dip) .LE. Tiny
+ .AND. ABS(Tolerance - 90.0) .LE. Tiny) Omni = 1
     Tol = Tolerance * DTR
C initialize local and global variables
     Numpair = 0
     Ptr = 0
     SumFrom = 0.0
     SumTo = 0.0
     SumSqFrom = 0.0
     SumSqTo = 0.0
     Sum = 0.0
     SumSq = 0.0
     Dmax = LagCutoff(I)
     AvgDist(I) = 0.0
     Mfrom(I) = 0.0
     Mto(I) = 0.0
     LagMean(I) = 0.0
     Vmin(I) = 0.0
$V_{max}(I) = 0.0$
$VQ1(I) = 0.0$
$V_{mean}(I) = 0.0$
$V_{Median}(I) = 0.0$
$VQ3(I) = 0.0$
$Estimate(I) = 0.0$
$GamPower1(I) = 0.0$
$GamPower2(I) = 0.0$
$GamRelative(I) = 0.0$
$NonErgodic(I) = 0.0$
$VarianceFrom(I) = 0.0$
$VarianceTo(I) = 0.0$
$LagVariance(I) = 0.0$

if (I .EQ. 1) then
    $D_{min} = \text{LagMinDist}$
else
    $D_{min} = \text{LagCutoff}(I-1)$
    if (I .GT. Nlag) $D_{max} = \text{MaxDist}$
endif

C use binary search to locate first pair in
distance interval and store this
C pair pointer into a temporary called $Ptr$
call VSearchB(Distance, Npair, Dmin, $Ptr$)
if (Ptr .EQ. 0) $Ptr = 1$
5 if (Distance(Ptr).LE.Dmin.AND.Ptr.LT.Npair) then
    $Ptr = Ptr + 1$
go to 5
endif
if (Ptr .GT. Npair) return

C process each pair in the lag
10 if (Distance(Ptr) .LE. $D_{max}$ .AND. + NumPair.LT.MaxPairLag.AND.Ptr.LE.Npair) then
C store From and To values to temporaries
    $Z1 = \text{data(from(ptr),kdVar)}$
    $Z2 = \text{data(to(ptr),kdVar)}$
    if($z1$.lt.Missing.and.$z2$.lt.Missing) then
        continue
    else
        write(*,*)'Missing value: from or to:'
        $ptr$,from(ptr),to(ptr)
        $Ptr = Ptr + 1$
go to 10
endif
if(chckpt.ne.0.and.i.eq.nlag)then
  write(un12,*)'Vcompute ptrs at nlag:',nlag
  write(un12,*)'ptr,z1,z2; Dist,from,to:',ptr,zl,z2
  write(un12,*)Distance(Ptr)
endif
if (Omni.eq.1) then
  Retain = 1
else
  omni set to one
endif
if (Retain.eq.1) then
  if (NumPair.LT. MaxPairLag) then
    NumPair = NumPair + 1
  else
    write(6,*) 'Maximum pairs reached'
go to 20
  endif
if (Isign.EQ.-1) then
  SumFrom = SumFrom + Z2
  SumTo = SumTo + Z1
  SumSqFrom = SumSqFrom + (Z2 * Z2)
  SumSqTo = SumSqTo + (Z1 * Z1)
else
  SumFrom = SumFrom + Z1
  SumTo = SumTo + Z2
  SumSqFrom = SumSqFrom + (Z1 * Z1)
  SumSqTo = SumSqTo + (Z2 * Z2)
endif
Incr = ABS (Z1-Z2)
Incr2 = Incr * Incr
Difference(NumPair) = Incr2
DiffPtr(NumPair) = Ptr
GamPower1(I) = GamPower1(I) + Incr
GamPower2(I) = GamPower2(I) + Incr2
VMean(I) = VMean(I) + Incr2
AvgDist(I) = AvgDist(I) + Distance(Ptr)
NonErgodic(I) = NonErgodic(I) + (Z1 * Z2)
endif
Ptr = Ptr + 1
go to 10
endif

20 continue
if (NumPair .LE. 0) go to 999
C compute Variogram, Madogram, Relative Variogram,
C and NonErgodic Variogram
Mfrom(I) = SumFrom / float(NumPair)
Mto(I) = SumTo / float(NumPair)
LagMean(I) = (Mfrom(I) + Mto(I)) / 2.0
AvgDist(I) = AvgDist(I) / float(NumPair)
VMean(I) = VMean(I) / float(NumPair)
GamPower1(I) = GamPower1(I)/float(NumPair+NumPair)
GamPower2(I) = GamPower2(I)/float(NumPair+NumPair)
if (LagMean(I) .NE. 0) GamRelative(I) =
+ GamPower2(I) / (LagMean(I) * LagMean(I))
Sum = SumFrom + SumTo
SumSq = SumSqFrom + SumSqTo
Rnp = float (NumPair * (Numpair - 1))
Np2 = float (NumPair+NumPair)
if (Omni.eq.1) then
  Mfrom(I) = LagMean(I)
  Mto(I) = LagMean(I)
endif
NonErgodic(I) = NonErgodic(I) / float(NumPair) -
+ Mfrom(I) * Mto(I)
NonErgodic(I) = dVariance - NonErgodic(I)
if (NumPair .GT. 1) then
  LagVariance(I) = (Np2*SumSq-Sum*Sum)
+ / (Np2 * (Np2 - 1.0))
  VarianceTo(I) = (float(NumPair)*SumSqTo -
+ SumTo*SumTo) / Rnp
  VarianceFrom(I) = (float(NumPair)*SumSqFrom -
+ SumFrom*SumFrom) / Rnp
if (Omni.eq.1) then
  Temp = (VarianceTo(I) + VarianceFrom(I))/2.0
  VarianceTo(I) = Temp
  VarianceFrom(I) = Temp
  LagVariance(I) = Temp
endif
endif
if (GammaType .EQ.1) then
  Estimate(I) = GamPower1(I)
else if (GammaType .EQ. 2) then
  Estimate(I) = GamPower2(I)
else if (GammaType .EQ. 3) then
  Estimate(I) = GamRelative(I)
else if (GammaType .EQ. 4) then
  Estimate(I) = NonErgodic(I)
endif
write(un12,*9'NumPairs,i,estimate(i),'
     NumPair,i,estimate(i)
if (NumPair .GT. 1) then
C sort the differences (along with pointers)
C into ascending order
    call VQsort(Difference, DiffPtr, NumPair)
C compute the minimum, maximum, quartiles,
C and median differences
if (MOD(NumPair, 2) .EQ. 0) then
  K = NumPair / 2
  Vmedian(I) = (Difference(K) + Difference(K+1))/2.0
else
  VMedian(I) = Difference((NumPair+1)/2)
endif
VMin(I) = Difference(1)
arg=jnint(numpair*0.25)
arg=max(1,arg)
VQ1(I) = Difference(arg)
arg=jnint(numpair*0.75)
arg=max(1,arg)
VQ3(I) = Difference(arg)
VMax(I) = Difference(NumPair)
else
write(*,*)'no pairs at distance(ptr):',
     i,distance(ptr)
  VMin(I) = Difference(1)
  VQ1(I) = Difference(1)
  VMedian(I) = Difference(1)
  VQ3(I) = Difference(1)
  VMax(I) = Difference(1)
endif
999 continue
return
end
SUBROUTINE Calculate_default_Variograms(gamma, x,two,imon)

C gamma is input variable to model and intermediate
C variable. Using this subroutine may destroy original
C estimate and avgdist variable fitting 3 monotonic
C and 1 non monotonic finite models to exp variogram
C with independently fitted nugget and sill
C Written by M.M.Moody, Sept, 1992

INTEGER one, NTEMP, two, IERR, six, five,
+       ten,i,j,iker,
+       globalifk,k,minifk
REAL*4 X(52),
+       zero, sqdiff(401), minsqdiff, range0,
+       minnug, gamma(52), globalminsqdiff, pi, arg,
+       globallambda, delta, globalrange, minrange
INCLUDE 'vario.inc'
pi=4.0*atan(1.0)
one=0
five=5
six=6
ten=0

C complete model comparison, use best:
C first calculate default parameters
C try finite model fit set variables
DO 2 i = 1,nlag
  estimate(i) = gamma(i)
  avgdist(i) = x(i)
2 continue

C set nugget
  nug0 = 2.0*estimate(1)-estimate(2)
  IF(nug0.lt.0.0)nug0=0.0
  IF(imon.eq.1)then
    IF(nug0.gt.estimate(1))nug0=(estimate(1)
+       + estimate(2))/2.0
  endif

C set sill
  ntemp = nlagfit/3
  IF (NTEMP.LT.2) NTEMP = 2
  SIL0 = estimate(nlagfit)
  DO 5 i = 1,ntemp-1
SIL0=SIL0+ESTIMATE(NLAGfit-I)
5 continue
sil0=sil0/float(ntemp)
C  EPS = 0.0001*sil0
if(nug0.ge.sil0)then
  nug0=sil0
  write(*,*)'pure nugget variogram'
  ifk=1
  Range0=x(1)
  lambda=1
  go to 90
endif
delta=2.0*LagMaxDist/400.
globalminsqdiff=1.e31
C  try each of 3 diff finite model types in turn
if(imon.eq.1)then
  nvarfit=3
else
  nvarfit=3
endif
do 82 k=1,nvarfit
if(imon.eq.1)then
  ifk=k+1
  kend=400
else
  ifk=k+6
  kend=100
endif
minsqdiff=1.e31
do 80 j = 1,kend
if(ifk.le.6)then
  range0=toatl(j)*delta
else
  range0=toatl(j)*delta
endif
sqdiff(j)=0.0
do 70 i=1,nlagfit
arg = avgdist(i)/range0
if(ifk.eq.2)then
  if(avgdist(i).gt.range0)then
    gamma(i)=sil0
  else
    gamma(i)=0.0
  endif
else
  if(avgdist(i).gt.range0)then
    gamma(i)=sil0
  else
    gamma(i)=0.0
  endif
endif
\[
\text{gamma}(i) = \text{nug0} + (\text{sil0} - \text{nug0}) \times (1.5 \times \text{arg} - 0.5 \times \text{arg} \times 3)
\]

\text{endif}

\text{elseif}(\text{ifk} = 3)\text{then}
\text{arg} = \text{arg} \times \text{arg}
\text{gamma}(i) = \text{nug0} + (\text{sil0} - \text{nug0}) \times (1.0 - \exp(-3.0 \times \text{arg}))
\text{elseif}(\text{ifk} = 7)\text{then}
\text{gamma}(i) = \text{nug0} + (\text{sil0} - \text{nug0}) \times (\sin(\pi \times \text{arg}) / (\pi \times \text{arg}))
\text{elseif}(\text{ifk} = 9)\text{then}
\text{gamma}(i) = \text{nug0} + (\text{sil0} - \text{nug0}) \times (1.0 + (\pi \times \text{arg}) \times \sin(\pi \times \text{arg}))
\]

\text{endif}

\text{sqdiff}(j) = \text{sqdiff}(j) + (\text{estimate}(i) - \text{gamma}(i)) \times 2
\]

\text{70 continue}
\text{sqdiff}(j) = \text{sqdiff}(j) / \text{float}(\text{nlagfit})
\text{if}(\text{sqdiff}(j) < \text{minsqdiff}) \text{then}
\text{minsqdiff} = \text{sqdiff}(j)
\text{Lambda} = \text{float}(j)
\text{minrange} = \text{range0}
\text{minifk} = \text{ifk}
\text{endif}
\text{80 continue}
\text{C have completed each model calculation}
\text{if}(\text{minsqdiff} < \text{globalminsqdiff}) \text{then}
\text{globalminsqdiff} = \text{minsqdiff}
\text{globallambda} = \text{Lambda}
\text{globalifk} = \text{minifk}
\text{globalrange} = \text{minrange}
\text{endif}
\text{82 continue}
\text{C have completed all model calculations}
\text{C range1}(1) = \text{Range0}
\text{C reset parameters}
\text{ifk} = \text{globalifk}
\text{range0} = \text{globalrange}
\text{if}(\text{range0} \leq 0.0) \text{then}
\text{lambda} = 1.e31
else
    lambda = globallambda
endif

reset nugget to nug0
90 nugget = nug0
GamMax = sII0
sill(1) = sII0-nug0
write(12, *) 'using finite model', ifk
write(12, *) ' LAMBDA =', Lambda
write(12, *) ' Nugget =', Nug0
write(12, *) ' SII0 =', SII0
write(12, *) ' range0 =', range0
write(12, *) ' minsqdiff =', minsqdiff
write(*, *) ' using finite model', ifk
write(*, *) ' LAMBDA =', Lambda
write(*, *) ' Nugget =', Nug0
write(*, *) ' SII0(1) =', SII0(1)
write(*, *) ' range0 =', range0
write(*, *) ' minsqdiff =', minsqdiff
85 minsqdiff = 0.0

C calculate final finite model
write(12, *) ' ifk; model, est, minsqdiff, arg, lamb:', ifk
DO 100 1 = 1, NLAG
    arg = avgdist(i)/range0
    if(ifk.eq.1) then
        model(i) = sII0
    elseif(ifk.eq.2) then
        if(avgdist(i).gt.range0) then
            model(i) = sII0
        else
            model(i) = nug0 + (sII0-nug0)*(1.5*arg-0.5*arg**3)
        endif
    elseif(ifk.eq.3) then
        model(i) = nug0 + (sII0-nug0)*(1.0-exp(-3.0*arg))
    elseif(ifk.eq.4) then
        model(i) = nug0 + (SII0-NUG0)*(1.0-exp(-3.0*arg))
    elseif(ifk.eq.5) then
        model(i) = nug0 + (sII0-nug0)*(7.0*arg**2 +
                                   -8.75*arg**3 + 3.5*arg**4 - 0.75*arg**7)
    elseif(ifk.eq.6) then
        model(i) = nug0 + (sII0-nug0)*(2.0*arg-arg**2)
100 continue

elseif(ifk.eq.7)then
    model(i)=nugO+(silO-nugO)*(1.0-sin(pi*arg)/(arg*pi))
elseif(ifk.eq.8)then
    model(i)=nugO+(silO-nugO)*(1.0+sin(pi*arg))/2.0
elseif(ifk.eq.9)then
    model(i)=nugO+(silO-nugO)*(1.0+(pi*arg)*sin(pi*arg))/2.0
endif
Imodel(1)=ifk-1
minsqdiff=minsqdiff+(estimate(i)-model(i))**2
write(12,*)i,model(i),estimate(i),minsqdiff, arg,lmbda(i)
100 continue
minsqdiff=minsqdiff/float(nlag)
write(*,*)'final error for nlag pts:',minsqdiff
write(12,*)'GamMax,final error for nlag pts:', GamMax,minsqdiff
end model fitting
C set variogram anisotropy to isotropy
write(*,*)'enter any number to continue'
read(*,*)ntemp
return
end

