

Utah State University

DigitalCommons@USU

All Graduate Theses and Dissertations

Graduate Studies

5-2009

Simulation, Kriging, and Visualization of Circular-Spatial Data

William James Morphet
Utah State University

Follow this and additional works at: <https://digitalcommons.usu.edu/etd>



Part of the [Mathematics Commons](#)

Recommended Citation

Morphet, William James, "Simulation, Kriging, and Visualization of Circular-Spatial Data" (2009). *All Graduate Theses and Dissertations*. 386.
<https://digitalcommons.usu.edu/etd/386>

This Dissertation is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.



SIMULATION, KRIGING, AND VISUALIZATION
OF CIRCULAR-SPATIAL DATA

by

William J. Morphet

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mathematical Sciences

Approved:

Jürgen Symanzik
Major Professor

Adele Cutler
Committee Member

Donald H. Cooley
Committee Member

John R. Stevens
Committee Member

Daniel C. Coster
Committee Member

Byron Burnham
Dean of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2009

Copyright © William Morphet 2009

All Rights Reserved

ABSTRACT

Visualization, Kriging, and Simulation
of Circular-Spatial Data

by

William J. Morphet, Doctor of Philosophy
Utah State University, 2009Major Professor: Dr. Jürgen Symanzik
Department: Mathematics and Statistics

The circular dataimage is defined by displaying direction as the color at the same direction in a color wheel composed of a sequence of two-color gradients with color continuity between gradients. The resulting image of circular-spatial data is continuous with high resolution. Examples include ocean wind direction, Earth's main magnetic field, and rocket nozzle internal combustion flow. The cosineogram is defined as the mean cosine of the angle between random components of direction as a function of distance between observation locations. It expresses the spatial correlation of circular-spatial data. A circular kriging solution is developed based on a model fitted to the cosineogram. A method for simulating circular random fields is given based on a transformation of a Gaussian random field. It is adaptable to any continuous probability distribution. Circular random fields were implemented for selected circular probability distributions. An R software package was created with functions and documentation.

(391 pages)

ACKNOWLEDGMENTS

Study of the dataimage of Minotte and West (1998), for the imaging of ordered multiple linear variables and observational units, motivated the problem of how to display directional-spatial data as an image, and what this method should be called. I am thankful for the helpful discussions with Dr. Mike Minnotte regarding color wheels. I would also like to thank the JCGS associate editor and referee for their helpful suggestions regarding circular dataimages. I acknowledge and express thanks to ATK Launch Systems, Inc., and NASA for authorizations to display model flow in the Space Shuttle solid rocket motor nozzle in Figure 2-17 (b) and to display nozzle vectoring data in Figure 2-18, and to Technical Artist Alan Eaton for Figure 2-19 (Property number A045477a, ATK Launch Systems, Inc., copyright © 2005) showing the Space Shuttle roll maneuver with left/right solid rocket motor labels added.

Most of the figures were generated in R, versions 2.8.0 (R Development Core Team 2008). R was originally created by Ihaka and Gentleman (1996) and is now a collaborative worldwide effort. The binary distributions of R and R contributor packages are freely downloadable from <http://www.r-project.org/>, and are supported on Windows (NT, 95 and later) and in some versions for other operating systems. Figure 1-4 was produced using functions of R package Fields (Furrer, Nychka, and Sain 2009), software for simulation of random fields. Figure 1-5, Appendix Figure N-1 (b), and Figure N-2 (a) were produced using functions of R package CircStats (Lund and Agostinelli 2007), software for circular statistics, and manually enhanced. The functions of CircStats were used extensively in the codes written for this dissertation. Figures N-2 (b) and (c) were produced using a demo version of Oriana 2, software for the analysis and display of circular data (Kovach Computing 2004), and manually enhanced. Oriana is available at <http://www.kovcomp.co.uk/oriana/oribroc.html>. Oriana is supported on Microsoft

Windows 98/Windows NT 4 or later, including Windows ME/2000/XP/2003/Vista.

Figures 2-15, 2-16, and N-4 were constructed using functions of R package RGL (Adler 2009), software for 3-D real time visualization. Other R packages used extensively include geoR (Ribeiro and Diggle 2001), and RandomFields (Schlather 2001).

I would like to acknowledge and thank:

ATK Launch Systems, Inc.:

- Dr. Suresh Kulkarni – Provided critical endorsement following rejection of application for the master's program of study.
- Management and supervision – Approved adjusted working hours to accommodate class schedules.

Utah State University professors:

- Dr. Donald Cooley – Head of Computer Science and defense chairman - Suggested programming to have some fun when needing a break.
- Dr. Dan Coster – Provided a listening ear and encouragement.
- Dr. Adele Cutler – Provided a critical evaluation leading to a productive topic of research and a thorough critique of my dissertation.
- Dr. Richard Cutler – Defended me and encouraged high GPA.
- Dr. Robert Heal – Provided a listening ear and encouragement.
- Dr. John Stevens – Suggested important improvements to the dissertation.
- Dr. Jürgen Symanzik, advisor – Provided guidance, persuaded me to remain in the doctoral program of study following withdrawals, and provided critical review.
- Dr. Russell Thompson - Was highly supportive as department head (without him, one withdrawal would have been terminal).

Others:

- Kenneth Johnson, NASA/ NESC Systems Engineering Office – For review of Chapter 1 of the dissertation.
- Terry, wife and mother of 5 children – Sacrificed many hours of companionship and performed many supportive efforts.

William J. Morphet

CONTENTS

	Page
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	xii
LIST OF FIGURES.....	xiii
SYMBOLS, TERMINOLOGY, ACRONYMS.....	xxi
CHAPTERS	
1. INTRODUCTION	
1.1 Introduction to the Circular Random Field and Circular Random Variables.....	1
1.2 A Motivational Example	7
1.3 Problem Description	7
1.4 Literature Review.....	9
1.5 Dissertation Overview.....	15
2. CIRCULAR DATAIMAGE, A HIGH RESOLUTION CONTINUOUS IMAGE OF CIRCULAR-SPATIAL DATA	
2.1 Introduction.....	19
2.2 Overview of Vectorial-Spatial Displays	20
2.3 Cross Over.....	24
2.4 The Circular Dataimage and Color Wheel.....	25
2.5 Comparison of Methods.....	27
2.6 Calculation of a BGYR Color Wheel.....	32
2.7 Color Considerations and Variations	33
2.8 Other Examples.....	43
2.9 Chapter Summary and Future Work.....	50
3. COSINEOGRAM, A MEASURE OF CIRCULAR-SPATIAL CORRELATION	
3.1 Introduction.....	51
3.2 The Cosineogram	52
3.3 Derivation of the Sill.....	54
3.4 Expectation of the Cosines.....	59
3.5 Verification of the Sill by Simulation.....	63
3.6 Cosine Models.....	67
3.7 Cosineogram of Ocean Wind in a South Polar Region.....	73
3.8 Chapter Summary and Future Work.....	75

CONTENTS

	Page
4. CIRCULAR KRIGING	
4.1 Introduction	78
4.2 Solution	79
4.3 Verification of Optimality	88
4.4 Computationally Efficient Formula	90
4.5 Kriging Behavior Around a Sampled Location	91
4.6 Circular Kriging Variance, σ_{CK}^2	94
4.7 How Distance and the Cosine Model Affect $\hat{\sigma}_{CK}^2$	98
4.8 Chapter Summary and Future Work	101
5. SIMULATION OF CIRCULAR RANDOM FIELDS	
5.1 Introduction	103
5.2 Background	104
5.3 New Method of Generating a CRF	108
5.4 Mathematical Properties of the CRF	112
5.5 Qualitative Evaluations of Method of Simulating a CRF	124
5.6 Extension of the Method	128
5.7 Chapter Summary and Future Work	129
6. COMPREHENSIVE EXAMPLE	
6.1 Outline of Circular-Spatial Processes	131
6.2 Simulation of a CRF	131
6.3 Estimation of the Spatial Trend	133
6.4 Computation of the Residuals	134
6.5 Plotting and Modeling the Cosineogram	135
6.6 Kriging the Residuals	136
6.7 Interpolation of the Trend Estimate	137
6.8 Computing The Circular-Spatial Estimate	138
6.9 Imaging the Circular-Spatial Estimate	139
6.10 Computing the Circular Kriging Variance	139
7. SUMMARY	141
CITATIONS	147
APPENDICES	151
A Notation	152

CONTENTS

	Page	
B	Linear Algebra	
B.1	Identities for Vectors	153
B.2	Some Properties of the Positive Definite Matrix \mathbf{K}	153
B.3	Theorem: The P. D. Matrix Has an Inverse	154
B.4	Theorem: The Inverse of P. D. Matrix Is Symmetric.....	154
B.5	Theorem: The Inverse of P. D. Matrix Is P. D.....	155
B.6	Some Properties of the Negative Definite Matrix.....	155
B.7	Derivatives Required for Kriging	156
B.8	The Requirements for Maximization	158
B.9	Expectation	159
C	Qualitative Evaluations of Other CRFs with Standardization	160
D	Qualitative Evaluations of CRFs Near Parameter Extremes	165
E	Derivations of the CDF Formulae for Support $[0, 2\pi)$	
E.1	Cardioid	184
E.2	Triangular.....	184
E.3	Uniform	186
E.4	Von Mises	186
E.5	Wrapped Cauchy	187
F	Verification by Evaluation of the CDF Formulae with Support $[0, 2\pi)$	
F.1	Cardioid	188
F.2	Triangular.....	188
F.3	Uniform	189
F.4	Von Mises	189
F.5	Wrapped Cauchy	190
G	Modification of the PDF and CDF Formulae for Rotated Support $[-\pi, +\pi)$	
G.1	Cardioid	192
G.2	Triangular.....	193
G.3	Uniform	194
G.4	Von Mises	194
G.5	Wrapped Cauchy	195
H	Wrapped Cauchy CDF	
H.1	Additional Forms of the CDF	196
H.2	Alternate Forms with Support $[0, 2\pi)$	198

CONTENTS

	Page
H.3	Evaluation of Alternate Forms 199
H.4	Selected Form for Rotated Support $[-\pi, +\pi)$ 201
I	Triangular Inverse CDF 202
J	R Package Documentation
J.1	Introduction and Installation 204
J.2	SimulateCRF 206
J.3	CircResidual 211
J.4	CosinePlots 214
J.5	KrigCRF 222
J.6	InterpDirection 228
J.7	TestPattern 233
J.8	OceanWind 234
J.9	WorldMask 234
J.10	CircDataimage 235
J.11	PlotVectors 244
K	R Functions
K.1	TestPattern 248
K.2	CircDataimage 249
K.3	SimulateSill 271
K.4	CorrelationTransfer 273
K.5	SimulateCRF 274
K.6	AssessCRF 278
K.7	PlotVectors 282
K.8	CircResidual 284
K.9	CosinePlots 285
K.10	KrigCRF 288
K.11	InterpDirection 291
K.12	CircMedianPolish 296
K.13	AssessStandardization 298
K.14	MakeCosineData 302
K.15	FitCosineData 305
K.16	FitOceanWind 306
K.17	3DPolarMainMagnetic 307
K.18	Circular Kriging Variance 310
L	R Command Line Input
L.1	Figures 3-5 to 3-9 311
L.2	Figure 3-13 312
L.3	Figure 5-1, Simulated CRF 314

CONTENTS

	Page
L.4 Figure 5-3, Image of GRF	314
L.5 Figure 5-4, Variogram and Inverted Cosineogram Similar	314
L.6 Figure 5-5, Standardization	315
L.7 Figure 5-6, Variability vs. ρ	315
L.8 Figure 5-8 and the Figures in Appendices C and D.....	315
L.9 Figures 6-2 to 6-11	317
L.10 Make Cosine Datasets.....	321
L.11 Figure M-1, Fitted Covariogram an Unbiased Estimator	326
L.12 Plot Figures M-2, M-3, and M-4, Families of Curves	327
L.13 Plot Figures M-6 to M-10	333
L.14 Plot Figures 4-3 and 4-4	336
 M Cosine Curves of Simulated Circular Random Fields (CRF)	
M.1 Review	338
M.2 Generation of Cosine vs. Distance Curves	338
M.3 Families of Cosine Curves.....	341
M.4 Characterization of the Cosine Curves	345
M.5 Expressions for the Cosine Models of Table M-2	348
M.6 Generalization of the Generation and Characterization of the Cosine Curves	353
 N Additional Graphics for Circular Data	
N.1 Summary Plots for Circular Data	354
N.2 Histograms for Circular Data	355
N.3 Nonparametric Density Plots for Circular Data	357
N.4 New Cylindrical Plot of the Circular Probability Density.....	358
 O Permissions	360
 CURRICULUM VITAE.....	368

LIST OF TABLES

Table		Page
2-1	BGYR Color Wheel Formulae for RGB Space	32
3-1	Circular Probability Distributions, $\mu = 0$, $0 \leq \theta < 2\pi$ Radians.....	56
3-2	The Sill of Selected Distributions.....	64
5-1	Circular Probability Distributions, $\mu = 0$, $-\pi \leq \theta < \pi$ Radians.....	111
5-2	CDFs and Inverse CDFs for Circular Distributions, $\mu = 0$, $-\pi \leq \theta < \pi$ Radians	112
D-1	Spatial Property Scores of Figures D-1 to D-16	182
J-1	Output of CosinePlots	220
J-2	Output of KrigCRF	224
J-3	Output of PlotVectors	245
M-1	Mean Resultant Vector Length ρ of Circular Distributions for Figures M-2, M-3, and M-4	339
M-2	Cosine Models Approximating CRF Cosine Curves.....	347

LIST OF FIGURES

Figure	Page
1-1 Circular PDF of the Triangular Circular Probability Distribution.....	2
1-2 The Arithmetic Mean of 180° Does Not Point in the Central Direction of 0°	6
1-3 The Effect of the Population Resultant Vector Mean Length ρ on the Sample Mean Resultant Vector (Black) of a Sample (Tan) from the von Mises Circular Distribution.....	6
1-4 Circular and Vector Spatial Data and Their Means for the Direction the Ocean Wind Blows Toward.....	8
1-5 Rose Plot of the Circular Data Derived from the Data of Figure 1-3	11
1-6 Kriging, Estimation of Spatial Data Based on Spatial Correlation	14
1-7 Flow Chart of Methods for Circular-Spatial Data.....	17
2-1 Circular Dataimage of the Direction Wind Is Blowing Toward, Coded with Yellow-Red-Green-Blue (YRGB) Color Wheel (Right)	19
2-2 Some Existing Methods for Display of Circular and Vectorial-Spatial Data Using the Smoothed Ocean Wind Data.....	23
2-3 Evolution of the YRGB Color Wheel.....	25
2-4-1 Plots of Average Ocean Wind Direction	28
2-4-2 Plots of Smoothed Ocean Wind Direction	29
2-5 Comparison of Arrow and Circular Dataimage Plots of Ocean Wind Average Direction	31
2-6 Normal and Simulated Deuteranopic Views of Images	35
2-7 Variety of Continuous and Discrete Color Wheels	36
2-8 Effects of GYRB Color Wheel and Smoothness of Data.....	38
2-9 Effects of Color Wheel Rotation, Color Wheel Labeled with Rotation.....	38
2-10 Focus Plots of Smoothed Average Direction with Focal Directions 0° (Top) and 180° (Bottom).....	40

LIST OF FIGURES

Figure	Page
2-11 Axial Focus Plots of Smoothed Average Direction with Axial Focal Directions 0° (Top) and 90° (Bottom)	41
2-12 Strength Binned by Quartiles and Coded as Value (V) in HSV Scheme	42
2-13 Circular Dataimage of Wind with Direction Coded Using HSV Color Wheel and Magnitude (m/s) Plotted as Contour Curves	43
2-14 Circular Dataimage of Earth Main Magnetic H Field Direction	44
2-15 3D Polar Plot of Earth Main Magnetic H Field Model with Direction as a Color and Magnitude as Radius for 1/1/1900, 1/1/1950, and 1/1/2000	45
2-16 Asymmetry of Earth Main Magnetic H Field Model 1/1/2000 Demonstrated by 45° Rotations about the Horizontal Axis Through 0°-180° Longitude at the Equator	45
2-17 Space Shuttle Booster, Nozzle, and Nozzle Internal Combustion Flow	46
2-18 Time Series of the Space Shuttle Booster Nozzle Direction Angle	48
2-19 Illustration of the Space Shuttle Roll Maneuver vs. Time from Ignition	49
3-1 Distance Between Locations vs. Angular Distance Between Observations	52
3-2 Features of the Cosine Model	53
3-3 Acute and Obtuse Cases of Random Directions	58
3-4 Mean Cosine of the Angle Between Independent Cardioid CRV, $\rho^2 = 0.062$, Is Consistent with the Theoretical Sill	65
3-5 Mean Cosine of the Angle Between Independent Triangular CRV, $\rho^2 = 0.041$, Is Consistent with the Theoretical Sill	65
3-6 Mean Cosine of the Angle Between Independent Uniform CRV, $\rho^2 = 0$, Is Consistent with the Theoretical Sill	66
3-7 Mean Cosine of the Angle Between Independent Von Mises CRV, $\rho^2 = 0.798$, Is Consistent with the Theoretical Sill	66

LIST OF FIGURES

Figure	Page
3-8 Mean Cosine of the Angle Between Independent Wrapped Cauchy CRV, $\rho^2 = 0.135$, Is Consistent with the Theoretical Sill.....	67
3-9 The Exponential Cosine Model	69
3-10 The Gaussian Cosine Model	70
3-11 The Spherical Cosine Model	70
3-12 Circular Dataimage of Model of Ocean Wind Direction for South Polar Region	74
3-13 Cosineocloud, Cosineogram, and Exponential Model of South Polar Ocean Wind.....	75
4-1 Circular Kriging, the Interpolation of Circular-Spatial Data Based on Spatial Correlation.....	78
4-2 Directions Represented by the Unobserved \mathbf{u}_0 , Estimate $\hat{\mathbf{u}}_0$, and Error $\mathbf{e}_0 = \hat{\mathbf{u}}_0 - \mathbf{u}_0$ Vectors	81
4-3 Effect of Cosine Model on the Kriging Estimate Around the Measurement Location.....	92
4-4 Effect of Range, Mean Resultant Length ρ , and nugget n_g on the Circular Kriging Variance $\hat{\sigma}_{CK}^2$	100
5-1 Simulated Sample of a von Mises CRF, $\rho = 0.8$, Range $r = 10$	108
5-2 Mapping a GRV to a CRV via the CDFs F_Z and G_θ	109
5-3 Simulated GRF with Spherical Covariance Model and Range $r = 10$ Corresponding to Figure 5-1.....	109
5-4 Similar Shapes of Variograms and Inverted Cosineogram Reflect Transformations of the Spatial Correlation of the GRF	116
5-5 Standardization of the GRV Increases Fit of the GRV and the CRV.....	121
5-6 Variability of Fit of the Simulated Triangular CRV Increases as ρ Decreases	123

LIST OF FIGURES

Figure	Page
5-7 Standardization of the GRV Biases GRF the Covariance	124
5-8 Evaluation of a von Mises CRF, $\rho = 0.8$, Overfit, Range $r = 10$	126
6-1 Comprehensive Example - The Trend Model, or the Underlying First Order Component of Variation.....	132
6-2 Comprehensive Example - Simulated Sample of a von Mises CRF, $\mu = 0, \rho = \sqrt{0.5}$ with Underlying Trend.....	132
6-3 Comprehensive Example - Comparison of the Trend Estimate (Tan) with the True Trend (Blue).....	133
6-4 Comprehensive Example - Enlarged View of the Data (Black), Trend Estimate (Tan), and Residual Rotation (Dashed Red) Corresponding to the Green Highlighted Area in Figures 5-9 to 5-11	134
6-5 Comprehensive Example - Points of the Cosineogram, and the Exponential, Gaussian, and Spherical Cosine Models of Circular-Spatial Correlation.	135
6-6 Comprehensive Example - Enlarged View of the Kriging (Light Grey) and the Residual Rotations (Red) Corresponding to the Green Highlighted Area in Figures 5-9 to 5-11	136
6-7 Comprehensive Example - Enlarged View of the Interpolation (Purple) of the Trend Estimate (Tan) Corresponding to the Green Highlighted Area in Figures 5-9 to 5-11	137
6-8 Comprehensive Example - Enlarged View of the Circular Spatial Data Estimate (Gold) and the Sample (Black) Corresponding to the Green Highlighted Area in Figures 5-9 to 5-11.....	138
6-9 Comprehensive Example – Circular Dataimage (Left) of the Circular Spatial Data Estimate with HSV Color Wheel (Right) of Direction.	139
6-10 Comprehensive Example - Circular Kriging Variance with Measurements on a Regular Grid.....	140
C-1 Evaluation of a Cardioid CRF, $\rho = 0.25$, Overfit, Range $r = 10$	161

LIST OF FIGURES

Figure		Page
C-2	Evaluation of a Triangular CRF, $\rho = 0.203$, Overfit, Range $r = 10$	162
C-3	Evaluation of a Uniform CRF, Overfit, Range $r = 10$	163
C-4	Evaluation of a Wrapped Cauchy CRF, $\rho = 0.5$, Overfit, Range $r = 10$	164
D-1	Evaluation of a Cardioid CRF, $\rho = 0.05$, Overfit, Range $r = 10$	166
D-2	Evaluation of a Cardioid CRF, $\rho = 0.05$, Range $r = 10$	167
D-3	Evaluation of a Cardioid CRF, $\rho = 0.475$, Overfit, Range $r = 10$	168
D-4	Evaluation of a Cardioid CRF, $\rho = 0.475$, Range $r = 10$	169
D-5	Evaluation of a Triangular CRF, $\rho = 0.05$, Overfit, Range $r = 10$	170
D-6	Evaluation of a Triangular CRF, $\rho = 0.05$, Range $r = 10$	171
D-7	Evaluation of a Triangular CRF, $\rho = 0.385$, Overfit, Range $r = 10$	172
D-8	Evaluation of a Triangular CRF, $\rho = 0.385$, Range $r = 10$	173
D-9	Evaluation of a von Mises CRF, $\rho = 0.05$, Overfit, Range $r = 10$	174
D-10	Evaluation of a von Mises CRF, $\rho = 0.05$, Range $r = 10$	175
D-11	Evaluation of a von Mises CRF, $\rho = 0.95$, Overfit, Range $r = 10$	176
D-12	Evaluation of a von Mises CRF, $\rho = 0.95$, Range $r = 10$	177
D-13	Evaluation of a Wrapped Cauchy CRF, $\rho = 0.05$, Overfit, Range $r = 10$	178
D-14	Evaluation of a Wrapped Cauchy CRF, $\rho = 0.05$, Range $r = 10$	179
D-15	Evaluation of a Wrapped Cauchy CRF, $\rho = 0.95$, Overfit, Range $r = 10$	180
D-16	Evaluation of a Wrapped Cauchy CRF, $\rho = 0.95$, Range $r = 10$	181

LIST OF FIGURES

Figure	Page
G-1 Visual Verification of Cardioid CDF, $\rho = 0.30$, Support $[-\pi, +\pi)$ Radians.....	192
G-2 Visual Verification of Triangular CDF, $\rho = 0.30$, Support $[-\pi, +\pi)$ Radians.....	193
G-3 Visual Verification of von Mises CDF, $\rho = 0.30$, Support $[-\pi, +\pi)$ Radians.....	195
H-1 Incorrect Wrapped Cauchy CDF, $\rho = 0.75$, Support $[0, 2\pi)$ Radians	196
H-2 Dataplot WCACDF of Wrapped Cauchy CDF, $\rho = 0.75$, Support $[0, 2\pi)$ Radians	197
H-3 Three Forms of the Wrapped Cauchy CDF, $\rho = 0.75$, Support $[-\pi, +\pi)$ Radians.....	199
H-4 Iterated Wrapped Cauchy CDF, $\rho = 0.95$, Support $[-\pi, \pi)$ Radians, 15 Iterations.....	200
H-5 Visual Verification of Wrapped Cauchy CDF, $\rho = 0.75$, Support $[-\pi, \pi)$ Radians.....	201
I-1 Visual Verification of Triangular Inverse CDF, $\rho = 0.95 * 4 / \pi^2$, Support $[-\pi, +\pi)$ Radians.....	203
J-1 Mapping a GRF to a CRF via CDFs.....	207
J-2 Shapes of Variograms and Inverted Cosineogram Show Spatial Correlation Transformed from the GRF with Spherical Covariance and Range $r = 10$	208
J-3 Plots of True Model, Simulated CRF, Data, Fitted Model, and Residuals.....	213
J-4 Distance Between Locations (Red) vs. Angular Distance (Grey) Between Observations	214
J-5 Features of the Cosineogram Model	215
J-6 Cosineocloud.....	215

LIST OF FIGURES

Figure		Page
J-7	Empirical Cosineogram	216
J-8	Cosine Models for Circular-Spatial Data, Range $r = 8$	218
J-9	Fitted Cosine Models.....	222
J-10	Residual Rotations (Black) Overplotted on the Circular Kriging (Tan)	224
J-11	Smoothing via the Nugget Not Effective at Data Locations.....	225
J-12	Smoothing the Kriging Components Is Effective at All Locations.....	226
J-13	Variability of the Circular Kriging Estimate with Locations on a Regular Grid	227
J-14	Variability of the Circular Kriging Estimate with Random Locations	228
J-15	Six Cases of Interpolation Location Indicated by Labeled Red Dots.....	229
J-16	Effect of Interpolation on Smoothed Average Wind Direction with BGYR Color Wheel	229
J-17	Fitted Model (Black) Overplotted on the Fitted Model Interpolation (Tan).....	231
J-18	Original Data (Black) Overplotted on the Estimates (Tan)	232
J-19	Enlargement of Figure J-16.....	233
J-20	Image Plot of WorldMask	235
J-21	Comparison of Arrow and Circular Dataimage Plots of Ocean Wind Average Direction	236
J-22	Evolution of the YRGB Color Wheel.....	238
J-23	Initial Display of the GUI, the Circular Dataimage Window (R Graphics Device 2), and the Color Wheel Window.....	239
J-24	Display with Circular Dataimage of Average Direction after Inputs Entered ..	240
J-25	GYRB Color Wheel Rotated 90°, Data Smoothed with Bandwidth 2.5, and Display Coordinates Changed (Zoomed)	242
J-26	HSV Color Wheel Rotated 90°, Data Smoothed with Bandwidth 2.5, Color Scale Gap 0.20, and Arrows on	243

LIST OF FIGURES

Figure	Page
J-27 Mask Restores Land Mass Shapes in Smoothed Data	244
J-28 Unit Vector Plot of Ocean Wind Data	246
J-29 Vector Plot of Ocean Wind Data	247
J-30 Triangle Icon Plot of Ocean Wind Data	247
M-1 Fitted Covariogram an Unbiased Estimator of Spherical Covariance	340
M-2 Family of Cosine vs. Distance Curves from the GRF with Exponential Covariance	342
M-3 Family of Cosine vs. Distance Curves from the GRF with Gaussian Covariance	343
M-4 Family of Cosine vs. Distance Curves from the GRF with Spherical Covariance	344
M-5 Whittlematern Cosine Model ($a=.493$) Approximates the Cosine Curve of the von Mises CRF, $\rho = 0.95$, Transformed from an Exponential GRF, Range=5	346
M-6 Whittlematern Cosine Models for $\rho = 0$	348
M-7 Cauchytm Cosine Models for $\rho = 0$	349
M-8 Generalized Cauchy Cosine Models for $\rho = 0$	350
M-9 Hyperbolic Cosine Models for $\rho = 0$	351
M-10 Stable Cosine Models for $\rho = 0$	352
N-1 Summary Plots of the Ocean Wind Data.....	354
N-2 Circular Histograms of the Ocean Wind Data.....	356
N-3 Kernel Density Plots of the Ocean Wind Data.....	358
N-4 New Cylindrical Plot of PDFs of von Mises Probability Densities.....	359

LIST OF SYMBOLS, TERMINOLOGY, ACRONYMS

Symbols

- σ_{CK}^2 : Circular kriging variance
 Θ : Circular random variable (CRV)
 θ : Observation (realization) or simulation of a CRV
 \mathbf{C} : Matrix of cosines of angles between observations of direction
 \mathbf{c} : Vector of cosines of angles between observations and unobserved direction to be estimated
 $\zeta(d)$: Model of the mean cosine of the angle between random components of direction as a function of distance between observation locations
 $\hat{\zeta}(d)$: Cosineogram estimate of $\zeta(d)$
 κ : Concentration parameter of the von Mises distribution
 n_g : Nugget
 κ : Population concentration about the mean direction for von Mises CRV
 μ : Population mean resultant vector direction
 ρ : Population resultant vector mean length and concentration about the mean direction
 \bar{R}_n : Sample resultant mean vector length
 \mathbf{x} : Vector of spatial coordinates

New Terminology

Circular Dataimage
 Cosineocloud
 Cosineogram
 Cosine Model
 Circular Random Field

Terminology from Linear Kriging

Covariogram
 Covariance Model
 Nugget
 Range
 Sill
 Variogram

LIST OF SYMBOLS, TERMINOLOGY, ACRONYMS

Acronyms

CDF : Cumulative Distribution Function
CRF : Circular Random Field
CRV : Circular Random Variable
CCW: Counterclockwise
GRV : Gaussian Random Variable
GRF : Gaussian Random Field
GUI : Graphical User Interface
GYRB: Green Yellow Red Blue
HSV : Hue Saturation Value
KBWR: Black Blue White Red
MAD : Mean Absolute difference
PDF : Probability Density Function
RF : Random Field
RGB : Red Green Blue
RV : Random Variable
YRGB: Yellow Red Green Blue

CHAPTER 1

INTRODUCTION

1.1 Introduction to the Circular Random Field and Circular Random Variables

This dissertation addresses related and practical aspects of the circular random field (CRF) including extracting the spatial correlation, modeling the spatial correlation, estimation, simulation, and plotting.

A random field (RF) is a stochastic process operating over a space of dimension ≥ 1 . A CRF is defined as a RF containing circular random variables (CRVs) at multiple observation locations which are spatially correlated. With Θ the CRV and \mathbf{x} the location, in 2-dimensional space, the CRF is the set $\{\Theta(\mathbf{x}), \mathbf{x} \in \mathbb{R}^2\}$. Circular-spatial correlation is defined here as the mean cosine of the angle between random components of directions (nonrandom component removed) vs. distance between observation locations. Spatial correlation increases as distance between observation locations decreases. Hence, random components of direction tend to be more similar as distance between observation locations decreases.

A CRV takes random directions with the total probability of all possible directions distributed on the circular support (unit circle, $[0, 2\pi)$, or $[-\pi, \pi)$). The starting point of the support is the same direction as the ending point. A CRV or direction is expressed as either a scalar in units of radians or degrees ($^\circ$), or as a unit vector (Chapter 4). Since trigonometric functions require angles in radian units, the input for functions of direction will be expressed in radian units with values in $[0, 2\pi)$ until Chapter 5, where a new method requires values in the equivalent support of $[-\pi, +\pi)$ radians. Maps and compasses will use $^\circ$ units, which may be obtained from radian units by multiplying by

$180^\circ/\pi$. On a circle, the 0s of the $[0, 2\pi)$ radians, the $[-\pi, +\pi)$ radians, and the $[0, 360)^\circ$ scales are aligned. 0 radians, 0° , and the east direction will be aligned to 3 o'clock. 90° , $\pi/2$ radians, and the north direction will be aligned to 12 o'clock. π radians, 180° , and the west direction will be aligned to 9 o'clock. These scales of direction or angle are shown in Figure 1-1. Figure 1-1 is a typical plot of the probability density function (PDF) of the triangular CRV (density increases linearly toward the maximum density at 0).

Other types of directional random variables include the spherical, axial, and vectorial random variables. A spherical random variable takes random locations on a unit sphere. An axial random variable takes random axis orientations in a plane where there is no reason to distinguish a direction from its opposite. A vectorial random variable has both random direction and random magnitude. Hence, random fields may also be defined for axial, vectorial, and spherical random variables.

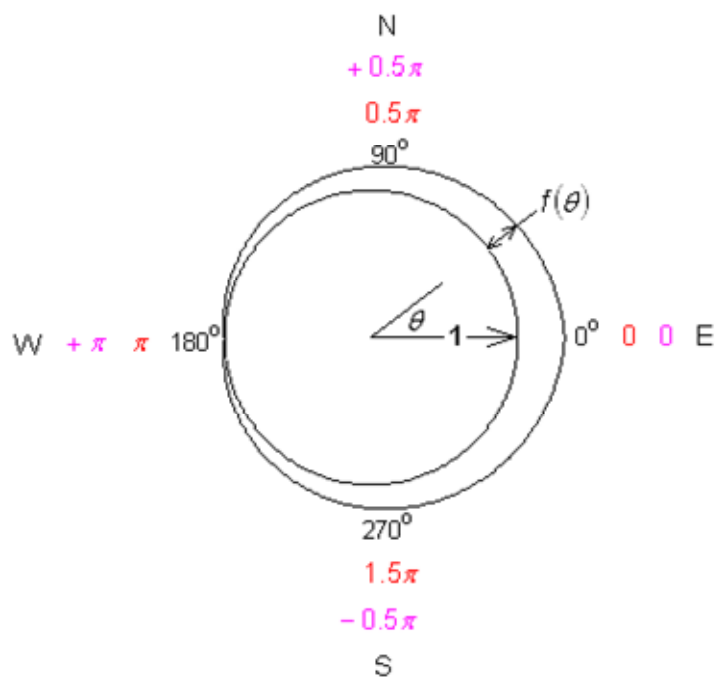


Figure 1-1. Circular PDF of the Triangular Circular Probability Distribution. The density $f(\theta)$ is often plotted on the outside of a unit circle.

Applications of circular random variables and circular random fields include:

- Astronomy - Planet orbit inclination
- Biology - Creature migration and navigation by sun, wind, magnetic fields, etc
- Chronotherapeutics - Response to a treatment relative to the time of treatment
- Geography - Compass directions
- Geology - Crystal and fault orientation
- Geophysics - Magnetic field direction
- Meteorology - Wind direction
- Oceanography - Ocean currents
- Periodic phenomena - Births/month, deaths/month, eggs produced/month, coats sold/month, accidents per hour, accidents per month, sunspots/year, biorhythms
- Paleomagnetism – Direction of magnetism locked into lava
- Physics – Dihedral (having or formed by two planes) angles in molecules
- Rounding errors – Integer atomic weights
- Structural Geology - Fracture pattern in a region

This dissertation will treat the cardioid, triangular, uniform, von Mises, and wrapped Cauchy circular distributions in alphabetical order in all sections:

- The cardioid distribution models the direction marbles roll off when dropped on a plane inclined to the horizontal. A 0° inclination of the plane produces the circular uniform PDF.
- The triangular distribution has a PDF that increases linearly toward mean direction.
- The uniform distribution models an honest roulette wheel and provides the null model to test the alternatives of unimodal (a single cluster of directions in the data) and multimodal distributions (two or more clusters in the data).

- The von Mises distribution is practically interchangeable with the wrapped normal circular probability distribution. The “wrapped normal” distribution is obtained by “wrapping” the tails of the normal PDF around a unit circle in opposite directions. The probability density at an angle increases with each revolution of the tails by the densities of the PDF that overlap the angle. Originally, the von Mises distribution modeled experimental errors arising from determination of atomic weights. Other applications of the von Mises distribution now include the direction of the sum of unit vectors representing observations of direction or periodic phenomena. The wrapped normal distribution dominates geology and models Brownian motion on the circle. However, inference is easier with the von Mises distribution.
- The wrapped Cauchy distribution is obtained by “wrapping the tails” of the Cauchy distribution on a circle in opposing directions. The Cauchy distribution is used to indirectly simulate the von Mises distribution.

In this dissertation, an observation is a measurement of direction or a realization of a circular random variable, expressed as a unit vector or as an angle, with an angle from 0 to 360°, from 0 to 2π , or from $-\pi$ to π (see Figure 1-1 for details). The main circular statistics are based on computing with direction in unit vector format. A sample consisting of observations of direction as unit vectors is summarized as the resultant vector. The vector resultant is the sum of the unit vectors representing directions. Unit vectors are summed by attaching the tail of one vector to the head of another. The main circular statistics include the resultant vector mean direction and the resultant vector mean length.

The resultant vector mean direction, $\bar{\theta}_n$, which is the direction of the resultant vector, is the measure of central direction. Why is it necessary to use vectors to

determine the average direction? Figure 1-2 shows that the average or central direction of 15° and 345° is not the arithmetic mean = 180° as on a linear scale.

In Figure 1-2, the sum of these directions is the vector

$$\begin{aligned} & (\cos(15^\circ), \sin(15^\circ)) + (\cos(345^\circ), \sin(345^\circ)) = \\ & (\cos(15^\circ), \sin(15^\circ)) + (\cos(15^\circ), -\sin(15^\circ)) = \\ & (2\cos(15^\circ), 0) \end{aligned}$$

which has a direction of 0° . Hence, the average direction is 0° . The extensive use of trigonometry distinguishes circular statistics from the statistics of linear random variables.

The other main circular statistic is the resultant vector mean length \bar{R}_n . With n the number of observations of direction, the resultant vector mean length \bar{R}_n of n observations of direction is $1/n$ times the vector resultant length R_n . It is a measure of concentration about the mean direction, where the sense of concentration is the opposite the sense of variability (a measure of spread). When variability increases, concentration decreases and vice versa. If all n observations have the same direction, the variability is zero, the resultant vector length $R_n = n$ (the unit vector observations of direction added tail to head are aligned and n long), and the resultant vector mean length $\bar{R}_n = \frac{1}{n} R_n = 1$, which is the theoretical maximum. When direction takes random values, the variability is greater than 0, $R_n < n$, and $\bar{R}_n = \frac{1}{n} R_n < 1$. If n is even, and the angles between all pairs of adjacent observations of direction are equal, the variability (spread) is the theoretical maximum, the horizontal and vertical components of the unit vectors cancel, $R_n = 0$, $\bar{R}_n = \frac{1}{n} R_n = 0$, and the resultant vector mean direction $\bar{\theta}_n$ is undefined.

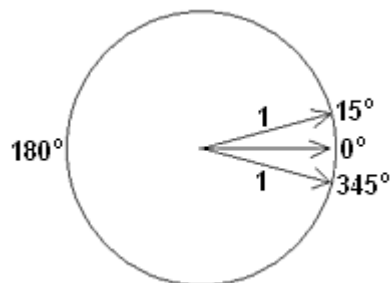


Figure 1-2. The Arithmetic Mean of 180° Does Not Point in the Central Direction of 0° .

The parameter of the circular distribution corresponding to the resultant vector mean length of the sample is the population resultant vector mean length ρ . The effects of ρ on the sample observations of direction and the sample mean resultant vector (sample resultant vector scaled by $1/n$) are illustrated in Figure 1-3. The sample observations are indicated by tan arrows and the sample mean resultant vectors by black arrows. Circles with unit radius are over plotted in black to indicate a distance of 1. Going left-to-right in Figure 1-3, the population resultant vector mean length ρ increases, concentration about the mean direction increases, and the length of the mean resultant vector of the sample tends to increase. In the right hand plot with $\rho = 0.99$, the length of the sample mean resultant vector gets close to 1, but is not exactly 1 as can be seen in the zoom view on the right.

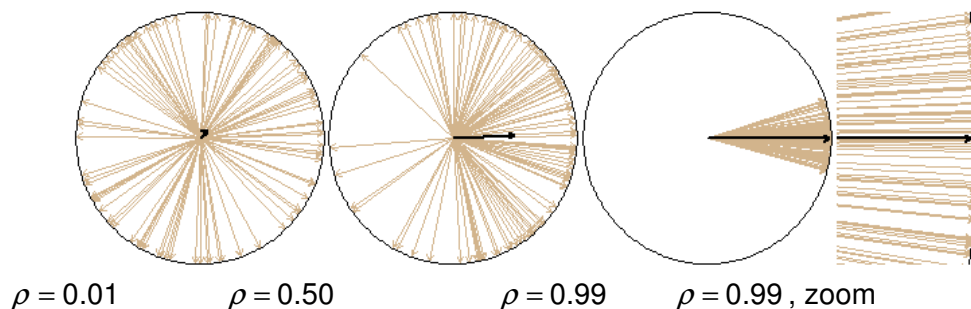


Figure 1-3. The Effect of the Population Resultant Vector Mean Length ρ on the Sample Mean Resultant Vector (Black) of a Sample (Tan) from the von Mises Circular Distribution. As ρ goes toward 1, concentration about the mean direction increases and the length of the mean resultant vector goes toward 1.

1.2 A Motivational Example

Figure 1-4 shows traditional arrow plots of ocean wind data as circular-spatial data (top), which is the focus of the dissertation, and vector-spatial data (bottom). The data are plotted as tan colored arrows and the means as black arrows. The data were freely extracted from the Comprehensive Ocean Atmosphere Data Set (Chapter 2, Subsection 2.2.1) at <http://iridl.ldeo.columbia.edu/SOURCES/.COADS/.mean/> for 1980 to 1990, December to March, and for the area of latitude -3° N to $+3^{\circ}$ N and longitude -93° E to -87° E. Note that -3° N means 3° south of the equator, and -93° E means 93° west of the Greenwich prime meridian. The data contain 1934 observations of month, year, longitude, latitude, and east and north components of wind velocity. In the vector-spatial plot, the mean resultant vectors were computed from the average horizontal and vertical velocity components by location. The circular-spatial data were obtained from the vector-spatial data by scaling the vector observations to unit length losing the magnitude information. In the circular-spatial plot, the mean resultant vectors of the circular-spatial data were computed by location, and scaled to unit length. The difference between differently computed means is 9.96° at -87° E and $+3^{\circ}$ N. Average wind direction is changing smoothly in the south-north direction, rotating clockwise as latitude increases and evidencing a global trend.