SUBROUTINE Calculate_Nugget
C calculate nugget and sill from pair data directly
C choose min nugget and maximum sill from series of calculations. Written by M.M.Moody, Sept, 1992

integer i,j,nugpts,ptr,k,index,ntemp
REAL*4 delta1,delta2,minnug0,maxsil0,nug1,nug2,z1,z2
include 'vario.inc'
C cycle through set number of pts
nugpts=npair/20
if(nugpts.lt.20)then
    write(*,*)'min no of max no pts = 20,'
    write(*,*)'set index to 20'
    nugpts=20
endif
minnug0=1.e31
C cycle through several numbers of points
do 50 j=15,nugpts
nug1=0.0
delta1=0.0
nmiss=0
C set first nugget point
  do 10 ptr=1,j
    Z1 = data(from(ptr),kdVar)
    Z2 = data(to(ptr),kdVar)
    if(z1.lt.Missing.and.z2.lt.Missing)then
      nug1=nug1+(z1-z2)**2
      delta1=delta1+distance(ptr)
    else
      write(*,*)'Missing value: from or to:',
      + ptr,from(ptr),to(ptr)
      nmiss=nmiss+1
      GOTO 10
    endif
  10 continue
  nugl =0.5*nugl/float(nugpts-nmiss)
  deltal = delta1/float(nugpts-nmiss)
C set second nugget point
  nuggest=0.0
deltal2=0.0
  nmiss=0
  do 20 ptr=j+1, 2*j
    Z1 = data(from(ptr),kdVar)
    Z2 = data(to(ptr),kdVar)
    if(z1.lt.Missing.and.z2.lt.Missing)then
      nug2=nug2+(z1-z2)**2
      delta2=delta2+distance(ptr)
    else
      write(*,*)'Missing value: from or to:',
      + ptr,from(ptr),to(ptr)
      nmiss=nmiss+1
      GO TO 20
    endif
  20 continue
  nug2=0.5*nug2/float(2*nugpts-nmiss)
  deltast2= delta2/float(2*nugpts-nmiss)
C calculate nugget
  nug0=nug1-deltal *(nug2-nugl )/( delta2-deltal)
C save minimum nugget over range
if(nugO.lt.minnugO)then
  minnugO=nugO
  index=j
endif
50 continue
nugO=minnugO
if(nugO.lt.0.0)nugO=0.0
write(*,*)'nugpts,index,nugO: ',nugpts,index,nugO
write(12,*)'nugpts,index,nugO: ',nugpts,index,nugO
C set sill
maxsilO=-1.e31
j=npair/10
IF (j.LT.15)j=15
k=npair/4
IF (k.LT.20)then
  k=20
  j=10
endif
do 100 ntemp=j,k
silO=0.0
nmiss=0
do 60 i=1,ntemp
  ptr=npair-i+1
  Z1 = data(from(ptr),kdVar)
  Z2 = data(to(ptr),kdVar)
  if(z1.lt.Missing.and.z2.lt.Missing)then
    silO=silO+(z1-z2)**2
  else
    write(*,*)'Missing value: from or to: ',
    + ptr,from(ptr),to(ptr)
    nmiss=0
  GO TO 60
endif
60 continue
silO=0.5*silO/float(ntemp-nmiss)
C save maximum sill over range
if(silO.gt.maxsilO)then
  maxsilO=silO
  index=ntemp
endif
100 continue
sil0=maxsil0
if(nug0.ge.sil0)then
  nug0=sil0
  write(*,*)'pure nugget variogram'
endif
write(*,*)'j,k,index,sil0:',j,k,index,sil0
write(12,*)'j,k,index,sil0:',j,k,index,sil0
return
end
B.3 Kriging Routines

C geostat.inc - global variables for combined programs
C  prevar and vario of GEOEAS with GKS graphics.
C  Description of Global variables and program
C  limits. GEOEAS routines modified by H.R.Rhodes
C  August, 1990 and M.M.Moody Sept, 1992

C MAXDATA - maximum samples allowed
C MAXVAR - maximum variables allowed
C MAXPAIR - maximum pairs allowed
C MAXPAIRLAG - maximum pairs in a lag
C MAXCUTOFF - maximum number of lag cutoffs
C MAXLAG - maximum number of lag intervals
C MAXSTRUC - max. no. of nested variogram structures
C GAM - estimator type (used with GammaType)
C MAXEQ - Maximum number of equations in system
C MAXNEIGHBOR - Max. no. of neighbors in kriging system
C MAXESTIMATE - maximum number of estimates in output
C MAXPTS - maximum number of points for kriged grid
C UN10 - UN16 - input output unit numbers

INTEGER MAXVAR, MAXDATA, MAXPAIR, MAXPAIRLAG
INTEGER MAXCUTOFF, MAXLAG, MAXSTRUC, MAXEQ
INTEGER MAXNEIGHBOR, MAXESTIMATE, MAXPTS
INTEGER GAMABS1, GAMABS2, GAMREL, GAMERGO
INTEGER UN10, UN11, UN12, UN13, UN14, UN15
INTEGER UN16, UN17

PARAMETER (MAXVAR = 6, MAXDATA= 250)
PARAMETER (MAXPAIR = 20000, MAXPAIRLAG = 10000)
PARAMETER (MAXCUTOFF = 45, MAXLAG = 44)
PARAMETER (MAXSTRUC = 4, MAXNEIGHBOR = 80)
PARAMETER (MAXEQ = MAXNEIGHBOR+6)
PARAMETER (MAXESTIMATE = 1000, MAXPTS = 1000)
PARAMETER (GAMABS2 = 0, GAMABS1 = 2)
PARAMETER (GAMREL = 1, GAMERGO = 3)
PARAMETER ( UN10 = 10, UN11 = 11, UN12 = 12)
PARAMETER ( UN13 = 13, UN14 = 14, UN15 = 15)
PARAMETER ( UN16 = 16, UN17 = 17)

C VarNam - array of variable names
C Units - array of variable units
C Data_File - data file
C Pair_File - pair comparison file
C Nbhd_File - opt.parameters for neighborhoods file
C Sort_File - file of sorted arrays
C OutPut_File - Output file of run parameters and data
C Results_File - Output file of estimates for plotting
C Title - array of titles for variogram plot
C Header - Title for output file and graphs

CHARACTER VarNam(MAXVAR)*10, Units(MAXVAR)*10,
+ Data_File*14, Pair_File*14, OutPut_File*14,
+ Results_File*14, Title(4)*60, Header*80,
+ Nbhd_File*14, Sort_File*14

COMMON /CHARS/ VarNam, Units, Data_File, Pair_File,
+ Nbhd_File, Sort_File, Output_File, Results_File,
+ Title, Header

C Ndata - the number of samples (rows in matrix)
C Nvar - the total number of variables (columns)
C Xvar - column number for x coordinate
C Yvar - column number for y coordinate
C Evar - column number for e coordinate
C kVar - column no. for current var. from KrigVar
C kdVar - kvar-datadim: current data matrix col. no.
C Npair - the number of pairs in pair file
C TotalPairs - the total number of pairs retained
C Nlag - the number of lag cutoffs
C Nlagfit - the number of lags used in model fitting
C GammaType - code for the type of estimator
C Nkept - the no. of values used in variogram calc.
C NMiss - the number of missing values in data
C NumNeg - the number of sample values <=0.0
C Current_Lag - current lag number
C Mtype - array of variogram type codes for modeling
C Pairs - a matrix for the no. of lag pairs by var.
C From - ptr to data for 1st sample in pair
C To - ptr to data for second sample in pair
C DiffPtr - array of pointers to squared differences
C Flpairs - floating value of real pairs
C PairPtr - holds sorted order index (by distance)
C Ppair - the number of possible pairs
C Dimension - the dimension of the problem space
C VarDimen - the dimension of the variogram space
C DataDim - The dimension of the data set coordinates
C Layers - the number of layers in a pseudo 2D space
C Language - Language code 1 = english.
C NumEstimate - the number of estimates produced
C Neq - the number of equations to be solved
C Nrecord - the number of Ndata records kept
C KrigVar - array of dependent variables to be kriged
C NkVar - the number of variables to be kriged
C Mvar - index for the variable being kriged
C numptsx - the number of grid pts in x direction
C numptsy - the number of grid pts in y direction
C numptse - the number of grid pts in e direction
C Same - the no. of interp. pt.s equal to data pt.s
C Npoints - the number of intepolation points
C Num_Errors - the number of errors in system solution
C Num_NoFound - the no. of no neighbors found in solution
C Num_Neg - the no. of negative varriances calculated
C chckpt - the data pt. for printout check, 0 = none

C Variogram parameters:
C Imodel - the model type for each struc. (up to 4)
C NZeroDis - number of zero distances in pcf
C Nstruc - the no. of structures in variogram model
C Nsills - the number of sills in finite models

C Neighborhood Search Parameters:
C YValPtr - Pointer list to data (parallel to Sptr)
C Zptr - Pointer ro data set entry
C Nbhdpts - Array of # of pts in each nbrhd.
C Min_Nbhd_points - min number of points per nbhd to solve
C Evalptr - Pointer list to data (parallel to Sptr)

C Option codes:
C Ikrig - code for kriging system
C Layers - code for number of layers in 2-D system
C Ilocal - code for local or global neighborhood
C Ioption - code for calculation options
C Iso - code for isotropic(1) or anisotropic(2)
C xval - gives no. of xvalid point, else 0
C inbhd - neighborhood code
C icalc - code smoothing parameter calculation
C iresid - code for using residuals or data
C ivgm - variogram code
C ifk - smoothing kernel function array
C iker - current smoothing kernel function
C ikr - estimating kernel function

INTEGER Ndata, Nvar, Xvar, Yvar, Evar, kVar
INTEGER kdVar, Mvar, Npair, TotalPairs, Nlag
INTEGER Nlagfit, GammaType, icalc, Nkept, NMiss
INTEGER NumNeg, Current_Lag, Language, ivgm
INTEGER Ppair, PairPtr(MAXPAIR), Dimension
INTEGER VarDimen, DataDim, Mtype(MAXSTRUC)
INTEGER Pairs(MAXLAG,MAXVAR), inbhd, iresid
INTEGER From(MAXPAIR), To(MAXPAIR), NModel
INTEGER DiffPtr(MAXPAIRLAG), Imodel(MAXSTRUC)
INTEGER Flpairs(MAXLAG), Nstruc, NZeroDist, iso
INTEGER Nbhdpts(MAXESTIMATE), NumEstimate, Neq
INTEGER Nsector, Maxp, Minp, Maxps, Maxempty
INTEGER Nsills, Ikrig, Layers, Ilocal, Ioption
INTEGER Min_Dhpts, Nrecord, KrigVar(MAXVAR)
INTEGER NkVar, xval, Numptsx, Numptsy, Numptse
INTEGER Same, Npoints, chkpt, Num_Errors, ifk
INTEGER Num_NoFound, YValPtr(MAXDATA), Num_Neg
INTEGER Zptr(MAXDATA), Evalptr(MAXDATA)
INTEGER kfun(MAXVAR)

COMMON /INTS/ Ndata, Nvar, Xvar, Yvar, Evar, kdVar,
+ Npair, TotalPairs, Nlag, Nlagfit, kvar, Mvar,
+ Nkept, Nmiss, NumNeg, Current_Lag, Neq, NModel,
+ Pairs, From, To, DiffPtr, Flpairs, DataDim,
+ Ppair, PairPtr, Dimension, Layers, Language,
+ Nstruc, NZeroDist, Imodel, Nbhdpts, NumEstimate,
+ Nsector, Maxp, Minp, Maxps, Maxempty, Ikrig,
+ Ilocal, Ioption, Min_Nbhd_Points, Iso, xval,
+ VarDimen, Nrecord, KrigVar, NkVar, Numptsx,
+ Numptsy, Numptse, Mtype, Same, Npoints, Nsills,
+ Num_Errors, Num_NoFound, Zptr, YValptr, Evalptr,
+ inbhd, icalc, ivgm, ifk, chkpt, iresid,
+ GammaType, Num_Neg, ifk
C Xdata - the array of X values
C Ydata - the array of Y values
C Edata - the array of E values
C Data - the matrix of Data Values
C Resid - the matrix of residuals or data
C Smooth - the matrix of smoothed data
C MaxDist - the max. interpair distance used
C MinDist - the min. interpair distance used
C VarMin - minimum data value
C VarMax - maximum data value to use
C XCMin - minumum x coordinate value
C XCMax - maximum x coordinate value
C YCMin - minimum y coordinate value
C YCMax - maximum coordinate value
C ECMin - minimum e coordinate value
C ECMax - maximum e coordinate value
C Distance - distance for each pair
C Vector - coordinates of vectors by pair
C Azimuth - horizontal direction parameter
C Dip - vertical ...
C Tolerance - tolerance for calculation
C BandWidth - maximum bandwidth parameter
C Missing - missing value
C LagMinDist - first value for lag cutoffs
C LagMaxDist - last value for lag cutoffs
C LagIncDist - lag Increment Value
C LagCutoffs - lag cutoffs values
C Tcutoff - individual lagcutoff value
C Fraction - value used [in prevar] to subset pairs
C Estimate - array of estimated values
C StdDev - array of standard deviations
C Xdist - length over x axis for estimate
C Ydist - length over y axis for estimate
C Edist - length over e axis for estimate
C Delx - distance between estimated points
C Dely - distance between estimated points
C Dele - distance between estimated points
C DataRec - temporary data array for input

C Experimental Variograms:
C AvgDist - mean distance per lag
C LagData - values used in differences
C (Estimate - also the values retained for models )
C Difference - array of differences squared in xvalid
C Zscore - (Difference/Variance) in xvalid
C from, Mto - mean of from pairs, to pairs
C Vmin, Vmax - min max for Variogram values
C Emin, Emax - min max for Estimated values
C Smin, Smax - min max for StdDev values
C Dmin, Dmax - min max for Differences
C Zmin, Zmax - min max for Zscores
C VQ1, VQ3 - 1st and 3rd Quartiles of Variograms
C EQ1, EQ3 - 1st and 3rd Quartiles of Variograms
C SQ1, SQ3 - 1st and 3rd Quartiles of Variograms
C DQ1, DQ3 - 1st and 3rd Quartiles of Variograms
C ZQ1, ZQ3 - 1st and 3rd Quartiles of Variograms
C VMean, VMedian- increment mean, median of Variograms
C EMean, EMedian- increment mean, median of Estimates
C SMean, SMedian- increment mean, median of StdDevs
C DMean, DMedian- increment mean, median of Differences
C ZMean, ZMedian- increment mean, median of Zscores
C Estd - Standard Deviation of estimates
C Sstd - Standard Deviation of standard dev.
C Dstd - Standard Deviation of Differences
C Vstd - Standard Deviation of Data
C Zstd - Standard Deviation of zscores
C Mean - mean of data values
C dVariance - Variance of data
C xVariance - error of estimate in xvalid
C LagMean - mean of all values in the lag
C LagVariance - variance of all values in the lag
C VarianceFrom - variance of FROM pairs
C VarianceTo - variance of TO pairs
C GamPower1 - madogram values
C GamPower2 - variogram values
C GamRelative - relative variogram values
C NonErgodic - nonErgodic variogram values
C Scale - scale of kriging nbhd to vgm nbhd

C the variogram and model:
C Sill - array of variogram sills for models
C Range1 - array of variogram ranges for models
C Range2 - array of variogram ranges for models
C Range3 - array of variogram ranges for models
C Varazimuth - Array of variogram ellipsoid azimuths
C Vardip - Array of variogram ellipsoid dips
C Nugget - variogram nugget effect
C Mglobal - Global mean for simple kriging
C Model - the finite model values for AvgLagDist
C GamMax - maximum value of variogram model
C Lambda - inverse of pseudo range for some models
C SParam - array of smoothing parameters

C Neighborhood Search Variables and Estimation Variables
C 3-D parameters follow:
C Sradius1 - array of radii for search ellipsoids
C Sradius2 - array of radii for search ellipsoids
C Sradius3 - array of radii for search ellipsoids
C Sazimuth - array of angles in x-y plane of axes
C Sdip - array of dip angles from x-y plane of axes
C B - vector for solving kriging system
C B2 - vector for solving kriging system
C W - array for solving kriging system
C Zptr - pointer array to data set entry
C Vk - working value of estimate
C Sk - working value of stddev
C xorigin - kriging grid x origin
C yorigin - kriging grid y origin
C eorigin - kriging grid e origin
C EVValues - List of ascending E coordinate values

C 2-D parameters follow:
C Yvalues - Array of values in Y ascending order
C SrMajor - a axis of ellipse
C SrMinor - b axis of ellipse
C SrAngle - ellipse angle
C aaxis - direction cosines of a axis
C baxis - direction cosines of b axis
C caxis - direction cosines of c axis
C a0,b0,c0 - axes lengths
C theta - angle
C alpha - angle

REAL*4 XCmin, XCmax, YCmin, YCmax, ECmin, ECmax,
+ theta, alpha, eps, asq, bsq, csq, xi, yi, zi, xp, yp, zp, rsq,
+ MinDist, MaxDist, Missing, VarMin, VarMax, Mean,
Azimuth, Dip, Tolerance, BandWidth, PA(3),
LagMinDist, LagMaxDist, LagIncDist, Fraction,
Xdata(MAXDATA), Ydata(MAXDATA), Edata(MAXDATA),
Data(MAXDATA,MAXVAR), Distance(MAXPAIR),
Difference(MAXPAIRLAG), LagCutoff(MAXCUTOFF),
Vector(3,MAXPAIR), AvgDist(MAXLAG), Zscore(MAXDATA),
Estimate(MAXESTIMATE), Zscore(MAXDATA),
StdDev(MAXDATA), Mfrom(MAXLAG), Mto(MAXLAG),
Vmin(MAXLAG), Vmax(MAXLAG), Evaluate(MAXESTIMATE),
VMean(MAXLAG), VMedian(MAXLAG), DataRec(MAXVAR),
VQ1(MAXLAG), VQ3(MAXLAG), Resid(MAXDATA,MAXVAR),
LagMean(MAXLAG), LagVariance(MAXLAG), Scale,
VarianceFrom(MAXLAG), VarianceTo(MAXLAG),
GamPower1(MAXLAG), GamPower2(MAXLAG), dVariance,
GamRelative(MAXLAG), NonErgodic(MAXLAG), Mglobal,
Model(MAXLAG), Sill(MAXSTRUC), Nugget, xVariance,
Range1(MAXSTRUC), Range2(MAXSTRUC), Delx, Delx,
Range3(MAXSTRUC), Varazimuth(MAXSTRUC), Dele,
Vardip(MAXSTRUC), Sradius1(MAXESTIMATE), Lambda,
Sradius2(MAXESTIMATE), Sradius3(MAXESTIMATE),
Sazimuth(MAXESTIMATE), Sdip(MAXESTIMATE), Zmin,
Xdist, Ydist, Edist, Smooth(MAXDATA,MAXVAR),
Tcutoff, EQ1, EQ3, SQ1, SQ3, DQ1, DQ3, ZQ1,
ZQ3, Emin, Emax, Smin, Smax, Dmin, Dmax, Zmax,
Emean, Smean, Dmean, Zmean, Estd, Sstd, Dstd,
Vstd, Zstd, Emedian, Smedian, Dmedian, Zmedian,
xorigin, yorigin, eorigin, Yvalues(MAXDATA),
SrMajor, SrMinor, SrAngle, SParam(MAXVAR),
aaxis(3), baxis(3), caxis(3), a0, b0, c0

double precision W(maxeq,maxeq), B(maxeq),
B2(maxeq), Vk, Sk, GamMax

COMMON /FLOATS/ XCmin, XCmax, YCmin, YCmax, ECmin,
ECmax, MinDist, MaxDist, Missing, Fraction,
xVariance, VarMin, VarMax, Mean, dVariance,
Azimuth, Dip, Tolerance, BandWidth, PA, DataRec,
LagMinDist, LagMaxDist, LagIncDist, Zscore,
Xdata, Ydata, Edata, Data, Distance, Vector,
Difference, LagCutoff, AvgDist, Estimate,
Mfrom, Mto, Vmin, Vmax, VMean, VMedian,
VQ1, VQ3, LagMean, LagVariance, Smooth,
C Calculates pair comparison file, standard experimental variograms, default model calculations, validation, simple and ordinary kriging. May smooth data with kernel smoother and analyse smoothed data of residuals.
C Taken from GEOEAS adapted and extended by M.M.Moody from 1991 to 1993

PROGRAM Geostatistics
character*14 Grid_File, Chckpt_File
character*14 Xvalid_File
integer i, j, k, l, Zeoropairs, plotmodel, nomodel,
+ three, gflag, modify, transform_specified,
+ Plot_id, toggle, two, numpts, iest, ierr, kk,
+ minptrs, maxptrs, npts, flag, dummy, ibasis,
+ dptr, igrd, unt
real Dh, Hx, Hy, He, Tx1, Tx2, Ty1, Ty2, Te1,
+ Te2, Direction(2), Hxp, Hyp, Hep, varlm,
+ Dx, Dy, De, x0, y0, e0, arg, vsq, gamma(24)
include 'geostat.inc'
Data_File = 'Vario.dat'
Pair_File = 'Vario.pcf'
Plot_File = 'Vario.plt'
Output_File = 'Vario.out'
Chckpt_File = 'Chckpt.out'
Xvalid_File = 'Xvalid.out'
Results_File = 'Estimate.out'
Nbhd_File = 'Vario.nhd'
Grid_File = 'Grid.nhd'
Language = 1
two=2
three=3
open (unit=un10, File=Data_File, status=
+ 'unknown, errr=9999)
open (unit=un12, File=Output_File, status=
+ 'unknown', err=9999)
open (unit=un16, File=Chckpt_File, status=
+ 'unknown', err=9999)
open (unit=un17, File=Xvalid_File, status=
+ 'unknown', err=9999)
C prevar sequence follows:
C initialize prevar subroutine follows:
DO 5 I = 1, MaxPair
  PairPtr(I) = I
5 continue
DO 10 I = 1, MaxVar
  VarNam(I) = ',
10 continue
Missing = 1.E31
Npair = 0
Ppair = 0
NumNeg = 0
Ndata = 0
Nrecord = 0
NMiss = 0
C the following needs to be in control file or
C introductory menu in reading data.
Xvar = 1
Yvar = 2
Evar = 3
XCmin = +1.E30
XCmax = -1.E30
YCmin = +1.E30
YCmax = -1.E30
ECmin = +1.E30
ECmax = -1.E30
MinDist = +1.E30
MaxDist = -1.E30
C end initialize subroutine
C specify unit of data file
unt=un10
call read_data(unt)
close(unt)

C establish codes for processing:
write(*,*)'enter variogram calculation code:'
write(*,*)' 1 use input data,'
write(*,*)' 2 use kernel smoothed data'
write(*,*)' 3 use kernel residuals'
read(*,*)Iresid
write(*,*)'Input Variogram Nbrhood Definition:'
write(*,*)' 1 = Global Variogram Nbhd,'
write(*,*)' 2 = Opt Nbhd,'
write(*,*)' 3 = Limited Var,'
write(*,*)' 4 = Fixed N Nbhd,'
write(*,*)' 5 = Stand Nbhd'
read(*,*)Ivgm
if(Ivgm.eq.5)then
  write(*,*)'enter relative nbhd parameters -
  write(*,*)' will be renormalized'
  write(*,*)'a,b,c'
  write(*,*)'aaxis(i),i= 1,3'
  write(*,*)'baxis(i),i= 1,3'
  write(*,*)'caxis(i),i= 1,3'
  write(*,*)'use zeros if dimension is less than 3'
read(*,*)a0,b0,c0
read(*,*)a(j)=1,3)
read(*,*)b(j)=1,3)
read(*,*)c(j)=1,3)
arg=sqrt(aaxis(1)**2+aaxis(2)**2+aaxis(3)**2)
aaxis(1)=aaxis(1)/arg
aaxis(2)=aaxis(2)/arg
aaxis(3)=aaxis(3)/arg
if(data_dim.gt.1)then
  arg=sqrt(baxis(1)**2+baxis(2)**2+baxis(3)**2)
baxis(1)=baxis(1)/arg
baxis(2)=baxis(2)/arg
baxis(3)=baxis(3)/arg
endif
if(data_dim.gt.2)then
  arg=sqrt(caxis(1)**2+caxis(2)**2+caxis(3)**2)
caxis(1)=caxis(1)/arg
caxis(2)=caxis(2)/arg
caxis(3) = caxis(3)/arg
endif
write(*,*)'renormalized dir cos'
write(*,*) (aaxis(i), i=1,3)
write(*,*) (baxis(i), i=1,3)
write(*,*) (caxis(i), i=1,3)
write(un12,*)'renormalized dir cos'
write(un12,*) a0, b0, c0
write(un12,*) (aaxis(i), i=1,3)
write(un12,*) (baxis(i), i=1,3)
write(un12,*) (caxis(i), i=1,3)
endif
if(lvgm.eq.2.or.lvgm.eq.3.or.lvgm.eq.4)then
  open (unit=un15, File=nbhd_file, status='unknown', err=9999)
  if(lvgm.eq.2.or.lvgm.eq.4)then
    read(un15,*,err=999,end=9999)
    dummy, inbhd, iso, iest, ierr, minptrs, maxptrs, npnts
  elseif (lvgm.eq.3)then
    read(un15,*,err=999,end=9999)
    dummy, inbhd, iso, iest, ierr, minptrs, maxptrs, varlm
  endif
endif
C calc_possible_pairs subroutine follows:
if(lvgm.eq.1)then
  Ppair = nrecord * (nrecord-1) / 2
  write(*,*) 'Possible number of pairs:', Ppair
  write(*,*) 'Maximum number of pairs:', Maxpair
C end calc_possible_pairs
if(Ppair.gt.MaxPair)then
  write(*,*)'Poss. No. of Pairs Exceeds Capacity'
  write(*,*)'Partition Data Set or Use Nbrhoods'
go to 99999
endif
C Compute_pairs subroutine follows
ZeroPairs = 0
Npair = 0
C read first nbhd and throw away
if(lvgm.eq.2.or.lvgm.eq.3.or.lvgm.eq.4)then
  read(un15,*,err=999,end=9999) dummy, flag
  read(un15,*,err=999,end=9999) dummy, v, vsq
read(un15,*,err=999,end=9999)a0,b0,c0
read(un15,*,err=999,end=9999)(aaxis(j),j=1,3)
if(flag.eq.0)then
  read(un15,*,err=999,end=9999)(baxis(j),j=1,3)
  read(un15,*,err=999,end=9999)(caxis(j),j=1,3)
endif
endif

DO 120 i = 2,Nrecord
  Tx1 = Xdata(i)
  Ty1 = Ydata(i)
  Te1 = Edata(i)
  
  C if(Ivgm.eq.2.or.Ivgm.eq.3.or.Ivgm.eq.4)then
  read(un15,*,err=999,end=9999)dummy,flag
  if(dummy.ne.i)then
    write(*,*)'nbhd not matched to pt.',dummy,i
  endif
  read(un15,*,err=999,end=9999)dummy,v,vsq
  read(un15,*,err=999,end=9999)(a0,b0,c0)
  read(un15,*,err=999,end=9999)(aaxis(j),j=1,3)
  if(flag.eq.0)then
    read(un15,*,err=999,end=9999)(baxis(j),j=1,3)
    read(un15,*,err=999,end=9999)(caxis(j),j=1,3)
  endif
  else
    flag=0
  endif

  DO 110 j = 1, i-1
    Tx2 = Xdata(j)
    Ty2 = Ydata(j)
    Te2 = Edata(j)
    
    C calculate distances
    Hx = Tx2 - Tx1
    Hy = Ty2 - Ty1
    He = Te2 - Te1
    
    C zero distance pairs case
    if (Hx .eq. 0. and. Hy .eq. 0.) then
      if (MinDist .eq. 0.0)
        ZeroPairs = ZeroPairs + 1
      Dh = 0.0000001
    else
        C further processing here
\[ Dh = \sqrt{Hx^2 + Hy^2 + He^2} \]

C check if distance is within distance limits
if (Dh > MaxDist .OR. Dh < MinDist) then
    write(*,*)'zero distance'
go to 110
endif
C exclude if not in nbhd except global
C convert coordinates and calc ellipse
if(lvgm.ne.1) then
    Hxp=Hx*aaxis(1)+Hy*aaxis(2)+He*aaxis(3)
    Hyp=Hx*baxis(1)+Hy*baxis(2)+He*baxis(3)
    Hep=Hx*caxis(1)+Hy*caxis(2)+He*caxis(3)
elseif(datadim.eq.1) then
    arg=(Hxp/a0)*(Hxp/a0)
elseif(datadim.eq.2) then
    arg=(Hxp/a0)*(Hxp/a0)
    if(flag.eq.0) then
        arg=arg+(Hyp/b0)*(Hyp/b0)
    endif
elseif(datadim.eq.3) then
    arg=(Hxp/a0)*(Hxp/a0)+(Hyp/b0)*(Hyp/b0)
    if(flag.eq.0) then
        arg=arg+(Hep/c0)*(Hep/c0)
    endif
endif
if(arg.gt.1.001) go to 110
C may be better to find machine accuracy
endif
C store pair
if (Npair .LE. MaxPair) then
    Npair = Npair + 1
    From(Npair) = i
    To(Npair) = j
    Distance(Npair) = Dh
    Vector(1, Npair) = Hx
    Vector(2, Npair) = Hy
    Vector(3, Npair) = He
    PairPtr(Npair) = Npair
else
    write(*,*)'Number of pairs exceeds max.'
go to 99999
endif
110 continue
120 continue
Nkept=Nrecord
C sort according to increasing distances
call VQsort(Distance, PairPtr, Npair)
if (ZeroPairs .gt. 0) then
    write(*,*) 'Zero Distance Pairs encountered'
endif
C check for no pairs
if (Npair .eq. 0) then
    write(*,*) 'No pairs were computed'
go to 99999
endif
close(15)
C end compute_pairs
C Write_Pairs subroutine follows
C open data file and pair comparison file
open (unit =un11, File =Pair_File, status=
    + 'unknown', err =9999)
only (unit =un10, File =Data_File, status=
    + 'unknown', err =9999)
C
C read header string
read(un10, '(A)', err = 9999, end = 999)
    + Header
C read number of variables in file
read(un10, *, err = 9999, end = 999) Nvar,datadim
C read variable names and units for each variable in file
DO 200 I = 1, Nvar
    read(un10, '(2A10)', err = 999, end = 999)
        + VarNam(I), Units(I)
200 continue
C write file signature, header information
write (un11,*, err = 999) 'PREVAR'
write (un11,990, err = 999) Header
write (un11,*,err = 999) Xvar, Yvar, Evar, XCmin, YCmin, YCmax, ECmin, ECmax, MinDist, MaxDist
C write number of variables, data, pts kept, pairs
write(un11,*,err = 999)
    + Nvar, DataDim, Ndata, Nkept, Npair
C write variable names and units to pair file
write(un11,930,err = 999)  
+  (VarNam(I),Units(I), I = 1, Nvar)  
C write data matrix to pair file  
DO 210 I = 1, Ndata  
   read (un10, *) (DataRec(J), J = 1, Nvar)  
   write (un11,*,err = 999)  
+  (DataRec(J), J = 1, Nvar)  
210   continue  
C write pair comparison data to pair file  
write (un11,*, err = 999)  
+  (From(PairPtr(I)), I = 1, Npair)  
write (un11,*, err = 999)  
+  (To(PairPtr(I)), I = 1, Npair)  
write (un11,*, err = 999)  
+  (Distance(I), I = 1, Npair)  
write (un11,*, err = 999)  
+  ((Vector(I,Pai:Pairptr(J)))  
  +  I = 1, datadim),J = 1, Npair)  
close (10)  
close (11)  
C   end Write_Pairs  
do 220 i=1,nstruc  
sill(i)=0.0  
rang1(i)=lagmaxdist  
220 continue  
call Display_Location_Map  
if(loption.ge.4)then  
call Display_Pairs_Histogram  
endif  
call Get_Geostat_Main_Menu  
C   nkvar=1 not recycling now  
C   do 10000 kk=1,nkvar  
C   kvar=KrigVar(kk)  
KdVar=KrigVar(1)  
kvar=KdVar+datadim  
910 format(A80)  
920 format(I3,A14)  
930 format(1x,2A14)  
if(loption.eq.5)then  
C     statistics sequence follows:  
call CALCULATE_DATA_STATISTICS  
call DISPLAY_DATA_STATISTICS
if(TRANSFORM_SPECIFIED.gt.0)
  call TRANSFORM_VARIABLES
endif
endif