1.3 Problem Description

The problems addressed in this dissertation include:

- How may circular-spatial data be efficiently interpolated based on spatial correlation? Jammalamadaka and SenGupta summarized many expressions of nonspatial circular correlation (2001, Chapter 8). How could spatial correlation be extracted from circular-spatial data and modeled to be useful for the interpolation

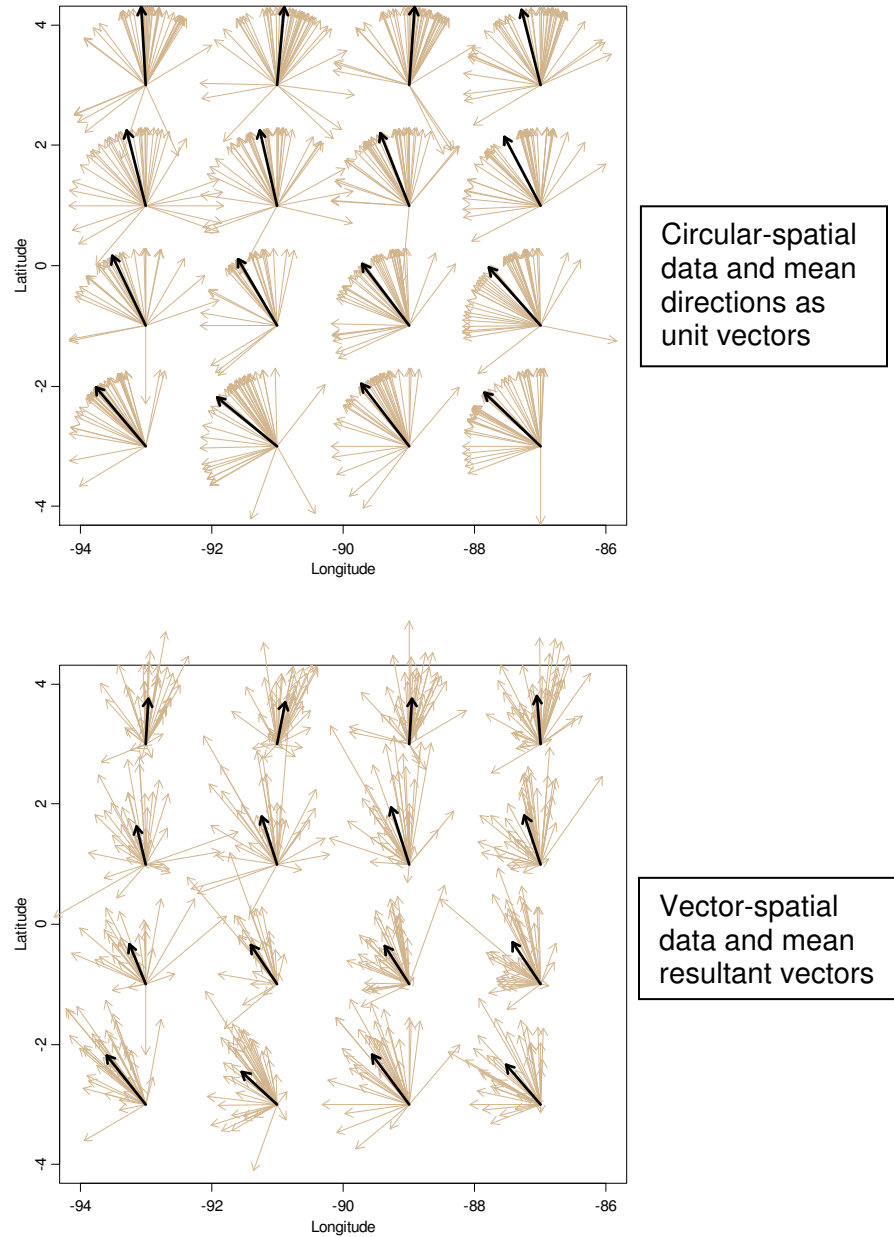


Figure 1-4. Circular and Vector Spatial Data and Their Means for the Direction the Ocean Wind Blows Toward. Data from 1980 to 1990, December to March, in the Area of Latitude -3° N to $+3^{\circ}$ N and Longitude -93° E to -87° E. At each sampling location, the raw data are indicated by tan arrows and the means by black arrows.

of direction between observation locations?

- How can a CRF be simulated based on input specifications of a circular probability distribution and spatial properties such as the distance at which CRVs are no longer correlated, and what are the properties of the simulated CRF?
- The intelligibility of arrow plots (Figure 1-4) decreases as data density and random variation increase. Intelligibility is also affected by missing values. How can circular-spatial data, interpolations of circular-spatial data, or simulations of a CRF be plotted as a heat map that is not color discontinuous at any direction, e.g., where the color encoding direction at 0° and 360° are the same? How can these data be plotted with high data density such that both large scale and small scale directional structure can be easily recognized?

1.4 Literature Review

1.4.1 Brief History of Circular Statistics

Circular statistics, the statistics of direction, is a relatively new statistical domain as indicated by some history extracted from Fisher (1993, chap. 1). Circular-spatial statistics is very new. In 1767, John Mitchell, FRS (Fellow of the Royal Society), tested the hypothesis that the distribution of angular separations of stars is uniform. He determined that the number of close stars were too many to support this hypothesis. In 1802, John Playfair noted that directional data should be analyzed differently from linear data, recommending that average direction be the direction of the resultant vector. In 1858, Florence Nightingale, chief nurse in the British Army during the Crimean War, created the rose diagram (for example, see Figure 1-5, a rose plot of ocean wind direction) displaying the effect of sanitation vs. month of year, saving thousands of lives in military hospitals. In 1880, Lord Rayleigh created a statistical test for the hypothesis

of the uniform circular distribution vs. the unimodal alternative. In 1918, von Mises defined the circular normal, or von Mises distribution, which is a basis of parametric statistical inference for circular data. In 1938, Reiche introduced what is now called the CUSUM chart, which plots cumulative vector direction and average magnitude, to indicate when a sufficient amount of vectorial data has been acquired. In 1939, Krumbein introduced the transformation of axial to vectorial data for analysis, and back transformation to axial results. The paper by Watson and Williams (1956) about statistical inference for the mean and variability of a sample from the von Mises distribution and methods for comparing two or more samples started a period of significant theoretical development. Following developments of the 1960s, Mardia (1972) published a comprehensive account of methods for display, summarization, goodness of fit, and parametric/nonparametric analyses of circular data. Batschelet (1981) studied methods for bio-circular data analysis. Large sample theory was introduced about a decade after Mardia's book. Developments in circular correlation and regression, time series analysis, large sample and bootstrap methods, and nonparametric density estimation are found in Jupp and Mardia (1989). McNeill (1993) extended geostatistics to circular data. Thus, most of the theoretical developments in the field of circular statistics are relatively recent. Additional past contributors are listed in Mardia (1972).

The latest books on circular statistics include those written by Fisher (1993), Mardia and Jupp (2000), and Jammalamadaka and SenGupta (2001).

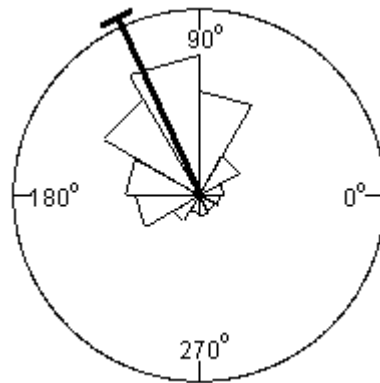


Figure 1-5. Rose Plot of the Circular Data Derived from the Data of Figure 1-4. The angle of the wedge is the bin width and the area of a wedge is proportional to the bin count. The heavy radial line indicates the mean of 106.3° and the short arc indicates the 95% confidence interval of $(104.8^\circ, 107.8^\circ)$.

1.4.2 Literature Review for Imaging Circular-Spatial Data

Sources were examined for examples of imaged circular-spatial data including:

- Visualization displays of computational fluid dynamics (CFD) software:
 - FLUENT (FLUENT 2008) – Software for simulation of fluid flow, heat and mass transfer, and related phenomena involving turbulence, reactions, and multiphase (liquid and gas) flow.
 - FIELDVIEW (Intelligent Light 2008) – Post-processing software for identification of important flow features and characteristics in simulations, and for interactive exploration of results.
 - Ensight (CEI 2008) – General tools for visualizing complex datasets.
- Software for the analysis of circular data:
 - Axis (Pisces Conservation Ltd 2008) – Implements the principal graphical methods and statistical tests described by Fisher (1993) for the analysis of circular data.

- CircStats (Lund and Agostinelli 2007) – This R package implements the graphical methods and statistical tests described by Jammalamadaka and SenGupta (2001) for the analysis of circular data.
- Oriana 2 (Kovach Computing 2004) – Calculates statistics, tests, and correlations for circular data. Graphics include the rose diagram, linear and circular histograms, the arrow plot with arrow length as frequency or magnitude, stacked raw data plots, and circular QQ plots.
- Surfer 8 (Scientific Software Group 2008) – Converts vector-spatial data into contour, surface, wireframe, vector, and shaded relief maps.
- Vector Rose 3.0 (Zippi 2001) – Calculates circular statistics, tests, and graphics (including the rose diagram and the circular histogram) for circular data.

None of these software packages provide a method of imaging circular-spatial data similar to the new circular dataimage of Chapter 2.

1.4.3 Literature Review for Circular-Spatial Correlation

Bivariate or multivariate data involving CRV is common. However, the study of association or correlation is newer than the relatively new area of circular statistics. Further, the study of circular-spatial correlation is newest. Jammalamadaka and SenGupta (2001) described several methods for computing the association and correlation of nonspatial CRV and circular data. These include:

- The population circular correlation coefficient

$$\rho_c(\alpha, \beta) = \frac{E\{\sin(\alpha - \mu)\sin(\beta - \nu)\}}{\sqrt{Var(\sin(\alpha - \mu))Var(\sin(\beta - \nu))}}$$

with E the expectation operator, angle α , $\mu = E\{\alpha\}$, angle β , $\nu = E\{\beta\}$, and Var the variance.

- Parametric cases of ρ_c involving specific circular probability distributions.

- The sample circular correlation coefficient, $r_{c,n} = \frac{\sum_{i=1}^n \sin(\alpha_i - \bar{\alpha}) \sin(\beta_i - \bar{\beta})}{\sqrt{\sum_{i=1}^n \sin^2(\alpha_i - \bar{\alpha}) \sin^2(\beta_i - \bar{\beta})}}$ with n the sample size, sample $(\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n)$, and $\bar{\alpha}$ and $\bar{\beta}$ the sample mean directions.
- The nonparametric version of $r_{c,n}$ with α_i replaced by $\frac{\text{rank}(\alpha_i)}{n} 2\pi$, and β_i replaced by $\frac{\text{rank}(\beta_i)}{n} 2\pi$.

1.4.4 Literature Review for Kriging of Circular-Spatial Data

1.4.4.1 Terminology

Kriging is a body of techniques for predicting spatially correlated data. Figure 1-6 shows a heatmap before and after kriging. The name of the technique is derived from Daniel Krige, a South African mining geologist, who originated the method. Kriging uses the measurements, their distances apart, and a model of their spatial dependence based on the variogram or covariogram. The covariogram is the graph of the mean covariance between observations a distance d apart vs. d . The variogram is the graph of the mean squared difference of observations a distance d apart vs. d . In general, the variogram is less sensitive to minor departures from the assumption that the process mean is independent of location than the covariogram. The data are called isotropic, as opposed to anisotropic, when the spatial dependence is independent of the direction in which measurements are taken, and dependent on the distance d only.

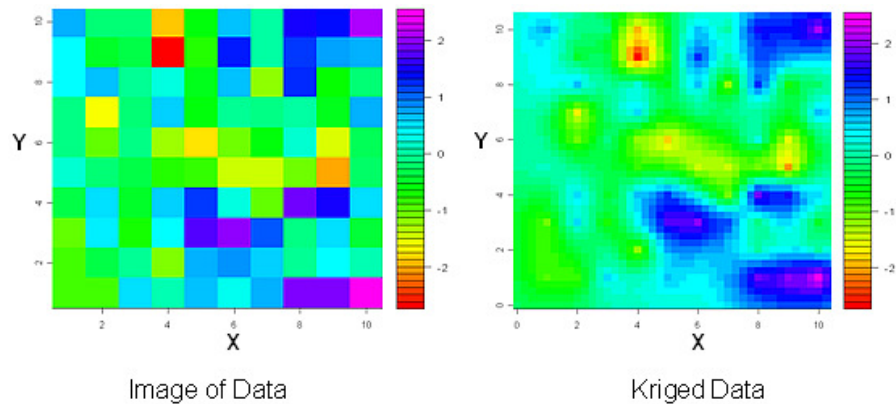


Figure 1-6. Kriging, the Estimation of Spatial Data Based on Spatial Correlation.

1.4.4.2 Literature Review for Kriging of Circular-Spatial Data

A significant body of literature exists for the kriging of circular and vector spatial data. Quimby (1986) estimated an n -vector at a location assuming each location has a different mean and variance, and using the multivariate auto covariance-cross covariance matrix. Young (1987) showed that kriging is applicable to 3D vectors describing rock fracture orientation assuming each location has the same unknown mean, and using a scalar variogram function of vectors. Young's method is evaluated using cross validation. Schaeben, Boogaart, and Apel (2001) predicted the polar unit vector at a given location, using multivariate variograms and covariance functions, assuming a constant mean, and defining different types of isotropy, which lead to different simplifications of the general cross-covariance function and kriging procedures. A measure of confidence in the estimate was given. Boogaart and Schaeben (2002a) extended prediction to direction, axis, or orientation by embedding a sphere/hemisphere in a real vector space. Boogaart and Schaeben (2002b) predicted rotation by embedding the rotations in a real vector space with assumptions of isotropy.

McNeill (1993) introduced kriging of circular random variables via trigonometric based circular statistics, assuming a common circular probability distribution, isotropic spatial correlation, and a variogram as a function of cosines.

1.4.5 Literature Review for Simulation of Circular-Spatial Data

A RF is a stochastic process operating over a space. A Gaussian RF (GRF) is a RF in which the random variables follow the multivariate normal distribution with covariance depending on distance between locations of the random variables.

References include Gneiting and Schlather (2004), Lantuejoul (2002), and Schlather (1999). The function `grf` in the R package `geoR` (Ribeiro and Diggle 2001) generates simulations of GRFs for a given covariance model. The function `GaussRF` in the R package `RandomFields` (Schlather 2001) generates spatial GRFs and spatial-temporal GRFs.

1.5 Dissertation Overview

Chapter 2 extends the graphical methods of spatial statistics. It details a new method for circular-spatial data that produces a continuous image with high resolution such that directional structure can be simultaneously recognized on both local and global scales.

Chapter 3 introduces a new graphical method called the cosineogram (graph of cosines) and related theory for the extraction of spatial correlation from circular-spatial data in the form required by the circular kriging method of Chapter 4. The empirical cosineogram plots the mean cosine of the angle between random components of direction a distance d apart vs. d . The cosineogram is replaced with a fitted positive definite function to achieve optimal fit of estimated direction to the actual, but

unobserved and unknown direction. Three main positive definite functions from linear kriging are adapted to the cosine behavior of the CRF. Additional functions are identified in Appendix M.

Chapter 4 provides a detailed linear-algebraic and trigonometric derivation of an estimator of direction in an isotropic CRF (correlation same in all directions). Building on the work of McNeill, the derivation proceeds without assuming the variogram as a function of cosines and avoids the Taylor series approximation. This is accomplished by minimization of the mean squared length of the error between the estimator and the actual, but unknown and unobserved direction. This derivation produces a new expression of circular-spatial correlation as the mean cosine of the angle between random components of direction observed at a distance d apart. Optimality of the estimator is proved. A computationally efficient form of the estimator is derived. Chapter 4 also derives a first order estimator of the imprecision of the direction estimator, correcting the result of McNeill. The interpolation is called “exact” in the sense that, although undesirable in the presence of noise, the estimate at a location where direction is observed equals the observed direction, and the imprecision or variability of the estimate goes to zero as distance to an observation location goes to zero.

In Chapter 5, the ideas of GRFs are extended to CRFs. A method is provided to simulate a CRF with a specified circular probability distribution from a GRF with a specified spatial covariance model. Some properties of the simulated CRF are argued and others involving one or two nonclosed form transformations are characterized. Figure 1-7 summarizes circular-spatial methods of Chapters 1-5. Chapter 6 provides a comprehensive example, which shows each step of Figure 1-7, and connects the chapters.

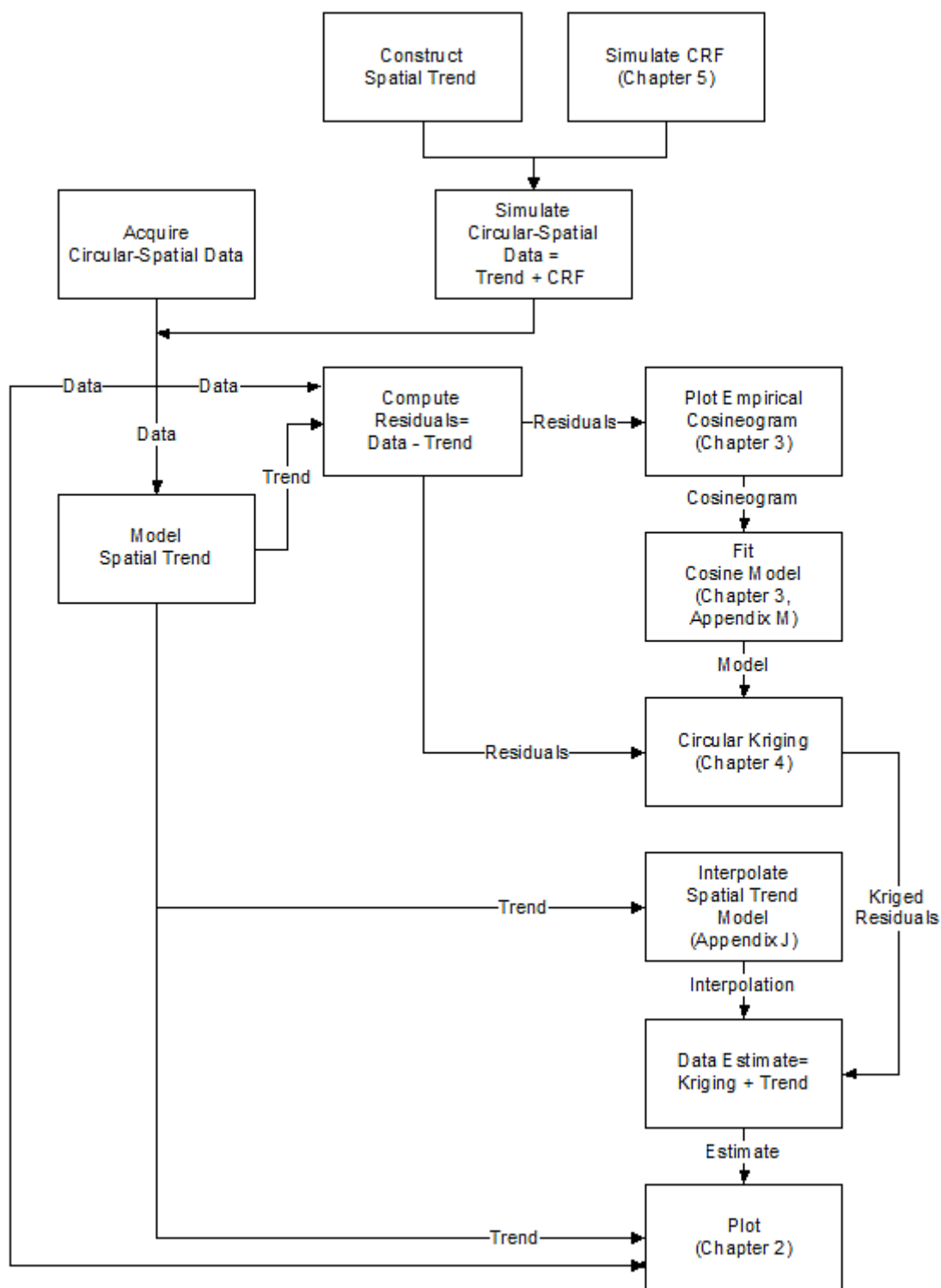


Figure 1-7. Flow Chart of Methods for Circular-Spatial Data.

Additional details are given in the appendices. Appendix A summarizes the mathematical notation used. Appendix B organizes the linear algebra theory and subordinate proofs required by the circular kriging derivations of Chapter 4.

Appendices C – M are relevant to the simulation of circular random fields of Chapter 5. Appendices C and D continue the qualitative evaluations of CRFs of Chapter 5, Section 5.5. Circular CDFs are derived in Appendix E for support $[0, 2\pi)$, verified by integration in Appendix F, and modified for the equivalent rotated support $[-\pi, +\pi)$ in Appendix G. Rotation of the support from $[0, 2\pi)$ to $[-\pi, +\pi)$ is required to map standard normal random variables to a CRV with mean direction 0 using the method of Chapter 5. Appendix H corrects a form of the wrapped Cauchy CDF, evaluates three forms of the CDF, and selects the form for implementation in the R package `CircSpatial` that is simple and does not have numerical issues at extreme low variability. Appendix I derives the inverse CDF of the triangular circular probability distribution. It is required to simulate triangular CRFs. The inverse CDFs of the cardioid, von Mises, and wrapped Cauchy circular distributions are nonclosed form transformations. Hence, Appendix M characterizes the spatial dependence of CRFs simulated by the method of Chapter 5.

Appendix J documents the R software package `CircSpatial`, which covers all the chapters and details a method of interpolation of global trend models based on circular-spatial data. Estimated direction is obtained by adding the kriging interpolation to the global trend model interpolation. Appendices K and L contain the R function code of the R package `CircSpatial` and the R command line input used to produce many of the figures in the dissertation. Appendix N has graphics for CRV and circular data introduced in Chapter 1 including a new cylindrical display of the probability density function of CRV.

CHAPTER 2

CIRCULAR DATAIMAGE, A HIGH RESOLUTION CONTINUOUS
IMAGE OF CIRCULAR-SPATIAL DATA

2.1 Introduction

The focus of this chapter is on a new method for imaging circular-spatial data, the circular dataimage. Figure 2-1 is a circular dataimage based on data introduced in Subsection 2.2.1 and further discussed in Section 2.4. The dataimage of Minnotte and West (1998), for the imaging of many ordered variables and observational units to show correlation structure, motivated the problem of how to image circular-spatial data and what this new method should be called. Moreover, dataimages for linear variables have been used extensively in various disciplines. Heatmaps, as they are called in genomics, have been made widely popular by Eisen, Spellman, Brown, and Botstein (1998).

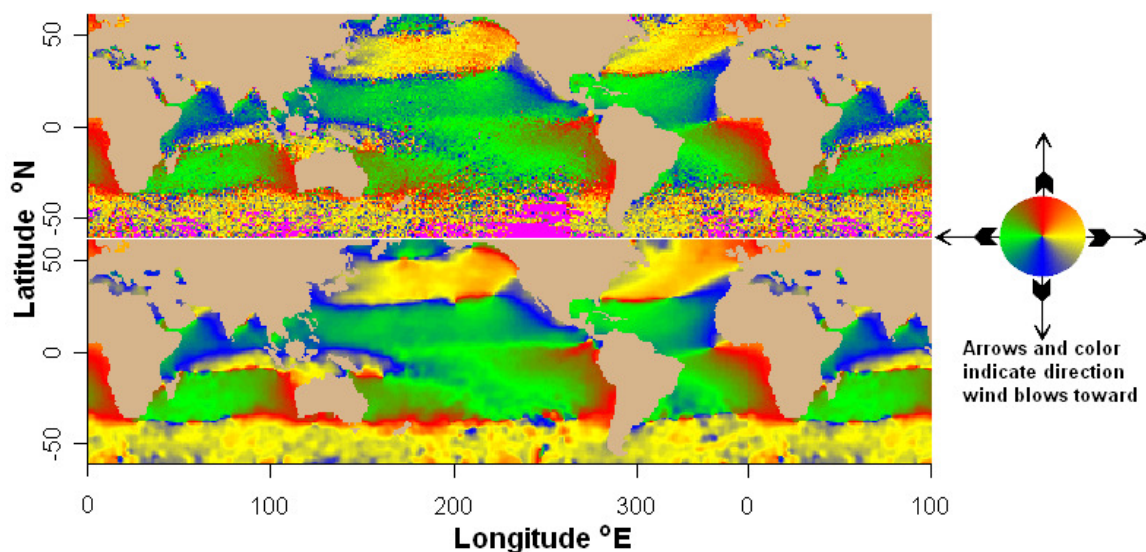


Figure 2-1. Circular Dataimages of the Direction Wind Is Blowing Toward, Coded with Yellow-Red-Green-Blue (YRGB) Color Wheel (Right). Average direction is shown in the top plot, and smoothed average direction in the bottom plot. The circular dataimage provides a continuous high resolution image of circular-spatial data with structure simultaneously recognizable on a broad range of scales.

This chapter is organized as follows: Section 2.2 introduces the data used throughout the chapter and provides an overview of some of the existing methods for the display of circular-spatial and vectorial-spatial data. Section 2.3 describes the “cross over” problem of circular data. Section 2.4 provides a solution to the cross over problem, the color wheel scale of direction, and the circular dataimage, with color continuity at the cross over point and capable of a high resolution image of circular-spatial data and statistics. Section 2.5 compares the circular dataimage with the existing method of arrow plots. Section 2.6 gives an example calculation of a circular color wheel for the Red-Green-Blue (RGB) color system. Section 2.7 provides color considerations, and color wheel and circular dataimage variations including a brief outlook of how magnitude might be encoded. Section 2.8 provides additional examples. Section 2.9 concludes with the summary and description of future work.

2.2 Overview of Vectorial-Spatial Displays

2.2.1 Data

The ocean wind data used throughout this paper were freely extracted from the International Comprehensive Ocean Atmosphere Data Set (ICOADS) at <http://dss.ucar.edu/datasets/ds540.1/data/msga.form.html> for the El Niño years 1972, 1976, 1982, 1987, 1991, 1994, and 1997, January through April, and in 1° increments for the area of longitude 0.5° E to $+359.5^\circ$ E by latitude -59.5° N to $+60.5^\circ$ N. Note that -3° N equals 3° south of the equator, and 359° E equals 1° W of the prime meridian. These data were selected to provide homogeneous data of the El Niño periods. El Niño (the child) refers to the Christmas season when changes in Pacific Ocean currents usually begin. These changes are often accompanied by severe climate disruptions to countries in and adjacent to the Pacific.

This dataset comprises 495,688 observations of month, year, longitude, latitude, and east and north components of wind velocity in units of 0.01 meters/second (m/s). With 0 to 28 observations per grid cell, average wind direction was computed as the quadrant specific inverse tangent of the average vertical component of wind velocity in a grid cell divided by the average horizontal component in the cell. Alternatively, hexagon binning has advantages for grouping data (Carr, Littlefield, Nicholson, and Littlefield 1987). All scaled figures in this paper have horizontal scale in $^{\circ}$ longitude and vertical scale in $^{\circ}$ latitude.

ICOADS began as COADS (Comprehensive Ocean Atmosphere Data Set) in 1981 as a cooperative project of the National Climatic Data Center (NCDC), the Environmental Research Laboratories, the Cooperative Institute for Research in Environmental Sciences, and the National Center for Atmospheric Research. COADS was renamed ICOADS in 2002 to recognize extensive international collaboration. The objective of ICOADS is to provide a consistent and easily used historical record of surface marine data beginning 1854. Seventy million unique reports of 28 variables obtained from ships of opportunity and ocean buoys were organized and cleaned of outliers. Trimmed monthly summaries give statistics for observed air and sea surface temperatures, wind east and north components in m/s, sea level pressure, humidity, cloudiness, and derived variables.

2.2.2 Smoothing Average Wind Data

The bottom subplot of Figure 2-1 and Figure 2-2 displays smoothed average wind data. The R package `fields` (Fields Development Team 2009) function `image.smooth` with a smoothing bandwidth of 2.5° was applied separately to the cosines and sines of average wind direction to avoid the cross over problem (see Section 2.3,

Crossover). The smoothed direction was computed as the inverse tangent of the smoothed vertical component divided by the smoothed horizontal component.

2.2.3 Existing Methods for Displaying Circular and Vectorial-Spatial Data

Some methods for the display for circular-spatial and vectorial-spatial data are shown in Figure 2-2. These methods were adapted from Ware (2004, pp. 200-205) using the R code in Appendix K.7. In all vectorial, arrow, and triangle icon plots in this and other chapters, flow or direction will be from tail to head. Figure 2-2 contains plots of:

- a) Arrows of fixed length on a regular grid (Figure 2-2 (a)).
- b) Jittered arrows of fixed length (Figure 2-2 (b)). Jittering is implemented here by adding values from a uniform distribution to the tail coordinates of the arrow. Thus, jittering only randomizes the origin of the arrow so jittered arrows have the same directions as nonjittered arrows. This helps to reduce overplotting and improves the sense of flow. Note that sufficiently large magnitude jitter can randomize the structure in directional-spatial data.
- c) Jittered arrows of length proportional to magnitude (Figure 2-2 (c)).
- d) Jittered triangle icons with icon area equal to vector magnitude (Figure 2-2 (d)).

Other methods discussed in Ware (2004, p. 204), but not shown, include: line integral convolution and large arrow heads along a streamline using a regular grid. Ware (2004, p. 205) stated "the display problem becomes ... to reveal important aspects of the data for a particular set of tasks" So, the choice and effectiveness of a method depends on the task to which the method is put. Within this paper, the task is to discover structure in circular-spatial data (all vectors have equal length). Hence, the circular dataimage will be compared to plots of arrows of equal length (method a).

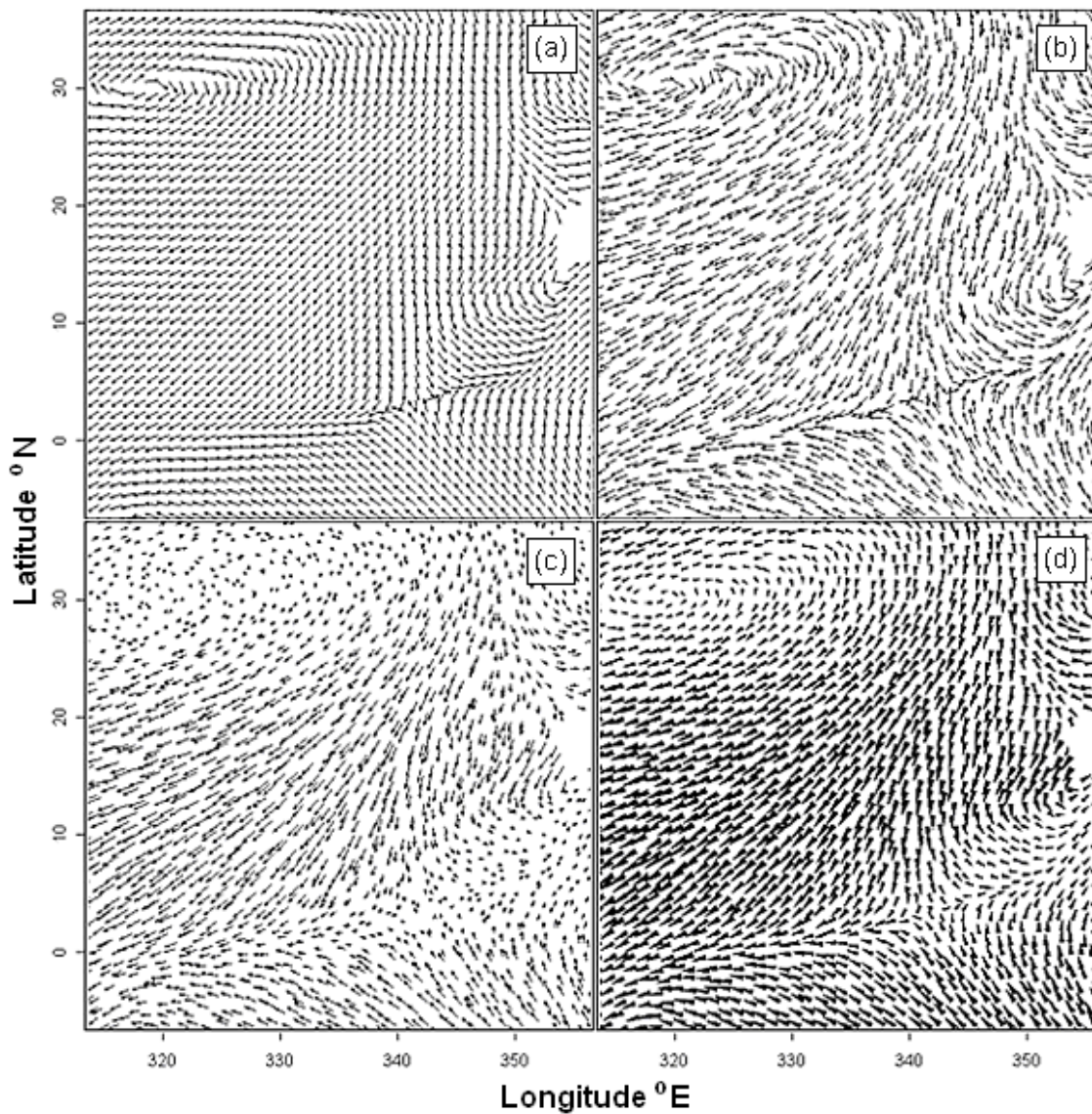


Figure 2-2. Some Existing Methods for Display of Circular and Vectorial-Spatial Data Using the Smoothed Ocean Wind Data. (a) Unit vectors, (b) jittered unit vectors, (c) jittered vectors, (d) jittered triangle icons with icon area equal to vector magnitude. Jittering helps to reduce overplotting improving the sense of flow.

2.3 Cross Over

Circular data, as opposed to linear data, is cyclic, i.e., the starting point at 0° equals the ending point at 360° . The distance between values of 0 and 360 on a linear scale is 360, but the distance on a circle between 0° and 360° is 0° . The distance on a unit circle between 0 and 2π radians is 0 radians. In 1802, John Playfair noted that directional data should be analyzed differently from linear data, recommending that the average direction should be the direction of the vector resultant. Thus, the average or central direction of 1° and 359° is not the arithmetic mean of 180° . The correct average is 0° , which is equal to the direction of the sum of the unit vectors $(\cos 1^\circ, \sin 1^\circ)$ and $(\cos 359^\circ, \sin 359^\circ)$. Summing these unit vectors, the equal and opposite vertical components annihilate and the equal horizontal components reinforce.

Historically, this problem arose in automating the summarization of wind data. It has been called the “cross over” problem because it occurs when crossing over 360° on a scale of 0° to 360° , or crossing over $+180^\circ$ on a scale of -180° to $+180^\circ$. In another example, cross over occurs in plotting a circular time series with direction on the vertical axis and time on the horizontal axis. As direction rotates counterclockwise (CCW) past 360° , direction vanishes at the top of the scale at 360° and instantly reappears at the bottom of the scale at 0° at the next time value resulting in a full scale vertical gap between plotted points. In vector field visualization, when direction is coded with a single color gradient, e.g. dark blue at -180° to bright red at $+180^\circ$, image discontinuity occurs where direction varies around 180° . At the cross over point the direction to color correspondence is one to two. An image using this scale would display the 180° direction as either red or blue depending on whether direction is increasing or decreasing. To avoid this problem, users may examine subregions where cross over

does not occur. Consequently, the ability to resolve both local and global structure in circular-spatial data is reduced, and the overall structure is not seen.

2.4 The Circular Dataimage and Color Wheel

2.4.1 The Color Wheel Solution to Cross Over

Suppose that a linear color scale, made from a blue-red color gradient to code direction from -180° to $+180^\circ$ (Figure 2-3 (a)), is wrapped on a circle with -180° (blue) connected to $+180^\circ$ (red) on the left side of the circle. A red-to-blue color discontinuity occurs at the cross over point where the ends of the wrapped scale intersect at $\pm 180^\circ$ (Figure 2-3 (b)). To eliminate the discontinuity, a three-color gradient, red to green to blue with continuity between gradients, was inserted and centered at $+180^\circ$. In Figure 2-3 (c), each two-color gradient (green-blue, blue-red, red-green) is adjusted to an equal arc length of 120° . Every point on the color wheels (c) and (d) is color continuous. The color scheme in (b) is discontinuous at the 180° location.

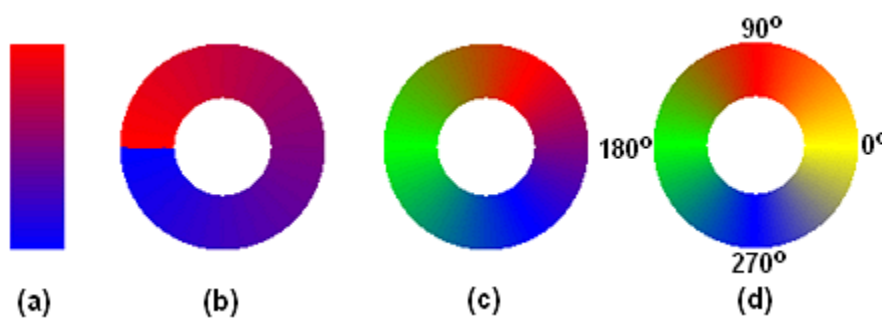


Figure 2-3. Evolution of the YRGB Color Wheel. (a) Blue-red linear color scale, (b) color scale (a) wrapped on circle, (c) red-green-blue linear color gradient inserted at 180° , and (d) blue-yellow-red gradient inserted at 0° and labels added. The final YRGB color wheel (d) aligns the 4 main colors (yellow, red, green, and blue) to the 4 main directions (0° , 90° , 180° , 270°).

An additional blue-yellow-red color gradient was inserted at 0° and the component two-color gradients (green-blue, blue-yellow, yellow-red, red-green) were adjusted to 90° arcs (Figure 2-3 (d)). This particular color wheel is called a YRGB Color Wheel. The four-color color wheel is more intuitive than the three-color color wheel because the number of main or pure colors between color gradients equals the number of main directions (0° , 90° , 180° , and 270° , or east, north, west, and south), aligning the four main colors to the four main directions. Thus, the information (number of perceived color boundaries and degree of color structure) in the dataimage (Subsection 2.4.2) of circular-spatial data is increased. In general, a color wheel can be defined as a sequence of three or more two-color gradients with color continuity between connecting color gradients. Hence, the color wheel is color continuous at any point on the color scale going clockwise or counterclockwise.

2.4.2 The Circular Dataimage

To image circular-spatial data, let a direction in a pixel be plotted as the color on a color wheel in the direction of the data. The result is called the circular dataimage. In all circular dataimages in this and other chapters, flow or direction will be from the center of the color wheel toward the color on the color wheel. In Figure 2-1, which was constructed using the R code in Appendix K.2, the average direction that ocean wind blows toward is coded with the YRGB color wheel at the right. The YRGB color wheel consists of color gradients yellow to red from 0° to 90° , red to green from 90° to 180° , green to blue from 180° to 270° , and blue to yellow from 270° to 360° . The resulting circular dataimage shows some interesting features. In the Pacific Ocean and around the equator, wind tends to blow from east to west, which is typical for any year. The pattern of direction on the west side of the Americas is similar to that on the west side of

Africa: wind flows toward the equator from both the north and the south, then turns to the west.

2.5 Comparison of Methods

Figures 2-4-1 and 2-4-2 show the direction ocean wind is blowing toward between 100° and 325° E longitude and between -59.5° and 60.5° N latitude. Figure 2-4-1 shows average direction as described in Subsection 2.2.1 computed with 0 to 28 observations of direction per location due to missing data. Figure 2-4-2 shows smoothed average direction as described in Subsection 2.2.2.

As black arrows were hardly visible in some regions when plotted on a dataimage using the previously discussed YRGB color wheel, direction is coded via the hue, saturation, and value (HSV) color wheel (saturation = 0.5) in the right margin of the subplots. Further discussion on various color spaces will be found in Subsection 2.7.1. Missing ocean data is coded by white, and structurally missing values over land by tan. Arrows are overplotted at a density of one arrow per 15 cells in the horizontal and vertical directions in subplots B and F, and at a density of one arrow per 5 cells in Subplots C, D, G, and H.

The ability to perceive structure via arrow plots depends on the variability of the data and the arrow density relative to the plot scale. In the relatively noisy latitudes south of Australia (135° E, -25° N) in Figure 2-4-1 B, the general direction of the arrows is difficult to recognize. The arrows point in random directions due to noise. Increasing the arrow density in Figure 2-4-1 C does not help. The arrows are more misleading than informative. The use of arrows alone on noisy data in Figure 2-4-1 D is worse.

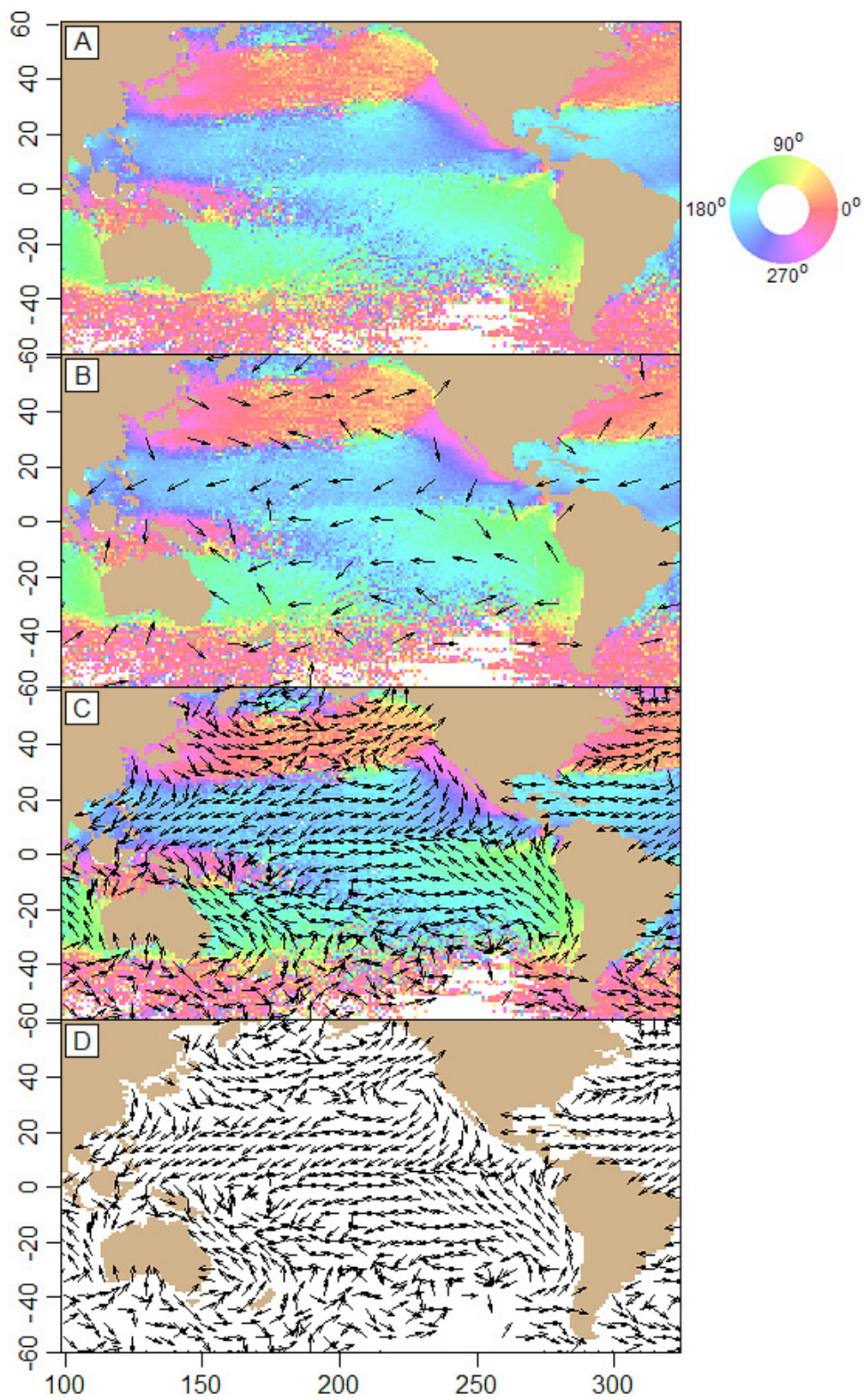


Figure 2-4-1. Plots of Average Ocean Wind Direction.

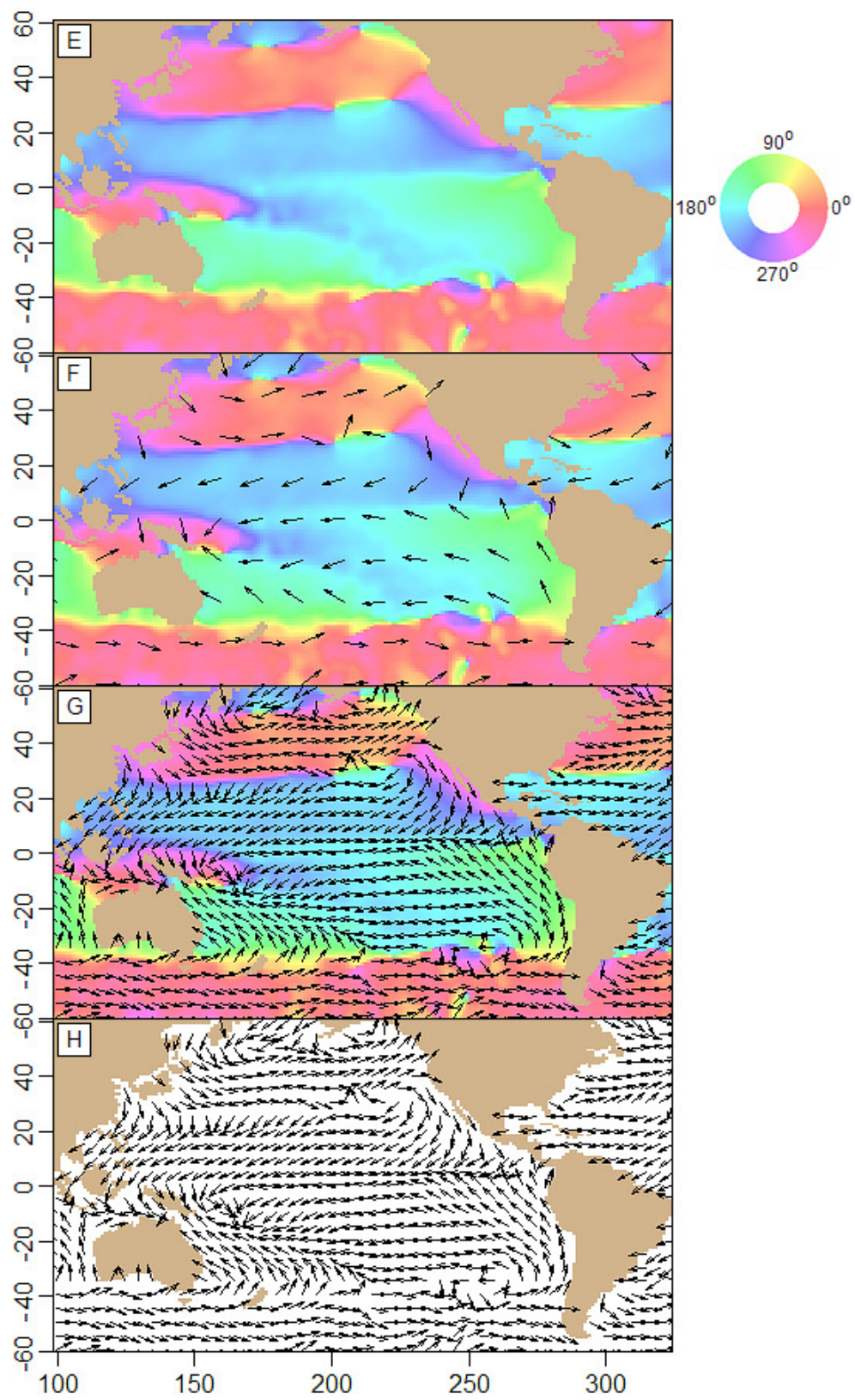


Figure 2-4-2. Plots of Smoothed Ocean Wind Direction.

In Figure 2-4-2 the data are smoothed, which replaces missing values. Figure 2-4-2 E shows maximum structural detail. In Figure 2-4-2 F the general direction in the extreme southern latitudes is apparent in the arrow structure. Increasing the arrow density on the smoothed data in Figure 2-4-2 G or using arrows alone in Figure 2-4-2 H further increases the perception of structure. Arrows work well for smooth data at low to moderate arrow density.

In contrast, the circular dataimage shows more structure in noisy data. The general direction in the latitudes south of Australia can be seen as the dominant color in Figure 2-4-1 A as approximately west to east. The colors alone in Figure 2-4-1 A give a better overall impression of the structure, general direction, noise, and missing data than colors and arrows in Figure 2-4-1 B.

Figure 2-5 shows average wind direction coded with the Blue-Green-Yellow-Red (BGYR) color wheel with missing values displayed in magenta (southern latitudes) and continents shown in tan. The data plotted in the left plots are identical to the data plotted in the right plots with decreasing scale (zooming in) top to bottom.

As we zoom into a smaller area, directional structure in the arrow plots eventually becomes recognizable, e.g., Figure 2-5 (e). With increasing scale (zooming out) and constant arrow spacing relative to the data, arrow plots eventually become unintelligible, e.g., in Figures 2-5 (a) and (c). In contrast, the circular dataimage shows overall and detailed structure on a wide range of scales, e.g., Figures 2-5 (b), (d), and (f). Even at a scale of a $50^{\circ} \times 50^{\circ}$ area (Figure 2-5 (f)), the circular dataimage easily shows structure that is not easily obtained from the arrow plot in Figure 2-5 (e).

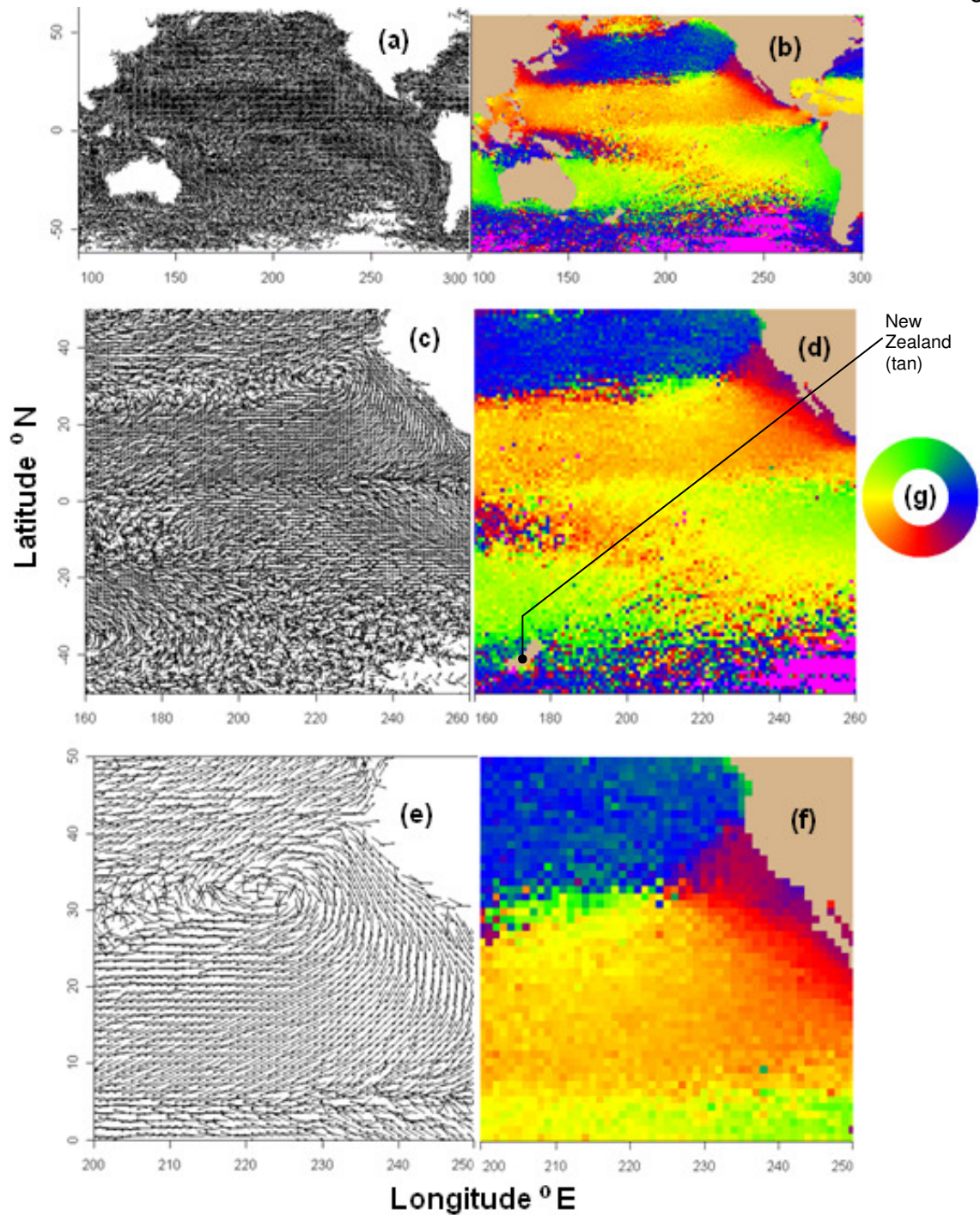


Figure 2-5. Comparison of Arrow and Circular Dataimage Plots of Average Ocean Wind Direction. Plots (a) and (b) cover 200° of longitude; (c) and (d) cover 100° Of longitude; (e) and (f) cover 50° of longitude; and (g) BGYR color wheel. The ability to recognize structure depends on plot type, smoothness of data and density and arrows, and distribution of missing data.

Additional characteristics of circular dataimages and arrow plots include:

- Missing values can easily be seen as a contrasting color not in the color wheel, or by plotting arrows over missing data plotted as an area in contrasting color.
- Arrows may overplot and obscure geographical boundaries as in Figure 2-5 (c) (New Zealand obscured), but boundaries may be overplotted on circular dataimages. New Zealand remains visible in (d).
- Circular dataimages fill the plot at pixel level resolution. Arrows cannot.

The circular dataimage has been defined, compared with arrow plots, and its capabilities have been demonstrated. In Sections 2.6 and 2.7, the use of color will be discussed in further detail.

2.6 Calculation of a BGYR Color Wheel

Table 2-1 shows how the color levels of red, green, and blue on a scale of 0 to 1 were computed for the BGYR color wheel in Figure 2-5. For example, as direction goes CCW from 0° to 90°, the corresponding color is obtained by decreasing the amount of blue and increasing the amount of green linearly while the level of red is constant at zero. The reader is invited to experiment with nonlinear color gradients.

Table 2-1. BGYR Color Wheel Formulae for RGB Space.

Angle (°) in	Color Range	Level of Red	Level of Green	Level of Blue
[0,90)	blue to green	0	Angle / 90	1 – Angle / 90
[90,180)	green to yellow	(Angle - 90) / 90	1	0
[180, 270)	yellow to red	1	1 - (Angle - 180) / 90	0
[270, 360)	red to blue	1 - (Angle - 270) / 90	0	(Angle - 270) / 90

2.7 Color Considerations and Variations

2.7.1 Color Space

The use of color in computer graphics is described in Foley, Van Dam, Feiner, and Huges (1992). The use of color in presentation graphics is described in Ihaka (2003). The use of color in statistical graphics is described in Zeileis, Hornik, and Murrell (2008). Color spaces include CIEXYZ, RGB, HSV, CIELUV, and HCL. CIEXYZ is one of the first color spaces based on measurements of visual perception and mathematically defined. CIE denotes the Commission Internationale de l'Éclairage, 2004, which is an international body of scientists whose standards provide for the accurate communication of color information. X, Y and Z values are the levels of the primary colors added to match a color. RGB is a version of CIEXYZ space. R, G, and B are the relative intensities of the red, green, and blue primaries. HSV is a transformation of RGB space to hue (H), saturation (S), and value (V). However, HSV colors are often not considered to be perceptually based because the brightness of colors is not uniform over hues and saturations. CIELUV is a transformation of the CIEXYZ space to the perceptual axes of luminance, and the coordinates u and v of the CIE chromaticity map of human color perception. HCL colors are obtained by transforming the rectangular coordinates of u and v in the CIELUV space to the polar coordinates of hue H and chroma C. Hue H takes values in the range 0 to 360° with 0° = red, 120° = green, 240° = blue, etc. Within the space's boundaries, the admissible levels of chroma and luminance depend on the hue chosen as some hues lead to light and others to dark colors.

The R contributor package `colorspace` (Ihaka, Murrell, Hornik, and Zeileis 2009) includes the above color spaces together with a variety of HCL based qualitative palettes for categorical data, and divergent and sequential palettes for numerical data. The colors of a color wheel, when applied to directional data, define a new class of palette,

the circular palette. The color wheels in this paper use color in the RGB and HSV color spaces. Color wheels based on the HCL color space were not included because they are less effective in highlighting and contrasting circular-spatial structure. Kosara, Healey, Interrante, Laidlaw, and Ware (2003) recommend a color sequence with a substantial luminance component to reveal form or if detailed patterns need to be displayed.

2.7.2 Color Functions

Different color gradients, different color gradient orders, nonlinear gradients, and/or color spaces may be more effective in a particular application. For scientific visualization, Brewer (1997, p. 210) suggested the spectral sequence of red purple, red, orange, yellow, green, blue, and purple to arrange adjacent darkest and lightest colors which mark hue changes to form visually prominent color boundaries through the color sequence. The modified spectral sequence of purple, red, orange, yellow, green, and blue divides the angular range into convenient 60° bins or gradients while enjoying most of the benefits of the Brewer spectral sequence. A diverging color sequence focuses attention on a band of directions. Other color schemes recommended by Brewer (1997) can be obtained from the ColorBrewer software tool at <http://ColorBrewer.org>.

Another function of color is to distinguish among nonstructural missing data (missing ocean data), structurally missing data (land areas), and nonmissing data. In the figures of this paper, structurally missing data over the landmasses, which results in the well known shape of the continents, are indicated by tan, and missing ocean data (most notably in the regions around the South Pole) are indicated by grey or magenta colors. If possible, colors for missing data should not duplicate colors for nonmissing data. Generally, for structurally missing data over a large area, we suggest neutral colors, e.g.,

tan (sandy brown), which do not duplicate color wheel colors for data and are comfortable to view compared to an intense color like dark blue.

Various forms of human color impairment exist. Deuteranopia (reduced capability to see green) affects about 5% of males and about 0.5% of females. Protanopia (reduced capability to see red) affects about 1% of males. The tritanopic has reduced capability to see blue. Colors for the color impaired are described at <http://www.toledo-bend.com/colorblind/index.asp>. For people with red or green color deficiency, Brewer (1997) recommended the spectral sequence red, orange, yellow, blue-green, blue, and purple-blue (ROYBgBPb).

In Figure 2-6, the bottom plots are based on a ROYBgBPb color scale (purple-blue was coded as red + blue). The top plots are based on the Green-Yellow-Red-Blue (GYRB) color scale. To view these images as a color-deficient person would see them, jpeg graphics were uploaded to <http://www.vischeck.com/vischeck/vischeckImage.php>. To the deuteranopic, the left side of Figure 2-6 appears as on the right side. The bottom right plot, with yellow-grey-light blue-blue, shows more structure than the upper right plot with indistinguishable yellows for red and green. Using the Vischeck simulation, colors may be varied to develop better color scales for the color impaired.

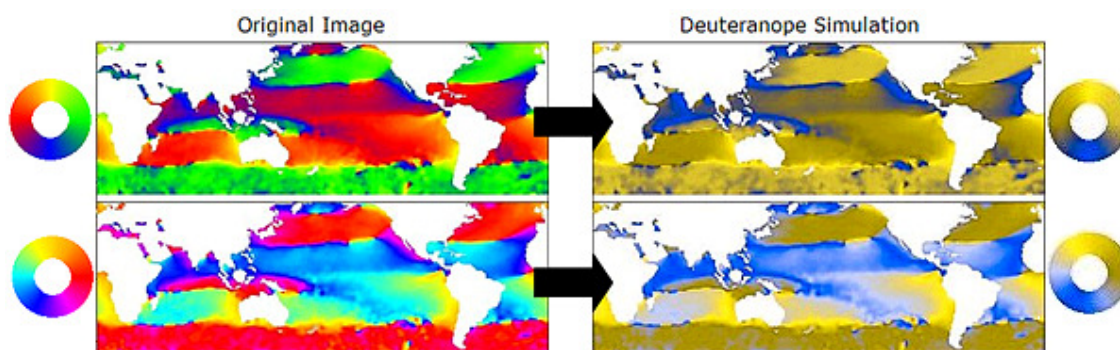


Figure 2-6. Normal and Simulated Deuteranopic Views of Images. To the deuteranope, the left plots appear as on the right with red and green being indistinguishable.

2.7.3 Color Wheel Variations

Figure 2-7 displays a variety of suitable color wheels in the RGB color space to show some of the many possibilities and to motivate experimentation to discover interesting structure in circular-spatial data. The top labels indicate which colors are used. From left to right, the continuous color wheels include the GYRB, the Black-Blue-White-Red (KBWR), and a modified Brewer Divergent, and the discrete color wheels include a modified Brewer Divergent, the KBWR, and the Rainbow color wheel. The modified Brewer divergent color wheels were constructed by connecting the ends of the Brewer 10-color divergent sequence #6 at <http://ColorBrewer.org> together, and replacing the dark color at one of the ends with an average of the dark colors at both ends. The KBWR discrete sequence was constructed from main colors of black, blue, white, and red inserting intermediate colors 1/3 and 2/3 of the way between main colors by varying levels of red, green, and blue. Brewer's (1994) 3 x 3 arrays of color in "generalized set of color schemes" at <http://www.personal.psu.edu/cab38/ColorSch/Schemes.html> provide additional sequences for discrete color wheels by omitting the center color, or by cycling around a pair of adjacent rows or columns.

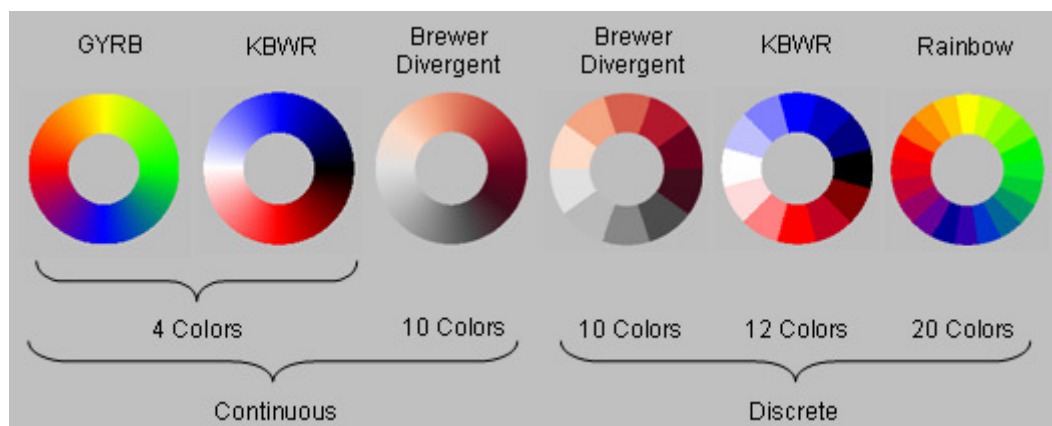


Figure 2-7. Variety of Continuous and Discrete Color Wheels.

Figure 2-8 shows the effects of different continuous and discrete color wheels on noisy and smoothed data. The continuous color wheels produce smoother overall pictures with "soft" or fuzzy color boundaries and show maximum directional detail. The discrete color wheels provide for exact quantification of regions with similar values via "hard" color boundaries between adjacent ranges of direction. Color wheels with 20 bins show more directional detail than color wheels with 10 or 12 bins. Similar to histogram binning, the appearance of the image will vary with the choice of the bin origin and the bin width. The choice of discrete or continuous color wheel depends on the function of the circular data image.

More color wheels and effects can be obtained by rotating a color wheel. In Figure 2-9, the amount and direction of rotation is shown at the center of the GYRB color wheels. Missing values are indicated by magenta and tan, which are not in the color wheel. For focus on the equatorial region using the GYRB color wheel, the bottom plot (90°) is best and top plot worst. In the bottom subplot, yellow at 180° indicates wind blowing from east to west. The adjacent colors of green or red shade yellow to indicate deviations toward the north or south, respectively. Choose a rotation that best contrasts and highlights structure in an area of interest and is comfortable to view.

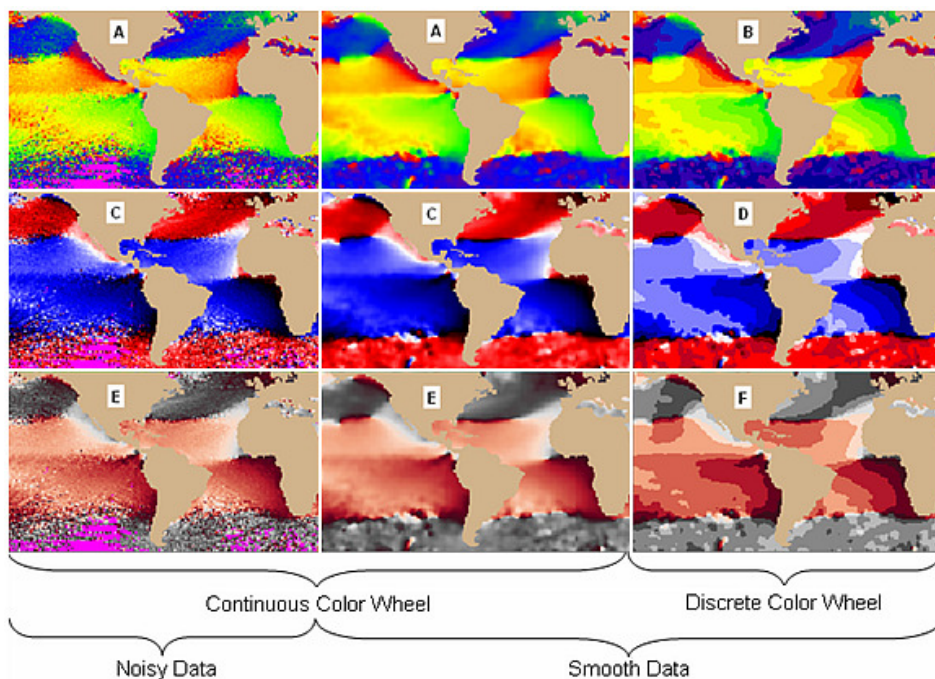


Figure 2-8. Effects of Color Wheel and Smoothness of Data. Color scales rotated 90° : (a) GYRB; (b) rainbow; (c), (d) KBWR; (e), (f) Brewer divergent. Choice of a color wheel depends on the objective of the circular dataimage.

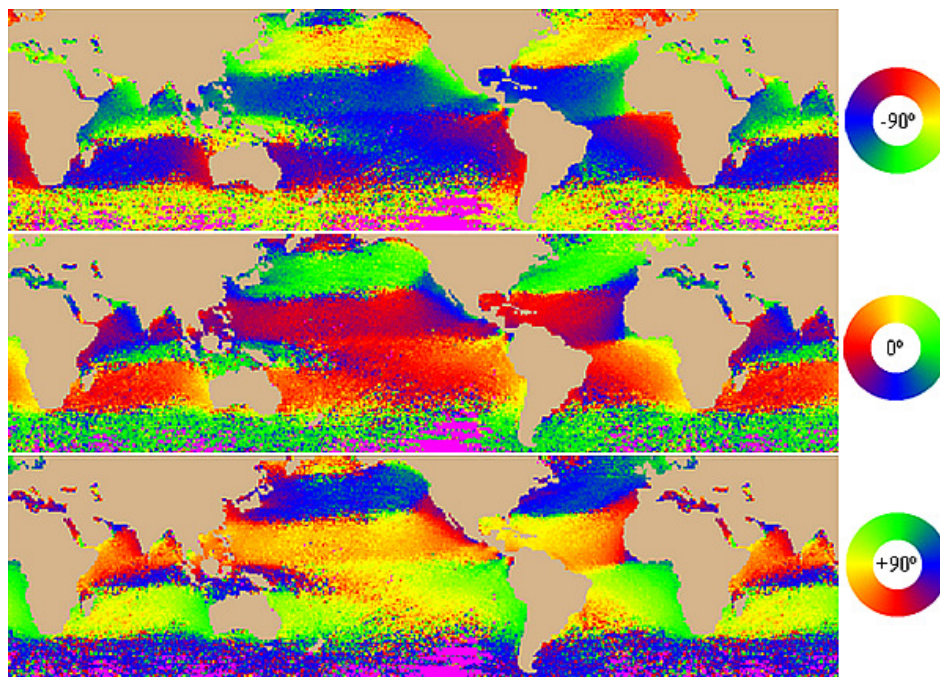


Figure 2-9. Effects of Color Wheel Rotation, Color Wheel Labeled with Rotation. The color wheel is rotated to effectively highlight structure in an area of interest. For focus on the equator using the GYRB color wheel, the bottom plot is best and top plot worst.

2.7.4 Summary of Benefits to Be Obtained from Using Different Color Schemes

- The Brewer spectral scheme red-purple, red, orange, yellow, green, blue, and purple forms visually prominent color boundaries adding structure to the circular dataimage.
- Diverging color schemes focus attention on a band of directions (Figure 2-7, middle four color wheels, and Figure 2-8, bottom).
- Colors not contained in the color wheel can be used to distinguish nonstructural missing data (e.g., missing ocean data), and structurally missing data (e.g., land areas) from nonmissing data (Figure 2-8). Neutral colors that are not used in the color wheel, e.g., tan, do not distract from colors used for data. A contrasting color not in the color wheel makes it easy to see missing data.
- The continuous color wheels give "soft" color boundaries and show maximum detail (Figure 2-8, left and middle plots). The discrete color wheels give "hard" color boundaries for the exact identification and quantification of regions with similar values (Figure 2-8, right plot) for low to moderate number of colors.
- Rotation helps to select a color sequence that contrasts and highlights structure in an area of interest (Figure 2-9, bottom, for the equatorial region). Changing from a light color, e.g., yellow or green, toward a dark color, e.g., blue or red, seems to be more effective than changing from a dark color toward a light color. Special color sequences and rotations increase the ability of color impaired viewers to recognize structure in an area of interest.

2.7.5 Circular Dataimage Variations

A single gradient of color could be used to interactively focus on a range of directions. Using the single color gradient black to white with black for the main direction and vanishing to white at 45° (or another arbitrarily selected cutoff) away from the main direction will give some sense of where the main direction lies, show some structure related to the main direction, and eliminate structure in the vanished areas.

Figure 2-10, which is here named a "focus" plot, augments the black-white color gradient by highlighting in green any direction that is within a tolerance of the main direction. Interactively, the user enters a "focal" direction and a tolerance in degrees ($^\circ$). In the bottom plot of Figure 2-10, the focal direction is 180° for wind blowing from the east to the west and the tolerance is 1° . The green pixels represent areas with directions in the range of 179° to 181° . The white areas have directions more than 45° away from the focal direction.

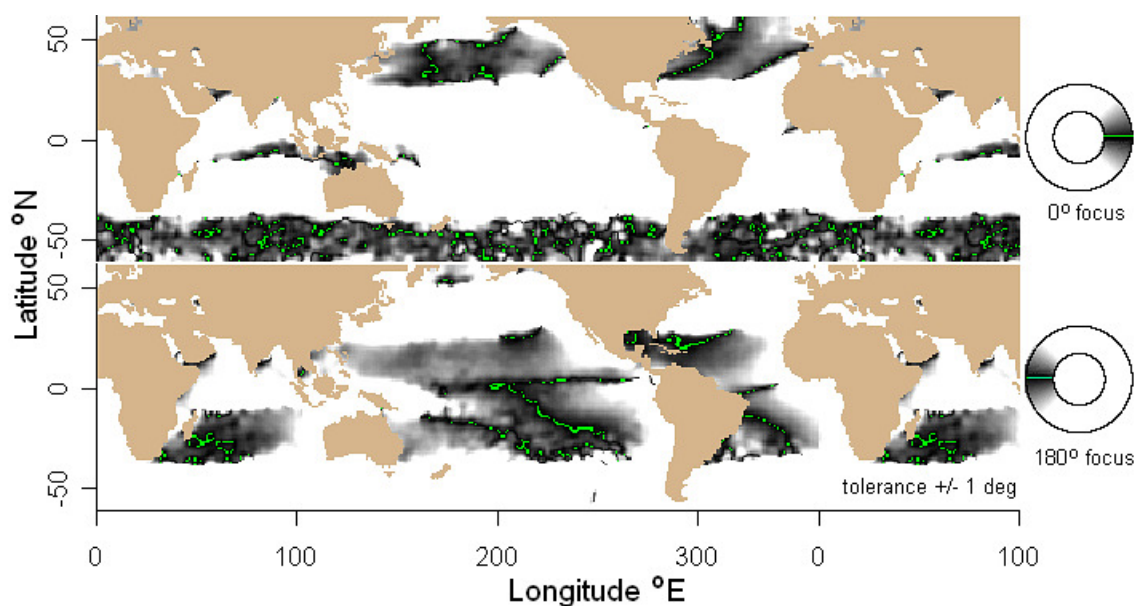


Figure 2-10. Focus Plots of Smoothed Average Direction with Focal Directions 0° (Top) and 180° (Bottom). Shading fades to white at 45° from focal direction. Areas within 1° tolerance of focal direction are colored green.

An axial random variable takes a random axis orientation when there is no reason to distinguish a direction from its opposite direction, e.g., fault lines. Figure 2-11 illustrates an axial focus plot of wind direction that highlights two specific axial directions or orientations. The top of Figure 2-11 combines the top and bottom of Figure 2-10 in one single plot. While this plot type is not particularly useful for the running example of ocean wind data, such a plot will make much more sense for true axial-spatial data.

Overall, this chapter has dealt with direction. Magnitude will now be briefly discussed. The use of arrows to represent direction and strength has been common practice. For example, see the “Vector Maps” at http://www.ssg-surfer.com/html/surfer_details.html (Scientific Software Group 2008). In CFD visualization, 3D perspectives of the paths of particles in a flow are colorized by magnitude, e.g., temperature or pressure. For example, see the flow curves at <http://www.fluent.com/solutions/examples/x209.htm>.

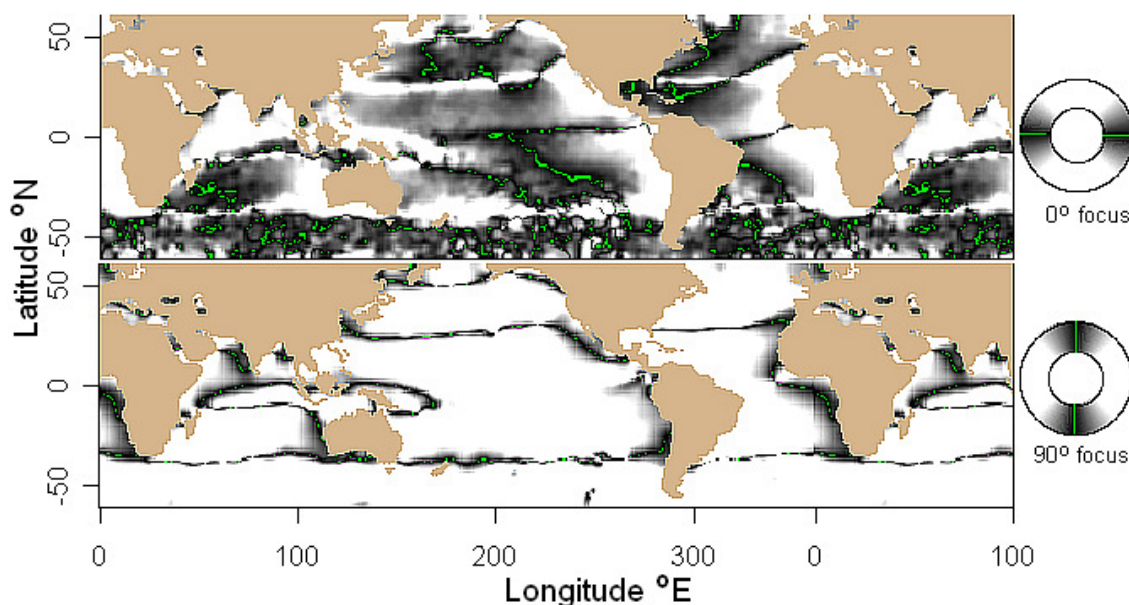


Figure 2-11. Axial Focus Plots of Smoothed Average Direction with Axial Focal Directions 0° (Top) and 90° (Bottom). Shading fades to white at 45° from focal direction. Areas within 1° tolerance of focal direction are coded green.

Going from arrows to the circular dataimage, vector magnitude is lost. However, circular dataimages can be enhanced with magnitude information. Suggestions include representing magnitude using alpha-blending or lightness. In Figure 2-12, wind speed is binned by quartiles and coded as quarter values (V) of the HSV scheme. Direction is binned in 45° intervals and coded as hue (H) in 45° increments. The outer ring represents the fourth quartile of wind strength. This enhancement provides some useful strength structure. Visual extraction of direction and strength was difficult when binning strength by few levels of value with hue a continuous function of direction. Hence, a few levels of value and a few levels of hue are suggested. Traditional methods, e.g., adding contour curves such as in Figure 2-13, adding arrows of variable length corresponding to magnitude to a circular dataimage, or adding a heatmap (linear dataimage) of magnitude to the side of a circular dataimage are effective.

Finally how can the circular dataimage be used to discover new patterns? Changes in pattern may be discovered by imaging the difference of directions with respect to two conditions, e.g., El Niño periods vs. other periods, and looking for changes in shape and color.

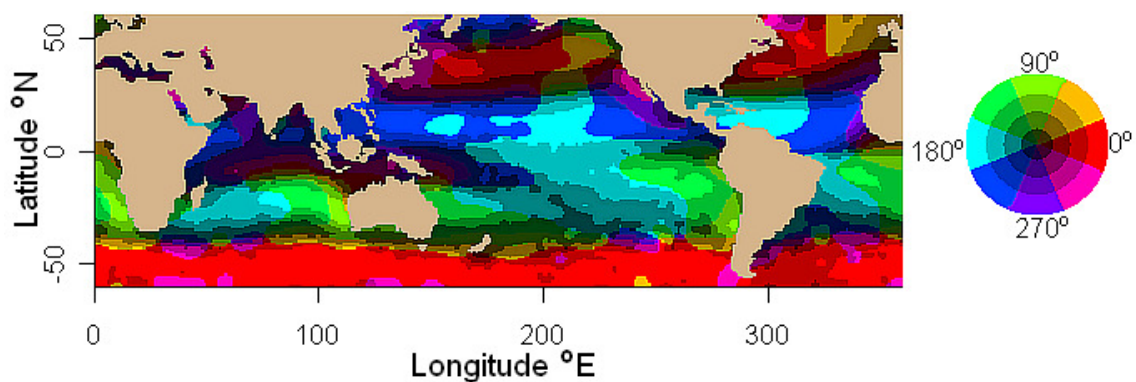


Figure 2-12. Strength Binned by Quartiles and Coded as Value (V) in HSV Scheme. In the color wheel, value increases from center outward in quarters corresponding to strength in quartiles. Direction is binned by hue into 45° bins.

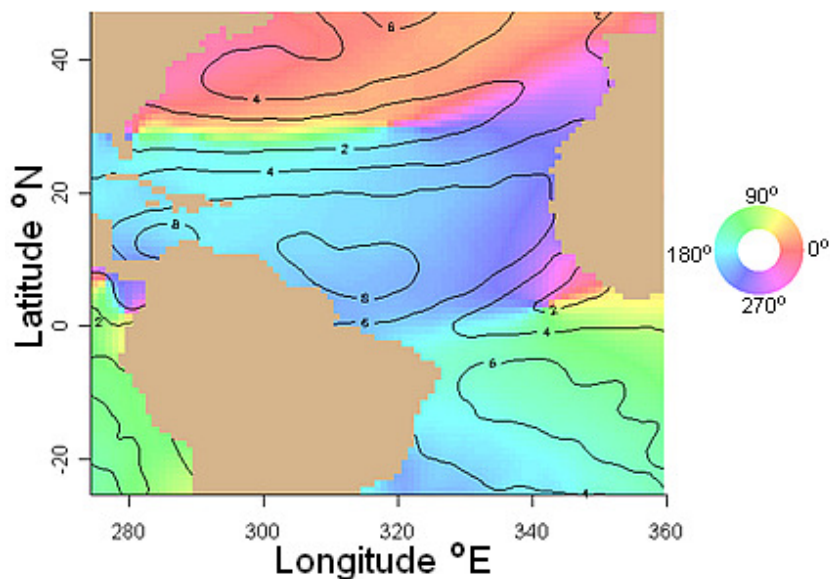


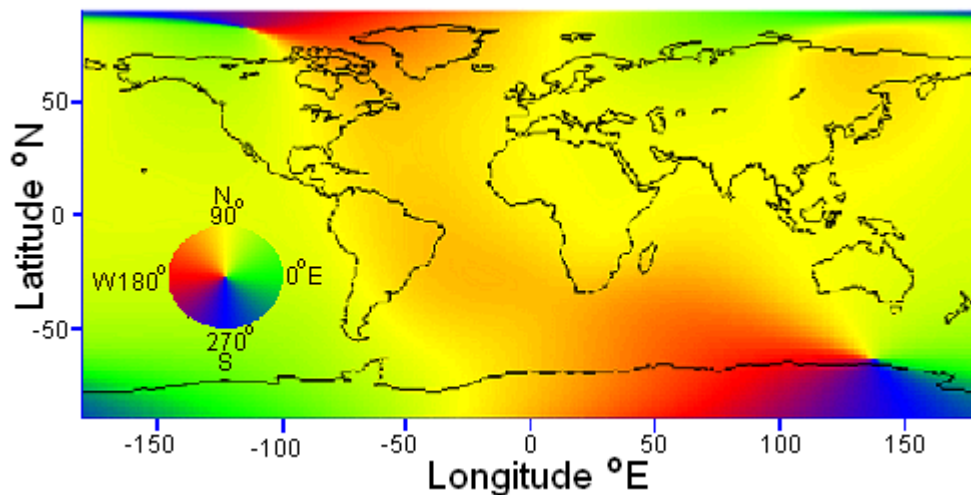
Figure 2-13. Circular Dataimage of Wind with Direction Coded Using HSV Color Wheel and Magnitude (m/s) Plotted as Contour Curves.

2.8 Other Examples

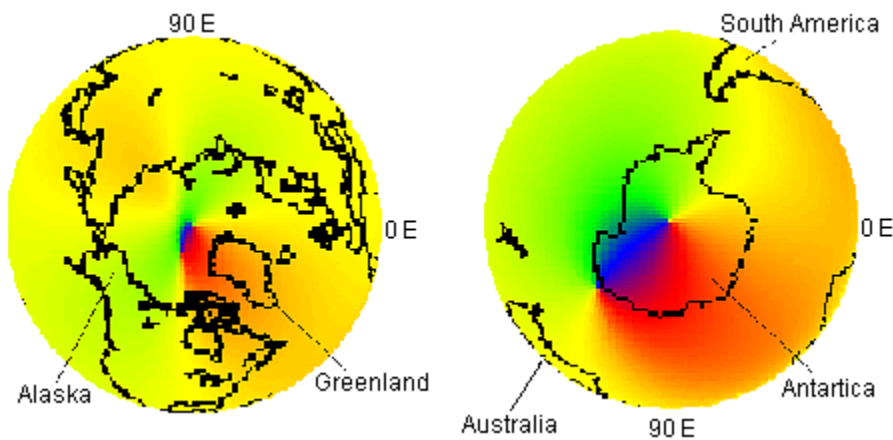
2.8.1 Earth Main Magnetic H Field Direction

The International Geomagnetic Reference Field (IGRF) models in Figures 2-14 and 2-15 were extracted without cost from the National Geophysical Data Center (NGDC) at <http://www.ngdc.noaa.gov/geomagmodels/IGRFGrid.jsp>. The scientific domain of the NGDC spans the distance from the bottom of the sea to the surface of the sun, providing data describing the marine, solid Earth, and terrestrial-solar environments. The total magnetic field at any point on the Earth's surface derives from multiple sources. The main field, which generates more than 90% of the total field, is generated in Earth's outer core. For more information, go to Frequently Asked Questions at <http://www.ngdc.noaa.gov/geomag/faqgeom.shtml>.