C vario sequence follows
if(Ioption.GE.1)
  initialize variables
  Azimuth = 0.0
  Dip = 0.0
  Tolerance = 90.0
  Direction(1) = Azimuth
  Direction(2) = Dip
  LagMinDist = 0.0
write(*,*)'max pair distance:',Distance(Npair)
write(*,*)'enter maxdistance for exp vgm:'
read(*,*)LagMaxDist

C LagMaxDist = Distance(Npair)
LagIncDist = LagMaxDist / float(nlag-0.01)
write(*,*)
write(*,*)'Lag Parameters:'
write(*,*)'LagMinDist',LagMinDist
write(*,*)'LagMaxDist',LagMaxDist
write(*,*)'LagIncDist',LagIncDist
write(un12,*),'LagMaxDist',LagMaxDist
write(un12,*),'LagIncDist',LagIncDist
if (Nlag.LE.1)
  write(*,*) 'too few lags'
goto 99999
endif
Tcutoff = LagMinDist + LagIncDist
do 300 i=1,nlag
  LagCutoff(i) = Tcutoff
  Tcutoff = Tcutoff + LagIncDist
300  continue
  call VRank(Lagcutoff,Nlag)
C compute experimental variogram
C
C calculate residuals or use input data
C for calculation of variograms
C
C not cycling now only one KrigVar value
C
do 360 j=1,NkVar
C KdVar=KrigVar(j)
do 350 i=1,Nkept
  if(Iresid.eq.1) then
    Smooth(i,KdVar)=data(i,KdVar)
    Resid(i,KdVar)=0.0
  else
    call filter(KdVar,nkept,i,SParam(KdVar))
  endif
350 continue
C 360 continue
  if(Ioption.GE.3) then
    note no offset
    call Compute_Vario
    if(Ioption.ge.4) then
      call Display_Variogram_Lags_Histogram
    endif
  endif
C automatic subroutines follow:
C set model to show experimental variogram values
  do 425 i=1,nlag
    model(i)=estimate(i)
  425 continue
C display experimental variogram
  call Display_Experimental_Variograms
  read(*,*)nlagfit
  write(un12,*)'Nlagfit',nlagfit
  call CALCULATE_DEFAULT_VARIOGRAMS(gamma,two,varlm)
  call Display_Variograms_With_Models
C note no offset
  if(Ioption.GE.2) then
    write(12,*)'nugget=', nugget
    write(12,*)'sill(1)=', sill(1)
    write(12,*)'GamMax=', GamMax
    write(12,*)'range=', range1(1)
    write(12,*)'Imodel(1)=', Imodel(1)
  endif
  if(Ioption.GE.2) then
    write(*,*)' enter 1 to xvalidate data;'
    read(*,*)igrid
    if(igrid.eq.1) then
      if(iiocal.gt.l) then

C xvalid sequence follows
  open (unit=un15, File=Nbhd_File, status=
  + 'unknown', err=9999)
if(ilocal.eq.2)then
  C optimal nbhd used (inbhd=1)
    read(un15,*,err=999,end=9999)
    dummy,nkvar,inbhd,iso,iest,ierr,minptrs,maxptrs,npts
elseif(ilocal.eq.3)then
  C limited variance nbhd used (inbhd=4)
    read(un15,*,err=999,end=9999)
    dummy,nkvar,inbhd,iso,iest,ierr,minptrs,maxptrs, varlm
elseif(ilocal.eq.4)then
  C fixed n nbhd used (inbhd=3)
    read(un15,*,err=999,end=9999)
    dummy,nkvar,inbhd,iso,iest,ierr,minptrs,maxptrs,npts
endif
if(dummy.ne.ndata)then
  C limited variance nbhd used (inbhd=4)
    write(*,*)'numbers in nbhd_file not same stopping',
    dummy,ndata
endif
else
  open (unit=un15, File = Grid_File, status=
  + 'unknown', err=9999)
  read(un15,*,err=999,end=9999)nkept
  write(*, *)'xvalidating no. of grid pts:',nkept
endif
C initialize krigier parameters
    Num_Neg = 0
    Num_Errors = 0
    Num_NoFound = 0
    Num_Neg = 0
    NZeroDist = 0
    call XVALID (igrid)
    write(12,*)'Xvalid Num_Neg Var :,Num_Neg
endif
C close(15)
C kriging sequence follows
C set grid points and solve eqns for each pt
C set xval code to zero
xval=0
open (unit=un14, File=Results_File, status='unknown', err=9999)
if(ilocal.gt.1)then
  open (unit=un15, File=Grid_File, status='unknown', err=9999)
endif

C set common data/into pts zero
Same=0
C Initialize VKRIGER
Num_Errors = 0
Num_NoFound = 0
Num_Neg = 0
NZeroDist = 0
C set code for select S.R. for grid

gflag=0
if(ilocal.eq.1)then
  write(12,*')'Numptsx,xcmin,delx,'
  + Numptsx,xcmin,delx
  write(12,*')'Numptsy,ycmin,dely,'
  + Numptsy,ycmin,dely
  write(12,*')'Numptse,ecmin,dele,'
  + Numptse,ecmin,dele
numestimate=0
do 450 k=1,Numptse
  if(dimension.gt.2) e0=ecmin+dele*float(k-1)
do 440 j=1,Numptsy
    if(dimension.gt.1) y0=ycmin+dely*float(j-1)
do 430 i=1,Numptsx
      x0=xcmin+delx*float(i-1)
      numestimate=numestimate+1
if(Ikrig.le.3)then
  call Vkriger(x0,y0,e0,gflag)
elseif(Ikrig.eq.4)then
  call kerest(x0,y0,e0,gflag)
endif
  continue
  continue
  continue
else
  numestimate=0
  read(15,*')numpts
  do 460 i=1,numpts
numestimate = numestimate + 1
if (Ikrig .le. 3) then
  call VKRIGER(x0,y0,e0,gflag)
elseif (Ikrig .eq. 4) then
  call kerest(x0,y0,e0,gflag)
end if
460  continue
endif
write(12,*) 'Vkriger Num_Neg_Var :', Num_Neg
close(12)
close(14)
close(15)
close(16)
close(17)
write(*,*) 'Same no. data and interp. points', same
write(*,*) 'DONE check output files'
C   display or write results
   call DISPLAY_KRIGED_VALUES
   call DISPLAY_KRIGING_VARIANCES
   go to 99999
999  write(*,*) 'error reading or writing file'
go to 99999
99999 write(*,*) 'error in opening files'
999999 continue
end
C
C DUMMY SUBROUTINES NEEDED TO COMPILE
C PRIOR TO FINAL DEVELOPMENT FOLLOW:
C
  subroutine DISPLAY_LOCATION_MAP
  return
  end

  subroutine CALCULATE_DATA_STATISTICS
  return
  end

  subroutine DISPLAY_DATA_STATISTICS
  return
  end

  subroutine TRANSFORM_VARIABLES
return
end

C subroutine DISPLAY_PAIRES_HISTOGRAM
return
end

C subroutine DISPLAY_VARIOGRAM_LAGS_HISTOGRAM
return
end

C subroutine DISPLAY_EXPERIMENTAL_VARIOGRAMS
return
end

C subroutine DISPLAY_VARIOGRAMS_WITH_MODELS
return
end

C subroutine DISPLAY_KRIGED_VALUES
return
end

C subroutine DISPLAY_KRIGING_VARIANCES
return
end

C subroutine DISPLAY_KRIGING_VARIANCES
return
end

C subroutine Get_Statistics_Menu
return
end

subroutine Get_Xvalid_Menu
return
end

subroutine Get_Interpolation_Menu
return
end
subroutine Get_Geostat_Main_Menu
include 'geostat.inc'
C set to add non english text later
Language=1
if(Language.eq.1)then
    call Get_Options_Menu
    if(Ioption.eq.1.OR.Ioption.eq.2)then
        call Get_Model_Menu
    endif
    if(Ioption.ge.4)then
        call Get_Statistics_Menu
    endif
    if(Ioption.GE.3)then
        call Get_Vario_Menu
    endif
    if(Ioption.GE.2)then
        call Get_Xvalid_Menu
    endif
    if(Ioption.GE.1)then
        call Get_Interpolation_Menu
    endif
endif
endif
return
end

subroutine Get_Options_Menu
include 'geostat.inc'
integer I, J, TEMP, numpts
if(Language.eq.1)then
    C doing only one variable at a time now
    NkVar=1
    write(*,*),'ENTER DEPENDENT VARIABLE NO OF INTEREST'
    DO 5 J=1,NkVar
        read(*,*)KrigVar(J)
        KdVar=KrigVar(J)
        if(Iresid.ne.1)then
            ifk=0
            write(*,*),'input kernel function for this var.:'
            call kernel(1,2,1.0,1.0,1.0,1)
            kfun(Kdvar)=ifk
            write(*,*),'input smoothing param. for this var.:'
            read(*,*)SParam(Kdvar)
        endif
    5  
endif
endif
5 continue
write(un12,*),'KRGVAR',(KrigVar(i),i=1,nkvar)
write(*,'INPUT CALCULATION OPTIONS:"
write(*,*)' 0 = GENERATE PAIR COMPARISON FILE ONLY'
write(*,*)' 1 = KRIG WITH KNOWN VARIOGRAM MODEL'
write(*,*)' 2 = ADD XVALID WITH KNOWN VGM MODEL'
write(*,*)' 3 = ADD VGM MODELING PRIOR TO XVALID'
write(*,*)' 4 = ADD HISTOGRAMS DURING CALCS'
write(*,*)' 5 = ADD STATS PRIOR TO VGM MODELING'
read(*,*)Ioption
write(un12,*)'Ioption',Ioption
if(Ioption.eq.0) then
write(*,*)'PAIR DATA IN Variopcf'
STOP
endif
write(*,*)' INPUT KRIGING SYSTEM DEFINITION:"
write(*,*)' 1 = SIMPLE KRIGING'
write(*,*)' 2 = ORDINARY KRIGING'
write(*,*)' 3 = SIMPLE KRIGING WITH NEIGHBORHOODS'
write(*,*)' 4 = INDEPENDENT KERNEL ESTIMATE'
read(*,*)Ikrig
write(un12,*)'Ikrig',Ikrig
if(Ikrig.eq.1) then
write(*,*)'input known global mean:'
read(*,*)Mglobal
else
Mglobal=0.0
endif
if(Ikrig.eq.4) then
 call kernel(1,2,1.0,1.0,1)
endif
write(*,*)'enter data check point for print out,
+ 0 for none:'
read(*,*)chkpt
write(*,*)' INPUT ESTIMATION PROBLEM DIMENSION:"
write(*,*)' 1 = 1D; 2 = 2D OR LAYERED 3D; 3 = 3D'
read(*,*)Dimension
if(Dimension.gt.DataDim) then
 write(*,*)' X IS ALWAYS FIRST COLUMN '
 write(*,*)' ENTER 1 TO LEAVE Y AS IS, '
+ 0 TO CREATE ZEROS'
read(*,*)TEMP
if(TEMP.eq.0)then
    DO 1=1,Ndata
    YDATA(I)=0.0
1    continue
endif
write(*,*)' E IS LAST COLUMN SET TO ZERO '
DO 4 I=1,Ndata
    EDATA(I)=0.0
4    continue
end if
if(Dimension.LT.DataDim)then
    write(*,*)' PROBLEM DIMEN. LESS THAN DATA DIMEN. '
    write(*,*)' SETTING DIMENSION TO DATA DIMENSION '
    Dimension=DataDim
endif
write(un12,*),Dimension,Dimension
LAYERS=0
if(Dimension.eq.2)then
    write(*,*)' ENTER NUMBER OF LAYERS (REGULAR 2D=1)'
    read(*,*)LAYERS
    write(un12,*),LAYERS,LAYERS
endif
write(*,*)' INPUT VARIOGRAM Dimension'
write(*,*)' 1 = 1D; 2 = 2D OR LAYERED 3D; 3 = 3D'
read(*,*)VARDIMEN
write(un12,*),VARDIMEN,VARDIMEN
write(*,*)' INPUT NUMBER OF LAGS FOR EXPER. VGM:'
read(*,*)Nlag
write(un12,*),Nlag,Nlag
write(*,*)' INPUT KRIGING NEIGHBORHOOD DEFINITION: '
write(*,*)' 1 = GLOBAL KRIGING, 2 = OPT NBHD '
write(*,*)' 3 = LIMITED VAR, 4 = FIXED N KRIGING '
if(datadim.eq.2)then
    write(*,*)' 5 = STAND NBHD KRIGING '
else
    if(Ivgm.eq.5)then
        write(*,*)' Need 2-D for stand. nbhd vgm '
        ilocal=1
        write(*,*)' have to use global krig nbhd',ilocal
    go to 7