Figures 2-14 (a) to (c) image the direction of the Earth main magnetic horizontal (H) field for 9/15/2004 and elevation 0 km using 65,340 observations of longitude, latitude, and east and north components of the magnetic field in nano Tesla (nT).



(a)



(b)

(c)

Figure 2-14. Circular Dataimage of Earth Main Magnetic H Field Direction. IGRF model for elevation 0 km on 9/15/2004 with a GYRB color wheel (green = East = 0°).

The choice of GYRB color wheel results in subtle green and red color shadings around yellow to emphasize north and show directional detail. In Figure 2-14 (a), the backward “S” shaped yellow band crossing North and South America codes north. The rectangular plot distorts the pattern of direction, especially at the poles. Figures 2-14 (b) and (c) correct this distortion by mapping the color onto a sphere and displaying it in perspective (spherical circular dataimage). Figure 2-14 (b) shows the northern hemisphere. Starting at the center and going outward, latitude decreases from $+90^\circ$ N

to 0° N at the equator. Figure 2-14 (c) shows the southern hemisphere. Starting at the outside of Figure 2-14 (c) and going toward the center, latitude decreases from 0° N to -90° N (+90° S).

Figure 2-15, which was constructed with rgl (Adler 2009) using the R code in Appendix K.17, plots models of Earth horizontal (H) magnetic field intensity for January 1900, 1950, and 2000. The surfaces are 3D polar plots of intensity as radius at angles of longitude and latitude. The plot surface color is direction on the GYRB color wheel. The heavy red, green, and blue lines are, respectively, 0° longitude and latitude, 90° longitude and 0° latitude, and 90° latitude. The bulge near the South Pole appears to be changing shape. Figure 2-16 shows the asymmetry of the Earth main magnetic H field model of 1/1/2000 via 45° rotations about the horizontal red axis, top out of the page.

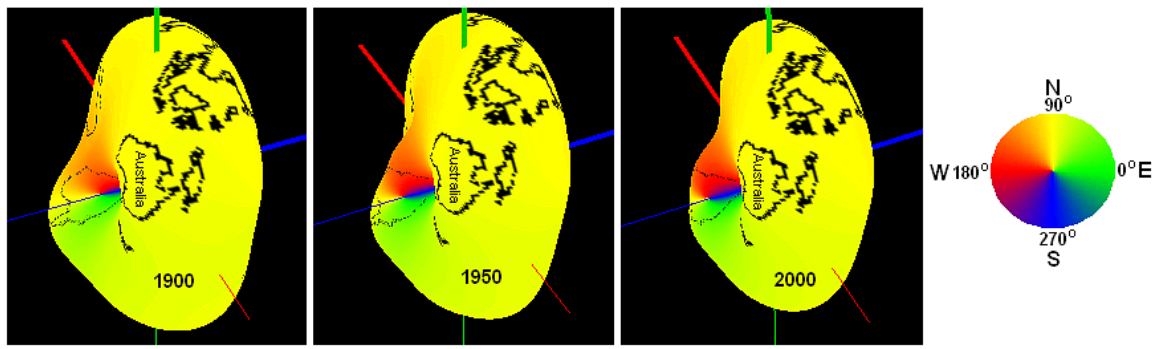


Figure 2-15. 3D Polar Plot of Earth Main Magnetic H Field Model with Direction as a Color and Magnitude as Radius for 1/1/1900, 1/1/1950, and 1/1/2000.

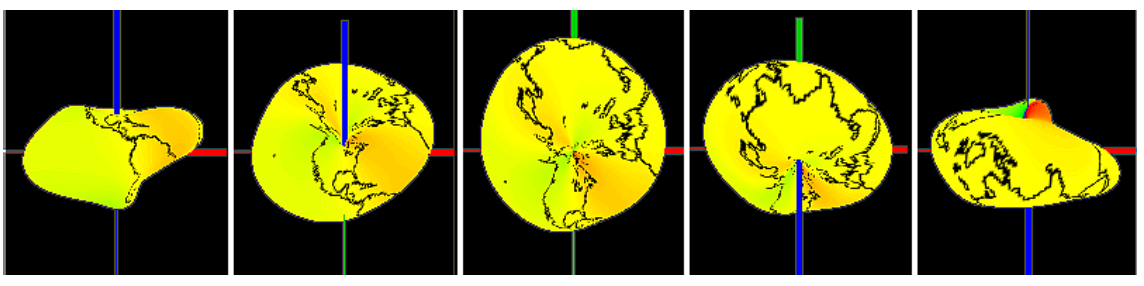


Figure 2-16. Asymmetry of Earth Main Magnetic H Field Model 1/1/2000 Demonstrated by 45° Rotations about the Horizontal Axis Through 0°-180° Longitude at the Equator.

2.8.2 Space Shuttle Solid Rocket Motor Nozzle Internal Flow

Each Space Shuttle is boosted by two solid rocket motors (Figure 2-17a), each 126 feet long and 12 feet in diameter. Figure 2-17 (b) shows an enlarged upper cross section of the nozzle. Gaseous combustion products with entrained liquid aluminum/alumina droplets enter the nozzle at subsonic speeds and accelerate to a Mach number of 1. Maximum compression occurs at the throat where the nozzle internal diameter is minimal. Aft (to the right) of the throat, nozzle diameter increases. Gases exiting the throat to the right expand, increase in velocity to supersonic speeds, and generate thrust.

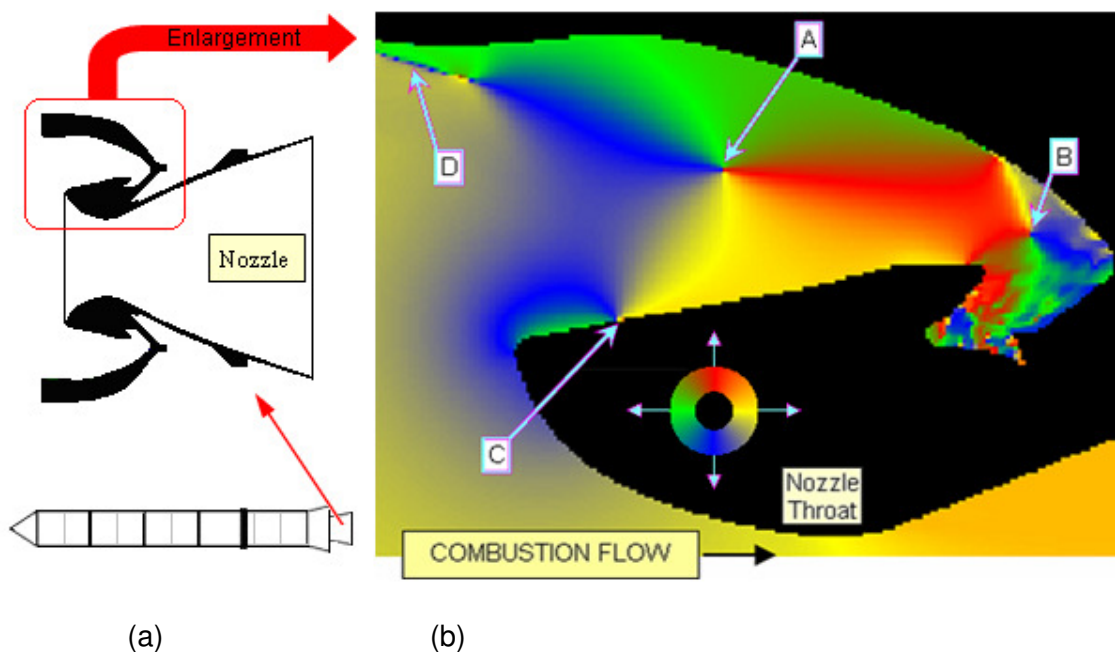


Figure 2-17. Space Shuttle Booster, Nozzle, and Nozzle Internal Combustion Flow. (a) Space Shuttle solid rocket motor booster and section view of nozzle. (b) enlarged nozzle section (black), YRGB color wheel with direction of flow, and circular dataimage of internal combustion flow. Interesting features include two counter-rotating vortices A and B, flow impingement on the nozzle surface at C, and a narrow particle shear zone at D.

The flow direction imaged in Figure 2-17 (b) was computed with FLUENT, CFD software (FLUENT 2008), for the Space Shuttle solid rocket motor nozzle at 67 seconds from ignition (See Appendix O for permissions). The dataset comprises 30,351 points of four variables (axial and radial coordinates in meters (m), and axial and radial speed in m/sec as computed from the CFD model). With direction of flow aft (right) = yellow, upward = red, forward (left) = green, and down = blue, the large CCW pattern in the cavity above the nozzle throat and centered at point A indicates a CCW flow. The smaller pattern at the right end of the cavity and centered at point B indicates a clockwise flow. These two vortices mesh like oppositely rotating gears. At point C, combustion products flowing down impact the nozzle surface, and rapidly turn forward. D is a high shear zone where particle breakup occurs. The circular dataimage easily shows much more circular-spatial structure than an arrow plot, although an arrow plot is frequently used to plot this rocket nozzle flow data. In particular, an arrow plot easily could miss the narrow high shear zone at D.

2.8.3 Space Shuttle Solid Rocket Motor Nozzle Circular Time Series

Nozzle direction angle is the angle a nozzle is pointing to in the plane perpendicular to the length of the motor (Figure 2-17 (a)). Figure 2-18 (a) shows the red-green-blue-yellow (RGBY) color wheel coding direction. Figure 2-18 (b) images the direction angle of a subset of 176 nozzles from 5 sec to 23 sec after ignition/liftoff in 0.04 second increments. Each narrow horizontal strip is a circular time series of the direction angle of a nozzle. The horizontal strips are vertically ordered to show circular-temporal structure. First, the horizontal strips are ordered vertically by left-side nozzle and right-side nozzle, second by angle in degrees of the Space Shuttle trajectory relative to the Earth equatorial plane, and last by orbital altitude in nautical miles (nm). In bottom half

of Figure 2-18 (b), the dominant red indicates that the left-side nozzle tends to vary about the 0° location, and the dominant blue above indicates that the right-side rocket nozzle tends to vary about the 180° location. This means that the left and right nozzles tend to be pointed toward Earth. The left-side blue vertical band beginning at about 7.5 sec signals the turning of the left-side nozzle to initiate the Space Shuttle roll maneuver as illustrated in Figure 2-19. In Figure 2-18 (a), the red diagonal structure of the right-side nozzle beginning about 15 sec signals the braking of the roll maneuver.

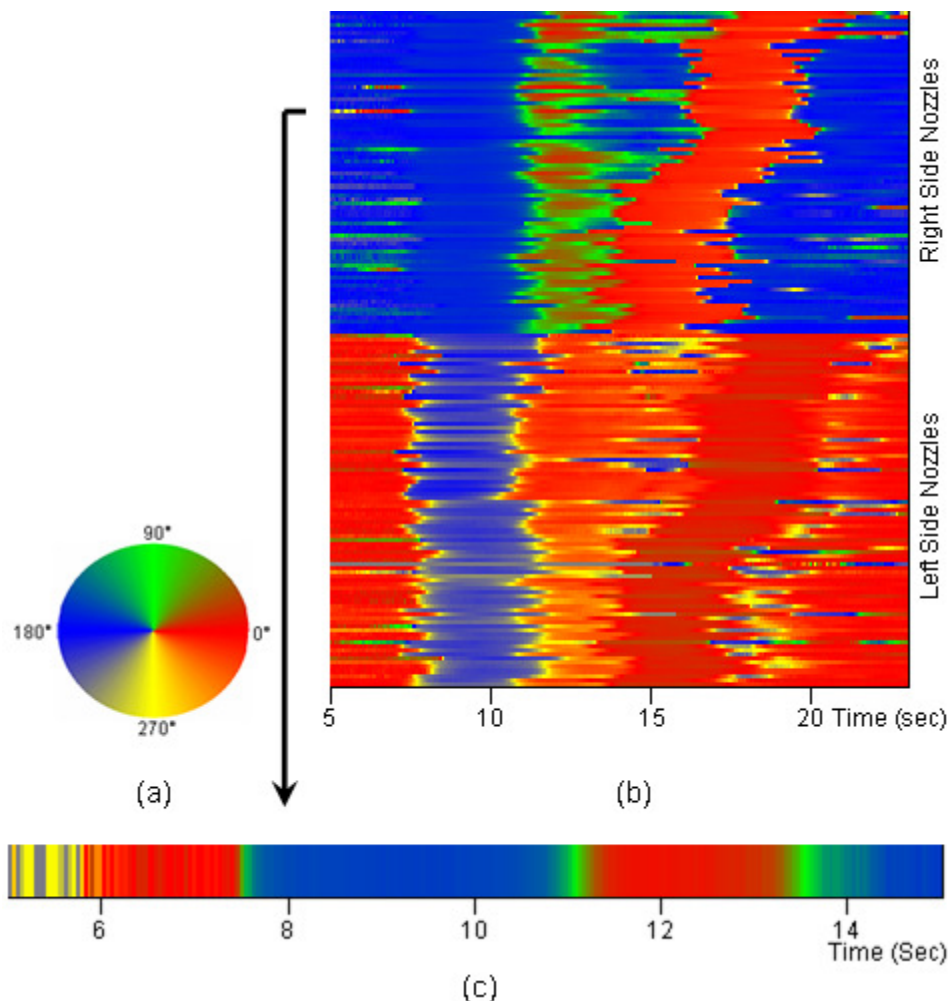


Figure 2-18. Time Series of the Space Shuttle Booster Nozzle Direction Angle. Direction angle is the direction the nozzle is pointing toward in a plane perpendicular to the booster axis. (a) RGBY color wheel, (b) circular time series families, and (c) enlargement of one time series. The vertical and diagonal structures in (b) reflect roll maneuver as influenced by inclination and altitude.

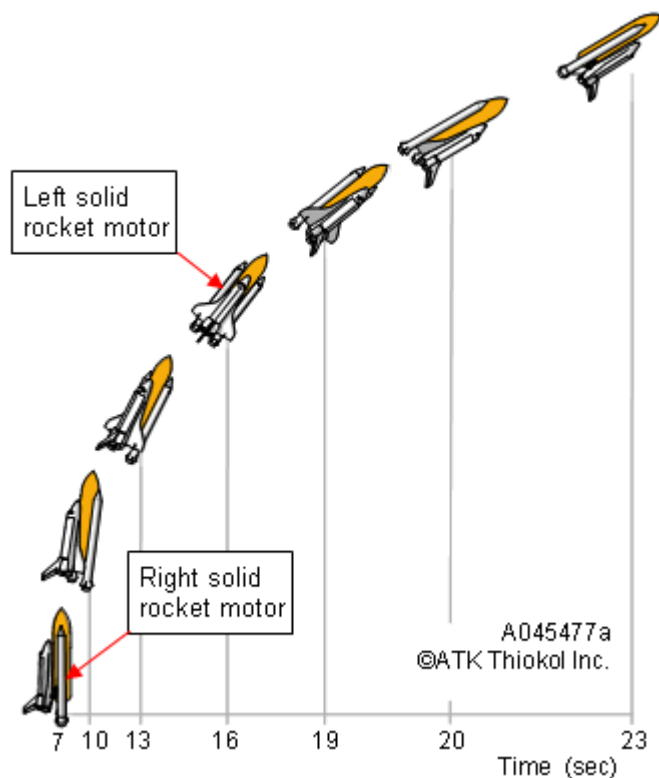


Figure 2-19. Illustration of the Space Shuttle Roll Maneuver vs. Time from Ignition. The time scale is nonlinear to show rotation in equal angular increments.

The diagonal structure in Figure 2-18 (b) from the vertical ordering of the horizontal strips shows a relationship between the nozzle direction angle and inclination and altitude, that without ordering would be obscured by numerical-alpha ordering of the nozzles. Figure 2-18 (c) is an enlargement of the horizontal strip in Figure 2-18 (b) near the tail of the large arrow connecting Figures 2-18 (b) and (c).

The roll maneuver illustrated in Figure 2-19 orients the cargo bay towards the Earth to satisfy communication, scientific, and Space Shuttle engineering requirements, and provides the astronauts with a spectacular view of Earth (Brown 2003).

2.9 Chapter Summary and Future Work

Traditional plots of circular-spatial data become less intelligible as random variation, missing data, and data density increase. These issues were solved by the circular dataimage. The circular dataimage was defined by coding direction as the color at the same angle on a color wheel, with the color wheel defined as a sequence of three or more two-color gradients with the same color between connecting gradients. The circular dataimage eliminated color discontinuity at the cross over point resulting in a continuous image of circular-spatial data, and provided an image in which fine detail on a small scale and large-scale structure on a global scale could be simultaneously recognized. Various suitable color wheels were shown and compared to motivate experimentation, the objective being to effectively contrast and comfortably view interesting circular-spatial structure. The discrete color wheel was constructed from a continuous color wheel by holding color in an angular interval to the start color of a continuous color wheel in the same interval. Variations on circular dataimages were given, e.g., the focus and axial focus plots, with interactive focus on a narrow band of directions or orientations, and direction and magnitude plots including a 3D polar plot with magnitude as radius and direction as color. Examples included global views of average wind direction, internal flow of the Space Shuttle solid rocket motor nozzle, families of circular time series of rocket nozzle vectoring, and the direction of the Earth main magnetic H field.

Future work includes R package CircSpatial implementation of an improved color wheel for deuteranopic color impairment, the focus and axial focus plots (Figures 2-10, 2-11), overlay of magnitude as contour curves (Figure 2-13) and as variable length arrows on circular dataimages, and 3D polar plots (Figure 2-15) with an overlay of features, e.g., geographical boundaries.

CHAPTER 3

COSINEOGRAM, A MEASURE OF CIRCULAR-SPATIAL CORRELATION

3.1 Introduction

This chapter defines the cosineogram, which is a graph expressing the empirical correlation of circular-spatial data. The positive definite cosine model with best fit to the cosineogram characterizes the spatial properties of circular-spatial data. This model will be used for circular kriging. Circular kriging (Chapter 4) is the estimation of circular-spatial data based on a model of circular-spatial correlation, which is a function of distance between measurement locations. Cosine models were adapted from three common covariance models from linear kriging (estimation of data of a continuous linear random variable (RV) based on a model of spatial covariance, which is a function of distance between measurement locations).

This chapter is organized as follows: Section 3.2 introduces the cosineogram and model with nugget, range, and sill similar to the nugget, range, and sill of the covariance model used for linear kriging. Section 3.3 derives the result that the theoretical sill is the square of the length of the mean resultant vector of the circular probability distribution underlying the circular-spatial data. Section 3.4 determines that the length of the mean resultant vector is the parameter, ρ , of the circular probability density function (PDF) for the selected circular distributions. Section 3.5 verifies the theoretical sill by simulation. Section 3.6 defines some cosine models (similar to covariance models used for linear kriging) for fitting to a cosineogram. Section 3.7 constructs an example cosineogram for ocean wind in a south polar region. Section 3.8 concludes with the summary and future work.

3.2 The Cosineogram

Circular-spatial data may have an underlying spatial trend where the mean direction depends on location. The random component of direction contains information about the spatial correlation. The empirical cosineogram is defined as the plot of the mean cosine of the angular distance (Figure 3-1, right) between the random components of direction vs. the Euclidean or linear distance between observation locations (Figure 3-1, left).

Let $\hat{\zeta}(d)$ be the estimate of the mean cosine, which depends on the Euclidean distance d between measurement locations, \mathbf{x}_i and \mathbf{x}_j vectors of location coordinates of observations i and j , respectively, $\|\mathbf{x}_j - \mathbf{x}_i\|$ the Euclidean distance between observation locations, $N(d)$ the number of pairs of observations separated by a distance $\|\mathbf{x}_j - \mathbf{x}_i\|$ within a tolerance ε of d , and θ_i and θ_j the measured directions at \mathbf{x}_i and \mathbf{x}_j , respectively. The cosineogram is the plot of $\hat{\zeta}(d) = \left(\frac{1}{N(d)} \right) \sum_{\|\mathbf{x}_j - \mathbf{x}_i\| - d | < \varepsilon} \cos(\theta_j - \theta_i)$ vs. d . The cosine model of spatial correlation underlying the sampling variation in the cosineogram are illustrated by Figure 3-2.

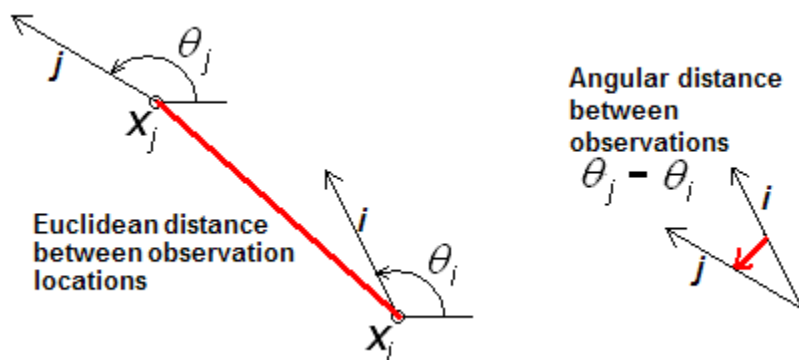


Figure 3-1. Euclidean Distance Between Locations vs. Angular Distance Between Observations. Euclidean and angular distances between observations are indicated by red lines.

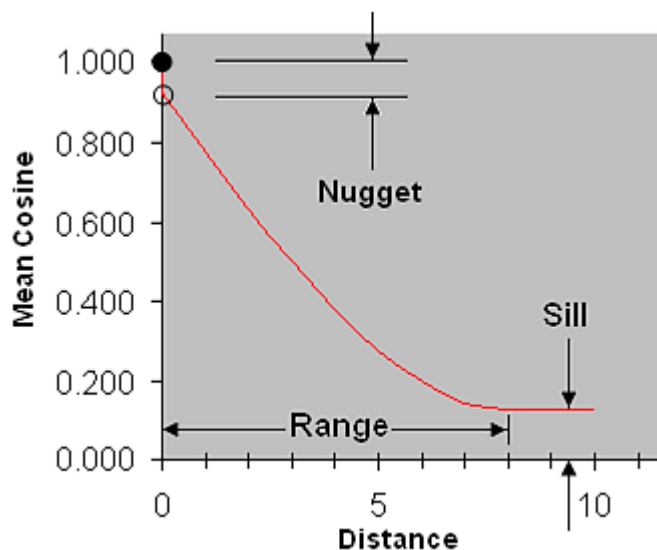


Figure 3-2. Features of the Cosine Model. The cosineogram characterizes the correlation of random components of direction vs. distance between locations. For the spherical cosine model, the sill is flat at distances beyond the range.

The value of the mean cosine of the angle between observations at zero distance is defined to be one because the angle between an observation and the same observation at the same location is zero. Measurement error may cause observations at locations which are close together (or even at the same location) to be more variable resulting in a decrease in the mean cosine. The difference between 1.0 and the mean cosine at distances approximately zero is called the nugget. As the distance between measurement locations increases, the nonrandom or spatial trend component of direction may change, and the random component of direction will have less correlation resulting in a decrease of the mean cosine of the random components of direction. For the spherical model shown in Figure 3-2, the range is defined as the distance at which the random components are no longer correlated. For other models, the practical range, which is a multiple of the range, is the distance at which random components are assumed to be uncorrelated. At distances where observations of direction are uncorrelated, the mean cosine is a constant, forming a plateau which is called the sill.

The next section will derive the result that the theoretical sill is the square of the resultant vector mean length parameter (Chapter 1, Section 1.1) of the circular distribution of the CRF (Chapter 1, Section 1.1). It is based on the new definition that the theoretical sill of the CRF is the expectation of the cosines of the small angles between pairs of independent CRVs. This definition parallels the definition of the sill of a covariogram in linear kriging. The covariogram is a plot of the empirical covariance vs. distance between observation locations. At distances where linear RVs are uncorrelated, the covariance is zero, forming a sill in the covariogram.

3.3 Derivation of the Sill

3.3.1 Review of Circular Probability Distributions and Statistics

A circular random variable (CRV) takes random directions with the total probability of all possible directions distributed on the circular support. In this chapter, direction will be expressed in radian units on the support $[0, 2\pi)$ since trigonometric functions require angles in radian units. To determine the properties of a circular probability distribution, imagine a point on a unit circle plotting a direction as the equivalent unit vector located at the origin of the unit circle with arrow head touching the unit circle. The main properties of a circular probability distribution include the resultant vector mean direction μ , which may depend on measurement location, and the resultant vector mean length ρ , which is a measure of concentration about the resultant vector mean direction and the opposite of variability about the mean, which is a measure of spread.

Let a vector be denoted by a bold lower case letter and a scalar by a nonbolded lower case letter. Let $\theta_1, \theta_2, \dots, \theta_n$ be a set of n observations of the corresponding CRVs

$\Theta_1, \Theta_2, \dots, \Theta_n$ measured in radians. With $C_n = \sum_{i=1}^n \cos \theta_i$ and $S_n = \sum_{i=1}^n \sin \theta_i$, the sample mean resultant vector direction $\bar{\theta}$ is

$$\bar{\theta} = \begin{cases} \tan^{-1}(v/h), & h > 0, v \geq 0 \\ \pi/2, & h = 0, v > 0 \\ \tan^{-1}(v/h) + \pi, & h < 0 \\ \frac{3}{2}\pi, & h = 0, v < 0 \\ \tan^{-1}(v/h) + 2\pi, & h > 0, v < 0 \\ \text{undefined}, & h = v = 0. \end{cases} \quad (3.1)$$

The population resultant vector mean direction is denoted by μ .

In terms of C_n and S_n , the sample resultant vector length is

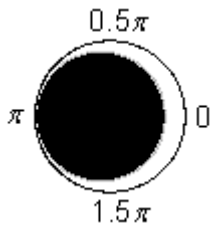
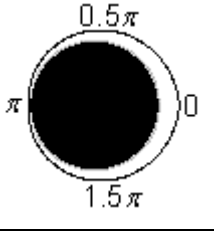
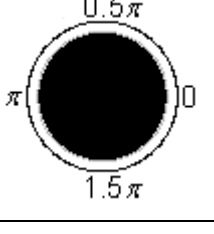
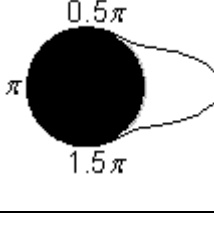
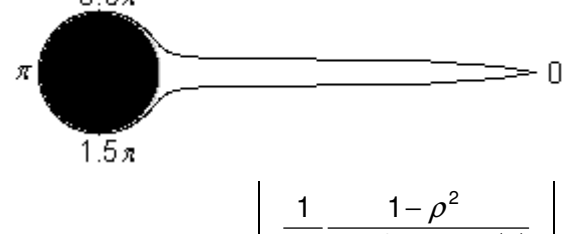
$$R_n = \sqrt{C_n^2 + S_n^2}, \quad (3.2)$$

and the sample resultant vector mean length is

$$\bar{R}_n = \frac{1}{n} R_n. \quad (3.3)$$

If all n observations have the same direction, the variability is zero, the resultant vector length $R_n = n$ (the unit vector observations of direction added tail to head are aligned and n long), and the resultant vector mean length $\bar{R}_n = \frac{1}{n} n = 1$, which is the theoretical maximum. When direction takes random values, the variability is greater than 0, the resultant vector length $R_n < n$, and $\bar{R}_n = \frac{1}{n} R_n < 1$. If n is even, and the angles between all pairs of adjacent observations of direction are equal, the variability (spread) is the theoretical maximum, the horizontal and vertical components of the unit vectors cancel, $R_n = 0$, $\bar{R}_n = \frac{1}{n} R_n = 0$, and the resultant vector mean direction $\bar{\theta}_n$ is undefined. The population mean resultant vector length is denoted by ρ . The circular distributions discussed in this dissertation were introduced in Chapter 1, Section 1.1, and are characterized in Table 3-1.

Table 3-1. Circular Probability Distributions, $\mu = 0$, $0 \leq \theta < 2\pi$ Radians. Circular density is plotted as the length of radial between black filled unit circle and outer curve.

Name of Distribution	Circular PDF Plot	Circular PDF Function	Range of Parameter	Value of ρ in PDF Plot
Cardioid		$\frac{1}{2\pi} [1 + 2\rho \cos(\theta)]$	$0 < \rho \leq 0.5$	$\rho = 0.95 \times 0.5$
Triangular		$\frac{4 - \pi^2 \rho + 2\pi\rho \pi - \theta }{8\pi}$	$0 < \rho \leq \frac{4}{\pi^2}$	$\rho = .95 \times \frac{4}{\pi^2}$
Uniform		$\frac{1}{2\pi}$	NA	NA
von Mises		$\frac{\exp(\kappa \cos(\theta))}{2\pi \sum_{j=0}^{\infty} \left(\frac{\kappa}{2}\right)^{2j} \left(\frac{1}{j!}\right)^2}$	$0 < \kappa < \infty$	$\kappa = 10.2696$ equivalent to $\rho = .95$
Wrapped Cauchy		$\frac{1}{2\pi} \frac{1 - \rho^2}{1 + \rho^2 - 2\rho \cos(\theta)}$	$0 < \rho < 1$	$\rho = 0.95 \times 1$

3.3.2 Assumptions

The dimension of the stochastic space is 2. The circular-spatial model consists of a spatial trend, with mean direction dependent on location or constant, plus a circular random field (CRF). With Θ the CRV and \mathbf{x} the location in R^2 , the CRF is the set $\{\Theta(\mathbf{x}), \mathbf{x} \in R^2\}$.

Spatial correlation increases as distance between measurement locations decreases, i.e., random components of direction (spatial trend removed) tend to be more similar as distance between observation locations decreases. In the form required by the circular kriging of Chapter 4, spatial correlation is the mean cosine of the angle between random components of directions vs. distance between observation locations. It is assumed that the spatial correlation is isotropic, i.e., it is independent of the geographic direction in which sampling is performed. If the spatial correlation varied with geographic direction (anisotropic) and sampling was performed in directions with different spatial correlation, the estimate of spatial correlation (mean cosine vs. distance) would be some average over geographic directions, and less accurate for a particular direction. Geometric anisotropy, where the sill is constant and the range varies with the spatial direction in which observations are taken, requires a directional cosineogram (mean cosine computed within a tolerance of a specified geographic direction) and applies to the geographic area over which the directional cosineogram is computed .

3.3.3 The Sill a Function of Expectations

Let Θ be the CRV of the circular distribution, Θ_i and Θ_j be two random directions, $0 \leq \Theta_i, \Theta_j < 2\pi$, with equivalent unit vector denoted by \mathbf{i} and \mathbf{j} . Also, let E be the expectation operator, and D be the smallest angle in radians between two independent random directions of a circular probability distribution, $0 \leq D \leq \pi$. Define the

will be the $E\{\cos(D)\}$. The result will be derived for the cases shown in Figure 3-3. Either

$0 \leq |\Theta_j - \Theta_i| < \pi$ (Case 1), or $\pi \leq |\Theta_j - \Theta_i| < 2\pi$ (Case 2).

Case 1, $0 \leq |\Theta_j - \Theta_i| < \pi$:

$$\begin{aligned}
 E\{\cos(D)\} &= E\{\cos|\Theta_j - \Theta_i|\} \\
 &= \begin{cases} E\{\cos(\Theta_j - \Theta_i)\}, & \Theta_j \geq \Theta_i \\ E\{\cos[-(\Theta_j - \Theta_i)]\}, & \Theta_j < \Theta_i \end{cases} \\
 &\stackrel{\cos(\alpha)=\cos(-\alpha)}{=} E\{\cos(\Theta_j - \Theta_i)\} \\
 &\stackrel{\text{TRIG IDENTITY}}{=} E\{\cos(\Theta_j)\cos(\Theta_i) + \sin(\Theta_j)\sin(\Theta_i)\} \\
 &= E\{\cos(\Theta_j)\cos(\Theta_i)\} + E\{\sin(\Theta_j)\sin(\Theta_i)\} \\
 &\stackrel{iid}{=} E\{\cos(\Theta)\}E\{\cos(\Theta)\} + E\{\sin(\Theta)\}E\{\sin(\Theta)\} \\
 &= [E\{\cos(\Theta)\}]^2 + [E\{\sin(\Theta)\}]^2
 \end{aligned}$$

Case 2, $\pi \leq |\Theta_j - \Theta_i| < 2\pi$:

$$\begin{aligned}
 E\{\cos(D)\} &= E\{\cos(2\pi - |\Theta_j - \Theta_i|)\} \\
 &= E\{\cos(2\pi)\cos|\Theta_j - \Theta_i| + \sin(2\pi)\sin|\Theta_j - \Theta_i|\} \\
 &= E\{\cos|\Theta_j - \Theta_i|\} \\
 &\stackrel{\text{above}}{=} [E\{\cos(\Theta)\}]^2 + [E\{\sin(\Theta)\}]^2
 \end{aligned}$$

$$\Rightarrow E\{\cos(D)\} = [E\{\cos(\Theta)\}]^2 + [E\{\sin(\Theta)\}]^2 \quad (3.4)$$

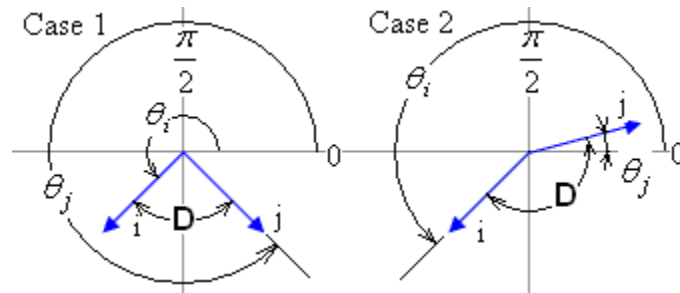


Figure 3-3. Cases of Random Directions. Directions are expressed in radian units.

Hence, complete evaluation of the sill requires knowledge of $E\{\cos(\Theta)\}$ and $E\{\sin(\Theta)\}$.

3.3.4 Expectation of the Sines

With g the PDF of Θ , for the symmetric circular distributions with $\mu = 0$, the PDF $g(\theta) = g(-\theta)$. Hence, the oppositely signed $\sin(\Theta)$ and $\sin(-\Theta)$ cancel when integrating over the full range and, hence, $E\{\sin(\Theta)\} = 0$ for $\mu = 0$. Let $\tilde{\rho}$ be the population mean resultant vector length. $E\{\sin(\Theta)\} = 0$ for $\mu = 0$, i.e., the vertical component of $\tilde{\rho}$ is zero. Hence

$$\tilde{\rho} = E\{\cos(\Theta)\}. \quad (3.5)$$

3.4 Expectation of the Cosines

From (3.5), the population mean resultant vector length $\tilde{\rho} = E\{\cos(\Theta)\}$ for $\mu = 0$. To evaluate $E\{\cos(\Theta)\}$, the PDFs for circular distributions summarized in Table 3-1 were obtained from Mardia (1972), Fisher (1993), and Jammalamadaka and SenGupta (2001). In the subsections 3.4.1 to 3.4.5, it will be shown that $\tilde{\rho}$ is the parameter ρ of the circular PDFs for the selected circular probability distributions. This step is necessary as it is not immediately obvious that the parameter ρ is the population mean resultant vector length for the selected distributions, as claimed by Fisher (1993).

3.4.1 Cardioid

$$\begin{aligned}
E\{\cos(\Theta) | \text{Cardioid}\} &= \int_0^{2\pi} \cos(\theta) \underbrace{\left(\frac{1}{2\pi} [1 + 2\rho \cos(\theta)] \right)}_{\text{DENSITY}} d\theta \\
&= \frac{1}{2\pi} \int_0^{2\pi} (\cos(\theta) + 2\rho \cos^2 \theta) d\theta \\
&= \frac{1}{2\pi} \int_0^{2\pi} \cos(\theta) d\theta + \frac{1}{2\pi} \int_0^{2\pi} 2\rho \cos^2 \theta d\theta \\
&= \frac{1}{2\pi} \sin(\theta) \Big|_0^{2\pi} + \frac{1}{\pi} \int_0^{2\pi} \rho \cos^2 \theta d\theta \\
&= 0 + \frac{\rho}{\pi} \int_0^{2\pi} \frac{1 + \cos(2\theta)}{2} d\theta \\
&= \frac{\rho}{2\pi} \int_0^{2\pi} 1 d\theta + \frac{\rho}{2\pi} \int_0^{2\pi} \cos(2\theta) d\theta \\
&= \frac{\rho}{2\pi} \theta \Big|_0^{2\pi} + \frac{1}{2} \frac{\rho}{2\pi} \int_0^{2\pi} \cos(2\theta) 2d\theta \\
&= \frac{\rho}{2\pi} 2\pi + \frac{\rho}{4\pi} \sin(2\theta) \Big|_0^{2\pi} \\
&= \rho + \frac{\rho}{4\pi} \sin(0) \\
&= \rho \Rightarrow
\end{aligned}$$

$$E\{\cos(\Theta) | \text{Cardioid}\} = \rho \quad (3.6)$$

3.4.2 Triangular

$$\begin{aligned}
E\{\cos(\Theta) | \text{Triangular}\} &= \int_0^{2\pi} \cos(\theta) \underbrace{\frac{1}{8\pi} (4 - \pi^2 \rho + 2\pi\rho |\pi - \theta|)}_{\text{DENSITY}} d\theta \\
&= \frac{1}{8\pi} \int_0^{2\pi} \cos(\theta) (4 - \pi^2 \rho) d\theta + \frac{1}{8\pi} \int_0^{2\pi} \cos(\theta) 2\pi\rho |\pi - \theta| d\theta \\
&= \frac{(4 - \pi^2 \rho)}{8\pi} \sin(\theta) \Big|_0^{2\pi} + 2\pi\rho \frac{1}{8\pi} \int_0^{\pi} \cos(\theta) (\pi - \theta) d\theta \\
&\quad + 2\pi\rho \frac{1}{8\pi} \int_{\pi}^{2\pi} \cos(\theta) (\theta - \pi) d\theta \\
&= 0 + 2\pi\rho \frac{\pi}{8\pi} \int_0^{\pi} \cos(\theta) d\theta - 2\pi\rho \frac{1}{8\pi} \int_0^{\pi} \theta \cos(\theta) d\theta \\
&\quad - 2\pi\rho \frac{\pi}{8\pi} \int_{\pi}^{2\pi} \cos(\theta) d\theta + 2\pi\rho \frac{1}{8\pi} \int_{\pi}^{2\pi} \theta \cos(\theta) d\theta
\end{aligned}$$

$$\begin{aligned}
&= \frac{\pi\rho}{4} \int_0^\pi \cos(\theta) d\theta - \frac{\rho}{4} \int_0^\pi \theta \cos(\theta) d\theta - \frac{\pi\rho}{4} \int_\pi^{2\pi} \cos(\theta) d\theta \\
&\quad + \frac{\rho}{4} \int_\pi^{2\pi} \theta \cos(\theta) d\theta \\
&= \frac{\pi\rho}{4} \sin(\theta) \Big|_0^\pi - \frac{\rho}{4} \{\cos(\theta) + \theta \sin(\theta)\} \Big|_0^\pi \\
&\quad - \frac{\pi\rho}{4} \sin(\theta) \Big|_\pi^{2\pi} + \frac{\rho}{4} \{\cos(\theta) + \theta \sin(\theta)\} \Big|_\pi^{2\pi} \\
&= 0 - \frac{\rho}{4} \{-1 - 1 + 0\} - 0 + \frac{\rho}{4} \{1 - -1 + 0\} \\
&= -\frac{\rho}{4} \{-2\} + \frac{\rho}{4} \{2\} \\
&= \frac{\rho}{2} + \frac{\rho}{2} \\
&= \rho \Rightarrow
\end{aligned}$$

$$E\{\cos(\Theta) | \text{Triangular}\} = \rho \quad (3.7)$$

3.4.3 Uniform

$$\begin{aligned}
E\{\cos(\Theta) | \text{Uniform}\} &= \int_0^{2\pi} \cos(\theta) \underbrace{[2\pi]^{-1}}_{\text{DENSITY}} d\theta = \\
&= \frac{1}{2\pi} \sin(\theta) \Big|_0^{2\pi} \\
&= 0 \\
&\stackrel{\rho=0}{=} \rho \Rightarrow
\end{aligned}$$

$$E\{\cos(\Theta) | \text{Uniform}\} = \rho \quad (3.8)$$

3.4.4 von Mises

$$\text{With } I_0(\kappa) = \frac{1}{2\pi} \int_0^{2\pi} \exp(\kappa \cos(\theta)) d\theta \text{ and } I_1(\kappa) = \frac{1}{2\pi} \int_0^{2\pi} \cos(\theta) \exp(\kappa \cos(\theta)) d\theta$$

(Jammalamadaka and SenGupta 2001, p. 288), $A_1(\kappa) = \frac{I_1(\kappa)}{I_0(\kappa)}$ (Fisher 1993, p. 50, eq.