end if
endif
read(*,*)Ilocal

7 if(Ilocal.eq.1) then
   if(Ndata.gt.MaxNeighbor) then
      write(*,*)'MORE DATA POINTS THAN ALLOWED:
             + SET Ilocal 2-5'
      go to 6
   endif
endif
if(Ilocal.eq.2.or.Ilocal.eq.3.or.Ilocal.eq.4) then
   write(*,*)'NBHD SET TO MATCH VGM NBHD OPT.',Ivgm
   Ilocal=Ivgm
   write(*,*)'ENTER KRIG NBHD SCALE FACTOR
             + [0.2 - 2.0 (*.VGM)]'
   read(*,*)SCALE
endif
if(Ilocal.eq.5) then
   if(datadim.eq.2) then
      write(*,*)'enter standard search ellipse:
             + a,b,theta(deg)'
      read(*,*)SrMajor,SrMinor,SrAngle
   else
      write(*,*)'this opt combination not allowed'
      stop
   endif
endif
write(un12,*)'Ilocal',Ilocal
write(un12,*)'SrMajor,SrMinor,SrAngle:',srmajor,srminor,srangle
if(Ilocal.eq.1) then
   NUMPTSY = 1
   NUMPTSE = 1
   write(*,*)'TOTAL POINTS ALLOWED IN KRIGE GRID IS:'
   + MaxEstimate
if(Dimension.eq.1) then
   write(*,*)'INPUT NUMBER OF GRID POINTS IN X DIRECTION:'
   + read(*,*)NUMPTSX
if(NUMPTSX.gt.maxestimate) then
   write(*,*)'numpts limited to',maxestimate
   NUMPTSX=maxestimate
endif
write(un12,*)'NUMPTS: X',NUMPTSX
elseif(Dimension.eq.2)then
  write(*,*)'INPUT NUMBER OF GRID POINTS IN X,Y DIRECTIONS:'
  read(*,*)NUMPTSX,NUMPTSY
  numpts=numptsx*numptsy
  if(numpts.gt.maxestimate)then
    write(*,*)'total numpts limited to',maxestimate
    go to 8
  endif
  write(un12,*)'NUMPTS: X,Y',NUMPTSX,NUMPTSY
elseif(Dimension.eq.3)then
  write(*,*)'INPUT NUMBER OF GRID POINTS IN X,Y,E DIRECTIONS:'
  read(*,*)NUMPTSX,NUMPTSY,Numptse
  write(un12,*)'NUMPTS: X,Y,E',
  NUMPTSX,NUMPTSY,Numptse
  numpts=numptsx*numptsy*numptse
  if(numpts.gt.maxestimate)then
    write(*,*)'total numpts limited to',maxestimate
    go to 10
  endif
endif
endif
write(*,*)'INPUT VARIOGRAM CALCULATION TYPE'
write(*,*)' 1 = GAMABS1'
write(*,*)' 2 = GAMABS2'
write(*,*)' 3 = GAMREL'
write(*,*)' 4 = GAMERGO'
read(*,*)GammaType
write(un12,*)'GammaType',GammaType
endif
return
deend

integer I
if(Language.eq.1)then
  DO 10 I=1,MAXSTRUC
    Imodel(I)=0
  10 continue
10 continue
C
if(Ioption.GE.3)then
ISO=1
Nsills=1
else
write(*,*),'INPUT MODEL STRUCTURE:
write(*,*),' 1 = NUGGET PLUS 1 FINITE SILL MODEL'
write(*,*),' 2 = NUGGET PLUS 2 FINITE SILL MODELS'
write(*,*),' 3 = NUGGET PLUS 3 FINITE SILL MODELS'
write(*,*),' 4 = NUGGET PLUS 4 FINITE SILL MODELS'
write(*,*),' 5 = NUGGET PLUS 1 POWER MODEL'
read(*,*)Nsills
if(Nsills.LE.4)then
write(*,*),'INPUT MODEL TYPE FOR MODEL # 1'
write(*,*),' 1 = SPHERICAL, 2 = GAUSSIAN,
+ 3 = EXPONENTIAL, 4 = CUBIC,
+ 5 = QUADRATIC, 6 = HOLE_EFFECT'
read(*,*)Imodel(1)
write(un12,*),'ImodeI1',Imodel(1)
if(Nsills.gt.1)then
write(*,*),'INPUT MODEL TYPE FOR MODEL # 2'
write(*,*),' 1 = SPHERICAL, 2 = GAUSSIAN,
+ 3 = EXPONENTIAL, 4 = CUBIC,
+ 5 = QUADRATIC, 6 = HOLE_EFFECT'
read(*,*)Imodel(2)
write(un12,*),'ImodeI2',Imodel(2)
endif
if(Nsills.gt.2)then
write(*,*),'INPUT MODEL TYPE FOR MODEL # 3'
write(*,*),' 1 = SPHERICAL, 2 = GAUSSIAN,
+ 3 = EXPONENTIAL, 4 = CUBIC,
+ 5 = QUADRATIC, 6 = HOLE_EFFECT'
read(*,*)Imodel(3)
write(un12,*),'ImodeI3',Imodel(3)
endif
if(Nsills.gt.3)then
write(*,*),'INPUT MODEL TYPE FOR MODEL # 4'
write(*,*),' 1 = SPHERICAL, 2 = GAUSSIAN,
write(*,*),' 3 = EXPONENTIAL, 4 = CUBIC'
write(*,*),' 5 = QUADRATIC, 6 = HOLE_EFFECT'
read(*,*)Imodel(4)
write(un12,*),'Imode14',Imodel(4)
endif
else
   Imodel(1)=5
write(un12,*),'Imode11',Imodel(1)
endif
endif
if(loption.LE.2)then
  write(*,*)'INPUT VGM MODEL NUGGET:
read(*,*)Nugget
write(un12,*),'NUGGET',Nugget
if(Dimension.gt.1)then
    write(*,*)' ENTER 1 FOR ISOTROPIC,  
            2 FOR ANISOTROPIC MODELS'
    read(*,*)ISO
else
    ISO=1
endif
GamMax=nugget
DO 81=1,Nsills
write(*,*)'ENTER SILL FOR MODEL NO.:',I
read(*,*)SILL(I)
GamMax=GamMax+sill(i)
write(un12,*),'SILL:',I,SILL(I)
if(ISO.eq.1)then
  write(*,*)'INPUT RANGE FOR ISO. MODEL NO.:',I
  read(*,*)Range1(I)
  Range2(I)=Range1(I)
  Range3(I)=Range1(I)
  write(un12,*),'ISOTROPIC RANGE',Range1(I)
else
  write(*,*)'INPUT RANGES FOR ANIS. MODEL NO.:',I
  write(*,*)'INPUT SMALLEST RANGE'
  read(*,*)Range1(I)
  if(Dimension.eq.2)then
    write(*,*)'INPUT LARGEST RANGE'
    read(*,*)Range2(I)
  endif
  if(Dimension.gt.2)then
    write(*,*)'INPUT INTERMEDIATE RANGE'
    read(*,*)Range2(I)
    write(*,*)'INPUT LARGEST RANGE'
read(*,*)Range3(I)
endif
endif
8 continue
endif
endif
return
end

subroutine Get_Vario_Menu
include 'geostat.inc'
integer Ismth
Ismth = 1
return
end

Subroutine select_points(iptr,rptr,x0,y0,z0,
+   ivalue,nvalue,apt,bpt,cpt)
C
C find points inside given ellipsoid parameters
C xval gives no of xvalid pt else zero;
C x0,y0,z0, central points of selection ;
C if 2d column 3 of data and z0 are 0
C
real x0,y0,z0,arg
integer i, n, nvalue, rptr(200), iptr(200), ivalue,
+   ptr, apt, bpt, cpt
include 'geostat.inc'
C find ptr(l) from distance to origin
n=ivalue
nvalue=0
do 10 i=1,n
ptr=iptr(i)
xp= (xdata(ptr)-x0)*aaxis(1)+(ydata(ptr)-y0)*aaxis(2)
+   +(edata(ptr)-z0)*aaxis(3)
yp= (xdata(ptr)-x0)*baxis(1)+(ydata(ptr)-y0)*baxis(2)
+   +(edata(ptr)-z0)*baxis(3)
zp= (xdata(ptr)-x0)*caxis(1)+(ydata(ptr)-y0)*caxis(2)
+   +(edata(ptr)-z0)*caxis(3)
if(abs(xp).le.0.0.and.abs(yp).le.0.0.
+   and.abs(zp).le.0.0)then
C
points are not included in own nbhds
if(xval.ne.0) then
   write(un12,*) i, 'xvaliding:
   + point at origin, not included '  
   endif
   go to 10
endif
arg = (xp/a0)**2
if(data_dim.gt.1) arg = arg + (yp/b0)**2
if(data_dim.gt.2) arg = arg + (zp/c0)**2
if(xval.ne.0 and xval.eq.ckpt) then
   write(un16,*) xp, yp, zp, arg
endif
if(arg.le.1.0 or apt.eq.ptr or bpt.eq.ptr + or.cpt.eq.ptr) then
   C point generates or is in ellipse; save in results ptr
   nvalue = nvalue + 1
   rptr(nvalue) = ptr
   if(xval.ne.0 and xval.eq.ckpt) then
      write(un16,*) 'select, nvalue:', nvalue
      write(un16,*) 'i, rptr(nvalue):', i, ptr
   endif
endif
10 continue
return
end

subroutine xvalid (igrid)
C - Krigs each data point and calculates error of 
C estimation for one data set at a time given by 
C value of kvar - may also validate reference 
C data set in Grid.nhd 
C Converted From GEOEAS by M.M.Moody oct 1991 
C
include 'geostat.inc'
integer I, TN, NdMiss, igrid, gflag
real X0, Y0, E0, sigma, est, std, V0
save V0, sigma, est, std, I, TN, NdMiss

C Main kriging loop - krig each sample in data file
NdMiss = 0
gflag = 0
xVariance = 0.0
C Initialize vkriger

Num_Errors = 0
Num_NoFound = 0
NZeroDist = 0
DO 10 I = 1, Nkept
   if(igrid.eq.1)then
      gflag=1
   C xvalidate data
      if(data(i,kdvar).ge.missing)then
         write(*, *)'missing value at data point:',I
         ndmiss=ndmiss+1
         go to 10
      endif
      XO=Xdata(I)
      YO=Ydata(I)
      E0=Edata(I)
      VO=data(i,kdvar)
   else
      C xvalidate reference pts in grid file
      gflag=0
      read(un15,*)X0,Y0,E0,V0
   endif
   numestimate = I
   xval=I
      write(12, *)'xvalidate',I,XO,YO,E0,V0
   call vkriger(X0,Y0,E0,gflag)
   xVariance = xVariance + (Estimate(i)-VO)**2
   est=estimate(i)
   std=stddev(i)
      write(un12, *)'i,data, est, kstd',I,VO,est,std
   10 continue
   write(un12,*')no. of points with no neighbors'
      Num_NoFound
   write(un12,*')no. of points with missing values'
      NdMiss
   xVariance=xVariance/float(Nkept-ndmiss-Num_NoFound)
   sigma = sqrt(xVariance)
      write(un12,*') mean squared xvalidation error='
      xVariance
      write(un12,*') root mean squared xvalidation error='
      sigma
   if(ioption.eq.4)then
C compute stats
  call Stats (Estimate, Nkept,
+     EMin, EQ1, Emedian, EQ3, EMax,
+     EMean, Estd)
  call Stats (StdDev, Nkept,
+     SMin, SQ1, Smedian, SQ3, SMax,
+     SMean, Sstd)
  call Stats (Difference, Nkept,
+     DMin, DQ1, Dmedian, DQ3, DMax,
+     DMean, Dstd)
  call Stats (Zscore, Nkept,
+     ZMin, ZQ1, Zmedian, ZQ3, ZMax,
+     ZMean, Zstd)
endif
  return
end

subroutine VKriger(xO,yO,eO,gflag)

C GEOEAS S.R. - Performs kriging for each data location
C or interpolation point. gflag set on entry for code
C to use on select subroutine option. gflag set on
C exit to code for condition of calculation or search
C Modified by M.M.Moody, Sept, 1992
  include 'geostat.inc'

real Zscore(MaxEstimate)
real en, zn, dn, sn, vn, est, Dtr, v, vsq, totalsum
real Tmax, Dx, Dy, De, xO, yO, eO, v0, MinDist2
integer Neighbors(MaxNeighbor), N, NZero, Tmpdata
integer TNeighbors(MaxNeighbor), TN, IPtr,
integer Inptr(MaxData), npts, dummy, Tmpdata
integer I, J, K, nvalue, rptr(200), minflag, ivalue
integer Error, Failed, apt, bpt, cpt, gflag, nume
save Tmax, Dx, Dy, De, Neighbors, N, NZero,
+     I, J, K, TN, TNeighbors, IPtr, Tmpdata
data Dtr /0.0174533/
  Dx = Delx
  Dy = Dely
  De = Dele
  Tmax = AMAX1(Dx, Dy)
  MinDist2 = MinDist
I = numestimate
Error = 0
if(xval.ne.0.and.xval.eq.chckpt)write(un16,*)
+ 'x0,y0,e0',x0,y0,e0
if(ilocal.EQ.2.OR.ilocal.EQ.3.or.ilocal.eq.4)then
C read optimal nbhd parameters from file for search
if(xval.eq.0)then
read(un15,*, end=999, err=9999)xO,yO,eO
endif
read(un15,*, end=999, err=9999)dummy,minflag,
  + apt,bpt,cpt
read(un15,*, end=999, err=9999)npts,v,vsq
read(un15,*, end=999, err=9999)a0,b0,c0
read(un15,*, end=999, err=9999)(aaxis(j),j=1,3)
if(minflag.ne.1)then
read(un15,*, end=999, err=9999)(baxis(j),j=1,3)
read(un15,*, end=999, err=9999)(caxis(j),j=1,3)
endif
endif
C set kriging nbhd in scale to vgm nhd
if(xval.eq.0)then
a0=scale*a0
b0=scale*b0
c0=scale*c0
endif
if(xval.ne.0.and.xval.eq.chckpt)then
write(un16,*)'chkpt nbhd parameters a0,b0,c0:',
  a0,b0,c0
write(un16,*)(aaxis(j),j=1,3)
if(minflag.ne.1)then
write(un16,*)(baxis(j),j=1,3)
write(un16,*)(caxis(j),j=1,3)
endif
endif
do 5 j = 1,ndata
inptr(j)=j
5 continue
if(gflag.eq.1)then
  call select_points(inptr,rptr,xO,y0,e0,
    + ndata,nvalue,apt,bpt,cpt)
else
  call xselect_points(inptr,rptr,xO,y0,e0,
    + ndata,nvalue)
endif
n=nvalue
if(Ikrig.eq.3)then
  Mglobal=0.0
  do 6 j=1,n
    Mglobal=Mglobal+data(rp(j),kdvar)
  continue
  Mglobal=Mglobal/float(n)
endif
  do 10 j=1,n
    Neighbors(j)=rp(j)
    if(xval.ne.0.and.xval.eq.chckpt)then
      write(un16,*)'check points:',j,rp(j)
    endif
  continue
elseif(ilocal.EQ.3)then
  C else use standard nbhd for search
  C but only for 2-D per input options
  C geoease version only 3 statements deleted with little
  C
  C Set temporary coordinate and sample values,
  C initialize neigbors to 0
  C
  Zptr(I) = numestimate
  N = 0
  C
  C perform neighborhood search (for non-missing values)
  C call Search (X0, Y0, Neighbors, N, Failed)
  C
  C replaced geoeas version with search by select_points
  do 12 j = 1,ndata
    inptr(j)=j
  continue
  C convert standard ellipse to pseudo 3-D
  a0=SrMajor
  b0=SrMinor
  c0=0.0
  aaxis(1)=cos(SrAngle*Dtr)
  aaxis(2)=sin(SrAngle*Dtr)
  aaxis(3)=0.0
  baxis(1)=-1.*aaxis(2)
  baxis(2)=aaxis(1)
  baxis(3)=0.0
caxis(1)=0.0
caxis(2)=0.0
caxis(3)=0.0
call select_points(inptr,rptr,x0,y0,e0,ndata,
   nvalue,apt,bpt,cpt)
n=nvalue
write(12,*)'number of nbrs selected:',n
do 14 j=1,n
Neighbors(j)=rptr(j)
   if(xval.ne.0.and.xval.eq.cbckpt)then
      write(un16,*)'check points:',j,rptr(j)
   endif
14 continue
c
else
   C use global nbhd (nbrs = all other pts) for search
   NZero=0
   NZeroDist=0
   if(xval.eq.0)then
      C set neighbors and find data points same as int. point
      npoints=Numptsx*Numptsy*Numptse
      DO 20 J=1,Nkept
         if(xO.eq.xdata(j).and.yO.eq.ydata(j).and.eO.eq.edata(j))then
            Estimate(I)=Data(J,kdvar)
            if(iresid.eq.2)Estimate(i)=Smooth(J,kdvar)
            if(iresid.eq.3)Estimate(i)=Resid(J,kdvar)
            STdDev(I)=0.0
            Difference(I) = 0.0
            Zscore(I) = 0.0
            Same=Same+1
            Tn = 0
            N=TN
            Neighbors(J)=0
            NBHDPTS(I)=0
            go to 100
         endif
         Neighbors(J)=J
      20 continue
   N = Nkept
else
   C find neighbors for xvalidation with deleted point