3.36).

$$\begin{aligned}
E\{\cos(\Theta) | \text{von Mises}\} &= \int_0^{2\pi} \cos(\theta) \underbrace{\frac{1}{2\pi I_0(\kappa)} \exp(\kappa \cos(\theta))}_{\text{DENSITY}} d\theta \\
&= \frac{1}{I_0(\kappa)} \frac{1}{2\pi} \int_0^{2\pi} \cos(\theta) \exp(\kappa \cos(\theta)) d\theta \\
&= \frac{1}{I_0(\kappa)} I_1(\kappa) \\
&= A_1(\kappa) \\
&\stackrel{\text{(Fisher 1993, p. 49)}}{=} \rho \Rightarrow
\end{aligned}$$

$$E\{\cos(\Theta) | \text{von Mises}\} = \rho \quad (3.9)$$

3.4.5 Wrapped Cauchy

Jammalamadaka and SenGupta (2001, p. 45) prove the equivalence of the PDF from Table 3-1 and the form of the PDF used below.

$$\begin{aligned}
E\{\cos(\Theta) | \text{WrCauchy}\} &= \int_0^{2\pi} \cos(\theta) \underbrace{\frac{1}{2\pi} \left[1 + 2 \sum_{k=1}^{\infty} \rho^k \cos(k\theta) \right]}_{\text{DENSITY}} d\theta \\
&= \frac{1}{2\pi} \int_0^{2\pi} \left[\cos(\theta) + 2 \sum_{k=1}^{\infty} \rho^k \cos(\theta) \cos(k\theta) \right] d\theta \\
&= \frac{1}{2\pi} \int_0^{2\pi} \cos(\theta) d\theta + \frac{1}{2\pi} \int_0^{2\pi} 2 \sum_{k=1}^{\infty} \rho^k \cos(\theta) \cos(k\theta) d\theta \\
&= \frac{1}{2\pi} \sin(\theta) \Big|_0^{2\pi} + \frac{1}{\pi} \sum_{k=1}^{\infty} \rho^k \int_0^{2\pi} \cos(\theta) \cos(k\theta) d\theta \\
&= 0 + \frac{1}{\pi} \rho^1 \int_0^{2\pi} \cos(\theta) \cos(\theta) d\theta + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \int_0^{2\pi} \cos(\theta) \cos(k\theta) d\theta \\
&= \frac{1}{\pi} \rho \left[\underbrace{.5 \sin(\theta) \cos(\theta) + .5\theta}_{\text{Weast 1972, p. A-136, \#302}} \right]_0^{2\pi} \\
&\quad + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \left[\underbrace{\frac{1}{2(k-1)} \sin(\theta(k-1)) + \frac{1}{2(k+1)} \sin(\theta(k+1))}_{\text{Weast 1972, p. A-137, \#317}} \right]_0^{2\pi} \\
&= \frac{1}{\pi} \rho [0 - 0 + \pi - 0] + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k [0 - 0 + 0 - 0] \\
&= \frac{1}{\pi} \rho \pi + \frac{1}{\pi} \sum_{k=2}^{\infty} 0 = \rho \Rightarrow
\end{aligned}$$

$$E\{\cos(\Theta) | \text{WrCauchy}\} = \rho \quad (3.10)$$

3.4.6 Summary of Individual Results

In summary, from (3.6) to (3.10) for the selected distributions from Table 3-1, the population mean resultant vector length $\tilde{\rho}$ equals the parameter ρ of the circular probability distribution. Let the parameter ρ also denote the mean resultant vector length. Hence, the theoretical sill is

$$E\{\cos(D)\} = \rho^2. \quad (3.11)$$

Result (3.11) has two implications. First, two circular distributions of different families (Cardioid, Triangular, von Mises, Wrapped Cauchy) with the same population mean resultant vector length, ρ , will have the same theoretical sill, ρ^2 . Hence, with the exception of the uniform circular distribution, the correspondence between the sill and a circular probability distribution is not unique. For the uniform circular distribution, $\rho = 0$ because all directions have equal probability density. The second implication is that zonal anisotropy (sill varies with direction) cannot occur in a pure CRF with one underlying circular probability distribution.

3.5 Verification of the Sill by Simulation

The theoretical sill was computed as ρ^2 for five circular probability distributions from Table 3-1. The results have been summarized in Table 3-2. The function A_1 in the Sill column of the table for the von Mises distribution is given in Subsection 3.4.4. The value of the sill of each distribution was verified by simulation.

Figures 3-4 to 3-8 were computed using the R code in Appendices K.3 and L.1. For each distribution, 1000 simulations were computed. In each simulation, 100 independent CRV were computed, the cosines of the angles between all pairs of CRV were collected, and the averages were computed for the cumulative collection of

Table 3-2. The Sill of Selected Distributions.

Distribution	Parameter Range	Selected Parameter Value	Sill, or $E\{\cos(D)\} = \rho^2$ (3.11)
Cardioid	$0 < \rho \leq 0.5$	$\rho = 0.25$	$\rho^2 = 0.25^2 = 0.062$
Triangular	$0 < \rho \leq 4/\pi^2$	$\rho = 2/\pi^2$	$\rho^2 = (2/\pi^2)^2 = 0.041$
Uniform	$\rho = 0$	$\rho = 0$	$\rho^2 = 0$
von Mises	$0 < \kappa < \infty$ ($0 < \rho < 1$)	$\kappa = 5$ (concentration)	$\rho^2 = (A_1(5))^2 = 0.798$
Wrapped Cauchy	$0 < \rho < 1$	$\rho = \exp(-1)$	$\rho^2 = (\exp(-1))^2 = 0.135$

cosines. Hence, the size of the collection increases with each simulation. Figures 3-4 to 3-8 plot the mean cosine of the angle between independent CRV vs. the number of simulations, and show that the mean cosine tends to the theoretical sill as the number of simulations increases and is consistent with the theoretical sill. In Figure 3-6, a slightly negative mean cosine developed at about 200 simulations. After 300 simulations, the mean cosine trended toward zero. With the uniform circular distribution all directions are equally likely. Hence, half of the angles are likely to occur between 0.5π and 1.5π . These have negative cosines. A negative mean cosine means there were more negative than positive cosines at the completion of a simulation in the sequence of simulations.

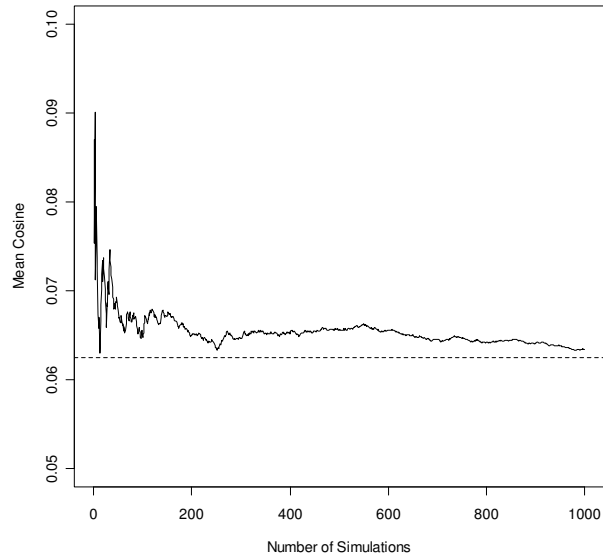


Figure 3-4. Mean Cosine of the Angle Between Independent Cardioid CRV, $\rho^2 = 0.062$, Is Consistent with the Theoretical Sill. The dashed line represents the theoretical sill.

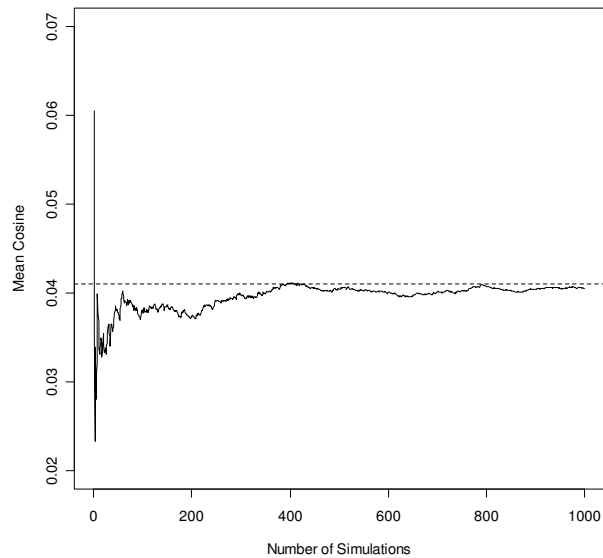


Figure 3-5. Mean Cosine of the Angle Between Independent Triangular CRV, $\rho^2 = 0.041$, Is Consistent with the Theoretical Sill. The dashed line represents the theoretical sill.

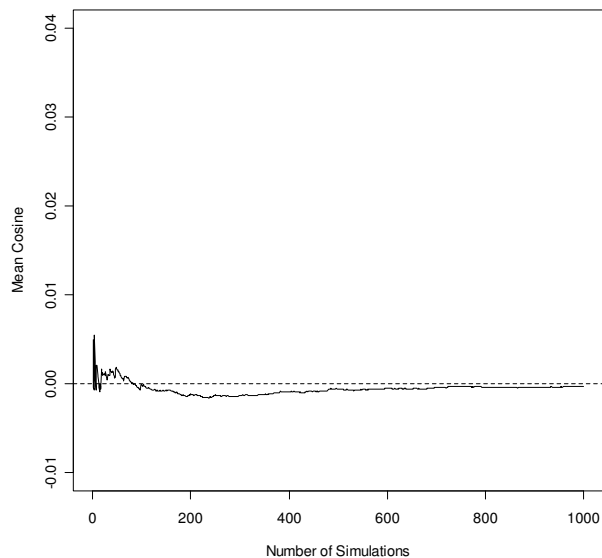


Figure 3-6. Mean Cosine of the Angle Between Independent Uniform CRV, $\rho^2 = 0$, Is Consistent with the Theoretical Sill. The dashed line represents the theoretical sill.

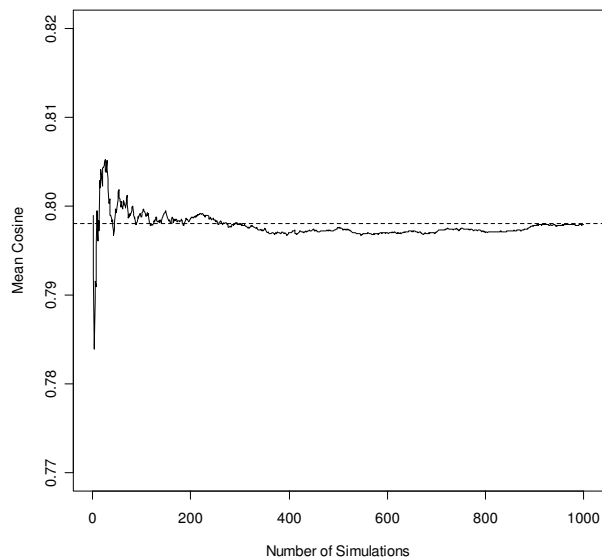


Figure 3-7. Mean Cosine of the Angle Between Independent von Mises CRV, $\rho^2 = 0.798$, Is Consistent with the Theoretical Sill. The dashed line represents the theoretical sill.

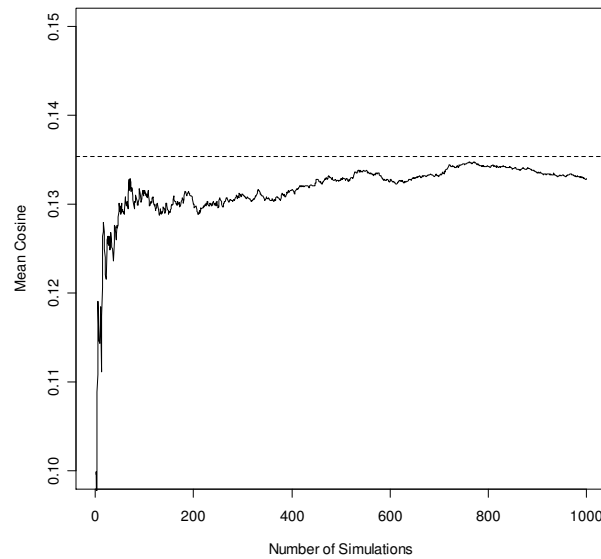


Figure 3-8. Mean Cosine of the Angle Between Independent Wrapped Cauchy CRV, $\rho^2 = 0.135$, Is Consistent with the Theoretical Sill. The dashed line represents the theoretical sill.

3.6 Cosine Models

3.6.1 Cosine Model Properties

In the previous sections, the mean cosine of the angle between independent CRV was determined to be ρ^2 . We now will consider the CRF. Covariance models used for linear kriging were derived from the semivariance models in Bailey and Gatrell (1995, pp. 179-180). These covariance models are monotonic decreasing and positive definite. Cosine models were adapted from covariance models by scaling and shifting to conform to the circular-spatial correlation in a CRF:

- At distance 0, the mean cosine equals 1.
- At distance not exactly 0, but close to 0, the mean cosine equals 1 minus the nugget.
- As distance increases, the mean cosine decreases monotonically to the sill.
- The sill equals the square of the mean resultant vector length parameter ρ of the circular probability distribution.

- Applying the cosine model to the matrix of pairwise distances produces a symmetric and positive definite matrix, which will be proved. A positive definite matrix is required for linear kriging.

3.6.2 Introductory Cosine Models Adapted from Linear Kriging

Let ρ be the mean resultant vector length of the circular probability distribution, $0 \leq \rho < 1$, and n_g be the nugget. Since the minimum value of the mean cosine is ρ^2 , the maximum nugget (Figure 3-2) is $1 - \rho^2$. Hence, $0 \leq n_g < 1 - \rho^2$. With $\zeta(d)$ the mean cosine of the angle between random components of direction a distance d apart and r the range, some introductory cosine models adapted from Bailey and Gatrell (1995, pp. 179-180) by scaling and shifting, are:

- The Exponential Cosine Model

$$\zeta(d) = \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2) \exp(-3d/r), & d > 0 \end{cases} \quad (3.12)$$

- The Gaussian Cosine Model

$$\zeta(d) = \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2) \exp(-3[d/r]^2), & d > 0 \end{cases} \quad (3.13)$$

- The Spherical Cosine Model

$$\zeta(d) = \begin{cases} 1, & d = 0 \\ 1 - n_g - (1 - n_g - \rho^2) \left(\frac{3}{2}[d/r] - \frac{1}{2}[d/r]^3 \right), & 0 < d \leq r \\ \rho^2, & d > r. \end{cases} \quad (3.14)$$

Note that the symbol $\hat{\zeta}(d)$ is used for the empirical version of the model, which is the cosineogram (Section 3.2).

Figures 3-9 to 3-11 show plots of cosine models for the selected circular distributions with range $r = 8$ and nugget $n_g = 0$. Values for the parameters ρ and κ have been chosen in accordance with Table 3-2 and with Figures 3-4 to 3-8. The parameter κ of the von Mises (vM) distribution is a measure of concentration about the mean direction equal to one half the log of the ratio of the maximum density at the mean to the minimum density at the opposite direction. The exponential model in Figure 3-9 is concave up, the Gaussian model in Figure 3-10 is “S” shaped with an inflexion point, and the spherical model in Figure 3-11 has a plateau (sill) at distances beyond the range. Additional suitable cosine models are given in Appendix M.

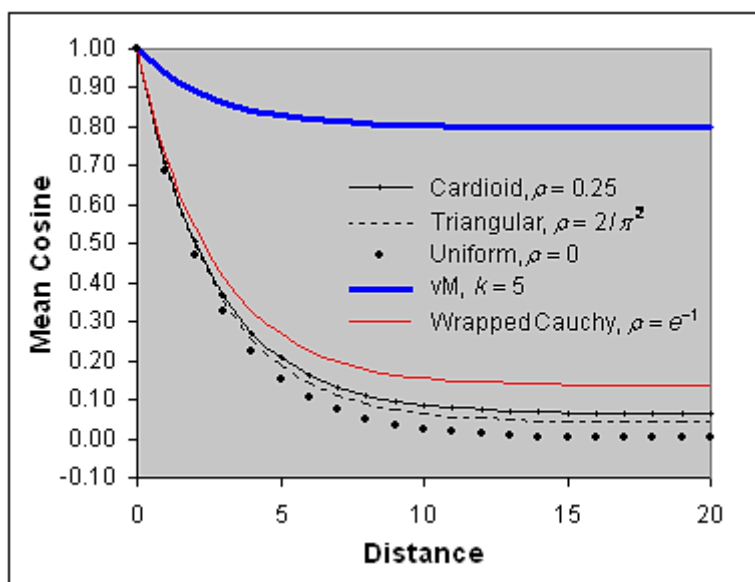


Figure 3-9. The Exponential Cosine Model.

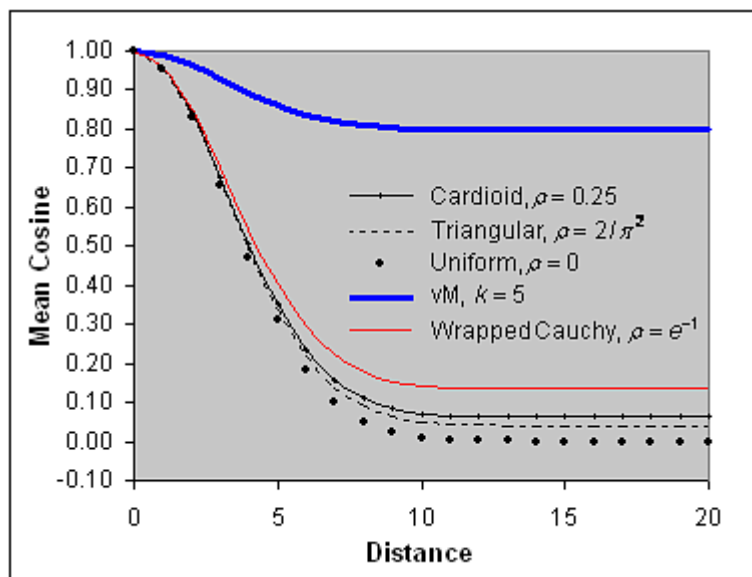


Figure 3-10. The Gaussian Cosine Model.

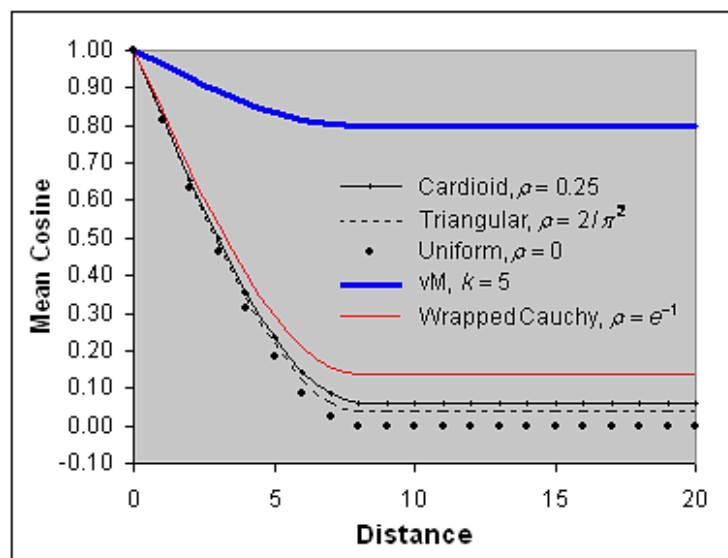


Figure 3-11. The Spherical Cosine Model.

3.6.3 The Adapted Cosine Models Are Positive Definite

Positive definiteness of the matrix of cosines is required by the circular kriging solution (Chapter 4, Section 4.3) for an optimal estimate of direction. In this subsection, it will be proven that the cosine models adapted from the positive definite covariance functions of linear kriging are positive definite. For this proof, the equivalent shifted and scaled form of the spherical cosine model in (3.14) is required.

$$\begin{aligned}
 \zeta(d) &\stackrel{(3.14)}{=} \begin{cases} 1, & d = 0 \\ 1 - n_g - (1 - n_g - \rho^2) \left(\frac{3}{2} [d/r] - \frac{1}{2} [d/r]^3 \right), & 0 < d < r \\ \rho^2, & d \geq r \end{cases} \\
 &= \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2) - (1 - n_g - \rho^2) \left(\frac{3}{2} [d/r] - \frac{1}{2} [d/r]^3 \right), & 0 < d < r \\ \rho^2, & d \geq r \end{cases} \\
 &= \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2) \left(1 - \left(\frac{3}{2} [d/r] - \frac{1}{2} [d/r]^3 \right) \right), & 0 < d < r \\ \rho^2, & d \geq r \end{cases} \Rightarrow \\
 \zeta(d) &= \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2) \left(1 - \left(\frac{3}{2} [d/r] - \frac{1}{2} [d/r]^3 \right) \right), & 0 < d < r \\ \rho^2, & d \geq r \end{cases} \quad (3.15)
 \end{aligned}$$

With $d_{ij} = d_{ji}$ the spatial distance between locations of observations i and j , let

$f(d_{ij}) = a_{ij}$ be a positive definite function of distance, i.e., a covariance function of linear kriging. Then, with k_1 and k_2 constants, the cosine models structured such as (3.12), (3.13), and (3.15) can be expressed as $k_1 + k_2 f(d_{ij})$ with $k_1 = \rho^2$ and $k_2 = (1 - n_g - \rho^2)$.

The circular uniform distribution has $\rho = 0$, which is the minimum mean resultant vector

length parameter of all circular distributions. The degenerate distribution has $\rho = 1$, which is the theoretical maximum. For CRV, the range of ρ^2 is $0 \leq \rho^2 < 1 \Rightarrow 0 \leq k_1 < 1$.

$k_2 = (1 - n_g - \rho^2)$ is the multiplier of the covariance function in (3.12), (3.13), and (3.15). $k_2 = (1 - n_g - \rho^2) = (1 - n_g) - \rho^2$ is the change in the mean cosine from the sill to the nugget. If spatial correlation does not exist, i.e., there is a “pure nugget,” $k_2 = (1 - n_g) - \rho^2 = 0$, and the mean cosine vs. distance is flat. If spatial correlation exists, $k_2 = (1 - n_g) - \rho^2 > 0$. When distance $d = 0$, k_2 is increased to $1 - \rho^2$ because the nugget $n_g = 0$ when $d = 0$. A nonzero nugget applies at distances $d > 0$.

With n the number of observations and \mathbf{J} the square $n \times n$ matrix of element 1, the $n \times n$ matrix of cosines \mathbf{C} , resulting from the element-wise application of a cosine model of the form $k_1 + k_2 f(d_{ij})$ to the matrix of pairwise distances is

$$\begin{aligned} \mathbf{C} &= \begin{bmatrix} k_1 + k_2 f(d_{11}) & k_1 + k_2 f(d_{12}) & \cdots & k_1 + k_2 f(d_{1n}) \\ k_1 + k_2 f(d_{12}) & k_1 + k_2 f(d_{22}) & \cdots & k_1 + k_2 f(d_{2n}) \\ \vdots & \vdots & \vdots & \vdots \\ k_1 + k_2 f(d_{1n}) & k_1 + k_2 f(d_{2n}) & \cdots & k_1 + k_2 f(d_{nn}) \end{bmatrix} \\ &= \begin{bmatrix} k_1 & k_1 & \cdots & k_1 \\ k_1 & k_1 & \cdots & k_1 \\ \vdots & \vdots & \vdots & \vdots \\ k_1 & k_1 & \cdots & k_1 \end{bmatrix} + k_2 \begin{bmatrix} f(d_{11}) & f(d_{12}) & \cdots & f(d_{1n}) \\ f(d_{12}) & f(d_{22}) & \cdots & f(d_{2n}) \\ \vdots & \vdots & \vdots & \vdots \\ f(d_{1n}) & f(d_{2n}) & \cdots & f(d_{nn}) \end{bmatrix} \\ &= k_1 \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} + k_2 \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{12} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{bmatrix} \\ &= k_1 \mathbf{J} + k_2 \mathbf{A} \end{aligned}$$

Now it will be proven that the matrix \mathbf{C} is positive definite with \mathbf{y} any n -element non zero vector.

$$\begin{aligned}
\mathbf{y}^T \mathbf{C} \mathbf{y} &= \mathbf{y}^T (k_1 \mathbf{J} + k_2 \mathbf{A}) \mathbf{y} = \mathbf{y}^T k_1 \mathbf{J} \mathbf{y} + \mathbf{y}^T k_2 \mathbf{A} \mathbf{y} \\
&= k_1 \mathbf{y}^T \underbrace{\begin{bmatrix} 1_{11} & \cdots & 1_{1n} \\ \vdots & \ddots & \vdots \\ 1_{n1} & \cdots & 1_{nn} \end{bmatrix}}_{\mathbf{J}} \mathbf{y} + k_2 \mathbf{y}^T \mathbf{A} \mathbf{y} \\
&= k_1 \left[\underbrace{\sum_{i=1}^n y_i \quad \cdots \quad \sum_{i=1}^n y_i}_{\Sigma \text{ for each column of } \mathbf{J}} \right] \mathbf{y} + k_2 \mathbf{y}^T \mathbf{A} \mathbf{y} \\
&= k_1 (y_1 [\sum_{i=1}^n y_i] + y_2 [\sum_{i=1}^n y_i] + \cdots + y_n [\sum_{i=1}^n y_i]) + k_2 \mathbf{y}^T \mathbf{A} \mathbf{y} = \\
&= k_1 (y_1 + y_2 + \cdots + y_n) [\sum_{i=1}^n y_i] + k_2 \mathbf{y}^T \mathbf{A} \mathbf{y} \\
&= k_1 [\sum_{i=1}^n y_i] [\sum_{i=1}^n y_i] + k_2 \mathbf{y}^T \mathbf{A} \mathbf{y} \\
&= \underbrace{k_1}_{0 \leq k_1 < 1} \underbrace{[\sum_{i=1}^n y_i]^2}_{\geq 0} + \underbrace{k_2}_{> 0} \underbrace{\mathbf{y}^T \mathbf{A} \mathbf{y}}_{\mathbf{A} \text{ P. D. } \Rightarrow > 0} \\
&> 0 \Rightarrow \\
\mathbf{y}^T \mathbf{C} \mathbf{y} &> 0 \quad \forall \mathbf{y} \neq \mathbf{0}
\end{aligned}$$

Hence, \mathbf{C} is positive definite by Appendix B, Section B.2, point 3.

3.7 Cosineogram of Ocean Wind in a South Polar Region

In this section, circular-spatial correlation will be extracted from ocean wind of a south polar region. A model was computed for longitude 69.5° E to 109.5° E by latitude -59.5° N to -40.5° N in 1° increments by averaging the data of Chapter 2, Subsection 2.2.1 via R package CircSpatial function CircDataimage (Chapter 2, and Appendices J.10 and K.2) with input as in Appendix J, Subsection J.10.6, step 1, and smoothing the averages with bandwidth 2.5° in the plane of longitude and latitude (J.10.6, step 8). The smoothed average directions from CircDataimage were output to the R workspace in the list object Globals. Appendix L, Section L.2 shows usage of the elements of Globals. Figure 3-12 shows the image of this model. The spatial correlation is expressed in the cosineocloud and cosineogram in Figure 3-13. The cosineocloud (grey points) is

here defined as the plot of the cosines of the angles between all pairs of directions vs. distance between measurement locations. The cosineocloud is useful for examination of the individual cosines. The cosineogram (red curve) as defined in Section 3.2, reduces the cosineocloud to the plot of the mean cosine vs. distance. Between distances of 0 and 1 on the horizontal axis, spatial correlation is changing rapidly. At a distance of about 3.8, the mean cosine tends to be constant at about 0.45 indicating that direction is not correlated. Hence, the range is 3.8 and the sill is 0.45. Circular kriging, as described in Chapter 4, requires a cosine model which is smooth, continuous, and positive definite. The shape of the cosineogram in Figure 3-13 suggests the exponential cosine model in (3.12), which is overplotted as a blue dashed curve over the full range of distances for comparison with the empirical cosineogram. The fit of the model to the cosineogram is adequate. The exponential cosine model characterizes the circular-spatial correlation in this region of ocean wind data as asymptotic, but without the inflexion (S shape) of the gaussian cosine model. The exponential cosine model is fairly linear near the origin and falls to the sill much more quickly than the spherical cosine model.

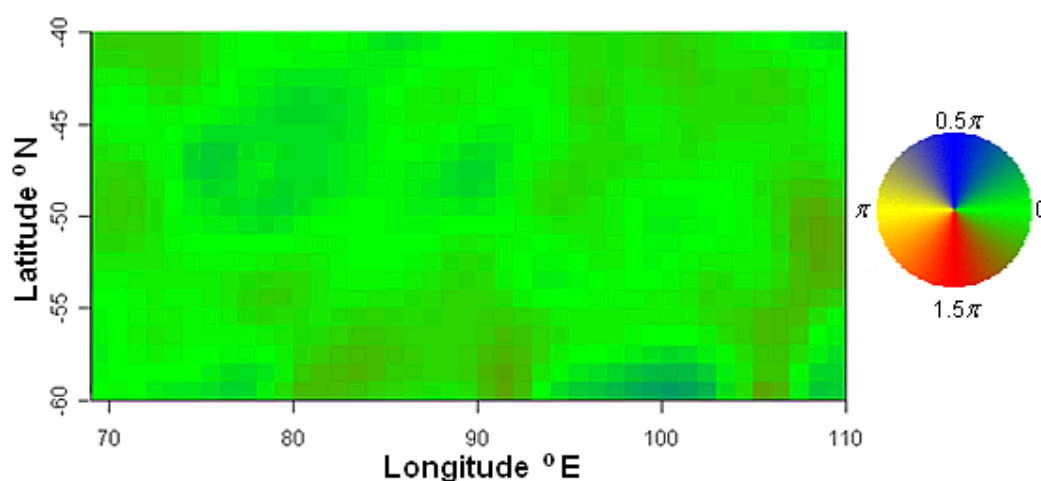


Figure 3-12. Circular Dataimage of Model of Ocean Wind Direction for South Polar Region. Direction, which is coded by the color wheel, is relatively homogeneous and varies about the direction of 0 radians.

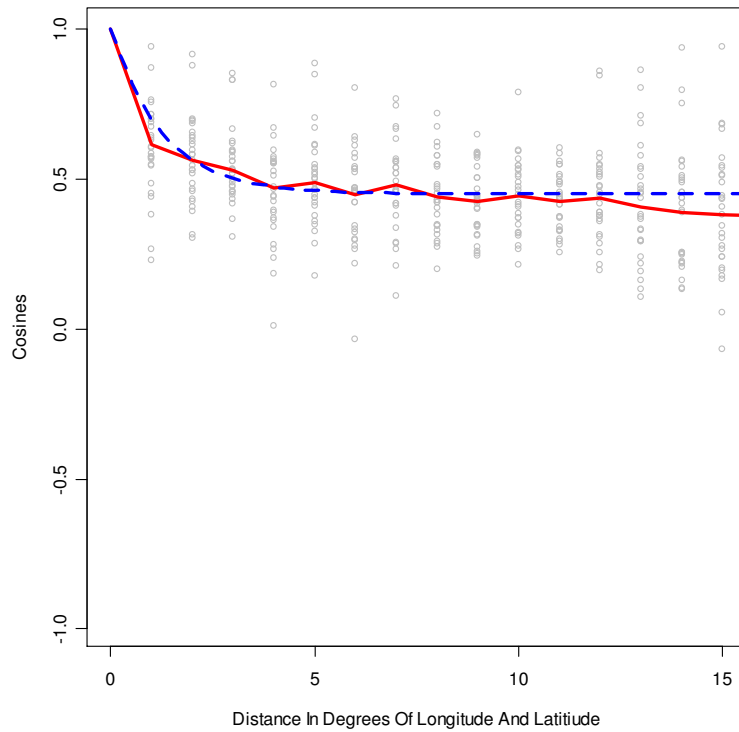


Figure 3-13. Cosineocloud, Cosineogram, and Exponential Model of South Polar Ocean Wind. The cosineocloud (grey points) shows the cosines of the angles between all pairs of directions vs. distance between measurement locations. The cosineogram (red solid curve) reduces the cosineocloud to the plot of the mean cosine vs. distance. The sill (plateau), where the random components of direction are uncorrelated, occurs at a distance of about 3.8 at a mean cosine of about 0.45. The exponential model (blue dashed curve) is overplotted for comparison.

3.8 Chapter Summary and Future Work

In this chapter, we discussed the cosineocloud, the empirical cosineogram, and theoretical cosine models. The cosineogram plots the empirical spatial correlation in circular-spatial data as the mean cosine of the angle between random components of direction at locations vs. distance d between observation locations. With $\hat{\zeta}(d)$ the mean cosine, \mathbf{x}_i and \mathbf{x}_j vectors of location coordinates of observations i and j , respectively, $\|\mathbf{x}_j - \mathbf{x}_i\|$ the linear distance between locations of observations i and j , and $N(d)$ the number of pairs of observations of direction separated by a distance within a tolerance

ε of d , the cosineogram is a plot of $\hat{\zeta}(d) = \frac{1}{N(d)} \sum_{\|\mathbf{x}_j - \mathbf{x}_i\| - d < \varepsilon} \cos(\theta_j - \theta_i)$ vs. d . For an example, a cosineogram was constructed from homogeneous ocean wind data in a south polar region.

The cosine model fitted to the cosineogram characterizes the spatial correlation in a form useful for circular kriging.

- The mean cosine equals 1 at zero distance.
- The mean cosine at distances close to 0 may be reduced by measurement error.

This reduction n_g is called the nugget effect.

- The range r is a scale parameter. The range of the spherical cosine model is the distance beyond which CRV are uncorrelated.
- The sill is the mean cosine at distances where CRV are uncorrelated. The theoretical sill is ρ^2 .

The theoretical sill was derived as the square of the mean resultant vector length of the circular probability distribution underlying the circular-spatial data. For the circular probability distributions cardioid, triangular, uniform ($\rho = 0$), von Mises, and wrapped Cauchy, the mean resultant vector length equals the parameter ρ of the underlying circular probability distribution. The theoretical sill was verified by simulation.

Introductory cosine models, which are required for circular kriging, were adapted from the exponential, Gaussian, and spherical covariance functions used for linear kriging by shifting and scaling the covariance function. With d the distance between measurement locations, ρ the mean vector resultant length parameter of the circular probability distribution, $0 \leq \rho < 1$, n_g the nugget, $0 \leq n_g < 1 - \rho^2$, r the range, and $c(d, r)$ the covariance function from linear kriging with a maximum of 1,

the general form of the cosine model is

$$\zeta(d) = \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2)c(d, r), & d > 0. \end{cases}$$

This form was proven to produce a positive definite cosine matrix.

Future work includes the development of theoretical foundations of directional cosineograms for anisotropic circular-spatial data where the range varies with the geographic direction.

CHAPTER 4

CIRCULAR KRIGING

4.1 Introduction

4.1.1 Objective

The objective of this chapter is to develop a circular kriging estimate of direction with optimum properties based on a sample of circular-spatial data. Kriging is a body of techniques for estimating continuous and spatially correlated data. The name of the technique is derived from Daniel G. Krige, a South African mining geologist, who originated the method for linear-spatial data. The circular kriging estimate is a linear combination of observations of direction with weights based on the spatial correlation as estimated by the cosine model fitted to the empirical cosineogram (Chapter 3), and may be imaged using arrow plots or the circular dataimage (Chapter 2). Figure 4-1 illustrates the kriging of simulated circular-spatial data. The R package `CircSpatial` function `KrigCRF` is documented in Appendix J, Section J.5.

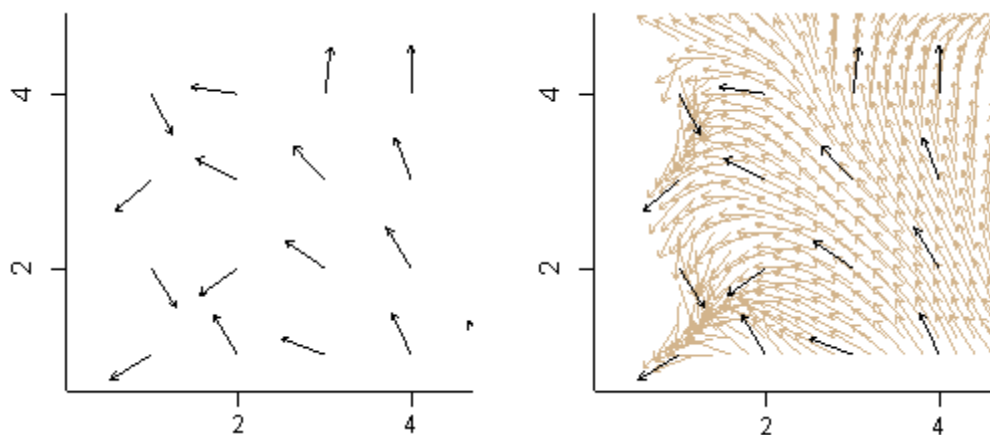


Figure 4-1. Circular Kriging, the Interpolation of Circular-Spatial Data Based on Spatial Correlation. The left plot shows simulated circular-spatial data. In the right plot, the simulated data (black) is superimposed on the kriged estimate of the simulated data (tan).

4.1.2 Chapter Organization

This chapter is organized as follows: Section 4.2 derives the circular kriging estimator, correcting the result from McNeill (1993). Section 4.3 proves that the estimator is optimal. Section 4.4 gives an alternate formula, which is computationally efficient. Section 4.5 shows the kriging behavior around a sampled location, and proves that the estimated direction at a sampled location is the observed direction. This is called “exact interpolation” in linear kriging. Section 4.6 derives the circular kriging variance of the circular kriging estimator, correcting the result from McNeill (1993). Section 4.7 shows how the circular kriging variance varies with distance and the circular-spatial correlation model. Section 4.8 concludes with the summary and description of future work.

4.2 Solution

4.2.1 A Linear Combination of Observations

The estimated spatial correlation parameters (see nugget, range, and sill in Chapter 3) are assumed to be reasonably accurate. Inaccuracy increases error and the variability of the estimate. The circular random field (CRF), as introduced in Chapter 1 and further discussed in Chapter 5, is assumed to be isotropic (spatial correlation independent of direction).

See Appendix A for a description of the notation and Appendix B for referenced Equations (B.1) to (B.12). In Chapter 3, the direction at location \mathbf{x}_0 was denoted by the scalar θ_0 in radians. The conversion from the scalar to the unit vector representation of direction is $\mathbf{u}_0 = \begin{bmatrix} \cos(\theta_0) \\ \sin(\theta_0) \end{bmatrix}$. Let $\hat{\mathbf{u}}_0$ be the unit vector estimate of the direction \mathbf{u}_0 at the

unmeasured location \mathbf{x}_0 , and $\mathbf{u}_i, i = 1, 2, \dots, n$ be observations of direction as unit vectors at measured locations $\mathbf{x}_i, i = 1, 2, \dots, n$, respectively.

Spatially correlated observations contain information about \mathbf{u}_0 . Because spatial correlation increases as distance decreases, observations nearer to the estimation location carry more information about \mathbf{u}_0 than observations more distant. Hence a weighted linear combination of the observations $\mathbf{u}_i, i = 1, 2, \dots, n$, is required. Let $w_i, i = 1, 2, \dots, n$, be the weights with $w_i \in R$ (the set of real numbers).

$$\hat{\mathbf{u}}_0 = \mathbf{u}_1 w_1 + \mathbf{u}_2 w_2 + \dots + \mathbf{u}_n w_n$$

$$= [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_n] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

$$\equiv \mathbf{U} \mathbf{w} \Rightarrow$$

$$\hat{\mathbf{u}}_0 = \mathbf{U} \mathbf{w} \quad (4.1)$$

\mathbf{w} will ultimately be determined by a constrained optimization in Subsection 4.2.7 such

that $\hat{\mathbf{u}}_0$ is a unit vector. In general, the length of the vector estimate $\hat{\mathbf{u}}_0$ for any $\mathbf{w} \in R^n$

is

$$\begin{aligned} \|\hat{\mathbf{u}}_0\| &\stackrel{(B.2)}{=} \sqrt{\hat{\mathbf{u}}_0^T \hat{\mathbf{u}}_0} \\ &\stackrel{(4.1)}{=} \sqrt{(\mathbf{U} \mathbf{w})^T (\mathbf{U} \mathbf{w})} \\ &= \sqrt{\mathbf{w}^T \mathbf{U}^T \mathbf{U} \mathbf{w}} \\ &= \sqrt{\mathbf{w}^T \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_n] \mathbf{w}} = \sqrt{\mathbf{w}^T \begin{bmatrix} \mathbf{u}_1^T \mathbf{u}_1 & \mathbf{u}_1^T \mathbf{u}_2 & \dots & \mathbf{u}_1^T \mathbf{u}_n \\ \mathbf{u}_2^T \mathbf{u}_1 & \mathbf{u}_2^T \mathbf{u}_2 & \dots & \mathbf{u}_2^T \mathbf{u}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_n^T \mathbf{u}_1 & \mathbf{u}_n^T \mathbf{u}_2 & \dots & \mathbf{u}_n^T \mathbf{u}_n \end{bmatrix} \mathbf{w}} \end{aligned}$$

$$\begin{aligned}
 & \equiv \sqrt{\mathbf{w}^T \begin{bmatrix} 1 & \mathbf{u}_1^T \mathbf{u}_2 & \cdots & \mathbf{u}_1^T \mathbf{u}_n \\ \mathbf{u}_2^T \mathbf{u}_1 & 1 & \cdots & \mathbf{u}_2^T \mathbf{u}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_n^T \mathbf{u}_1 & \mathbf{u}_n^T \mathbf{u}_2 & \cdots & 1 \end{bmatrix} \mathbf{w}} \\
 & \equiv \sqrt{\mathbf{w}^T \tilde{\mathbf{K}} \mathbf{w}} \Rightarrow
 \end{aligned}$$

$$\|\hat{\mathbf{u}}_0\| = \sqrt{\mathbf{w}^T \tilde{\mathbf{K}} \mathbf{w}}. \quad (4.2)$$

* For unit vectors \mathbf{u}_i , the diagonal elements of $\mathbf{U}^T \mathbf{U}$ are

$$\mathbf{u}_i^T \mathbf{u}_i \stackrel{(B.1)}{=} \cos(\theta_i) \cos(\theta_i) + \sin(\theta_i) \sin(\theta_i) = \cos^2(\theta_i) + \sin^2(\theta_i) = 1.$$

4.2.2 Optimality

Let \mathbf{e}_0 be the error vector equal to the unit vector estimate of direction $\hat{\mathbf{u}}_0$ minus the unobserved direction \mathbf{u}_0 (unit vector), or $\mathbf{e}_0 = \hat{\mathbf{u}}_0 - \mathbf{u}_0$. \mathbf{e}_0 , $\hat{\mathbf{u}}_0$, and \mathbf{u}_0 are located at \mathbf{x}_0 . The addition of the unit vector \mathbf{u}_0 and \mathbf{e}_0 is shown in Figure 4-2 as

$\mathbf{u}_0 + \mathbf{e}_0 = \mathbf{u}_0 + (\hat{\mathbf{u}}_0 - \mathbf{u}_0) = \hat{\mathbf{u}}_0$. In words, the unobserved direction plus the error vector equals the estimate of direction. Hence, \mathbf{e}_0 is a vector from the head of \mathbf{u}_0 to the head of $\hat{\mathbf{u}}_0$, and the length of the error vector is the distance from the head of \mathbf{u}_0 to the head of $\hat{\mathbf{u}}_0$. Let θ be the angle between these vectors in $[0, \pi]$. When $\theta = 0$, $\|\mathbf{e}_0\| = 0$, and when $\theta = \pi$, $\|\mathbf{e}_0\| = 2$. Hence, $0 \leq \|\mathbf{e}_0\| \leq 2$.

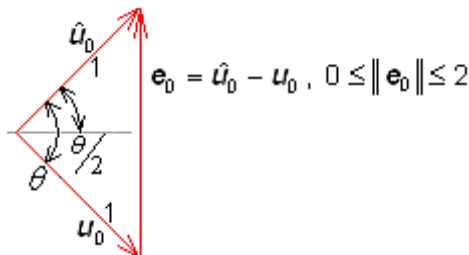


Figure 4-2. Directions Represented by the Unobserved \mathbf{u}_0 , Estimate $\hat{\mathbf{u}}_0$, and Error $\mathbf{e}_0 = \hat{\mathbf{u}}_0 - \mathbf{u}_0$ Vectors. θ is the angle between \mathbf{u}_0 and $\hat{\mathbf{u}}_0$.

The unit vectors \mathbf{u}_0 and $\hat{\mathbf{u}}_0$ can be visualized as the hands of a clock when the tails of the vectors are positioned at the center of a clock. For any \mathbf{u}_0 and $\hat{\mathbf{u}}_0$, while holding θ and $\|\mathbf{e}_0\|$ constant, the minute and hour hands (vectors) can be transposed so $\hat{\mathbf{u}}_0$ is counterclockwise of \mathbf{u}_0 , and the hands can be rotated so that the line joining the hands is vertical with \mathbf{e}_0 pointing upward. Since we are interested in $\|\mathbf{e}_0\|$ for minimization, Figure 4-2 illustrates the case of any \mathbf{u}_0 and $\hat{\mathbf{u}}_0$.

$\hat{\mathbf{u}}_0 = \mathbf{U}\mathbf{w}$, the estimate of the unobserved direction, will be considered optimal when the choice of \mathbf{w} results a unit vector estimate $\hat{\mathbf{u}}_0$ with an error vector \mathbf{e}_0 of minimum squared length over all such estimates. With θ the angle between $\hat{\mathbf{u}}_0$ and \mathbf{u}_0 , the squared length of the error vector is

$$\begin{aligned}\|\hat{\mathbf{u}}_0 - \mathbf{u}_0\|^2 &= [2\sin(\theta/2)]^2 \\ &= 4[\sin(\theta/2)]^2 \Rightarrow \\ &\|\hat{\mathbf{u}}_0 - \mathbf{u}_0\|^2 = 4[\sin(\theta/2)]^2.\end{aligned}\tag{4.3}$$

The result (4.3) can be further transformed.

$$\begin{aligned}\cos(\theta) &= \cos(\theta/2 + \theta/2) \\ &= \cos(\theta/2)\cos(\theta/2) - \sin(\theta/2)\sin(\theta/2) \\ &= [\cos(\theta/2)]^2 - [\sin(\theta/2)]^2 \\ &= \{1 - [\sin(\theta/2)]^2\} - [\sin(\theta/2)]^2 \\ &= 1 - 2[\sin(\theta/2)]^2 \Rightarrow \\ -\cos(\theta) &= -1 + 2[\sin(\theta/2)]^2 \Rightarrow \\ 1 - \cos(\theta) &= 2[\sin(\theta/2)]^2 \Rightarrow \\ 2(1 - \cos(\theta)) &= 4[\sin(\theta/2)]^2 \\ &\stackrel{(4.3)}{=} \|\hat{\mathbf{u}}_0 - \mathbf{u}_0\|^2 \Rightarrow \\ &\|\hat{\mathbf{u}}_0 - \mathbf{u}_0\|^2 = 2(1 - \cos(\theta))\end{aligned}\tag{4.4}$$

Hence, the squared length of the error vector (4.4) is minimized by maximizing $\cos \theta$, or minimizing the angle between unit vectors $\hat{\mathbf{u}}_0$ and \mathbf{u}_0 . With $\cos(\theta_{i_0})$ denoting the cosine of the angle between observation \mathbf{u}_i and unobserved \mathbf{u}_0 , and referencing Appendix B, Equation (B.1), it follows that

$$\begin{aligned}
 \cos(\theta) &\stackrel{(B.1)}{=} \hat{\mathbf{u}}_0^T \mathbf{u}_0 \\
 &\stackrel{(4.1)}{=} (\mathbf{U}\mathbf{w})^T \mathbf{u}_0 \\
 &= \mathbf{w}^T \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} \mathbf{u}_0 \\
 &= \mathbf{w}^T \begin{bmatrix} \mathbf{u}_1^T \mathbf{u}_0 \\ \mathbf{u}_2^T \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_n^T \mathbf{u}_0 \end{bmatrix} \\
 &\stackrel{(B.1)}{=} \mathbf{w}^T \begin{bmatrix} \cos(\theta_{1_0}) \\ \cos(\theta_{2_0}) \\ \vdots \\ \cos(\theta_{n_0}) \end{bmatrix} \\
 &\equiv \mathbf{w}^T \tilde{\mathbf{c}} \Rightarrow
 \end{aligned}$$

$$\cos \theta = \mathbf{w}^T \tilde{\mathbf{c}}. \quad (4.5)$$

4.2.3 Cosineogram

$\tilde{\mathbf{c}}$ must be estimated since \mathbf{u}_0 is unknown. Also, the eigenvalues of $\tilde{\mathbf{K}} = \mathbf{U}^T \mathbf{U}$ have been observed to be not all positive indicating that $\tilde{\mathbf{K}}$ is generally not positive definite according to Appendix B, Equation (B.3). However, positive definiteness is required for maximum fit (Section 4.3). This is accomplished by replacing $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{K}}$ with estimates \mathbf{c} and \mathbf{K} , respectively, which are computed on smooth and positive definite functions as is done in the kriging of linear-spatial data (Bailey and Gatrell 1995).

For circular-spatial data, the empirical cosineogram, a rough plot of circular-spatial correlation (Chapter 3, Section 3.2), is modeled by the positive definite cosine model (Chapter 3, Section 3.6 and Appendix M) with best fit. The elements of \mathbf{c} and \mathbf{K} are computed using the selected cosine model and the distances between measurement locations. \mathbf{c} , which depends on the estimation location, must be updated for each location to be estimated. \mathbf{K} , which depends only on the distances between observations, is computed once.

4.2.4 First Iteration of the Circular Kriging Solution \mathbf{w}

Recall that $\mathbf{w}^T \mathbf{c}$ is a linear combination of the expected cosines of the angles between the unobserved direction as a unit vector \mathbf{u}_0 and the sampled directions $\mathbf{u}_i, i = 1, 2, \dots, n$, with $\tilde{\mathbf{c}}$ replaced by \mathbf{c} in (4.5). It expresses the fit of $\hat{\mathbf{u}}_0$ to \mathbf{u}_0 . $\mathbf{w}^T \mathbf{K} \mathbf{w}$ is the squared length of $\hat{\mathbf{u}}_0$, with $\tilde{\mathbf{K}}$ replaced by \mathbf{K} in (4.2). $\mathbf{w}^T \mathbf{K} \mathbf{w} - 1 = 0$ expresses that the squared length of $\hat{\mathbf{u}}_0$ is constrained to 1. The vector of weights \mathbf{w} will be solved by maximizing $\mathbf{w}^T \mathbf{c}$ relative to \mathbf{w} with the maximization constrained such that $\mathbf{w}^T \mathbf{K} \mathbf{w}$ is equal to one. The method of Lagrange multipliers (Grossman 1988), for finding the extrema of a function of several variables subject to a constraint, introduces a new unknown scalar variable, which is called the Lagrange multiplier, and defines a new function, which is called the Lagrangian, in terms of the original function, the constraint, and the Lagrange multiplier. At the stationary point, the Lagrange multiplier is the proportionality of the gradient of the function to be maximized and the gradient of the constraint. Let ν be the Lagrange multiplier and q be the Lagrangian.

$$q = \mathbf{w}^T \mathbf{c} - \frac{\nu}{2} (\mathbf{w}^T \mathbf{K} \mathbf{w} - 1) \quad (4.6)$$

ν is divided by two to simplify a subsequent result.

Differentiating (4.6) with respect to \mathbf{w} according Appendix B, Equations (B.8) and (B.9),

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} \left(\mathbf{w}^T \mathbf{c} - \frac{\nu}{2} (\mathbf{w}^T \mathbf{K} \mathbf{w} - 1) \right) &= \mathbf{c} - \frac{\nu}{2} (2\mathbf{K} \mathbf{w} - 0) \\ &= \mathbf{c} - \nu \mathbf{K} \mathbf{w} \Rightarrow \\ \frac{\partial}{\partial \mathbf{w}} \left(\mathbf{w}^T \mathbf{c} - \frac{\nu}{2} (\mathbf{w}^T \mathbf{K} \mathbf{w} - 1) \right) &= \mathbf{c} - \nu \mathbf{K} \mathbf{w}. \end{aligned} \quad (4.7)$$

Setting the derivative (4.7) equal to the zero vector $\mathbf{0} = [0 \ 0 \ \dots \ 0]^T$,

$\mathbf{c} - \nu \mathbf{K} \mathbf{w} = \mathbf{0} \Rightarrow \nu \mathbf{K} \mathbf{w} = \mathbf{c}$. From the invertibility of \mathbf{K} (Appendix B, Equation (B.4)) it follows that

$$\mathbf{w} = \nu^{-1} \mathbf{K}^{-1} \mathbf{c}. \quad (4.8)$$

ν can be determined from the unit length constraint $\mathbf{w}^T \mathbf{K} \mathbf{w} = 1$.

$$\begin{aligned} 1 &= \mathbf{w}^T \mathbf{K} \mathbf{w} \\ &\stackrel{(4.8)}{=} (\nu^{-1} \mathbf{K}^{-1} \mathbf{c})^T \mathbf{K} (\nu^{-1} \mathbf{K}^{-1} \mathbf{c}) \\ &= \mathbf{c}^T (\mathbf{K}^{-1})^T \nu^{-1} \mathbf{K} \nu^{-1} \mathbf{K}^{-1} \mathbf{c} \\ &\stackrel{(B.5)}{=} \nu^{-1} \nu^{-1} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{K} \mathbf{K}^{-1} \mathbf{c} \\ &= \nu^{-2} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c} \Rightarrow \\ 1 &= \nu^{-2} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c} \Rightarrow \\ \nu^{+2} &= \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c} \Rightarrow \\ \nu &= \pm \sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}} \end{aligned}$$

The Lagrange multiplier ν is a scalar representing the proportionality of parallel gradient vectors of the function being optimized and the constraint. The positive sign of ν is selected to maximize fit as will be explained in detail in Section 4.3. It follows that

$$\nu = +\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}, \quad (4.9)$$

and
$$\mathbf{w} = \nu^{-1} \mathbf{K}^{-1} \mathbf{c} \quad (4.8)$$

$$= \frac{1}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \mathbf{K}^{-1} \mathbf{c} \Rightarrow \quad (4.9)$$

$$\mathbf{w} = \frac{1}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \mathbf{K}^{-1} \mathbf{c}. \quad (4.10)$$

(4.10) is McNeill's result (1993, p. 40, eq. 6).

4.2.5 Length of the Estimator $\hat{\mathbf{u}}_0$

Substituting the solution into the estimator, the squared length is

$$\begin{aligned} \|\hat{\mathbf{u}}_0\|^2 &\stackrel{(B.2)}{=} \left(\sqrt{\hat{\mathbf{u}}_0^T \hat{\mathbf{u}}_0} \right)^2 \\ &\stackrel{(4.1)}{=} \mathbf{w}^T \mathbf{U}^T \mathbf{U} \mathbf{w} \\ &\stackrel{(4.10)}{=} \left(\frac{1}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \mathbf{K}^{-1} \mathbf{c} \right)^T \mathbf{U}^T \mathbf{U} \left(\frac{1}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \mathbf{K}^{-1} \mathbf{c} \right) \\ &= \left(\mathbf{c}^T (\mathbf{K}^{-1})^T \frac{1}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \right) \mathbf{U}^T \mathbf{U} \left(\frac{1}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \mathbf{K}^{-1} \mathbf{c} \right) \\ &= \frac{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}} \\ &\equiv s \Rightarrow \end{aligned}$$

$$\|\hat{\mathbf{u}}_0\|^2 = \frac{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}. \quad (4.11)$$

Is $\hat{\mathbf{u}}_0$ a unit vector?

4.2.6 $\hat{\mathbf{u}}_0$ Is Likely Not a Unit Vector

If $\mathbf{U}^T \mathbf{U} = \mathbf{K} \stackrel{(4.11)}{\Rightarrow} \|\hat{\mathbf{u}}_0\|^2 = 1$. However, with $\mathbf{U}^T \mathbf{U}$ a realization of the continuous

random matrix $\mathbf{V}^T \mathbf{V}$,

$probability(\mathbf{V}^T \mathbf{V} = \mathbf{K}) = 0 \stackrel{(4.11)}{\Rightarrow} probability(\|\hat{\mathbf{u}}_0\| = 1) = 0$. Thus, it is likely that $\hat{\mathbf{u}}_0$ is not a unit vector (except for a set of Borel-measure 0).

4.2.7 Corrected Circular Kriging Solution \mathbf{w}

Equation (4.11) suggests that the matrix of cosines \mathbf{K} be scaled by s . It will be shown in Section 4.3 that scaling \mathbf{K} by s leads to a unit vector solution. The revised function to maximize is

$$q_s \equiv \mathbf{w}^T \mathbf{c} - \frac{\nu}{2} (\mathbf{w}^T s \mathbf{K} \mathbf{w} - 1). \quad (4.12)$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} (q_s) &\stackrel{(4.12)}{=} \mathbf{c} - \frac{\nu}{2} (2s \mathbf{K} \mathbf{w} - 0) \\ &= \mathbf{c} - \nu s \mathbf{K} \mathbf{w} \Rightarrow \end{aligned}$$

$$\frac{\partial}{\partial \mathbf{w}} (q_s) = \mathbf{c} - \nu s \mathbf{K} \mathbf{w} \quad (4.13)$$

Setting the derivative (4.13) equal to the zero vector $\mathbf{0}$,

$$\begin{aligned} \mathbf{c} - \nu s \mathbf{K} \mathbf{w} &= \mathbf{0} \Rightarrow \\ \nu s \mathbf{K} \mathbf{w} &= \mathbf{c} \Rightarrow \end{aligned}$$

$$\mathbf{w} = \nu^{-1} s^{-1} \mathbf{K}^{-1} \mathbf{c}. \quad (4.14)$$

From the constraint $\mathbf{w}^T s \mathbf{K} \mathbf{w} = 1$ in (4.12), and keeping the sign of ν from (4.9),

$$\begin{aligned} 1 &= \mathbf{w}^T s \mathbf{K} \mathbf{w} \\ &\stackrel{(4.14)}{=} (\nu^{-1} s^{-1} \mathbf{K}^{-1} \mathbf{c})^T s \mathbf{K} (\nu^{-1} s^{-1} \mathbf{K}^{-1} \mathbf{c}) \\ &= \mathbf{c}^T (\mathbf{K}^{-1})^T s^{-1} \nu^{-1} s \mathbf{K} \nu^{-1} s^{-1} \mathbf{K}^{-1} \mathbf{c} \\ &\stackrel{(B.5)}{=} \nu^{-1} \nu^{-1} s^{-1} s s^{-1} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{K} \mathbf{K}^{-1} \mathbf{c} \\ &= \nu^{-2} s^{-1} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c} \Rightarrow \\ 1 &= \nu^{-2} s^{-1} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c} \Rightarrow \\ \nu^2 &= s^{-1} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c} \Rightarrow \end{aligned}$$

$$\nu = +\sqrt{s^{-1} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}. \quad (4.15)$$

Substituting s (4.11) and v (4.15) into w (4.14), we arrive at the principal result of this chapter.

$$\begin{aligned}
 \mathbf{w} & \stackrel{(4.14)}{=} v^{-1} s^{-1} \mathbf{K}^{-1} \mathbf{c} \\
 & \stackrel{(4.15)}{=} \frac{1}{\sqrt{s^{-1} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} s^{-1} \mathbf{K}^{-1} \mathbf{c} \\
 & = \frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{s \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \\
 & \stackrel{(4.11)}{=} \frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{\left(\frac{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}} \right) \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \\
 & = \frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \Rightarrow \\
 \mathbf{w} & = \frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \tag{4.16}
 \end{aligned}$$

This result differs from McNeill (1993, p. 40, eq. 6), who obtained $\mathbf{w} = \frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}}$.

4.3 Verification of Optimality

In this section, it will be proven that the estimated direction, which is a linear combination of the observations of direction as unit vectors, \mathbf{Uw} , is also a unit vector, that the expression of constrained optimization of the cosine of the angle between the direction to be estimated and the estimator has derivatives of zero at the estimated direction, and that the estimated direction has a maximum fit as opposed to a minimum fit to the direction being estimated.

First, it will be shown that $\hat{\mathbf{u}}_0$ is a unit vector.

$$\begin{aligned}
 \|\hat{\mathbf{u}}_0\|^2 &\stackrel{(4.2)}{=} \mathbf{w}^T \mathbf{U}^T \mathbf{U} \mathbf{w} \\
 &\stackrel{(4.16)}{=} \left(\frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \right)^T \mathbf{U}^T \mathbf{U} \frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \\
 &= \frac{\mathbf{c}^T (\mathbf{K}^{-1})^T}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \mathbf{U}^T \mathbf{U} \frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \\
 &\stackrel{(B.5)}{=} \frac{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}} \\
 &= 1 \Rightarrow
 \end{aligned}$$

$$\|\hat{\mathbf{u}}_0\|^2 = 1 \quad (4.17)$$

Thus, the squared length of the estimate of direction as a vector is one. Hence, the direction estimate is a unit vector such as the observations.

Next, it will be shown that the vector of derivatives of the expression of constrained optimization (4.12) at the solution \mathbf{w} is the zero vector $\mathbf{0}$.

$$\begin{aligned}
 \frac{\partial}{\partial \mathbf{w}} (q_s) &\stackrel{(4.13)}{=} \mathbf{c} - \nu \mathbf{s} \mathbf{K} \mathbf{w} \\
 &\stackrel{(4.15), (4.16)}{=} \mathbf{c} - \sqrt{s^{-1} \mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}} \mathbf{s} \mathbf{K} \left(\frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \right) \\
 &= \mathbf{c} - s^{-1/2} \mathbf{s} \mathbf{K} \left(\mathbf{K}^{-1} \mathbf{c} / \frac{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \right) \\
 &= \mathbf{c} - s^{-1/2} \mathbf{s} \mathbf{K} \left(\mathbf{K}^{-1} \mathbf{c} / \sqrt{\frac{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \right) \\
 &\stackrel{(4.11)}{=} \mathbf{c} - s^{-1/2} \mathbf{s} \left(\frac{\mathbf{K} \mathbf{K}^{-1} \mathbf{c}}{\sqrt{s}} \right) \\
 &= \mathbf{c} - s^{-1/2} s^{1/2} \mathbf{c} \\
 &= \mathbf{c} - \mathbf{c} \\
 &= \mathbf{0} \Rightarrow
 \end{aligned}$$

$$\frac{\partial}{\partial \mathbf{w}} (q_s) = \mathbf{0} \quad (4.18)$$

Thus, the vector of derivatives at the solution \mathbf{w} is the zero vector, $\mathbf{0}$. Hence, the direction estimate has either a minimum or a maximum fit to the direction being estimated.

Last, it will be shown that the solution has a maximum fit. The quadratic part of (4.12) is $-0.5\nu(\mathbf{w}^T \mathbf{s} \mathbf{K} \mathbf{w}) = \mathbf{w}^T (-0.5\nu \mathbf{s} \mathbf{K}) \mathbf{w}$. \mathbf{K} is positive definite (Chapter 3, Subsection 3.6.3) and symmetric (Appendix B, Section B.2), hence, \mathbf{K} is orthonormally diagonalizable. Let \mathbf{Q} be the diagonalizing matrix of eigenvectors of \mathbf{K} , and $\mathbf{\Lambda}$ be the diagonal matrix of the eigenvalues of \mathbf{K} . Hence, the diagonalization of the symmetric matrix $-0.5\nu \mathbf{s} \mathbf{K}$ is $\mathbf{Q}^T (-0.5\nu \mathbf{s} \mathbf{K}) \mathbf{Q} = -0.5\nu \mathbf{s} \mathbf{Q}^T \mathbf{K} \mathbf{Q} = -0.5\nu \mathbf{s} \mathbf{\Lambda}$. By Appendix B, Equation (B.3), the eigenvalues of \mathbf{K} , which are the elements λ_i of $\mathbf{\Lambda}$, are all positive. Therefore, the eigenvalues of $-0.5\nu \mathbf{s} \mathbf{K}$, which are the diagonal elements of the matrix $-0.5\nu \mathbf{s} \mathbf{\Lambda}$, are all negative. Since all the eigenvalues are negative, $-0.5\nu \mathbf{s} \mathbf{K}$ is negative definite by equivalence (B.7). The Hessian of (4.12) is $-\nu \mathbf{s} \mathbf{K}$ by Equation (B.11). Hence, the Hessian is also negative definite. The point of zero derivatives (4.18) has a negative definite Hessian. According to Appendix B, Subsection B.8.1, the point of zero derivatives is a maximum. Hence, the direction estimate has maximum fit to the direction being estimated.

4.4 Computationally Efficient Formula

From (4.1) and (4.16), $\hat{\mathbf{u}}_0 = \mathbf{U} \mathbf{K}^{-1} \mathbf{c} / \sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}$. The denominator $\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}$ scales the vector $\mathbf{U} \mathbf{K}^{-1} \mathbf{c}$ to a unit vector, but does not affect the signs and the ratio of the magnitudes of the components of the vector $\mathbf{U} \mathbf{K}^{-1} \mathbf{c}$. Hence, computational efficiency may be obtained by eliminating the computation of

$\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}$ and computing direction directly from the components of $\mathbf{U} \mathbf{K}^{-1} \mathbf{c}$.

Let h and v be the horizontal and vertical components of the vector $\mathbf{U} \mathbf{K}^{-1} \mathbf{c}$, respectively,

i.e., $\begin{bmatrix} h \\ v \end{bmatrix} = \mathbf{U} \mathbf{K}^{-1} \mathbf{c}$. Then, the estimated direction in $[0, 2\pi)$ radians at location \mathbf{x}_0 is

$$\hat{\theta}_0 = \begin{cases} \tan^{-1}(v/h), & h > 0, v \geq 0 \\ \pi/2, & h = 0, v > 0 \\ \tan^{-1}(v/h) + \pi, & h < 0 \\ \frac{3}{2}\pi, & h = 0, v < 0 \\ \tan^{-1}(v/h) + 2\pi, & h > 0, v < 0 \\ \text{undefined}, & h = v = 0. \end{cases} \quad (4.19)$$

This is called the quadrant specific inverse tangent as in Chapter 3, Subsection 3.3.1, (3.1).

4.5 Kriging Behavior Around a Sampled Location

The kriging behavior around an observation location depends on which cosine model is used. Figure 4-3, which was constructed using the R code in Appendix L, Section L.14, shows the kriging estimate in degrees around a direction of 90° observed at location 0 with nearest observations of 0° at a distance of 10 units away, which is the range. The curve of estimated direction from the spherical cosine model is dashed and red, the curve from the gaussian cosine model is tan and thick, and the curve from the exponential model is solid and black. The exponential curve of direction vs. location has a discontinuity in the derivative of direction with respect to location at the observation location. With nugget=0.0, the Gaussian and spherical curves appear smooth. With nugget = 0.1, the spherical curve is smooth, but the exponential and gaussian curves spike with a discontinuity at the observation location. With or without a nugget, the kriging solution produces “exact interpolation” at a sampled location (estimate=observation). Exact interpolation will now be proven.

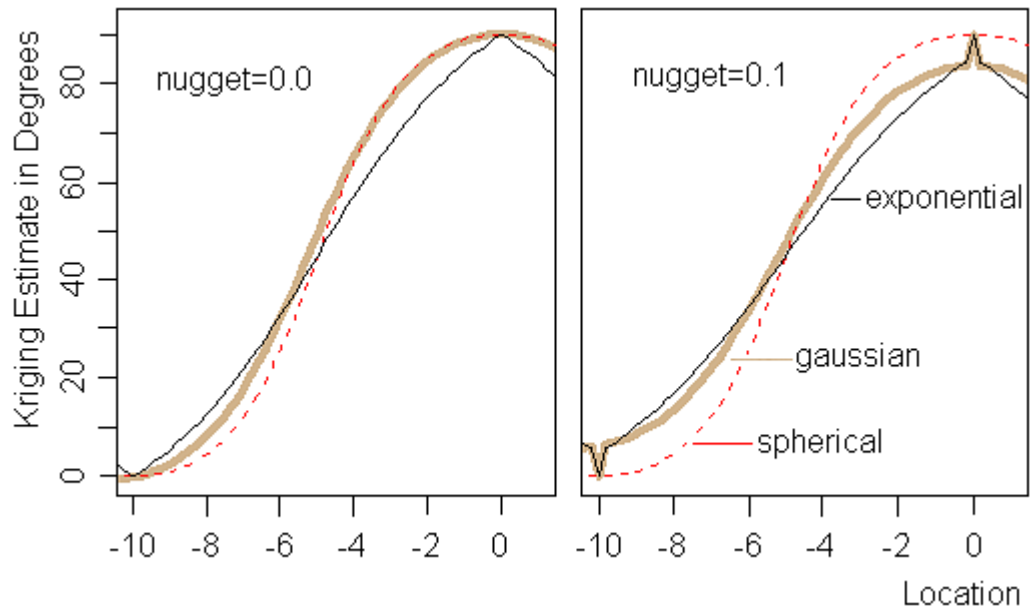


Figure 4-3. Effect of Cosine Model on the Kriging Estimate Around the Measurement Location. The curve from the spherical cosine model is dashed and red, the curve from the gaussian cosine model is tan and thick, and the curve from the exponential model is solid and black. Estimation at a sampled location produces exact interpolation.

With $\varsigma(d_{ij})$ being the mean cosine computed from the cosine model (Chapter 3, Subsection 3.6.2) on the distance from the i^{th} to the j^{th} observation d_{ij} , $i, j = 1, 2, \dots, n$, the unit vector estimate $\hat{\mathbf{u}}_j$ at sample location \mathbf{x}_j requires the vector of cosines

$\mathbf{c}_j^T = [\varsigma(d_{1,j}) \ \varsigma(d_{2,j}) \ \varsigma(d_{3,j}) \ \dots \ \varsigma(d_{j,j}) \ \dots \ \varsigma(d_{n-1,j}) \ \varsigma(d_{n,j})]$ and the symmetric matrix of cosines

$$\mathbf{K} = \begin{bmatrix} \varsigma(d_{1,1})=1 & \varsigma(d_{1,2}) & \varsigma(d_{1,3}) & \dots & \varsigma(d_{1,j}) & \dots & \varsigma(d_{1,n-1}) & \varsigma(d_{1,n}) \\ \varsigma(d_{2,1}) & \varsigma(d_{2,2})=1 & \varsigma(d_{2,3}) & \dots & \varsigma(d_{2,j}) & \dots & \varsigma(d_{2,n-1}) & \varsigma(d_{2,n}) \\ \varsigma(d_{3,1}) & \varsigma(d_{3,2}) & \varsigma(d_{3,3})=1 & \dots & \varsigma(d_{3,j}) & \dots & \varsigma(d_{3,n-1}) & \varsigma(d_{3,n}) \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots & \vdots \\ \varsigma(d_{j,1}) & \varsigma(d_{j,2}) & \varsigma(d_{j,3}) & \dots & \varsigma(d_{j,j})=1 & \dots & \varsigma(d_{j,n-1}) & \varsigma(d_{j,n}) \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots & \vdots \\ \varsigma(d_{n-1,1}) & \varsigma(d_{n-1,2}) & \varsigma(d_{n-1,3}) & \dots & \varsigma(d_{n-1,j}) & \dots & \varsigma(d_{n-1,n-1})=1 & \varsigma(d_{n-1,n}) \\ \varsigma(d_{n,1}) & \varsigma(d_{n,2}) & \varsigma(d_{n,3}) & \dots & \varsigma(d_{n,j}) & \dots & \varsigma(d_{n,n-1}) & \varsigma(d_{n,n})=1 \end{bmatrix}.$$

The key to this proof is to observe that \mathbf{c}_j is the j^{th} row or column of \mathbf{K} .

With \mathbf{P} the diagonalizing orthonormal matrix of eigenvectors (Appendix B, Section B.2, point 4) of the positive definite matrix \mathbf{K} (Chapter 3, Subsection 3.6.3), $\mathbf{\Lambda}$ the diagonal matrix of eigenvalues of \mathbf{K} (Appendix B, Section B.2, point 4), \mathbf{I} the diagonal matrix of 1s (the identity matrix), and $(\mathbf{M})_{\text{Col } j}$ the j^{th} column of matrix \mathbf{M} , and the fact (1) that the j^{th} column of the product of matrix \mathbf{P} post multiplied by a matrix $(\mathbf{\Lambda P}^T)$ equals the product of the matrix $(\mathbf{P\Lambda})$ post multiplied by the j^{th} column of \mathbf{P}^T , the numerator of the solution vector \mathbf{w}_j (4.16) is

$$\begin{aligned}
 \mathbf{K}^{-1}\mathbf{c}_j &= \\
 &= (\mathbf{P\Lambda P}^T)^{-1}\mathbf{c}_j \\
 &= \mathbf{P\Lambda}^{-1}\mathbf{P}^T(\mathbf{K})_{\text{Col } j} \\
 &= \mathbf{P\Lambda}^{-1}\mathbf{P}^T(\mathbf{P\Lambda P}^T)_{\text{Col } j} \\
 &\stackrel{(1)}{=} \mathbf{P\Lambda}^{-1}\mathbf{P}^T\mathbf{P\Lambda}(\mathbf{P}^T)_{\text{Col } j} \\
 &= \mathbf{P\Lambda}^{-1}\underbrace{\mathbf{P}^T\mathbf{P}}_{\substack{\text{P Orthonormal} \\ = \mathbf{I}}}\mathbf{\Lambda}(\mathbf{P}^T)_{\text{Col } j} \\
 &= \mathbf{P\Lambda}^{-1}\underbrace{\mathbf{I}\mathbf{\Lambda}}_{=\mathbf{I}}(\mathbf{P}^T)_{\text{Col } j} \\
 &= \mathbf{P}\mathbf{I}(\mathbf{P}^T)_{\text{Col } j} \\
 &\stackrel{(1)}{=} (\mathbf{P P}^T)_{\text{Col } j} \\
 &= (\mathbf{I})_{\text{Col } j} \\
 &= [0 \ 0 \ 0 \ \dots \ 1 \ \dots \ 0 \ 0]^T.
 \end{aligned}$$

So $\mathbf{K}^{-1}\mathbf{c}_j$ is a column vector of 0's with 1 in the j^{th} position. Using this result we will

see that the estimated direction is the observed direction.

$$\begin{aligned}
 \hat{\mathbf{u}}_j &\stackrel{(4.1)}{=} \mathbf{U}\mathbf{w}_j \\
 &\stackrel{(4.16)}{=} \mathbf{U} \frac{\mathbf{K}^{-1}\mathbf{c}_j}{\sqrt{\mathbf{c}_j^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}_j}}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{[\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3 \ \cdots \ \mathbf{u}_j \ \cdots \ \mathbf{u}_{n-1} \ \mathbf{u}_n][0 \ 0 \ 0 \ \cdots \ 1 \ \cdots \ 0 \ 0]^T}{\sqrt{[0 \ 0 \ 0 \ \cdots \ 1 \ \cdots \ 0 \ 0]\mathbf{U}^T\mathbf{U}[0 \ 0 \ 0 \ \cdots \ 1 \ \cdots \ 0 \ 0]^T}} \\
&= \frac{[\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3 \ \cdots \ \mathbf{u}_j \ \cdots \ \mathbf{u}_{n-1} \ \mathbf{u}_n][0 \ 0 \ 0 \ \cdots \ 1 \ \cdots \ 0 \ 0]^T}{\sqrt{\{[0 \ 0 \ 0 \ \cdots \ 1 \ \cdots \ 0 \ 0]\mathbf{U}^T\}\{\mathbf{U}[0 \ 0 \ 0 \ \cdots \ 1 \ \cdots \ 0 \ 0]^T\}}} \\
&= \frac{\mathbf{u}_j}{\sqrt{\mathbf{u}_j^T \mathbf{u}_j}} \\
&= \frac{\mathbf{u}_j}{\sqrt{1}} \\
&= \mathbf{u}_j
\end{aligned}$$

Thus, at a sampled location, the estimated direction equals the observed direction, which is a unit vector. This is called “exact interpolation” as in the kriging of linear RVs.

4.6 Circular Kriging Variance, σ_{CK}^2

In this section, let \mathbf{U} be the random sample of directions. Let $\mathbf{u}_i, i = 1, 2, \dots, n$ be the random direction at location $\mathbf{x}_i, i = 1, 2, \dots, n$, $\hat{\mathbf{u}}_i$ be the estimator of \mathbf{u}_i , and \mathbf{e}_i be the error vector of $\hat{\mathbf{u}}_i$ (Subsection 4.2.2, Figure 4-2). Let \mathbf{u}_0 be the random direction at unobserved (unsampled) location \mathbf{x}_0 , $\hat{\mathbf{u}}_0$ be the estimator of \mathbf{u}_0 , \mathbf{e}_0 be the random error vector equal to $\hat{\mathbf{u}}_0 - \mathbf{u}_0$ (Subsection 4.2.2, Figure 4-2), and Θ be the random angle between $\hat{\mathbf{u}}_0$ and \mathbf{u}_0 . Let σ_{CK}^2 be called the circular kriging variance and be defined as the mean squared length of the error vector, which is a measure of the variability of the circular kriging estimator.

At a sampled location, $\hat{\mathbf{u}}_i = \mathbf{u}_i$ (Section 4.5), $\mathbf{e}_i = \mathbf{0}$, and

$\sigma_{CK}^2 = E\{\|\mathbf{e}_i\|^2\} = E\{\|\mathbf{0}\|^2\} = 0$. If Θ were always the maximum of π , per Figure 4-2 $\|\mathbf{e}_0\|$ would always be 2 and σ_{CK}^2 would always be 4. However, in a circular random field, Θ is random and cannot always be π . Hence, $0 \leq \sigma_{CK}^2 < 4$.

The estimate of the circular kriging variance will now be derived. With vector \mathbf{c} and matrix \mathbf{K} containing real-valued constants computed from the cosine model ζ of circular-spatial correlation (Chapter 3, Subsection 3.6.2), let

$$\mathbf{X} = \mathbf{U}^T \mathbf{u}_0 \quad (4.20)$$

$$Y = \mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c} \quad (4.21)$$

$$g(\mathbf{X}, Y) = \frac{\mathbf{X}}{\sqrt{Y}}. \quad (4.22)$$

Note that \mathbf{X} is a random vector and Y is a random scalar. By the above definition of

σ_{CK}^2 ,

$$\begin{aligned} \sigma_{CK}^2 &= \mathbb{E} \left\{ \|\hat{\mathbf{u}}_0 - \mathbf{u}_0\|^2 \right\} \\ &\stackrel{(4.4)}{=} \mathbb{E} \{ 2(1 - \cos(\Theta)) \} \\ &= 2(1 - \mathbb{E} \{ \cos(\Theta) \}) \\ &\stackrel{(B.1)}{=} 2(1 - \mathbb{E} \{ \hat{\mathbf{u}}_0^T \mathbf{u}_0 \}) \\ &\stackrel{(4.1)}{=} 2(1 - \mathbb{E} \{ (\mathbf{U}\mathbf{w})^T \mathbf{u}_0 \}) \\ &= 2(1 - \mathbb{E} \{ \mathbf{w}^T \mathbf{U}^T \mathbf{u}_0 \}) \\ &\stackrel{(4.16)}{=} 2 \left(1 - \mathbb{E} \left\{ \left(\frac{\mathbf{K}^{-1} \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \right)^T \mathbf{U}^T \mathbf{u}_0 \right\} \right) \\ &\stackrel{(B.5)}{=} 2 - 2 \mathbb{E} \left\{ \frac{\mathbf{c}^T \mathbf{K}^{-1}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \mathbf{U}^T \mathbf{u}_0 \right\} \\ &= 2 - 2 \mathbf{c}^T \mathbf{K}^{-1} \mathbb{E} \left\{ \frac{\mathbf{U}^T \mathbf{u}_0}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}} \right\} \\ &\stackrel{(4.20), (4.21)}{=} 2 - 2 \mathbf{c}^T \mathbf{K}^{-1} \mathbb{E} \left\{ \frac{\mathbf{X}}{\sqrt{Y}} \right\} \\ &\stackrel{(4.22)}{=} 2 - 2 \mathbf{c}^T \mathbf{K}^{-1} \mathbb{E} \{ g(\mathbf{X}, Y) \} \Rightarrow \\ &\sigma_{CK}^2 = 2 - 2 \mathbf{c}^T \mathbf{K}^{-1} \mathbb{E} \{ g(\mathbf{X}, Y) \}. \quad (4.23) \end{aligned}$$

$g(\mathbf{X}, Y)$ is a nonlinear function of \mathbf{X} and Y . When confronted with a nonlinear function, we can approximate using a method in probability and statistics called the

“delta method” or “propagation of errors” (Rice 1995, p. 149). With (μ_X, μ_Y) being the expectation of (X, Y) , $g(X, Y)$ is approximated by a Taylor series about the fixed point (μ_X, μ_Y) with expansion to the random point (X, Y) . The expansion consists of one term which is nonrandom and random terms containing powers of the deltas $(X - \mu_X)$ and $(Y - \mu_Y)$. Hence, the method is called the “delta method.”

The Taylor series expansion of $g(X, Y)$ to a first order or linear approximation is

$$g(X, Y) \approx g(\mu_X, \mu_Y) + (X - \mu_X) \left\{ \frac{\partial g(X, Y)}{\partial X} \Big|_{(\mu_X, \mu_Y)} \right\} + (Y - \mu_Y) \left\{ \frac{\partial g(X, Y)}{\partial Y} \Big|_{(\mu_X, \mu_Y)} \right\}.$$

This approximation improves as the joint probability of X and Y increases in an area of the domain of $g(X, Y)$ where $g(X, Y)$ is approximately linear.

Taking the expectation of the first order linear approximation,

$$\begin{aligned} & E\{g(X, Y)\} \\ & \approx E\left\{g(\mu_X, \mu_Y) + (X - \mu_X) \left\{ \frac{\partial g(X, Y)}{\partial X} \Big|_{(\mu_X, \mu_Y)} \right\} + (Y - \mu_Y) \left\{ \frac{\partial g(X, Y)}{\partial Y} \Big|_{(\mu_X, \mu_Y)} \right\}\right\} \\ & = E\left\{\underbrace{g(\mu_X, \mu_Y)}_{\text{Constant}}\right\} + E\left\{\underbrace{(X - \mu_X)}_{\text{Random}} \underbrace{\left\{ \frac{\partial g(X, Y)}{\partial X} \Big|_{(\mu_X, \mu_Y)} \right\}}_{\text{Constant}}\right\} + E\left\{\underbrace{(Y - \mu_Y)}_{\text{Random}} \underbrace{\left\{ \frac{\partial g(X, Y)}{\partial Y} \Big|_{(\mu_X, \mu_Y)} \right\}}_{\text{Constant}}\right\} \\ & = g(\mu_X, \mu_Y) + \left\{ \frac{\partial g(X, Y)}{\partial X} \Big|_{(\mu_X, \mu_Y)} \right\} E\{X - \mu_X\} + \left\{ \frac{\partial g(X, Y)}{\partial Y} \Big|_{(\mu_X, \mu_Y)} \right\} E\{Y - \mu_Y\} \\ & = g(\mu_X, \mu_Y) + \left\{ \frac{\partial g(X, Y)}{\partial X} \Big|_{(\mu_X, \mu_Y)} \right\} \mathbf{0} + \left\{ \frac{\partial g(X, Y)}{\partial Y} \Big|_{(\mu_X, \mu_Y)} \right\} \mathbf{0} \\ & = g(\mu_X, \mu_Y) + 0 + 0 \\ & = g(E\{X\}, E\{Y\}) \Rightarrow \end{aligned}$$

$$E\{g(X, Y)\} \approx g(E\{X\}, E\{Y\}). \quad (4.24)$$

In general, $E\{g(X, Y)\} \neq g(E\{X\}, E\{Y\})$. What (4.24) means is that $g(E\{X\}, E\{Y\})$ is a first order linear approximation of $E\{g(X, Y)\}$.