N = Nkept-1
DO 30 J=1,Nkept
  if(xval.gt.j)then
    Neighbors(J)=J
  elseif(xval.eq.j)then
    go to 30
  elseif(xval.lt.j)then
    Neighbors(J-1)=J
  endif
30  continue
endif
c
C Make list of neighbors (samples) with non-missing values
TN = 0
if( xval.ne.0.and.xval.eq.chckpt)then
  write(un16,*),'Xvalid/Vkriger Tnbrs at checkpt:'
endif
DO 35 J = 1, N
  IPtr = Neighbors(J)
  if (Data(IPtr,kdvar) < Missing) then
    TN = TN + 1
    TNeighbors(TN) = IPtr
  endif
35 continue
C added TN counter = Nbhdpts
Nbhdpts(I) = TN
C If search failed to produce neighbors, then missing
if (TN .EQ. 0) then
  Num_NoFound = Num_NoFound + 1
if (Num_NoFound .GT. 25) then
  write ('*',*) 'Too many points no neighbors'
  GOTO 999
endif
if(xval.eq.0)then
  estimate(I)= Missing
  stddev(i)=Missing
else
  Estimate(I)=Data(J,kdvar)
  if(iresid.eq.2)Estimate(i)=Smooth(J,kdvar)

if(iresid.eq.3) Estimate(i) = Resid(J,kdvar)
    stddev(i) = abs(Data(i,kdvar) - Estimate(i))
endif
    gflag=-1
return
else
C Build system (NZero is no. of nbr pairs with 0 distance)
call VBuild_System(TNeighbors, TN, NZero, 
    + x0,y0,e0)
    NZeroDist = NZeroDist + NZero
C Solve the kriging system of equations, and compute results
call VSolve_System(TNeighbors, TN, Error, 
    + I,totalsum)
C Keep a count of system solver errors, check for too many
    if (Error.eq.1) then
        Estimate(I) = Missing
        Difference(I) = Missing
        Num_Errors = Num_Errors + 1
        if (Num_Errors .GT. 10) then
            write (*,*) 'Too many Solver errors'
            GOTO 999
        endif
    else
        endif
    endif
    Estimate(I) = vk
    stdDev(I) = sk
100 continue
    nume=numestimate
    if(xval.eq.0)then
        write(un14,*)x0,y0,e0,estimate(nume),
        + stddev(nume),Mglobal,totalsum
        write(un12,*)'Results:',x0,y0,e0,
        + estimate(nume),stddev(nume),Mglobal,totalsum
    else
        write(un17,*)x0,y0,e0,estimate(nume),stddev(nume),
        + Mglobal,totalsum
    endif
    gflag=0
return
C Error or quit exit
999 continue
subroutine VBuild_System(TNeighbors, TN, ZeroFlag,
+ x0,y0,e0)

C GEOES S.R.- Builds the system of equations used in
C kriging. The function VGamma is called to fill the
C kriging matrix.

include 'geostat.inc'

integer TNeighbors(MaxNeighbor), TN, ZeroFlag
integer I, J, Iptr, Jptr
real G, x0, y0, e0
double precision Dlx, Dly, D1e, D2x, D2y, D2e
save I, J, Iptr, Jptr

C Initialize
ZeroFlag = 0
DO 10 I = 1, MAXEQ
  B(I) = 0.0
  B2(I) = 0.0
DO 10 J = 1, MAXEQ
  W(I, J) = 0.0
10 continue
if (TN .LE. 0) then
  return
else
  if(checkpt.ne.0)then
    write(un16,*)'TN:',TN
  endif
endif
C Set up System of Equations for simple, or ordinary kriging
DO 25 I = 1, TN
  IPtr = TNeighbors(I)
  D1x = XData(Iptr) - X0
  D1y = YData(Iptr) - Y0
\[ D1e = EData(Iptr) - E0 \]
call VGamma(D1x,D1y,D1e, G)
B(I) = GamMax - G
if(xval.ne.0.and.xval.eq.chkpt)then
  write(un16,*)'x(i),d(i):',x0,y0,e0,d1x,d1y,d1e
call(vgamma,d2x,d2y,d2e, G)
endif
B2(I) = B(I)
DO 20 J = 1, I-1
  Jptr = TNeighbors(J)
  D2x = Xdata(Iptr) - XData(Jptr)
  D2y = YData(Iptr) - YData(Jptr)
  D2e = EData(Iptr) - EData(Jptr)
call VGamma(D2x,D2y,D2e, G)
C flag pairs with zero distance (duplicate pairs)
  if (G .LE. 1.E-20) then
    W(I, J) = GamMax * .999
    ZeroFlag = ZeroFlag + 1
  else
    W(I, J) = GamMax - G
  endif
W(J, I) = W(I, J)
20 continue
W(I, I) = GamMax
25 continue
Neq = TN
C Fill in non-bias conditions for ordinary Kriging
if (Ikrig .EQ. 2) then
  Neq = TN + 1
DO 30 I = 1, TN
  W(TN+1, I) = 1.0
  W(I, TN+1) = 1.0
30 continue
W(TN+1, TN+1) = 0.0
B(TN+1) = 1.0
B2(TN+1) = B(TN+1)
endif
return
end
subroutine VSolve_System(TNeighbors, TN, 
+ Error,inum,totalsum)

C GEOEAS S.R. Solves the system of equations and 
C computes the Kriging estimate and standard deviation 
C Modified by M.M.Moody, Sept, 1992

include 'geostat.inc'
integer TNeighbors(MaxNeighbor), TN, Ndim
integer j, K, I, Ptr, Error, inum
real totalsum
Double Precision Array(86,86), Sum
save Array

if (Ikrig.EQ. 2) then
  Ndim = TN + 1
else
  Ndim = TN
endif
write(12,*)'dimension used:', ndim
do 10 i=1,tn
  do 5 j=1,tn
    Array(i,j) = w(i,j)
  5 continue
10 continue
if (Ikrig.EQ. 2) then
  DO 11 I = 1, TN
    Array(tn+1,i)=1.0
    Array(i,tn+1)=1.0
  11 continue
  Array(tn+1,tn+1)=0.0
endif

if(inum.eq.chkpt) then
  write(16,*)'Input Array; ndim, tn:' , ndim, tn
  do 12 i = 1, ndim
    write(16,*) (W(i,j),j=1,ndim)
  12 continue
  write(16,*)'input b vector:' 

write(16,*) (b2(j),j=1,ndim)
endif

call VSimeq(Ndim, Error)

C Solve system of equations (LU - decomposition)
totalsum=0.0
do 20 j=1,tn
totalsum=totalsum+b(j)
20 continue
if(inum.eq.chckpt)then
  write(16,*)'output b vector: ' 
  write(16,*) (b(j),j=1,ndim)
endif

C Message for system error
if (Error.eq.1) then
  write (*,*) 'Error in system of equations'
  Vk = Missing
  Sk = Missing
else
  C compute kriging estimate and variance
  Vk = 0.0
  Sk = 0.0
  C variance calculation is basic equation for all options
  sum=0.0
  DO 35 K = 1, TN
  Ptr = TNeighbors(K)
  Vk = Vk + (Data(Ptr,kdvar) - Mglobal) * B(K)
  Sk = Sk + B2(K) * B(K)
do 30 j=1,tn
  sum=sum+b(k)*b(j)*Array(k,j)
30 continue
35 continue
  Vk = Vk + Mglobal
  c use gen formula for all as check
  c Sk = GamMax - Sk -b(tn+1)
  Sk = GamMax + sum - 2.0*Sk
C compute kriging standard deviation
  c if (Sk .GT. 0.) then
  c if (Sk .GE. 0.) then
  Sk = DSQRT(Sk)
else
  write (*,*) 'Negative kriging variance; vk,sk'
  + ,vk,sk
Num_Neg = Num_Neg + 1
Sk = Missing
endif
endif
return
de

dsubroutine VSimeq(N, Error)
C GEOEAS S.S. solves the system of equations with a modified LU decomposition.
c put into double precision by mmm

include 'geostat.inc'
integer I, J, K, N, Error
Double Precision Tiny, Sum, TotalSum
DATA Tiny /1.0E-6/
Total Sum = 0.0
C Solve system
Error = 1
DO 10 J = 1, N
if (J .GT. 1) then
  DO 20 I = 1, J - 1
  Sum = W(I, J)
  if (I .GT. 1) then
    DO 30 K = 1, I - 1
  30       Sum = Sum - W(I, K) * W(K, J)
  endif
  W(I, J) = Sum
20 continue
endif
DO 40 I = J, N
  Sum = W(I, J)
  if (J .GT. 1) then
    DO 50 K = 1, J - 1
  50       Sum = Sum - W(I, K) * W(K, J)
  endif
  W(I, J) = Sum
40 continue
if (J .LT. N) then
  if (W(J, J) .GE. Tiny) then
    DO 60 I = J + 1, N
    W(I, J) = W(I, J) / W(J, J)
  endif
10  continue
continue 
else
  return
endif
endif

continue
DO 70 I = 1, N
  Sum = B(I)
  if (I .GT. 1) then
    DO 80 J = 1, I - 1

80        Sum = Sum - W(I, J) * B(J)
  endif
  B(I) = Sum

continue
DO 90 I = N, 1, -1
  Sum = B(I)
  if (I .LT. N) then
    DO 100 J = I + 1, N

100       Sum = Sum - W(I, J) * B(J)
  endif
  B(I) = Sum / W(I, I)
  TotalSum = TotalSum + B(i)
  if(xval.ne.0.and.xval.eq_chkpt)then
    write(un16,*')'B(i)',i,b(i)
  endif

continue
correct total sum for mu in O.K.
if(Ikrig.eq.2)then
  totalsum=totalsum-b(n)
endif
if(xval.ne.0.and.xval.eq_chkpt)then
  write(un16,*')'Total Sum of B vector:;',TotalSum
endif
Error = 0
return
end

subroutine VGamma(Dx, Dy, De, G)

C GEOEAS S.R. function for calculating variogram
C value from increments. mmm converted to d.p.
include 'geostat.inc'
integer I
real Dx, Dy, De, G, pi
double precision Tiny, Teeny, Dist, Slope, Dtr,
      Hu, Hv, R, H
DATA Dtr /0.0174533/, Tiny /1.01E-6/,
      Teeny /1.E-20/
pi=4.0*atan(1.0)
C compute simple distance
Dist = Dx*Dx + Dy*Dy + De*De
if (Dist .GT. 0.0) then
   Dist = DSqrt (Dist)
else
   Dist = 0.0
endif
C variogram for 'zero' distance
if (Dist .LE. Teeny) then
   G = 0.0
   return
C variogram value for very small distances
else if (Dist .LE. Tiny) then
   G = Nugget
   return
endif
C variogram value for non-small distances
G = Nugget
DO 10 I = 1, maxstruc
   if (Imodel(I) .EQ. 0) GOTO 10
C Compute rotated distance components (Hu, Hv)
   if (Varazimuth(I) .GT. Teeny) then
      Azimuth = - Varazimuth(I) * Dtr
      Hu = Dx * COS(Azimuth) - Dy * SIN(Azimuth)
      Hv = Dy * COS(Azimuth) + Dx * SIN(Azimuth)
   else
      Hu = Dx
      Hv = Dy
endif
C Scale with anisotropy ratio, calc structural distance H
   Hu = Hu * (range2(I) / range1(I))
   H = SQRT(Hu * Hu + Hv * Hv)
   R = H / range2(I)
C Spherical
if (Imodel(I) .EQ. 1) then
  if (R .GT. 1.0) R = 1.0
  G = G + Sill(I) * (1.5*R - 0.5*R*R*R)
  C Gaussian (practical range is divided by SQRT(3))
else if (Imodel(I) .EQ. 2) then
  G = G + Sill(I) * (1.-EXP(-3.0*R*R))
  C Exponential (practical range is divided by 3)
else if (Imodel(I) .EQ. 3) then
  G = G + Sill(I) * (1.- EXP(-3.0*R))
  C Cubic
else if (Imodel(I) .EQ. 4) then
  G = G + Sill(I) * (7.0*R**2-8.75*R**3+
                      3.5*R**5-0.75*R**7)
  C Quadratic
else if (Imodel(I) .EQ. 5) then
  G = G + Sill(I) * (2.0*R - R*R)
  C Hole-Effect
else if (Imodel(I) .EQ. 6) then
  G = G + Sill(I) * (1.0 - sin(pi*R)/(R*pi))
  C Linear change to power including linear
else if (Imodel(I) .EQ. 7) then
  Slope = Sill(I) / range2(I)
  G = G + Slope * H changed to:
  G = nugget*((Slope*H)**sill(l))
endif
continue
return
end

Subroutine select_points(iptr,rptr,x0,y0,z0,
+ ivalue,nvalue,apt,bpt,cpt)

  c find points included inside given ellipsoid
  c xval gives no of xvalid pt else zero;
  c x0,y0,z0, central points about which selection
  c occurs; if 2d column 3 of data and z0 are 0
  c Written by M.M.Moody, Sept, 1992

  real x0,y0,z0,arg
  integer i, n, nvalue, rptr(200), iptr(200),
  + ivalue, ptr, apt, bpt, cpt
  include 'geostat.inc'
c determine points  find ptr(1) from distance to origin
n=ivalue
nvalue=0
do 10 i=1,n
ptr=iptr(i)
xp= (xdata(ptr)-xO)*aaxis(1)+(ydata(ptr)-y0)*aaxis(2)
+  +(edata(ptr)-z0)*aaxis(3)
yp= (xdata(ptr)-x0)*baxis(1)+(ydata(ptr)-y0)*baxis(2)
+  +(edata(ptr)-z0)*baxis(3)
zp= (xdata(ptr)-x0)*caxis(1)+(ydata(ptr)-y0)*caxis(2)
+  +(edata(ptr)-z0)*caxis(3)
if(abs(xp).le.0.0.and.abs(yp).le.0.0.
+    and.abs(zp).le.0.0)then
   c points are not included in own nbhds
    if(xval.ne.0)then
       write(un12,*)i,'xvaliding; exclude pt at origin'
    endif
   endif
arg=(xp/aO)**2
if(datadim.gt.1)arg= arg+(yp/bO)**2
if(datadim.gt.2)arg=arg+(zp/cO)**2
if(xval.ne.0.and.xval.eq.chkpt)then
   write(un16,*)xp,yp,zp,arg:' ,xp,yp,zp,arg
endif
if(arg.le.1.0.or.apt.eq.ptr.or.bpt.eq.ptr.
+   or.cpt.eq.ptr)then
   c point generates or is inside ellipse; save
   nvalue=nvalue+1
   rptr(nvalue)=ptr
   if(xval.ne.0.and.xval.eq.chkpt)then
      write(un16,*)'select, nvalue:','nvalue
      write(un16,*)'i,rptr(nvalue):',i,ptr
   endif
endif
10 continue
return
end

Subroutine xselect_points(iptr,rptr,x0,y0,z0,
+    ivalue,nvalue)
find points included inside given ellipsoid

x0,y0,z0, central points about which selection occurs; if 2d column 3 of data and z0 are 0
used for grid pts not exactly set by apt,bpt,and cpt
if data point is at point of interest it is included

Written by M.M.Moody, Sept, 1992

real x0,y0,z0, arg
integer i, n, nvalue, rptr(200), iptr(200),
+ ivalue, ptr, apt, bpt, cpt
include 'geostat.inc'

determine points  find ptr(1) from distance to origin
n=ivalue
nvalue=0

DO 10 i=1,n

ptr=iptr(i)

xp= (xdata(ptr)-x0)*aaxis(1)+(ydata(ptr)-y0)*aaxis(2)
+ (edata(ptr)-z0)*aaxis(3)
yp= (xdata(ptr)-x0)*baxis(1)+(ydata(ptr)-y0)*baxis(2)
+ (edata(ptr)-z0)*baxis(3)
zp= (xdata(ptr)-x0)*caxis(1)+(ydata(ptr)-y0)*caxis(2)
+ (edata(ptr)-z0)*caxis(3)

arg=(xp/a0)**2
if(datadim.gt.1)arg=arg+(yp/b0)**2
if(datadim.gt.2)arg=arg+(zp/c0)**2
if(xval.ne.0.and.xval.eq.checkpt)then
  write(un12,*)xp,yp,zp,arg',xp,yp,zp,arg
endif
if(arg.le.1.0)then
  point is inside ellipse; save in results pointer
  nvalue=nvalue+1
  rptr(nvalue)=ptr
endif
10 continue
if(xval.ne.0.and.xval.eq.checkpt)then
  write(un16,*)'select, nvalue:',nvalue
  write(un16,*)'i,rptr(nvalue):',i,ptr
dendif
return
end
323
BA Auxiliary Routines

PROGRAM gridlto3d
C
C
C
C
C
C
C
C
C

simple gridding and assignment of nearest data nbrhd
parameters: create the grid of interploation points
plus their nbhd parameters. May grid 3-D sets for
given z. All grids increase x then y, then z
May create 1-d transect through 3-d space ; or
may read in special grid pts.
may prepare file for validation of reference pts
code for graphics based on GKS not given
Written by M.M.Moody, 1992
include 'geostat.inc'
integer mx, my, mz, apt(300), bpt(300), cpt(300)
+ iest,ierr,minptrs,maxptrs,nptr,i,j,idim,ixvalid,
+ dum1,dum2,dum3,two,tpts,xpts,ypts,zpts,
+ vpts(300),gflag(300),unt
character*14 Input File
real xcen, ycen, delmx, delmy, delmz,
+
rad(3,300),drc(3,3,300),
+
valnbhd(300),varnbhd(300),disto

C

Data File = 'Vario.dat'
Plot-File = 'vario.plt'
output_File = 'Grid. out'
Input_File = 'Grid.dat l
Results File = 'Grid.nhd'
Nbhd File = 'Vario.nhd'
axes(1)
'X'
axes(2) = 'Y'
C

+
+

open (unit =un10,File= Data File,status=
'old',err = 9999)
open (unit=un15, File=Nbhd_File, status=
'old', err=9999)

C

C
prevar sequence follows:
C initialize prevar subroutine follows:
DO 5 I = 1, MaxPair
PairPtr(I) = I
continue
5
DO 10 I = 1, MaxVar
VarNam(I) = ,
10
continue
Missing = I.E31
Npair
= 0


C the following needs to be in control file or
C introductory menu in reading data.
Xvar   = 1
Yvar   = 2
Evar   = 3
XCmin  = +1.E30
XCmax  = -1.E30
YCmin  = +1.E30
YCmax  = -1.E30
ECmin  = +1.E30
ECmax  = -1.E30
MinDist = +1.E30
MaxDist = -1.E30
C end initialize subroutine
unt=un10
C call Read_Data(unt)
close(unt)
write(*,*)'INPUT FILE Vario.dat CONTAINS DATA FOR:
write(*,910)HEADER
write(*,*)'enter type of nbhd file:'
write(*,*)'1 = opt, 2 = fixed n, 3 = lim variance:'
read(*,*)inbhd
C Nbhd_File: read nbhd parameters for nkept data points
C
if(inbhd.eq.1.or.inbhd.eq.2)then
read(15,*)nkept,duml,dum2,dum3,iest,ierr,
    + minptrs,maxptrs,npts
else
read(15,*)nkept,duml,dum2,dum3,iest,ierr,
    + minptrs,maxptrs,varlm
endif
if(ndata.ne.nkept) then
write(*,*)'ndata not equal to nkept check it out'
endif
40   j=1,nkept
read(15,*)duml, gflag(j), apt(j), bpt(j),cpt(j)
if(duml.ne.j) then
write(*,*)'nbhd number not matching data point'
stop
else
read(15,*)vpts(j), valnbhd(j), varnbhd(j)
read(15,*)rad(1,j),rad(2,j),rad(3,j)
read(15,*)drc(1,1,j),drc(1,2,j),drc(1,3,j)
read(15,*)drc(2,1,j),drc(2,2,j),drc(2,3,j)
read(15,*)drc(3,1,j),drc(3,2,j),drc(3,3,j)
endif
continue
write(*,*)'Min and Max variable limits:
write(*,*)Xcmax,Xcmin,Ycmax,Ycmin,Ecmax,Ecmin
C plot location map
open (unit=un13, File=Plot_File, status=
+ 'unknown', err=9999)
    Plot_ID=1
    two=2
    write(13,* ) Plot_ID, ndata,two,
+ (xdata(i),ydata(i), i=1,nkept)
close(13)
call read_array(Plot_file)
call vario_twod (Plot_id)
open (unit=un12, File=Output_File, status=
+ 'unknown', err=9999)
open (unit=un13, File=Results_File, status=
+ 'unknown', err=9999)
write(*,*)'enter 1 for transect, 2 for x-y section,'
write(*,*)' 3 for x-y-z block, 4 for file input'
read(*,*)idim
if(idim.eq.4)then
    write(*,*)'enter 0 for gridding only,'
    write(*,*)' 1 for xvalidation of ref. pts'
    read(*,*)ixvalid
else
    write(*,*)'enter two end points of:'
    write(*,*)'transect, section, 3-D block '
    write(*,*)' use zeros if lower than 3-D space'
    read(*,*)x1,y1,z1
    read(*,*)x2,y2,z2
endif
if(idim.eq.1)then
    write(*,*)'enter number of pts in transect:
    read(*,*)tpts
elseif(idim.eq.2)then
    if (datadim.gt.2)then
        write(*,*)'enter z level of x-y section'
        read(*,*)z0
    else
        z0=0.0
    endif
    write(*,*)'enter number of pts in x direction'
    read(*,*)xpts
    write(*,*)'enter number of pts in y direction'
    read(*,*)ypts
    zpts=1
elseif(idim.eq.3)then
   write(*,*)'enter number of intervals in x direct.'
   read(*,*)xpts
   write(*,*)'enter number of intervals in y direct.'
   read(*,*)ypts
   write(*,*)'enter number of intervals in z direct.'
   read(*,*)zpts
endif
write(*,*)'Enter 1 for standard nbhd,
  + 2 for nearest data nbhd:'
read(*,*)igrid
if (igrid.eq.1)then
   write(*,*)'enter standard parmaeters a0,b0,theta:'
   read(*,*)SrMajor,SrMinor,SrAngle
endif
910 Format(A80)
if(idim.eq.1)then
   C transect can be through 3-D space
   mx=tpts+1
   my=1
   mz=1
   delmx=(x2-x1)/float(tpts)
   delmy=(y2-y1)/float(tpts)
   if(datadim.gt.2) then
      delmz=(z2-z1)/float(tpts)
   endif
else
   mx=xpts+1
   my=ypts+1
   mz=zpts+1
   delmx=(x2-x1)/float(xpts)
   delmy=(y2-y1)/float(ypts)
   if(mz.eq.1) then
      delmz=0.0
   else
      delmz=(z2-z1)/float(zpts)
   endif
endif
C now calculate grid pts and interpolate nbhd
C parameters for each grid pt
C node centered grid pts
if(idim.eq.4)then
   open (unit=un14, File=Input_File, status=
      + 'unknown', err=9999)
   read(14,*),mx
   my=1
   mz=1
   x1=0.0
y1=0.0  
z1=0.0  
delmx=0.0  
delmy=0.0  
delmz=0.0  
endif  
npts=mx*my*mz  
write(13,*)npts  
do 200 m=1,mz  
zcen=z1+float(m-1)*delmz  
do 190 l=1,my  
ycen=y1+float(l-1)*delmy  
do 180 k=1,mx  
if(idim.eq.4)then  
if(ixvalid.eq.0)then  
if(datadim.eq.1)then  
read(14,*)xcen  
elseif(datadim.eq.2)then  
read(14,*)xcen,ycen  
elseif(datadim.eq.3)then  
read(14,*)xcen,ycen,zcen  
endif  
else  
if(datadim.eq.1)then  
read(14,*)xcen,v0  
elseif(datadim.eq.2)then  
read(14,*)xcen,ycen,v0  
elseif(datadim.eq.3)then  
read(14,*)xcen,ycen,zcen,v0  
endif  
endif  
else  
if(idim.eq.1)then  
ycen=y1+float(k-1)*delmy  
zcen=z1+float(k-1)*delmz  
endif  
xcen=x1+float(k-1)*delmx  
endif  
c find nearest data point  
dist0=1.e31  
do 150 i=1,nkept  
arg=(xdata(i)-xcen)**2+(ydata(i)-ycen)**2  
+ (edata(i)-zcen)**2  
if(arg.lt.dist0)then  
dist0=arg  
nptr=i  
endif  
150 continue  
if(ixvalid.eq.0)then
SUBROUTINE Read_Data(unt)

C - GEOEAS S.R. Reads the header data, the variable names
C - and units, and the data values into memory
C - from the data or pair file. The coordinate
C - values are stored into memory, and the minimum
C - and maximum values are determined.
C - NOTE: include file depends on parent routine
C - file unit is unt

integer i,j,k,toggle,unt
real dx,dy,de
include 'filename.inc'
C Read identifier to check for a valid file
read(unt, 990, err=9999, end=999) Header
write(*,990) Header
if (Header .NE. ' PREVAR') then
   close(unt)
   write(*,*) 'Invalid file'
   stop
endif
C Read header information
read (unt, 990, end= 999, ERR = 9999) Header
write(*,990) Header
read (unt,*,ERR = 9999,end=999) Xvar, Yvar,
   Evar, XCmin, XCmax, YCmin, YCmax, ECmin,
   ECmax, MinDist, MaxDist
C write number of: variables, data, kept points, pairs
read (unt,*,ERR = 9999,end=999) Nvar, DataDim, Ndata, Nkept, Npair
write(unt,*)'Nvar,DataDim,Ndata,Nkept,Npair:', Nvar, DataDim, Ndata, Nkept, Npair
C error check
if (Nvar .LT. 2) then
  write(*,*) 'Too few variables in file'
  GOTO 9999
else if (Nvar .GT. MAXVAR) then
  write(*,*) 'Too many variables in file'
  GOTO 9999
endif
C read variable names and units for each variable in file
DO 20 I = 1, Nvar
  read(unt,995,ERR = 9999,end=999) VarNam(I), Units(I)
20  continue
C read coordinate data into memory
DO 30 J = 1, Ndata
  read (unt,*, end=35, ERR =9999) (DataRec(I), I=I,Nvar)
C compute minimums and maximums and store data values
  toggle=1
  if(DataRec(Xvar).LT. Missing)then
    if(datadim.gt.1)then
      if( DataRec(Yvar).LT. Missing)then
        if(datadim.gt.2)then
          if(DataRec(Evar).LT. Missing) then
            Nrecord=Nrecord+1
            Xdata(Nrecord)=DataRec(Xvar)
            if(XCmin.GT.Xdata(Nrecord))
              XCmin=Xdata(Nrecord)
            +
            if(XCmax.LE.Xdata(Nrecord))
              XCmax=Xdata(Nrecord)
            +
            Ydata(Nrecord)=DataRec(Yvar)
            if(YCmin.GT.Ydata(Nrecord))
              YCmin=Ydata(Nrecord)
            +
            if(YCmax.LE.Ydata(Nrecord))
              YCmax=Ydata(Nrecord)
            +
            Edata(Nrecord)=DataRec(Evar)
            if(ECmin.GT.Edata(Nrecord))
              ECmin=Edata(Nrecord)
            +
            if(ECmax.LE.Edata(Nrecord))
              ECmax=Edata(Nrecord)
            +
          else
            NMiss=NMiss + 1
            toggle=0
          endif
        endif
      endif
    endif
  endif
endif
else
    Nrecord=Nrecord+1
    Xdata(Nrecord)=DataRec(Xvar)
    if(XCmin.GT.Xdata(Nrecord))
        XCmin=Xdata(Nrecord)
    +
    if(XCmax.LE.Xdata(Nrecord))
        XCmax=Xdata(Nrecord)
    +
    Ydata(Nrecord)=DataRec(Yvar)
    if(YCmin.GT.Ydata(Nrecord))
        YCmin=Ydata(Nrecord)
    +
    if(YCmax.LE.Ydata(Nrecord))
        YCmax=Ydata(Nrecord)
    +
    Edata(Nrecord)=0.0
endif
else
    NMiss=NMiss + 1
    toggle=0
endif
else
    Nrecord=Nrecord+1

    Xdata(Nrecord)=DataRec(Xvar)
    if (XCmin .GT. Xdata(Nrecord))
        XCmin = Xdata(Nrecord)
    +
    if (XCmax .LE. Xdata(Nrecord))
        XCmax = Xdata(Nrecord)
    +
    Ydata(Nrecord)=0.0
    Edata(Nrecord)=0.0
endif
else
    NMiss=NMiss + 1
    toggle=0
endif
if(toggle.eq.1)then
    DO 25 I = 1, NVar-datadim
        k=datadim+I
        Data(Nrecord, I) = DataRec(k)
    continue
25
endif
30
continue
C error check
35 if (Nrecord .LE. 1) then
    write(*,*)'ERROR ZERO DATA POINTS'
    close(unt)
    GO TO 9999
endif
C compute extreme distances
if(datadim.lt.2)YCmax=0.0
if(datadim.lt.2)YCmin=0.0
if (datadim .lt. 3) Ecmax = 0.0
if (datadim .lt. 3) Ecmin = 0.0
Dx = XCmax - XCmin
Dy = YCmax - YCmin
De = ECmax - ECmin
if (Dx .LE. 0.) then
    write(*,*) 'ZERO X DISTANCE'
    GOTO 9999
endif
MinDist = 0.0
MaxDist = SQRT(Dx*Dx + Dy*Dy + De*De)
write(*,*) 'MAX AND MIN VARIABLE LIMITS:'
write(*,*) XCMAX, XCMIN, YCMAX, YCMIN, ECMAX, ECMIN
go to 1000
990 FORMAT(A80)
995 FORMAT(2A10)
999 write(*,*) 'unexpected end encountered in file'
go to 1000
9999 write(*,*) 'error reading data from file'
1000 return
end

SUBROUTINE read_Pairs(unt)
C GEOEAS S.R.
C include file must match that of read_data
C file unit is unt

integer unt
include 'filename.inc'
C Read pair comparison data
read(unt,*,end=999,ERR=9999)(From(I), I=1,Npair)
read(unt,*,end=999,ERR=9999)(To(I), I=1,Npair)
read(unt,*,end=999,ERR=9999)(Distance(I), I=1,Npair)
read (unt,*, end= 999, ERR =9999) ((Vector(I,J),
+ I = 1,DataDim), J = 1, Npair)
go to 1000
900 Format(A10)
910 Format(I3,A10)
9999 continue
write(*,*) 'Error reading Pair comparison file'
stop
999 continue
1000 continue
if (Ndata .GT. MAXDATA) then
    write(*,*) 'Excess data in pair compar. file'
+ ,Ndata,MAXDATA
    stop
else if (Npair .GT. MAXPAIR) then
C - GEOEAS utility S.R.
C - Sorts the first N values in the array A into increasing order. Also re-arranges the array P in parallel with A.
C - When the sort is complete P(i) will contain the original position of array element A(i).