Next, with $\varsigma(d_{ij})$ the expectation of the cosine of the angle between observations i and j estimated by the cosine model of circular-spatial correlation (Chapter 3, Section 3.6), $g(E\{\mathbf{X}\}, E\{Y\})$ is evaluated.

$$\begin{aligned}
 g(E\{\mathbf{X}\}, E\{Y\}) &\stackrel{(4.22)}{=} \frac{E\{\mathbf{X}\}}{\sqrt{E\{Y\}}} \\
 &\stackrel{(4.20), (4.21)}{=} \frac{E\{\mathbf{U}^T \mathbf{u}_0\}}{\sqrt{E\{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}\}}} = \\
 &= E\left\{ \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} \mathbf{u}_0 \right\} / \sqrt{\mathbf{c}^T \mathbf{K}^{-1} E\{\mathbf{U}^T \mathbf{U}\} \mathbf{K}^{-1} \mathbf{c}} \\
 &= E\left\{ \begin{bmatrix} \mathbf{u}_1^T \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_n^T \mathbf{u}_0 \end{bmatrix} \right\} / \sqrt{\mathbf{c}^T \mathbf{K}^{-1} E\left\{ \begin{bmatrix} 1 & \mathbf{u}_1^T \mathbf{u}_2 & \cdots & \mathbf{u}_1^T \mathbf{u}_n \\ \mathbf{u}_2^T \mathbf{u}_1 & 1 & \cdots & \mathbf{u}_2^T \mathbf{u}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_n^T \mathbf{u}_1 & \mathbf{u}_n^T \mathbf{u}_2 & \cdots & 1 \end{bmatrix} \right\} \mathbf{K}^{-1} \mathbf{c}} \\
 &\stackrel{(B.12)}{=} \begin{bmatrix} E\{\mathbf{u}_1^T \mathbf{u}_0\} \\ \vdots \\ E\{\mathbf{u}_n^T \mathbf{u}_0\} \end{bmatrix} / \sqrt{\mathbf{c}^T \mathbf{K}^{-1} \begin{bmatrix} E\{1\} & E\{\mathbf{u}_1^T \mathbf{u}_2\} & \cdots & E\{\mathbf{u}_1^T \mathbf{u}_n\} \\ E\{\mathbf{u}_2^T \mathbf{u}_1\} & E\{1\} & \cdots & E\{\mathbf{u}_2^T \mathbf{u}_n\} \\ \vdots & \vdots & \ddots & \vdots \\ E\{\mathbf{u}_n^T \mathbf{u}_1\} & E\{\mathbf{u}_n^T \mathbf{u}_2\} & \cdots & E\{1\} \end{bmatrix} \mathbf{K}^{-1} \mathbf{c}} \\
 &= \begin{bmatrix} \varsigma(d_{01}) \\ \vdots \\ \varsigma(d_{0n}) \end{bmatrix} / \sqrt{\mathbf{c}^T \mathbf{K}^{-1} \underbrace{\begin{bmatrix} \varsigma(d_{1,1}) & \varsigma(d_{1,2}) & \cdots & \varsigma(d_{1,n}) \\ \varsigma(d_{2,1}) & \varsigma(d_{2,2}) & \cdots & \varsigma(d_{2,n}) \\ \vdots & \vdots & \ddots & \vdots \\ \varsigma(d_{n,1}) & \varsigma(d_{n,2}) & \cdots & \varsigma(d_{n,n}) \end{bmatrix}}_{\mathbf{K}, \text{p. 89}} \mathbf{K}^{-1} \mathbf{c}} \\
 &= \begin{bmatrix} c_{01} \\ \vdots \\ c_{0n} \end{bmatrix} / \sqrt{\mathbf{c}^T \mathbf{K}^{-1} \underbrace{\mathbf{K} \mathbf{K}^{-1}}_I \mathbf{c}} \\
 &= \mathbf{c} / \sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}} \\
 &\stackrel{(4.24)}{\approx} E\{g(\mathbf{X}, Y)\} \Rightarrow
 \end{aligned}$$

$$E\{g(\mathbf{X}, Y)\} \approx \frac{\mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}} \quad (4.25)$$

Last, the first order linear approximation of the circular kriging variance will be completed.

$$\sigma_{CK}^2 \stackrel{(4.23)}{=} 2 - 2\mathbf{c}^T \mathbf{K}^{-1} \mathbf{E}\{g(\mathbf{X}, Y)\}$$

$$\stackrel{(4.25)}{\approx} 2 - 2\mathbf{c}^T \mathbf{K}^{-1} \frac{\mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}}$$

$$= 2 - 2 \frac{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}}$$

$$= 2 - 2\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}$$

$$\equiv \hat{\sigma}_{CK}^2 \Rightarrow$$

$$\hat{\sigma}_{CK}^2 = 2 - 2\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}} \quad (4.26)$$

4.7 How Distance and the Cosine Model Affect $\hat{\sigma}_{CK}^2$

When direction is estimated at a sample location \mathbf{x}_j , $j = 1, 2, \dots, n$, the vector of mean cosines \mathbf{c}_j is the j^{th} row or column of the positive definite matrix of mean cosines \mathbf{K} (Section 4.5). From p. 90, we know that $\mathbf{K}^{-1} \mathbf{c}_j = [0 \ 0 \ 0 \ \dots \ 1 \ \dots \ 0 \ 0]^T$ with 1 in the j^{th} position, hence,

$$\mathbf{c}_j^T \mathbf{K}^{-1} \mathbf{c}_j = \mathbf{c}_j^T \left[\begin{array}{cccc} 0 & 0 & 0 & \dots \\ & & & \underbrace{1}_{\text{Position } j} \\ & & & \dots \\ & & & 0 & 0 \end{array} \right]^T$$

$$= 1 \Rightarrow$$

$$\hat{\sigma}_{CK}^2(\mathbf{x}_j) \stackrel{(4.26)}{=} 2 - 2\sqrt{\mathbf{c}_j^T \mathbf{K}^{-1} \mathbf{c}_j}$$

$$= 2 - 2\sqrt{1}$$

$$= 0 \Rightarrow$$

$$\hat{\sigma}_{CK}^2(\mathbf{x}_j) = 0. \quad (4.27)$$

Further, in Section 4.5, it was proven that the estimate of direction at a sampled location is the observed direction. Then, at a sampled location, the error vector

$\mathbf{e}_j, j = 1, 2, \dots, n$ is always the zero vector $\mathbf{0}$ (Figure 4-2). Hence, at a sampled location, $\sigma_{CK}^2 = 0$, and $\hat{\sigma}_{CK}^2 = \sigma_{CK}^2 = 0$. In contrast to result (4.26), McNeill (1993, p. 46) obtained $\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}$ for the circular kriging variance. By (4.26), $\hat{\sigma}_{CK}^2$ decreases when $\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}$ increases. Hence, McNeill's result is a measure of concentration, which is opposite the sense of spread or variance.

The model parameters are the range, mean resultant length ρ of the circular distribution component of the CRF, and the nugget n_g from measurement error and close sampling. To see their effects, Figure 4-4 was constructed using the R code in Appendices K.18 and L.14, which simulates observation one with location at the origin of location coordinates, observation two with location at 1 unit of distance due north of observation one, and the estimation location at a variable distance due east of observation one. Hence, zero distance corresponds to estimation at the location of observation one.

The shape of the $\hat{\sigma}_{CK}^2$ curve resembles the inverted curve of the circular-spatial correlation model identified in the legend. The spherical curve attains the maximum at distances greater than and equal to the range. The exponential and gaussian curves appear to be asymptotic with the gaussian curve exceeding the exponential curve at a distance approximately equal to the range.

The upper left and lower right plots indicate that increasing the range increases the distance at which the maximum $\hat{\sigma}_{CK}^2$ occurs. The upper plots indicate that when ρ is increased (the distribution of the CRV is more concentrated), the estimator of direction, which depends on the variability of the CRV, is also more concentrated, and hence the $\hat{\sigma}_{CK}^2$ is reduced. The left plots indicate that the nugget introduces a

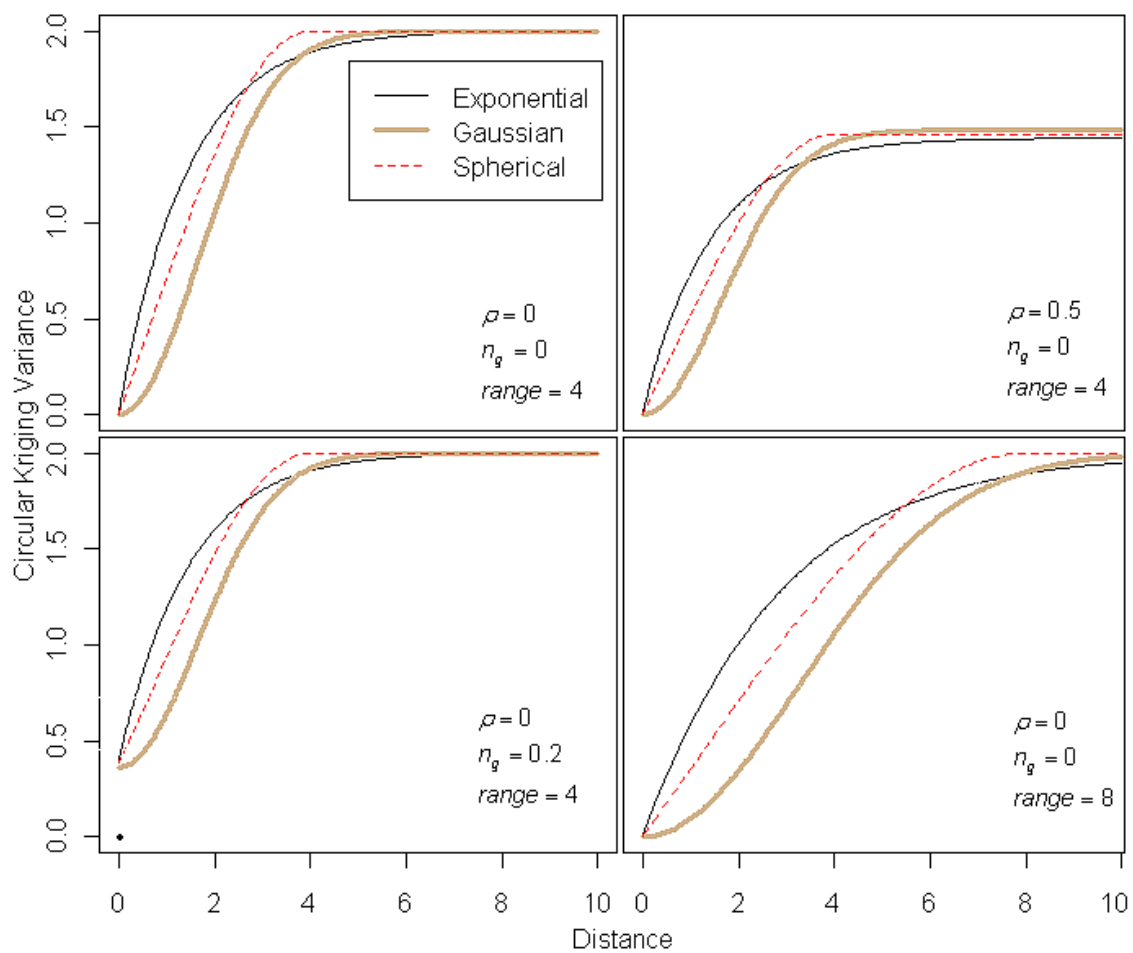


Figure 4-4. Effect of Range, Mean Resultant Length ρ , and nugget n_g on the Circular Kriging Variance $\hat{\sigma}_{CK}^2$.

discontinuity at zero distance and increases $\hat{\sigma}_{CK}^2$ at small distances relative to the range.

The maximum precision of the estimator at nonzero distances is limited by the nugget.

4.8 Chapter Summary and Future Work

The principal result of this chapter is the vector of weights \mathbf{w} of the circular kriging solution, which is

$$\mathbf{w} = \mathbf{K}^{-1}\mathbf{c} / \sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}.$$

It was derived in full detail, and shown to be optimal producing a unit vector estimate (4.17) at a stationary point (4.18) of maximum fit (Section 4.3). The approach avoided the first order Taylor series approximation of McNeill (1993, p. 39), which results in a nonunit vector estimator of direction (4.11).

A computationally efficient form of the estimator of direction (4.19) was derived with elimination of the scalar function $\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}$ in the denominator of the kriging solution. The scalar function does not affect the signs and the ratio of the magnitudes of the vector components. With h and v being the horizontal and vertical components of the vector $\mathbf{U} \mathbf{K}^{-1} \mathbf{c}$, respectively, i.e., $\begin{bmatrix} h \\ v \end{bmatrix} = \mathbf{U} \mathbf{K}^{-1} \mathbf{c}$, the estimated direction in $[0, 2\pi)$

radians at location \mathbf{x}_0 is

$$\hat{\theta}_0 = \begin{cases} \tan^{-1}(v/h), & h > 0, v \geq 0 \\ \pi/2, & h = 0, v > 0 \\ \tan^{-1}(v/h) + \pi, & h < 0 \\ \frac{3}{2}\pi, & h = 0, v < 0 \\ \tan^{-1}(v/h) + 2\pi, & h > 0, v < 0 \\ \text{undefined}, & h = v = 0. \end{cases}$$

The estimated direction at a sampled location was proven to be the observed direction. This is called “exact interpolation” as in the kriging of linear RV.

The variability of the circular kriging estimator σ_{CK}^2 was defined as the mean squared length of the error vector, $0 \leq \sigma_{CK}^2 < 4$. The first order linear approximation was derived (4.26) as

$$\hat{\sigma}_{CK}^2 = 2 - 2\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}.$$

It was proven that the circular kriging variance at a sampled location is zero. The effects on $\hat{\sigma}_{CK}^2$ of the distance to an observation and the cosine model parameters were shown.

Future work includes derivation of $\hat{\sigma}_{CK}^2$ to a higher order approximation to increase the accuracy. A nonzero nugget n_g such as from measurement error has the effect of smoothing the estimates at locations where data does not exist and not smoothing the estimates at observation locations (exact interpolation). This suggests deriving a circular kriging solution for estimation “without measurement error” as in linear kriging where the smoothing of estimates at all locations varies with the magnitude of the nugget.

CHAPTER 5

SIMULATION OF CIRCULAR RANDOM FIELDS

5.1 Introduction

This chapter defines a new method for simulation of a circular random field (CRF) by extending the inverse cumulative distribution function (CDF) method of generating a random variable (RV). A random field (RF) is a stochastic process operating over a space. A CRF is defined as a RF containing spatially correlated circular random variables (CRV). A CRV takes random directions on a unit circle with the total probability of all possible directions distributed on the unit circle with support $[0, 2\pi)$ or equivalent support $[-\pi, \pi)$ (Chapter 1, Figure 1-1). In this chapter, the support is $[-\pi, \pi)$. Spatial correlation, which is the correlation between RVs a distance d apart, increases as distance between measurement locations decreases, i.e., rotations from the mean direction tend to be more similar as distance decreases. In the form required by the circular kriging derivation of Chapter 4, spatial correlation is defined as the mean cosine of the angle between random components of directions (non random or trend component removed) vs. distance between measurement locations. An isotropic CRF is a CRF in which spatial correlation is the same in all directions of the sample space.

This chapter is organized as follows: the background is given in Section 5.2, the new method is defined in Section 5.3, the mathematical properties are discussed in Section 5.4, qualitative evaluations are given in Section 5.5, the method is extended to any continuous RV in Section 5.6, and the chapter summary and future work are given in Section 5.7.

5.2 Background

5.2.1 Random Field

To describe a type of spatial process, Besag (1974) described the RF as a stochastic model consisting of “a finite set of sites, each site having associated with it a univariate random variable.” Mathematically, let

- The dimension of the space be $d \geq 1$ (usually $d = 2$ or 3),
- \mathbf{x} be a vector of location coordinates of a measurement location in the d -dimensional space of real numbers \mathbb{R}^d ,
- $Y(\mathbf{x})$ be a RV at location \mathbf{x} ,
- $\mu(\mathbf{x})$ be the non random or trend component of $Y(\mathbf{x})$, which is the expected value of $Y(\mathbf{x})$ and a constant or a function of location \mathbf{x} , and
- $\varepsilon(\mathbf{x})$ be the random component of $Y(\mathbf{x})$ with mean zero.

Then, $Y(\mathbf{x}) = \mu(\mathbf{x}) + \varepsilon(\mathbf{x})$, and the RF is the set $\{Y(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$.

Let $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ be a set of locations in \mathbb{R}^d . Then, $\mathbf{Y} = (Y(\mathbf{x}_1), Y(\mathbf{x}_2), \dots, Y(\mathbf{x}_n))^T$ is a vector of RVs which map to $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ in \mathbb{R}^d . In a spatially correlated RF, the covariance of $Y(\mathbf{x}_i)$ and $Y(\mathbf{x}_j)$, $i, j = 1, 2, \dots, n$, is a function f which depends on the distance and direction between RVs and decreases as the distance between RVs increases. With E the expectation operator, the spatial covariance between $Y(\mathbf{x}_i)$ and $Y(\mathbf{x}_j)$ in the direction of $\mathbf{x}_j - \mathbf{x}_i$ is

$$c(Y(\mathbf{x}_i), Y(\mathbf{x}_j)) \equiv c(\mathbf{x}_i, \mathbf{x}_j) = E\{(Y(\mathbf{x}_i) - \mu(\mathbf{x}_i))(Y(\mathbf{x}_j) - \mu(\mathbf{x}_j))\} = f(\mathbf{x}_j - \mathbf{x}_i). \text{ The}$$

covariance of the vector \mathbf{Y} is the symmetric and positive definite matrix

$\mathbf{C} = (c(\mathbf{x}_i, \mathbf{x}_j))$, $i, j = 1, 2, \dots, n$. In an isotropic RF, covariance is a function of distance only, i.e., $c(\mathbf{x}_i, \mathbf{x}_j) = f(\|\mathbf{x}_j - \mathbf{x}_i\|)$.

For completeness, the spatial-temporal RV with the additional coordinate of time t is introduced. $Y(\mathbf{x}, t) = \mu(\mathbf{x}, t) + \varepsilon(\mathbf{x}, t)$. Hence, the spatial-temporal RF is the set $\{Y(\mathbf{x}, t), (\mathbf{x}, t) \in \mathbb{R}^d \otimes \mathbb{R}^+\}$. In the remainder of this chapter, RFs will be considered without the coordinate t and with $d = 2$.

5.2.2 Gaussian Random Field

Worsley (2002, p. 1674) states (with notation changed for consistency in this subsection):

The definition is simple: the Gaussian random field must be multivariate Gaussian at all finite sets of points, that is, $Y(\mathbf{x}_1), \dots, Y(\mathbf{x}_n)$ must be multivariate Gaussian for all $n > 0$ and all $\mathbf{x}_i \in \mathbb{R}^d$. Since the multivariate Gaussian is specified uniquely by its mean vector and variance matrix, then the Gaussian random field is defined uniquely by its mean function $\mu(\mathbf{x}) = E\{Y(\mathbf{x})\}$ and its covariance function $c(\mathbf{x}_i, \mathbf{x}_j) = \text{cov}(Y(\mathbf{x}_i), Y(\mathbf{x}_j))$.

Let $\mathbf{y} = (y(\mathbf{x}_1), y(\mathbf{x}_2), \dots, y(\mathbf{x}_n))^T$ be a sample from $\mathbf{Y} = (Y(\mathbf{x}_1), Y(\mathbf{x}_2), \dots, Y(\mathbf{x}_n))^T$, with expectation vector $\boldsymbol{\mu} = (\mu(\mathbf{x}_1), \dots, \mu(\mathbf{x}_n))^T$ and variance-covariance matrix \mathbf{C} . Note that \mathbf{Y} (the vector of RVs) has an expectation and covariance, and that \mathbf{y} (the vector of observations) does not. Then, a Gaussian random field (GRF) is a RF in which the RVs follow the multivariate normal distribution with density

$$\frac{1}{(2\pi)^{d/2} |\mathbf{C}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \mathbf{C}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right].$$

Quimby (1986, p. 21) states that simulation of a GRF is accomplished by finding a factorization of the desired variance-covariance matrix \mathbf{C} . Thus, the isotropic GRF can be simulated as follows:

- 1) Generate a sequence of regular or random locations $\{\mathbf{x}_i\}$, $i = 1, 2, \dots, n$, and an equal length sequence of a standard normal RV, $\{z_i\}$, $Z \sim iid N(0, 1)$.
- 2) Pair the sequence of realizations of Z with the sequence of locations. For example, z_1 is paired with location \mathbf{x}_1 , z_2 is paired with location \mathbf{x}_2 , etc. Denote the result as $z(\mathbf{x}_1), z(\mathbf{x}_2)$, etc.

- 3) With $\mathbf{x}_i = (x_{i1}, x_{i2})$, $i = 1, 2, \dots, n$, compute the pairwise distances

$$d_{ij} = \sqrt{(x_{j1} - x_{i1})^2 + (x_{j2} - x_{i2})^2}, \quad i, j = 1, 2, \dots, n,$$

- 4) With $c(d_{ij}; r, \sigma^2)$ the desired covariance function of distance d_{ij} between measurements locations, and parameters r the range and σ^2 the variance, the positive definite variance-covariance matrix $\mathbf{C} = (c_{ij})$, $i, j = 1, 2, \dots, n$ is computed with elements $c_{ij} = c(d_{ij}; r, \sigma^2)$. For example, some introductory covariance models are:

- Exponential: $c(d) = \sigma^2 \exp(-3d/r)$
- Gaussian: $c(d) = \sigma^2 \exp(-3[d/r]^2)$
- Spherical: $c(d) = \begin{cases} \sigma^2 - \sigma^2 \left(\frac{3}{2} [d/r] - \frac{1}{2} [d/r]^3 \right), & d \leq r \\ 0, & d > r. \end{cases}$

These models were derived from Bailey and Gatrell (1995, pp. 179-180). The computed covariance values in \mathbf{C} plotted vs. the corresponding pairwise distance coincides with a plot of the desired covariance model vs. distance. Two RVs are uncorrelated at a distance equal to the range parameter of the spherical covariance model, and they are assumed to be uncorrelated at the “practical range,” which is a distance $= 2r$ for the Gaussian model and a distance $= 3r$ for the exponential model.

- 5) The matrix \mathbf{C} is factorized such that $\mathbf{C} = \tilde{\mathbf{C}}\tilde{\mathbf{C}}^T$. Quimby (1986) states that the lower triangular method is computationally fast and numerically stable compared to the Cholesky decomposition.
- 6) With $\boldsymbol{\mu} = (\mu(\mathbf{x}_i))$, $i = 1, 2, \dots, n$, a vector of means, $\tilde{\mathbf{C}}$ a factorization of the desired variance-covariance matrix \mathbf{C} , $\mathbf{0}$ the $n \times n$ matrix of 0s, \mathbf{I} the $n \times n$ matrix with 1s in the main diagonal and 0 otherwise, and $\mathbf{Z} \sim N_n(\mathbf{0}, \Sigma = \mathbf{I})$ an n -vector of independent standard normal RVs, let $\mathbf{V} = \tilde{\mathbf{C}}\mathbf{Z} + \boldsymbol{\mu}$. Seber (1977, Theorem 1.1, Example 1.8, and Equation 1.4, pp. 8-11) proves that
- $E\{\mathbf{V}\} = E\{\tilde{\mathbf{C}}\mathbf{Z} + \boldsymbol{\mu}\} = \tilde{\mathbf{C}}E\{\mathbf{Z}\} + \boldsymbol{\mu} = \tilde{\mathbf{C}}\mathbf{0} + \boldsymbol{\mu} = \boldsymbol{\mu}$, and
 - $\text{Cov}(\mathbf{V}) = \text{Cov}(\tilde{\mathbf{C}}\mathbf{Z} + \boldsymbol{\mu}) = \tilde{\mathbf{C}}\text{Cov}(\mathbf{Z})\tilde{\mathbf{C}}^T = \tilde{\mathbf{C}}\mathbf{I}\tilde{\mathbf{C}}^T = \tilde{\mathbf{C}}\tilde{\mathbf{C}}^T = \mathbf{C}$.
- 7) With the vector $\mathbf{z} = (z_i)$, $i = 1, 2, \dots, n$, containing the realizations of a standard normal RV Z from step 1, compute the vector $\mathbf{v} = \tilde{\mathbf{C}}\mathbf{z} + \boldsymbol{\mu}$. Pair the elements of \mathbf{v} with the \mathbf{x}_i such that v_1 is paired with location \mathbf{x}_1 , v_2 is paired with location \mathbf{x}_2 , etc. Let this be denoted as $v(\mathbf{x}_1)$, $v(\mathbf{x}_2)$, etc. Then, the set $\{v(\mathbf{x}_i), i = 1, 2, \dots, n\}$ constitutes a simulation or realization of a GRF of mean vector $\boldsymbol{\mu}$ and variance-covariance matrix \mathbf{C} .

The function `grf` in the R package `geoR` (Ribeiro and Diggle 2001) generates simulations of GRFs for many covariance models. The function `GaussRF` in the R package `RandomFields` (Schlather 2001) generates simulations of GRFs for additional covariance models.

5.3 New Method of Generating a CRF

Figure 5-1 shows a simulation of a CRF with the von Mises CRV, parameter $\rho = .8$, which was transformed from a GRF with spherical covariance model and range $r = 10$. It was simulated using the R code in Appendices K.5 and L.3 with standardization of the realizations of the Gaussian random variable (GRV), $Z \sim N(0,1)$, to mean 0 and standard deviation 1 for demonstration. Realizations of the GRV, $z(\mathbf{x}_i), i = 1, 2, \dots, n$ with support $(-\infty, +\infty)$ are mapped to the θ with support $[-\pi, +\pi]$ as illustrated in Figure 5-2. $F_Z(z)$ the CDF of Z and $G_\theta(\theta)$ the CDF of Θ . The mapping of Z to θ is

- $z_i = -\infty$ has cumulative probability $p_i = F_Z(-\infty) = 0$ and maps to $\theta_i = G_\theta^{-1}(F_Z(-\infty)) = G_\theta^{-1}(0) = -\pi$ radians,
- $z_i = 0$ has cumulative probability $p_i = F_Z(0) = 0.5$ and maps to $\theta_i = G_\theta^{-1}(F_Z(0)) = G_\theta^{-1}(0.5) = 0$ radians,
- $z_i = +\infty$ has cumulative probability $p_i = F_Z(+\infty) = 1$ and maps to $\theta_i = G_\theta^{-1}(F_Z(+\infty)) = G_\theta^{-1}(1) = +\pi$ radians.

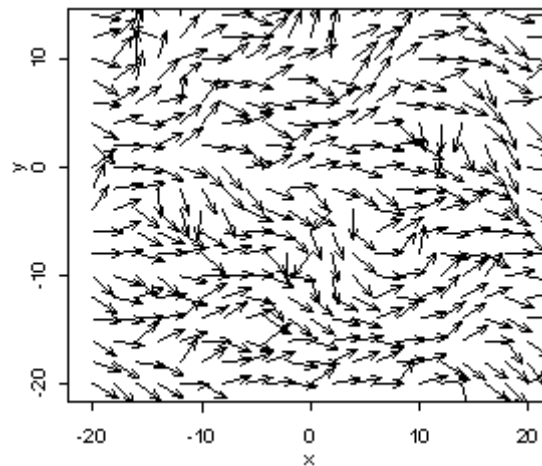


Figure 5-1. Simulated Sample of a von Mises CRF, $\rho = 0.8$, Range $r = 10$.

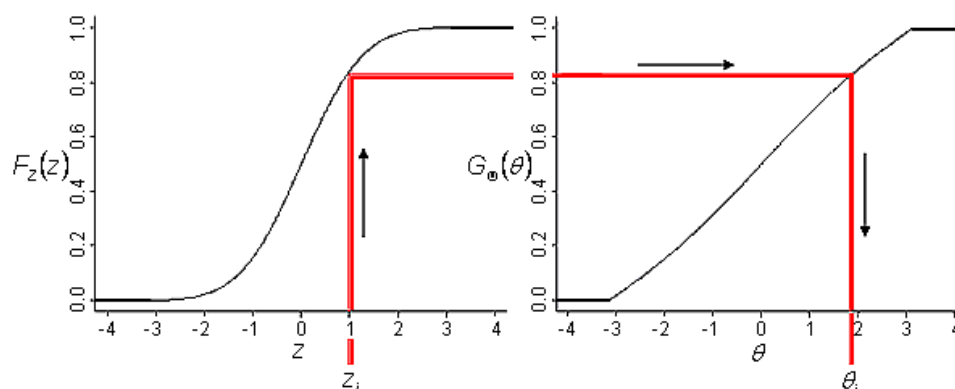


Figure 5-2. Mapping a GRV to a CRV via the CDFs F_Z and G_Θ . Direction of Θ is expressed in radian units.

A CRF may be simulated as follows:

- 1) Generate a GRF with the desired covariance model and variance $\sigma^2 = 1$. For visualization of a CRF with closer fit to the desired circular distribution, the observations $z(\mathbf{x}_i), i = 1, 2, \dots, n$ may be standardized to mean 0 and standard deviation 1. Figure 5-3 shows the standardized sample of the GRF with spherical covariance model, range $r = 10$, corresponding to Figure 5-1. Standardization should not be applied for simulation, analysis, or testing purposes as it produces undesirable effects (Subsection 5.4.4), but it may be used to obtain a single realization of an almost perfect CRF. Figure 5-3 was constructed with the R code in Appendices K.5 and L.4.

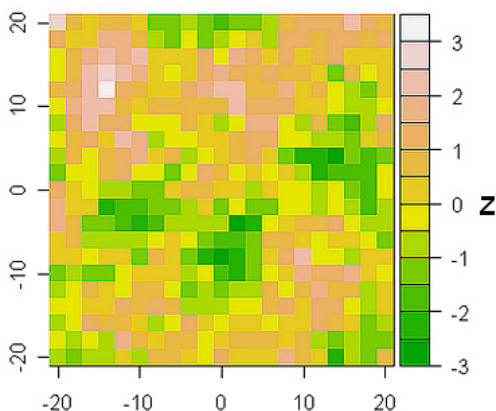


Figure 5-3. Simulated GRF with Spherical Covariance Model and Range $r = 10$ Corresponding to Figure 5-1.

- 2) For each realization $z(\mathbf{x}_i)$, $i = 1, 2, \dots, n$, compute the corresponding cumulative probability $p_i = F_Z(z(\mathbf{x}_i))$.
- 3) For the desired CRV Θ with support $[-\pi, +\pi)$ in Table 5-1, compute the inverse CDF $\theta_i = G_{\Theta}^{-1}\{F_Z(z(\mathbf{x}_i))\}$ per Table 5-2. Table 5-1 PDFs are derived in Appendix G, and Table 5-2 CDFs are derived in Appendices G and H, Equations (G.1) to (G.4), and (H.6). Note that, with exception to the triangular distribution, the PDFs of the selected distributions for support $[-\pi, +\pi)$ are identical to the PDFs for support $[0, +2\pi)$ (Chapter 3, Table 3-1).
- For the uniform CRV, the exact inverse CDF is $\theta_i = -\pi + 2\pi p_i$.
 - For the triangular CRV, θ_i is computed by applying the quadratic solution of Appendix I, Equations (I.2), (I.3), and (I.4).
 - For CDFs containing trigonometric functions, e.g., the cardioid, von Mises, and wrapped Cauchy distributions (Table 5-2), the inverse CDF does not have a closed form. For CDFs containing trigonometric functions,
 - a) Compute a table of the desired circular CDF per Table 5-2 using a sequence of θ from $-\pi$ to π .
 - b) Interpolate the θ_i on the table of the circular CDF at p_i . Let p_L and p_U be the lower and upper cumulative probabilities bounding $p_i = F_Z(z(\mathbf{x}_i))$, and θ_L and θ_U be the corresponding directions in radians in $[-\pi, +\pi)$. Then,

$$\theta_i = \theta_L + \frac{p_i - p_L}{p_U - p_L} (\theta_U - \theta_L).$$
 The R implementation is given in the last page of Section K.5, Appendix K.

Table 5-1. Circular Probability Distributions in R Package CircSpatial, $\mu = 0$, $-\pi \leq \theta < \pi$ Radians. Circular density is plotted as the length of radial between black filled unit circle and outer curve.

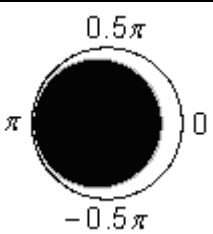
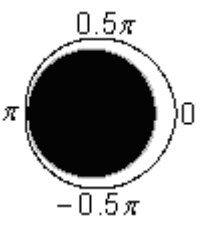
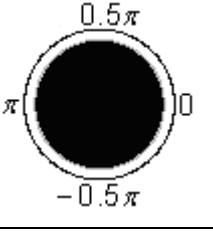
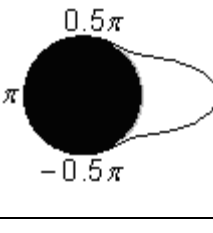
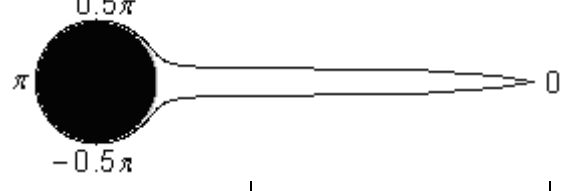
Name of Distribution	Circular PDF Plot	Circular PDF Function	Range of Parameter ρ	Value of ρ in PDF Plot
Cardioid		$\frac{1}{2\pi} [1 + 2\rho \cos(\theta)]$	$\rho = \text{mean resultant length,}$ $0 < \rho \leq 0.5$	$\rho = 0.95 \times 0.5$
Triangular		$\frac{4 - \pi^2 \rho + 2\pi\rho\delta}{8\pi}$ $\delta = \pi + \theta, -\pi \leq \theta < 0$ $\delta = \pi - \theta, 0 \leq \theta < \pi$	$0 < \rho \leq \frac{4}{\pi^2}$	$\rho = .95 \times \frac{4}{\pi^2}$
Uniform		$\frac{1}{2\pi}$	NA	NA
von Mises		$\frac{\exp(\kappa \cos(\theta))}{2\pi \sum_{j=0}^{\infty} \left(\frac{\kappa}{2}\right)^{2j} \left(\frac{1}{j!}\right)^2}$	$\kappa =$ concentration, $0 < \kappa < \infty$	$\kappa = 10.2696$ equivalent to $\rho = .95$
Wrapped Cauchy		$\frac{1}{2\pi} \frac{1 - \rho^2}{1 + \rho^2 - 2\rho \cos(\theta)}$	$0 < \rho < 1$	$\rho = 0.95 \times 1$

Table 5-2. CDFs and Inverse CDFs for Circular Distributions, $\mu = 0$, $-\pi \leq \theta < \pi$ Radians.

Distribution	CDF	Inverse CDF
Cardioid	$\frac{\theta + \pi + 2\rho \sin(\theta)}{2\pi}$	Interpolate CDF
Triangular	$+\frac{\rho}{8}\theta^2 + \frac{4 + \pi^2\rho}{8\pi}\theta + \frac{1}{2}, -\pi \leq \theta < 0$ $-\frac{\rho}{8}\theta^2 + \frac{4 + \pi^2\rho}{8\pi}\theta + \frac{1}{2}, 0 \leq \theta < \pi$	Solution in Appendix I
Uniform	$\frac{\theta + \pi}{2\pi}$	$\theta_i = -\pi + 2\pi F_Z(z(\mathbf{x}_i))$
von Mises	$\int_{-\pi}^{\theta} \frac{\exp(\kappa \cos(\phi))}{2\pi \sum_{j=0}^{\infty} \left(\frac{\kappa}{2}\right)^{2j} \left(\frac{1}{j!}\right)^2} d\phi$	Interpolate CDF
Wrapped Cauchy	$.5 - \frac{1}{2\pi} \cos^{-1}\left(\frac{(1 + \rho^2)\cos(\theta) - 2\rho}{1 + \rho^2 - 2\rho\cos(\theta)}\right), -\pi \leq \theta < 0$ $.5 + \frac{1}{2\pi} \cos^{-1}\left(\frac{(1 + \rho^2)\cos(\theta) - 2\rho}{1 + \rho^2 - 2\rho\cos(\theta)}\right), 0 \leq \theta < \pi$	Interpolate CDF

- 4) Let the pair (\mathbf{x}_i, θ_i) be denoted $\theta(\mathbf{x}_i)$. Then, the set $\{\theta(\mathbf{x}_i), i = 1, \dots, n\}$ is a simulation of the desired CRF. The function `SimulateCRF` in the R package `CircSpatial` (Appendix J, Section J.2) generates CRFs for the circular probability distributions in Table 5-1.

5.4 Mathematical Properties of the CRF

In the following subsections, the distributional and spatial properties of the circular-spatial data produced by the method of Section 5.3 will be discussed.

5.4.1 Distributional Properties of the CRF

Let Z be a continuous RV with a CDF F_Z , and define the random variable V as $V = F_Z(Z)$. Then, as shown by the CDF method in many textbooks in mathematical statistics (Rice 1995, p. 60), $V = F_Z(Z) \sim U(0,1)$, i.e., V is uniformly distributed. Also, as shown by the inverse CDF method in many textbooks in mathematical statistics (Rice 1995, p. 61), the distribution of Z can be generated by the inverse transformation $Z = F_Z^{-1}(V)$. This is a popular method for the generation of a random variable when F^{-1} is known in closed form and fast to calculate.

Now let Z be a GRV of a GRF, and $G_\Theta(\theta)$ be the CDF of the desired CRV Θ . By the CDF method, $F_Z(Z) \sim U(0,1)$, and by the inverse CDF method,

$$G_\Theta^{-1}(F_Z(Z)) \sim G_\Theta. \quad (5.1)$$

This is an extension of the inverse CDF method.

Given that a simulated GRF is a set of realizations of a GRV, it has a corresponding sample the uniform distribution equal to the cumulative probabilities of the realizations of the GRV. When this sample from the uniform distribution is input to the circular inverse CDF, the result is a sample from the desired circular distribution G_Θ .

5.4.2 Spatial Properties of the CRF

Let $\Theta(\mathbf{x})$ be a CRV at the location \mathbf{x} in 2-dimensional real space R^2 , $\mu(\mathbf{x})$ be the non random or trend component of $\Theta(\mathbf{x})$, which is the expected value of $\Theta(\mathbf{x})$ and a constant or a function of location, and $\varepsilon(\mathbf{x})$ be the random component of $\Theta(\mathbf{x})$, which follows a circular probability distribution. The parameters of the circular probability distribution, which are based on the unit vector form of the CRV, are the mean resultant direction μ , and the mean resultant length ρ , which is a measure of concentration

about the μ (Chapter 3, Subsection 3.3.1). Then, $\Theta(\mathbf{x}) = \mu(\mathbf{x}) + \varepsilon(\mathbf{x})$, and the CRF is the set $\{\Theta(\mathbf{x}), \mathbf{x} \in R^2\}$.

As required by the circular kriging derivation of Chapter 4, circular-spatial correlation is expressed as the mean cosine of the angle between random components of direction $\zeta(d)$ as a function of the distance d between measurement locations (isotropic CRF). Let D_{ij} (Chapter 3, Figure 3-3) be the angle or circular distance between the random components of direction of observations i and j , and n the number of observations of a sample,

$$D_{ij} = (\Theta(\mathbf{x}_j) - \mu(\mathbf{x}_j)) - (\Theta(\mathbf{x}_i) - \mu(\mathbf{x}_i)) = \varepsilon(\mathbf{x}_j) - \varepsilon(\mathbf{x}_i), \quad i, j = 1, 2, \dots, n, \text{ and}$$

$$\zeta(d) = E\{\cos(D_{ij})\} = E\{\cos(\varepsilon(\mathbf{x}_j) - \varepsilon(\mathbf{x}_i))\}, \|\mathbf{x}_j - \mathbf{x}_i\| = d.$$

5.4.2.1 Mean Cosine at Distance Zero

When the distance between measurement locations goes to zero, the mean cosine $\zeta(0)$ is taken of the angle between a CRV and itself, i.e., $D_{ij} \xrightarrow{\mathbf{x}_j \rightarrow \mathbf{x}_i} D_{ii} \equiv 0 \Rightarrow$

$E\{\cos(D_{ii})\} = E\{\cos(0)\} = 1$. Thus, the mean cosine at zero distance is one, which is the maximum. The mean cosine is observed to approach one as distance between measurement locations approaches zero.

Measurement error may cause measurements which are located close together to be more different, resulting in a cosineogram with a mean cosine less than one for distances close to zero. This reduction in the mean cosine is called the nugget as in the kriging of linear RVs.

5.4.2.2 Mean Cosine at Distances Where CRV Are Spatially Correlated

In a GRF with spatial correlation, observations of the GRV tend to be increasingly similar as the distance between measurement locations decreases. Because the CDF of the GRV Z is a one to one and strictly increasing function, the corresponding cumulative probabilities of the GRV will also tend to be increasingly similar. Thus, spatially correlated GRV map to spatially correlated cumulative probabilities. Conversely, by the extended inverse CDF method of Section 5.3, the transformation (5.1) of spatially correlated cumulative probabilities via the one to one and strictly increasing circular inverse CDF results in spatially correlated CRV. The measurement location coordinates are not transformed. Hence, the set of untransformed spatial coordinates of the GRF and the corresponding computed CRV constitute a simulated CRF.

Figure 5-4 contains two variograms and one cosineogram (Chapter 3). In the kriging of a linear RV, with σ^2 the variance of the RV, r the scale parameter, and $c(d, r)$ the covariance model dependent on the distance d between measurement locations, spatial dependence is expressed as the semivariance $\gamma(d) = \sigma^2 - c(d, r)$. The variogram is a plot of $\hat{\gamma}$ vs. d . It is a robust alternative to the empirical covariance. For the cosineogram, spatial correlation is expressed as the mean cosine of the angle between random components of directions. Where the CRV are uncorrelated, the mean cosine and the semivariance form a plateau, which is called the sill.

Figure 5-4 was constructed using the R code in Appendices K.4 and L.5 with standardization of the GRF of spherical covariance and range $r = 10$, and for a von Mises CRF of $\rho = 0.8$.