INTEGER N, M, I1, J1, I, J, K, L, I2
INTEGER P(N), TP, TP1, LOWER, UPPER
REAL*4 A, T, T1
DIMENSION LOWER(500), UPPER(500), A(N)
M = 1
I1 = 1
J1 = N
I = I1
J = J1
10 IF (I .GE. J) GOTO 110
20 K = I
I2 = INT((J + I) / 2)
T = A(I2)
TP = P(I2)
IF (A(I) .LE. T) GOTO 30
A(I2) = A(I)
P(I2) = P(I)
A(I) = T
P(I) = TP
T = A(I2)
TP = P(I2)
30 L = J
IF (A(J) .GE. T) GOTO 40
A(I2) = A(J)
P(I2) = P(J)
A(J) = T
P(J) = TP
T = A(I2)
TP = P(I2)
IF (A(I) .LE. T) GOTO 40
A(I2) = A(I)
P(I2) = P(I)
A(I) = T
P(I) = TP
T = A(I2)
TP = P(I2)

40
L = L - 1
IF (A(L) .GT. T) GOTO 40
T1 = A(L)
TP1 = P(L)

50
K = K + 1
IF (A(K) .LT. T) GOTO 50
IF (K .GT. L) GOTO 60
A(L) = A(K)
P(L) = P(K)
A(K) = T1
P(K) = TP1
GOTO 40

60
IF (L-I .LE. J-K) GOTO 70
LOWER(M) = I
UPPER(M) = L
I = K
M = M+1
GOTO 80

70
LOWER(M) = K
UPPER(M) = J
J = L
M = M + 1

80
IF (J-I .GE. I1) GOTO 20
IF (I .EQ. I1) GOTO 10
I = I - 1
90
I = I + 1
IF (I .EQ. J) GOTO 110
T = A(I+1)
TP = P(I+1)
IF (A(I) .LE. T) GOTO 90
K = I

100
A(K+1) = A(K)
P(K+1) = P(K)
K = K - 1
IF (T .LT. A(K)) GOTO 100
A(K+1) = T
P(K+1) = TP
GOTO 90

110
M = M - 1
IF (M .EQ. 0) GOTO 999
REAL*4 FUNCTION RMAX(A, N)
C RMAX - Return the maximum value for the first N elements
C of the array A.
INTEGER I, N
REAL*4 A, T
DIMENSION A(N)
T = A(1)
DO 10 I = 2, N
10 T = AMAX1(A(I), T)
RMAX = T
RETURN
END

REAL*4 FUNCTION RMIN(A, N)
C - Return the minimum value for the
C - first N elements of the array A.
INTEGER I, N
REAL*4 A, T
DIMENSION A(N)
T = A(1)
DO 10 I = 2, N
10 T = AMIN1(A(I), T)
RMIN = T
RETURN
END

SUBROUTINE VRank(A, N)
C -GEOEAS utility S.R.
C - Uses a shell sort to sort A into ascending order
INTEGER N, I, J, M, K, L
REAL*4 A(N), B
M = N
20 M = M / 2
   IF (M) 30, 40, 30
30 K = N - M
    J = 1
41 I = J
49 L = I + M
    IF (A(I)-A(L)) 60, 60, 50
50 B = A(I)
    A(I) = A(L)
A(L) = B
I = I - M
IF (I-1) 60, 49, 49
60 J = J + 1
IF (J - K) 41, 41, 20
40 RETURN
END

REAL FUNCTION DOT ( A, B )
C - Finds the dot product of 2 3-vectors (FIMAD)

REAL A (3), B (3)
DOT = A(1)*B(1) + A(2)*B(2) + A(3)*B(3)
RETURN
END

SUBROUTINE CROSS ( A, B, C)
C - Finds the cross product of 2 3-vectors (FIMAD)

REAL A (3), B (3), C(3)
C(1)= A(2)*B(3)-A(3)*B(2)
C(2)=-A(1)*B(3)+A(3)*B(1)
C(3)= A(1)*B(2)-A(2)*B(1)
RETURN
END

REAL FUNCTION VABS ( A )
C - Finds the absolute length of a 3-vector (FIMAD)

REAL A (3)
VABS = SQRT ( A(1)*A(1) + A(2)*A(2) + A(3)*A(3) )
RETURN
END

SUBROUTINE VSearchB(VECTOR, N, TARGET, INDX)
C GEOEAS S.R. - Binary search of the first N elements
C in the pre-sorted array VECTOR for a value <= TARGET
C and returns the position of this element in INDX.
C If INDX is returned as zero, the search failed.
C corrected by M.M.Moody Sept, 1992

INTEGER N, INDX, lower, upper, middle
REAL*4 VECTOR(N), TARGET

IF (TARGET .LT. VECTOR(1)) THEN
INDX = 0
RETURN
C
ELSE IF (TARGET .GE. VECTOR(n)) THEN **** GE ??
ELSE IF (TARGET .gt. VECTOR(n)) THEN
INDX = n+1
RETURN
ENDIF
lower = 1
upper = n
10 middle = (upper + lower) / 2

IF (VECTOR(middle) .GT. TARGET) THEN
upper = middle
ELSE IF (VECTOR(middle) .LT. TARGET) THEN
lower = middle
ELSE IF (VECTOR(middle) .EQ. TARGET) THEN
INDX = middle
RETURN
ENDIF

IF (lower+1 .GE. upper) THEN
INDX = lower
RETURN
ENDIF
GOTO 10
END

SUBROUTINE kernel(x0,y0,e0,nex)
C calculates estimate at pt (x0,y0,e0) using exponential
C kernel. nex is the point excluded in xvalidation or it
C is 0. n is the no, of pts used in calc.
C Written by M.M.Moody, sept, 1992

real fn, d, xO, yO, eO, sumfn, arg
integer i, n, nex, nume
include 'geostat.inc'

nume=numeestimate

estimate(nume)=0.0

sumfn=0.0

do 100 i=1,ndata
if(i.eq.nex)go to 100

d=(xdata(i)-x0)**2 + (ydata(i)-y0)**2
+ (edata(i)-e0)**2

d=sqrt(d)

arg=lambda*d

100
if(arg.lt.4.0)then
   fn=exp(-3.0*arg)
   n=n+1
   sumfn=sumfn+fn
endif
100 continue
if(n.gt.0)then
   estimate(nume)=estimate(nume)/sumfn
else
   call nnbr_est(x0,y0,e0,nex)
   n=1
endif
if(nex.eq.0)then
   write(14,*)x0,y0,e0,estimate(nume),n
else
   write(17,*)x0,y0,e0,data(nume,kdvar),
                      estimate(nume),n
endif
return
end

subroutine filter(mp,n,nex,lamb)
  c
  filters data with kernel functions
  c
  specified by the user
  c
  include 'geostat.inc'
  integer i,mp,n,nex,lamb,nz
  real fn,dn,fk
  c
  parameter (mx=20,mmx=21)
  fn=0.0
  dn=0.0
  nz=0
  c
  write(un12,*)' nex,smooth,resid:'
do 10 i=1,n
   call kernel(i,nex,lamb,fk,nz)
   fn=fn+fk*Data(i,mp)
    dn=dn+fk
10 continue
  Smooth(nex,kdvar)=fn/dn
  Resid(nex,kdvar)=Data(nex,kdvar)-Smooth(nex,kdvar)
  write(un12,*)nex,Smooth(nex,kdvar),Resid(nex,kdvar)
  c
  return
  end
  c
  subroutine estimate_point(mp,n,nex,lamb,est)
c estimates data with kernel functions and smoothers
specified by the user
c include 'geostat.inc'
integer i,mp,n,nex,lamb,nz
real fn,dn,fk,est
c parameter (mx=20,mmx=21)
  fn=0.0
  dn=0.0
  nz=0

do 10 i=1,n
call kernel(i,nex,lamb,fk,nz)
fn=fn+fksdata(i,mp)
   dn=dn+fks
10 continue
est=fn/dn
write(un12,*)'nex,estimate:',nex,est
write(*,*)'nex,fn,dn,estimate:',nex,fn,dn,est
return
end
subroutine kernel(i,nex,lamb,fk,nz)
c choose and evaluate kernel function for data pairs
where x is the pivot point and xi(i) are the rest of
the data points. xp is x trans/rotated.
c include 'geostat.inc'
integer nex,i,nz
real fk,lamb,arg
if(ifk.eq.0)then
5 write(*,*)'Input number (1-7) for desired kernel'
write(*,*)' 1: normal density function /2,D'
write(*,*)' 2: uniform density function w/2,3-D'
write(*,*)' 3: exponential function /2,3-D'
write(*,*)' 4: exponential decay function,s'
write(*,*)' 5: uniform, circle density function'
write(*,*)' 6: normal density function, radial'
write(*,*)' 7: inverse delta fn., rad. coords w/2
read(*,*)ifk
if(ifk.lt.1.or.ifk.gt.7) go to 5
if(ifk.eq.1.or.ifk.eq.2) then
  if(ifk.eq.1) then
10 write(*,*)' Input sigma(x) and sigma(y)'
read(*,*)a0,b0
if(a0.le.0.0.or.b0.le.0.0) go to 10
elseif(ifk.eq.2) then
  20 write(*,*)'input 1/2 length and 1/2 width of rect.'
  read(*,*)a0,b0
  if(a0.le.0.0.or.b0.le.0.0) go to 20
endif
  25 write(*,*)' Input (initial) theta (-90 to +90)'
  read(*,*)theta
  if(theta.lt.-90.0.or.theta.gt.90.0) go to 25
elseif(ifk.eq.3.or.ifk.eq.4) then
  30 write(*,*)'Input (initial) param [-x/a] (.001-100)'
  read(*,*) a0
  if(a0.lt.1.e-3.or.a0.gt.1.e2) go to 30
  35 write(*,*)'Input (initial) param [-y/b] (.001-100)'
  read(*,*) b0
  if(b0.lt.1.e-3.or.b0.gt.1.e2) go to 35
  37 write(*,*)'Input (initial) param [-z/c] (.001-100)'
  read(*,*) c0
  if(c0.lt.1.e-3.or.c0.gt.1.e2) go to 37
endif
elseif(ifk.eq.5) then
  40 write(*,*)' Input radius of circle (.001-1000)'
  read(*,*) a0
  if(a0.lt.1.e-3.or.a0.gt.1.e3) go to 40
elseif(ifk.eq.6) then
  50 write(*,*)' Input sigma '
  read(*,*) a0
  if(a0.le.0.0) go to 50
elseif(ifk.eq.7) then
  60 write(*,*)' Input denominator parameter a (.001-1000)'
  read(*,*) a0
  if(a0.lt.1.e-3.or.a0.gt.1.e3) go to 60
  65 write(*,*)' Input (initial) exponent epsilon (0-8)'
  read(*,*) eps
  if(eps.lt.0.0.or.eps.gt.8.0) go to 65
endif
asq=a0*a0
bsq=b0*b0
if ifk gt 0 calculate kernel value
else
    asq=a0*a0
    bsq=b0*b0
    csq=c0*c0
    xi=xdata(nex)
    yi=ydata(nex)
    zi=edata(nex)
    rsq=(xi-xdata(i))*(xi-xdata(i))
    if(datadim.gt.1) rsq=rsq+(yi-ydata(i))*(yi-ydata(i))
    if(datadim.gt.2) rsq=rsq+(zi-edata(i))*(zi-edata(i))

    coord translation and rotation
    xp= aaxis(1)*(xi-xdata(i))
    + aaxis(2)*(yi-ydata(i))
    + aaxis(3)*(zi-edata(i))
    yp= baxis(1)*(xi-xdata(i))
    + baxis(2)*(yi-ydata(i))
    + baxis(3)*(zi-edata(i))
    zp= caxis(1)*(xi-xdata(i))
    + caxis(2)*(yi-ydata(i))
    + caxis(3)*(zi-edata(i))
    endif

    if(ifk.eq.1) then
        arg=(xi-xp)*(xi-xp)/asq
        if(datadim.gt.1) arg=arg+(yi-yp)*(yi-yp)/bsq
        arg=arg/(2.0*lamb)
        if(arg.lt.30) then
            fk=exp(-arg)
            nz=nz+l
        else
            fk=0.0
        endif
    endif

    elseif(ifk.eq.2) then
        if(abs(xp).le.a0.and.abs(yp).le.b0) then
            fk=1.0/(4.d0*a0*b0)
            nz=nz+l
        else
            fk=0.0
        endif
    endif

    elseif(ifk.eq.3) then
        arg=(xi-xp)*(xi-xp)/a0+(yi-yp)*(yi-yp)/b0
        if(datadim.gt.2) arg=arg+(zi-zp)*(zi-zp)/c0
        arg=sqrt(arg)
        if(arg.lt.30) then
            fk=exp(-arg)
            nz=nz+l
        else
            fk=0.0
        endif
fk = exp(-arg)
nz = nz + 1
else
fk = 0.0
endif

c
elseif(ifk.eq.4) then
arg = sqrt(rsq)/a0
if(arg.lt.30.) then
fk = exp(-arg)
nz = nz + 1
else
fk = 0.0
endif

c
elseif(ifk.eq.5) then
if(rsq.le.asq) then
fk = 1.0
nz = nz + 1
else
fk = 0.0
endif

c
elseif(ifk.eq.6) then
arg = rsq/asq
if(arg.lt.30.) then
fk = exp(-arg)
nz = nz + 1
else
fk = 0.0
endif

c
elseif(ifk.eq.7) then
arg = rsq + asq
fk = 1.0/arg**(1.0+eps)
nz = nz + 1
endif
return
end
REFERENCES


Soc Ser B 34, 385-392.