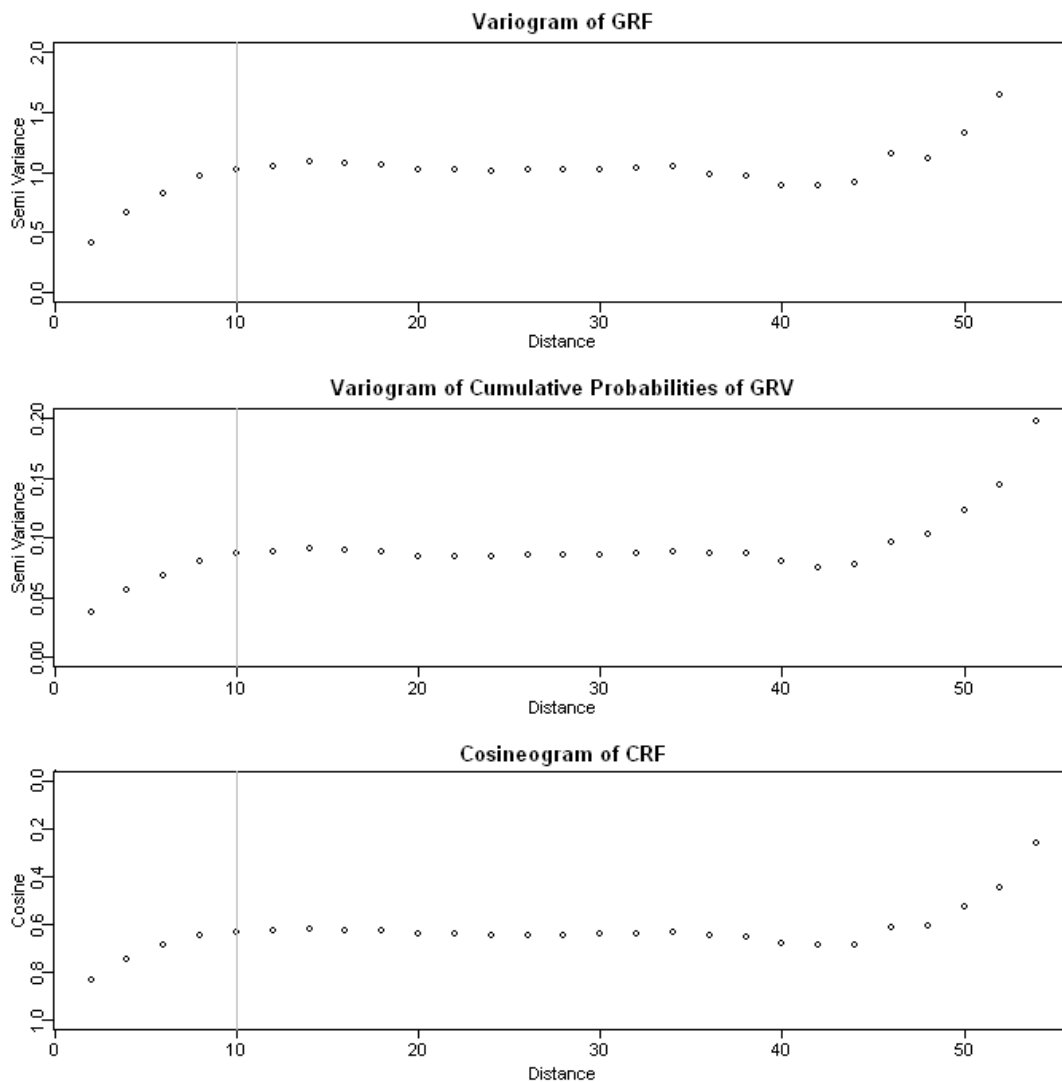


Figure 5-4. Similar Shapes of Variograms and Inverted Cosineogram Reflect Transformations of the Spatial Correlation of the GRF. This figure corresponds to Figures 5-1 and 5-3.

The cosineogram, an analogue of the covariogram (an inverted variogram), was inverted vertically to compare its shape to the shapes of the corresponding variograms. Note that the cosine axis labels are reversed in the bottom plot. The grey vertical lines are plotted at the value of the range input parameter of the GRF covariance model.

Figure 5-4, which corresponds to Figures 5-1 and 5-3, shows that the variogram of the sample of the GRF, the variogram of the cumulative probabilities of the

realizations of the GRV, and the inverted cosineogram of the CRF, are similar in shape. Hence, the computed CRVs have spatial correlation resembling, but not identical to the spatial correlation of the cumulative probabilities and of the GRF. How may the circular-spatial correlation be characterized?

The mean cosine function of distance is implicit in the relationship between the GRF covariance model and the simulated CRV. Let's build up an expression of that relationship. With (x_i, y_i) and (x_j, y_j) the coordinates of two measurement locations, the distance from location (x_i, y_i) to location (x_j, y_j) is $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Next, with $c(d_{ij})$ the covariance model of the GRF, E the expectation operator, and $Z(x_i, y_i)$ and $Z(x_j, y_j)$ mean 0 variance 1 GRVs of the GRF at (x_i, y_i) and (x_j, y_j) , respectively, the covariance function of distance is

$$\begin{aligned} c(d_{ij}) &= c\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right) \\ &= E\{Z(x_i, y_i) Z(x_j, y_j)\} - \underbrace{E\{Z(x_i, y_i)\}}_0 \underbrace{E\{Z(x_j, y_j)\}}_0 \\ &= E\{Z(x_i, y_i) Z(x_j, y_j)\} \end{aligned}$$

Next, with F_Z the CDF of the GRV Z , and G_Θ the CDF of the CRV Θ , apply the method of simulating a CRF (Section 5.3). Hence, $\Theta(x_i, y_i) = G_\Theta^{-1}(F_Z(Z(x_i, y_i)))$ and

$$\Theta(x_j, y_j) = G_\Theta^{-1}(F_Z(Z(x_j, y_j))). \text{ Conversely,}$$

$$F_Z^{-1}(G_\Theta(\Theta(x_i, y_i))) = F_Z^{-1}(G_\Theta(G_\Theta^{-1}(F_Z(Z(x_i, y_i)))))) = Z(x_i, y_i) \text{ and likewise}$$

$$F_Z^{-1}(G_\Theta(\Theta(x_j, y_j))) = Z(x_j, y_j). \text{ Substituting for } Z(x_i, y_i) \text{ and } Z(x_j, y_j) \text{ in the covariance}$$

expression, the expression of the general relationship is

$$c\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right) = E\left\{ \underbrace{F_Z^{-1}(G_\Theta(\Theta(x_i, y_i)))}_{Z(x_i, y_i)} \underbrace{F_Z^{-1}(G_\Theta(\Theta(x_j, y_j)))}_{Z(x_j, y_j)} \right\}.$$

For a specific example, let the covariance function be spherical (Subsection 5.2.2, step 4), point 3) with range r and $\sigma^2 = 1$. Then,

$$c(d_{ij}) = \begin{cases} 1 - 1.5 \frac{d_{ij}}{r} + 0.5 \left(\frac{d_{ij}}{r} \right)^3, & d_{ij} \leq r \\ 0, & d_{ij} > r. \end{cases}$$

Now let the CRV of the CRF mapping from the GRF have a cardioid circular CDF with parameter ρ (Table 5-2). Then, $G_\Theta(\Theta(x_i, y_i)) = \frac{\Theta(x_i, y_i) + \pi + 2\rho \sin(\Theta(x_i, y_i))}{2\pi}$. Hence,

the complete expression of the relationship between the spherical covariance of the GRF and the cardioid CRV of the CRF is

$$\begin{aligned} & E \left\{ F_Z^{-1} \left(\frac{\Theta(x_i, y_i) + \pi + 2\rho \sin(\Theta(x_i, y_i))}{2\pi} \right) F_Z^{-1} \left(\frac{\Theta(x_j, y_j) + \pi + 2\rho \sin(\Theta(x_j, y_j))}{2\pi} \right) \right\} \\ &= \begin{cases} 1 - 1.5 \frac{d_{ij}}{r} + 0.5 \left(\frac{d_{ij}}{r} \right)^3, & d_{ij} \leq r \\ 0, & d_{ij} > r. \end{cases} \end{aligned}$$

The problem is then to transform this nonclosed form relationship into an expression of the mean cosine of the angle between CRV vs. distance $\zeta(d_{ij})$ and the parameters r and ρ .

As an alternative to the derivation of an approximating expression characterizing the mean cosine vs. distance produced by the method of Section 5.3, the cosine curve may be described by an approximating covariance model of a GRF with translation and scaling (Chapter 3, Subsection 3.6.2). With ρ the mean resultant length of the circular probability distribution, $0 \leq \rho < 1$, n_g the nugget, $0 \leq n_g < 1 - \rho^2$, and $c(d)$ the covariance function of distance d from linear kriging with a maximum value of one,

the cosine model is

$$\zeta(d) = \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2)c(d), & d > 0. \end{cases} \quad (5.2)$$

Hence, the mean cosine is 1 at zero distance, 1 minus the nugget at distances close to 0, greater than ρ^2 and less than 1 minus the nugget at distances where CRV are correlated, and ρ^2 at distances where CRV are uncorrelated (Chapter 3, Equation (3.11)). Some introductory cosine models were given in Chapter 3, Subsection 3.6.2, and additional models are tabulated for a wide range of conditions in Appendix M, Section M.5.

Even though cosine models are not fully specified in the CRF domain as closed form expressions, but as transformations from the GRF domain, the cosine models acquire practical meaning as they apply to real world data (Chapter 3, Section 3.7, Figure 3-13).

5.4.2.3 Mean Cosine at Distances Where CRV Are Uncorrelated

Applying the method of Section 5.3, the location coordinates of the GRV are untransformed. At distances at which the GRV are uncorrelated, the transformations of the GRV are also uncorrelated. Thus, the corresponding cumulative probabilities $F_Z(z(\mathbf{x}_i))$ and simulated CRV $\theta(\mathbf{x}_i) = G_{\Theta}^{-1}(F_Z(z(\mathbf{x}_i)))$ are uncorrelated at distances where the GRV are uncorrelated. Hence, the distance at which the computed CRV are uncorrelated equals the distance at which the GRV are uncorrelated. In the example of Figure 5-4 with spherical covariance, the distance at which RVs are uncorrelated is approximately the range $r = 10$.

In Chapter 3, Section 3.3, the theoretical sill, which is the expectation of the cosine of the angle between independent random components of the CRVs was derived. The sill is equal to the square of the mean resultant length of the circular probability distribution. In Chapter 3, Section 3.4, it was determined that the vector resultant mean length is the parameter ρ of the circular distributions examined, which are the cardioid, triangular, uniform ($\rho = 0$), von Mises, and wrapped Cauchy distributions. Hence, the sill is ρ^2 . Hence, the mean cosine will be ρ^2 at distances at which the GRV are uncorrelated.

5.4.3 Overfitting Improves Fit of the Output CRV to the Desired CRV

The QQ (quantile-quantile) plot is a graphical method in which data are plotted against the expected values of a comparison distribution. The QQ plot shows a linear pattern when the data come from the comparison distribution. In Section 5.3, it is stated that the realizations of the GRV can be standardized to mean 0 and standard deviation 1. This was motivated by the observation that the variation in the mean and standard deviation of the GRV transforms to variation in the mean resultant direction and the mean resultant length, respectively, of the output CRF. Hence, standardizing the realizations of the GRV results in a closer fit of the output CRV to the desired CRV. The resulting CRF simulation is over fitted, but useful to demonstrate or visualize a closely fitting simulation.

Figure 5-5 was constructed using the R code in Appendices K.13 and L.6. For each of 30 simulations of a GRF (spherical covariance, range=10, variance=1, with standardization of the realizations of the GRV), the point coordinates of the QQ standard normal plot and the QQ circular uniform plot were accumulated separately. At the conclusion of the simulations, the point density was computed for each set of points.

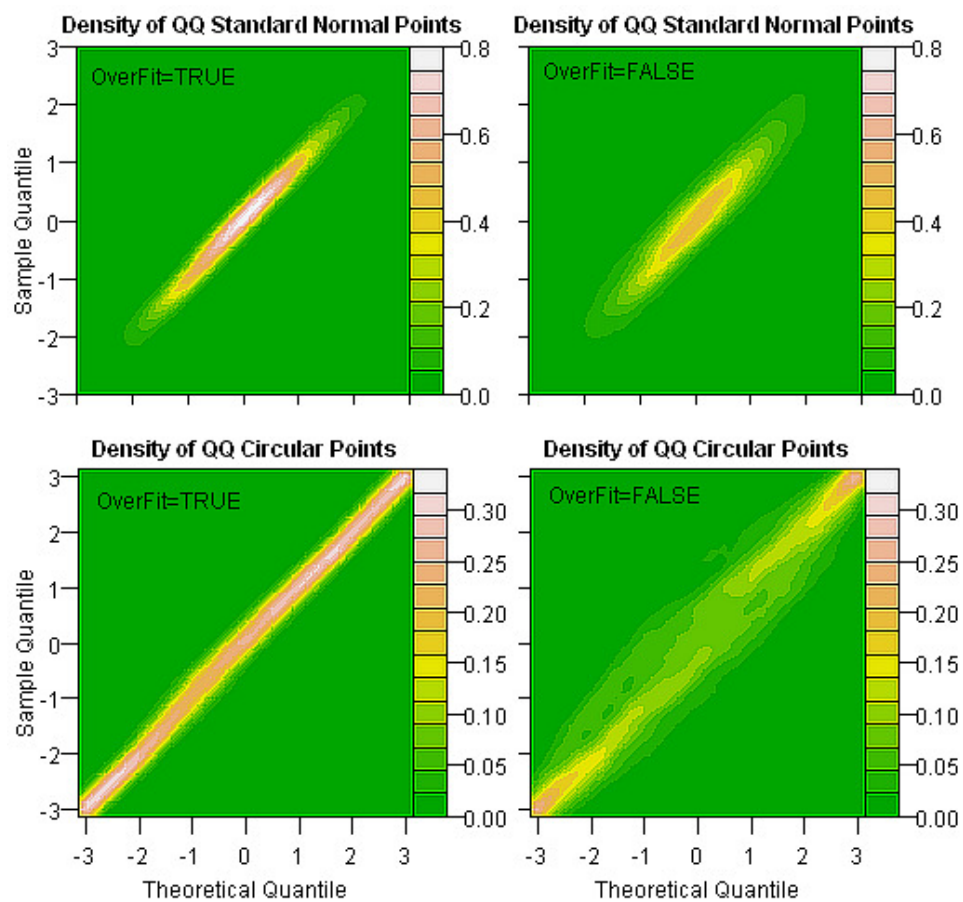


Figure 5-5. Standardization of the GRV Increases Fit of the GRV and the CRV. The uniform CRV was transformed from a GRF with spherical covariance and range=10. Circular quantiles are expressed in radian units. QQ density plots for other circular distributions with ρ at mid point of the parameter range allowed by the distribution showed results similar to Figure 5-5.

The same simulations were repeated without standardization by setting the seed of the random number generating process to the seed value prior to the first set of simulations. Again, the QQ plot points were separately accumulated, and the point densities computed for each of the two sets of points. In Figure 5-5, point density is shown as color. The greater the density about a straight line, the closer the realizations of a RV are to the comparison distribution. In the left plots, realizations of the GRV are standardized prior to transformation to the CRV (OverFit = TRUE in R package CircSpatial function SimulateCRF, Appendix J, Section J.2). In the right plots, realizations of the GRV are not standardized (OverFit = FALSE). The point mass in the left plots is more concentrated along the straight line of equality than the point mass in the right plots. This indicates that standardization results in a closer fit to the desired distributions.

Figure 5-6, which was constructed using the R code in Appendices K.13 and L.7, demonstrates the effect of decreasing ρ without standardization of the GRV on the fit of a CRV. The triangular CRV (Table 5-1) was arbitrarily selected. As ρ is decreased, dispersal of the QQ point mass increases. In the bottom plots with $\rho = 25\%$ of the maximum $(0.25 \times 4/\pi^2)$, standardization dramatically reduces dispersal of QQ point mass in the right plot. It is apparent that without standardization, the variability of the fit, which is indicated by the QQ point mass dispersal, increases as ρ decreases. This does not mean in general that standardization is desirable. In the next section, the undesirable effects of standardization are discussed.

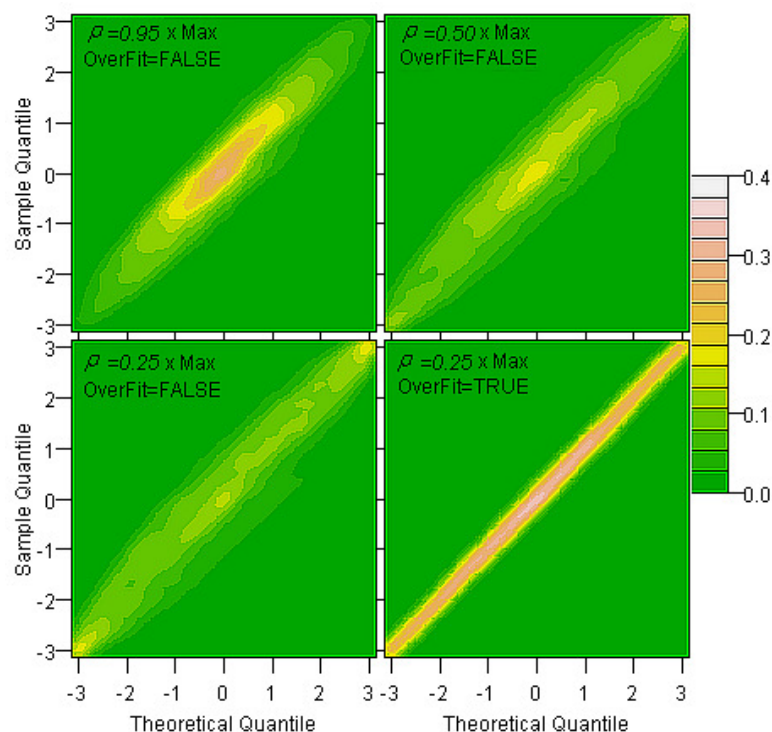


Figure 5-6. Variability of Fit of the Simulated Triangular CRV Increases as ρ Decreases. Circular quantiles are expressed in radian units.

5.4.4 Overfitting Has Unwanted Effects

Figure 5-7 was constructed by fitting 400 variograms made from the 400 simulations of a GRF with spherical covariance, range=10, variance=1. In the first set of simulations, realizations of the GRV were standardized. The sequence of simulations was repeated without standardization by setting the seed of the random number generating process to the seed value prior to the first set of simulations. The right plot without standardization has the correct variance of 1 at distance = 10. The left plot with standardization has a biased variance at distance = 10. If variation in the center and scale of the GRV is eliminated by standardization, everything that is derived from the CRF is altered. If a test were constructed based on simulations with standardization, a simulation without standardization would more likely appear as unusual and be rejected in a test of hypothesis, inflating the type 1 error.

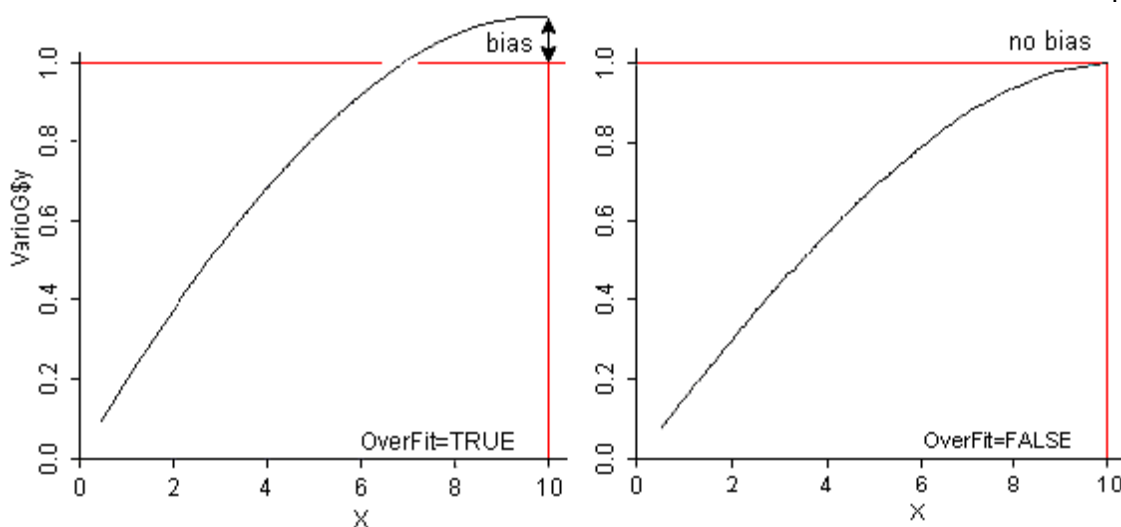


Figure 5-7. Standardization of the GRV Biases the GRF Covariance. The plots are averages of variograms (inverted covariance) of GRF with spherical covariance, range=10, and variance=1.

In summary, standardization is suitable for demonstration of a CRF with closer fit for visualization, but should not be used for the purposes such as simulation, analysis, and testing.

5.5 Qualitative Evaluations of Method of Simulating a CRF

5.5.1 Review

In this section, QQ plots will be used to show agreement with the desired probability distributions, and the variogram and the cosineogram will be used to show agreement with the desired spatial properties. In the kriging of a linear RV, with σ^2 the variance of the RV and $c(d,r)$ the covariance model with range (scale) parameter r and dependent on the distance d between measurement locations, spatial dependence is expressed as the semivariance $\gamma(d) = \sigma^2 - c(d,r)$. The variogram is a plot of $\hat{\gamma}$ vs. d . It is a robust alternative to the empirical covariance. For the cosineogram, spatial correlation is expressed as the mean cosine of the angle between random components

of directions. Where the CRV are uncorrelated, the mean cosine and the semivariance form a plateau, which is called the sill.

5.5.2 Construction of Figure 5-8

Figure 5-8 was computed using the R code in Appendices K.6 and L.8. The von Mises CRF, $\rho = 0.8$, was transformed from a realization of a GRF with spherical covariance, range $r = 10$, with standardization of the realizations of the GRV to 0 mean and standard deviation 1 for close fit. Figure 5-8, which closely corresponds with and is based on the same data as Figures 5-1, 5-3, and 5-4, provides qualitative evaluations of the standardized GRF and the CRF.

Per the R function “ppoints”, with k the index of the order statistic, n the number of observations, and $c_n = \begin{cases} 3/8, & n \leq 10 \\ 1/2, & n > 10 \end{cases}$, the theoretical quantile of the QQ plots is computed based on the corresponding cumulative probability $= (k - c_n)/(n + 1 - 2c_n)$.

5.5.3 Evaluations

In the QQ plots on the left of Figure 5-8, the degree of fit is indicated by proximity of the plotted curve to the straight line of equality through the origin. Although the QQ plot can show clear departures from the comparison distribution as a structured deviation from a straight line, minor departures may be indistinguishable from the typical variation of sampling from the comparison distribution.

The upper left plot is the QQ Standard Normal plot of the realizations of the GRV of the GRF. The blue line through the origin represents the standard normal probability distribution. The standardized realizations of the GRV (black curve) display a high degree of fit to the standard normal distribution. The GRV is over fit as described in Subsection 5.4.3.

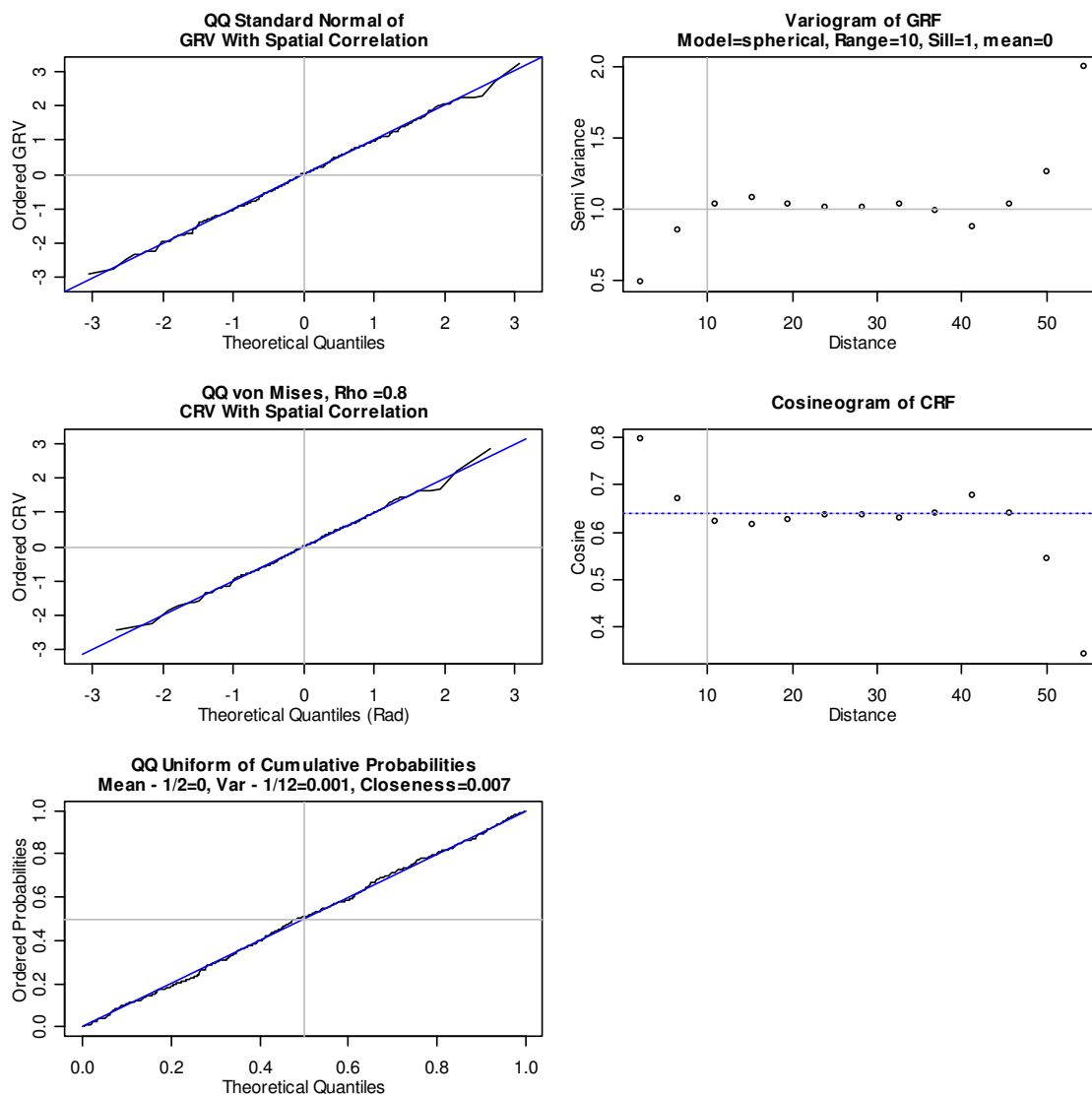


Figure 5-8. Evaluation of a von Mises CRF, $\rho = 0.8$, Overfit, Range $r = 10$. The CRF, which was transformed from a GRF with spherical covariance corresponds to Figures 5-1, 5-3, and 5-4. The QQ standard normal (top left) plot, the QQ von Mises plot (middle left), and the QQ uniform plot (bottom left) plots show simulations with close fits. The range of the variogram of the GRF (top right) and of the cosineogram (middle right) of the CRF match the input range 10. The square of the sample mean resultant length (blue dashed horizontal line) of the cosineogram is visually indistinguishable from ρ^2 (grey horizontal line).

The middle left plot is the QQ circular plot of the simulated CRV of the CRF. The title is automatically generated by the input specifications for subsequent reference. Thus, the CRF input specifications are von Mises distribution with $\rho = 0.8$. The blue line through the origin represents the desired distribution, which is the von Mises $\rho = 0.8$. The simulated CRV (black curve) displays a high degree of fit to the desired distribution.

The bottom left plot is the QQ Uniform plot of the cumulative probabilities of the realizations of the GRVs. The QQ Uniform plot displays a high degree of fit to the uniform distribution. According to the inverse CDF method, the high degree of fit predicts a high degree of fit for the CRV. The subtitle is automatically generated with three metrics of fit for evaluation. By the CDF method, the cumulative probabilities of a RV are uniformly distributed. When $X \sim U[0,1]$, $E\{X\} = 0.5$ and $Var(X) = 1/12$. Hence, the fit of the cumulative probabilities to the uniform distribution may be measured as the mean cumulative probability minus 0.5, and as the variance of cumulative probabilities minus 1/12. Then, if the variance minus 1/12 is negative, data are more concentrated in the middle. If the difference is positive, the data are more concentrated toward the tails. The deviation of the mean from 0.5 indicates an off center condition with positive deviation indicating a shift towards higher values. Additionally, "closeness" is defined as the mean vertical distance between QQ uniform plot points and the line through the origin. Hence, zero indicates a perfect fit. These metrics provide information to assess fit.

The upper right plot is a variogram reflecting the spatial properties of the GRF. The plot subtitle is generated by the input specifications for subsequent reference. Hence, the GRF input specifications are spherical covariance, range = 10, sill = 1, and mean=0. The grey vertical line is located at the input range of 10 and the grey horizontal

line is located at the input sill of 1. The variogram has a sill of about 1.0 and a range of about 10 consistent with the input specifications.

The bottom right plot is the cosineogram of the simulated CRF. The shape of the cosineogram is similar to the shape of the inverted variogram (upper right plot). The grey vertical line is the input range. The cosineogram matches the range input. The grey horizontal line, which is the theoretical sill ρ^2 , is visually indistinguishable from the blue dotted horizontal line, which is the square of the sample mean resultant length r . Thus, the cosineogram sill, the theoretical sill, and the sample mean resultant length squared are all close together. Hence, these observations evidence that the CRF has the correct range and sill.

Additional examples, with ρ set to one half of the maximum (Table 5-1), with standardization of the realizations of the GRV and selected with regard for fit, are shown in Appendix C. Further examples, with ρ set to the extremes of 5% and 95% of the maximum, with and without standardization of the GRV for comparison, and generated sequentially without regard for fit, are shown in Appendix D. The spatial properties were scored, and summarized in Table D-1. The conclusion was that the QQ plots with standardization indicated a high degree of fit. Standardization of the GRF had no apparent effect on agreement of the spatial properties of a simulation with the desired spatial properties.

5.6 Extension of the Method

The extension of the inverse CDF method to the simulation of circular random fields may be applied to any continuous RV whose CDF can be computed, or whose inverse CDF exists in closed form following the method of Section 5.3.

5.7 Chapter Summary and Future Work

The CRF was defined as a RF consisting of spatially correlated CRVs. The well known inverse CDF method, i.e., the fact that the inverse CDF of the desired RV operating on a uniform RV produces the probability distribution of the desired RV, was extended to the production of a CRF. The GRV component of a GRF with spatial correlation has spatially correlated cumulative probabilities. The inverse CDF of the desired circular probability distribution operating on the spatially correlated cumulative probabilities produces a spatially correlated CRV. The combination of the computed CRV and the untransformed coordinate locations of the corresponding realizations of the GRV is a simulation of the CRF. This method is applicable to any continuous RV.

The spatial properties of the simulated CRF were discussed. The spatial correlation of circular-spatial data is expressed as the mean cosine of the angle between random components of direction observed at a distance d apart vs. d as required by the circular kriging solution of Chapter 4. These properties include:

- 1) The mean cosine at distance zero is one. A discontinuity may exist near zero due to measurement error. The size of the discontinuity is called the nugget.
- 2) The mean cosine behavior between distance zero and the distance at which RVs are uncorrelated was characterized by closely fitting shifted and scaled positive definite covariance functions from linear kriging.
- 3) The mean cosine at distances where GRV and CRV are uncorrelated is the square of the mean resultant length of the CRV component of the CRF. For the circular probability distributions examined, it is ρ^2 , with ρ the mean resultant length parameter of the CRV.

Standardization of the realizations of the GRV to mean 0 and standard deviation 1 was examined. Standardization results in bias of the GRF variance, over fitting of the desired the CRF, and inflated type 1 error in tests based on simulation of over fitted realizations. Hence, over fitting may be used for the purpose of visualization of close fit, but should not be used for simulation, testing, or analysis. Qualitative evaluations of over fitted simulations demonstrated that CRFs produced were correct. The CRV component of the CRF had a close fit to the desired circular distribution, the sill matched ρ^2 , and the similarity of shape of the inverted cosineogram and variogram indicated that the range of the output CRF matched the desired range.

Metrics of fit were introduced based on the realizations of the uniformly distributed cumulative probabilities corresponding to the realizations of the GRV. The mean vertical distance between the QQ uniform plot points and the line of perfect fit through the origin measured the overall fit of the samples of the GRV and the CRV. The mean minus 1/2 measured shift (+ upward, - downward), and the variance minus 1/12 measured departure from the variance of the uniform RV generating the GRV and the CRV (+ more spread, - more concentrated).

Future work includes: Implementation of additional circular probability distributions for simulating CRFs; analysis of how the cosine behavior of the simulated CRF relates to the input spatial covariance model of the GRF; automatic fitting of the cosine models such as in Appendix M to the cosineogram with identification of best fit; and determination of what metrics of fit would be considered a good fit.

CHAPTER 6

COMPREHENSIVE EXAMPLE

6.1 Outline of Circular-Spatial Processes

This section combines the results of Chapters 2 - 5 to show that the theory and methods produce interpretable, and practical results. The processes involve:

- 1) Modeling the underlying trend component of the simulated circular-spatial data
- 2) Simulation of a CRF
- 3) Estimation of the spatially correlated random components of direction as the residual rotations from the trend estimate
- 4) Extracting the circular-spatial correlation as the cosineogram
- 5) Modeling the cosineogram for a smooth, continuous, and positive definite function
- 6) Kriging the residual rotations using the cosine model
- 7) Estimating the circular-spatial data
- 8) Plotting the circular-spatial estimate.

The R code used is located in Appendix L, Section L.9. References to the R package CircSpatial will be given. Arrow style (color, font, thickness) will be used consistently in closely related figures for the same type of information.

6.2 Simulation of a CRF

Figure 6-1, shows the trend model. Figure 6-2 shows a circular-spatial sample simulated by adding the trend model to a simulation of the von Mises CRF, $\mu = 0, \rho = \sqrt{0.5}$ transformed from a GRF with spherical covariance and range 4 (Section 5.3) using the R package CircSpatial function SimulateCRF (Appendix J, Section J.2). The green highlighted area will be enlarged in subsequent figures.

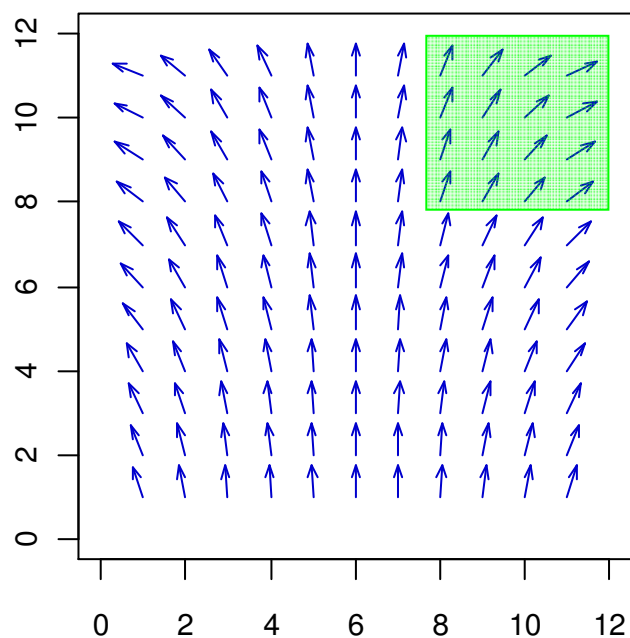


Figure 6-1. Comprehensive Example – The Trend Model, or the Underlying First Order Component of Variation. Closely related information of the green highlighted area will be enlarged in subsequent figures.

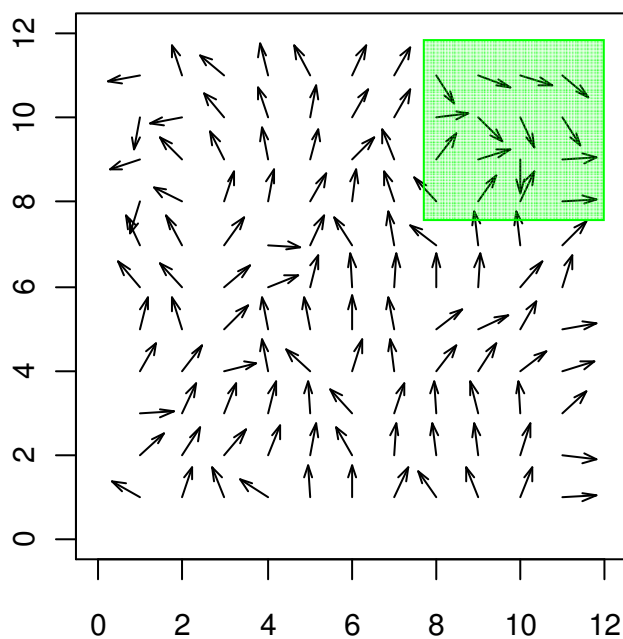


Figure 6-2. Comprehensive Example - Simulated Sample of a Von Mises CRF, $\mu = 0, \rho = \sqrt{0.5}$ with Underlying Trend. Closely related information of the green highlighted area will be enlarged in subsequent figures.

6.3 Estimation of the Spatial Trend

With \mathbf{x} the location of a measurement and $\theta(\mathbf{x})$ the direction at location \mathbf{x} , the estimate of the underlying trend was computed by regressing the $\cos(\theta(\mathbf{x}))$ and the $\sin(\theta(\mathbf{x}))$ on both the horizontal and vertical coordinates of \mathbf{x} to avoid the cross over issues of Chapter 2, Section 2.3. The estimates of $\cos(\theta(\mathbf{x}))$ and $\sin(\theta(\mathbf{x}))$ were combined using the quadrant specific inverse tangent of Chapter 3, Subsection 3.3.1, Equation (3.1). Figure 6-3 compares the trend estimate (tan arrows) vs. the true trend (blue arrows). The blue arrows have the same direction as the blue arrows in Figure 6-1. The trend estimate resembles the true trend. Closely related information of the green highlighted area will be enlarged in subsequent figures.

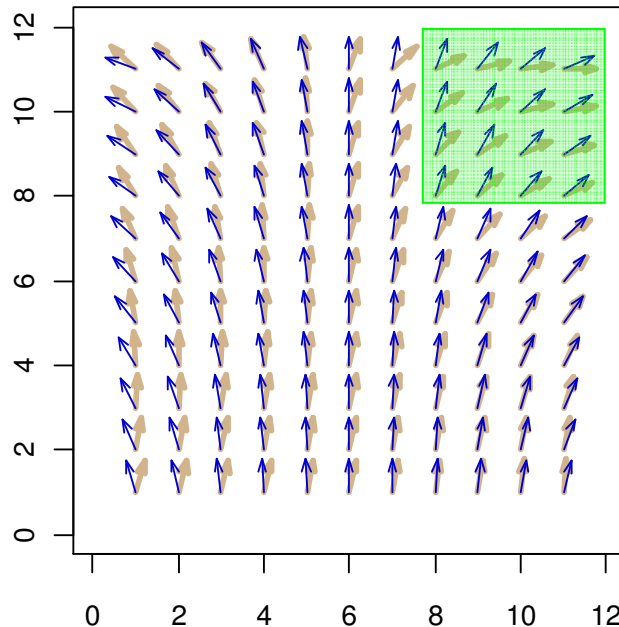


Figure 6-3. Comprehensive Example – Comparison of the Trend Estimate (Tan) with the True Trend (Blue). The trend estimate resembles the true trend. Closely related information of the green highlighted area will be enlarged in subsequent figures.

6.4 Computation of the Residuals

Spatial correlation is encoded in the residuals, which is the angular distance of the observed direction from the spatial trend. The residual at a location is plotted as a unit vector. Its direction equals the observed direction minus the estimated trend direction. The residual is positive [negative] if counterclockwise [clockwise] rotation is required to rotate the trend estimate vector into alignment with the observed direction vector. Figure 6-4 shows the observed direction as black-solid arrows corresponding to Figure 6-2, the trend estimate as tan arrows corresponding to Figure 6-3, and the residual rotation as red-dashed arrows. The plotted area corresponds to the green highlighted area in Figures 6-1 to 6-3. Figure 6-4 was computed using the R package CircSpatial function CircResidual (Appendix J, Section J.3).

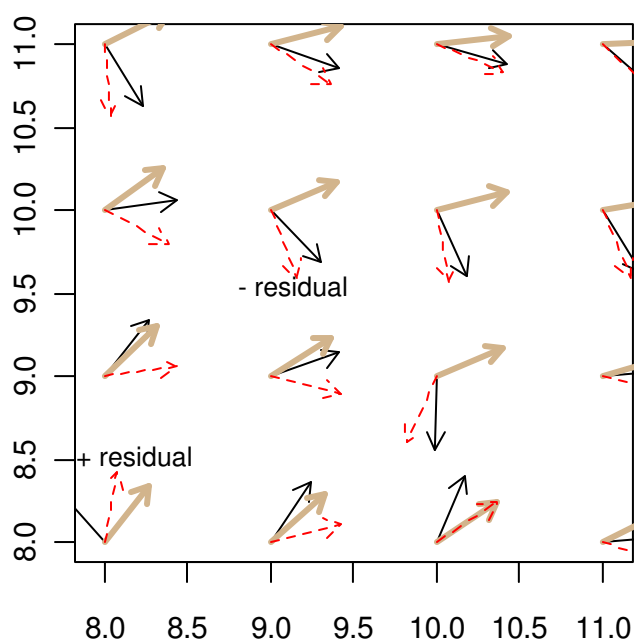


Figure 6-4. Comprehensive Example - Enlarged View of the Observed Direction (Black), Trend Estimate (Tan), and Residual Rotation (Dashed Red) Corresponding to the Green Highlighted Area in Figures 6-1 to 6-3. The residual at a location is plotted as a unit vector with direction equal to the observed direction minus the trend estimate direction. The residual is positive if counterclockwise rotation is required to rotate the trend estimate vector (tan) into alignment with the observed direction vector (black).

6.5 Plotting and Modeling the Cosineogram

Figure 6-5 shows the points of the cosineogram, and the exponential, gaussian, and spherical cosine models of circular-spatial correlation. To determine the best fit of the cosine models, the sill (plateau) was set to 0.674 to approximately center the sill of the models within the cosineogram points on the right. Then, the distance between points of evaluation of the cosineogram was varied to obtain a smooth sequence of points below the range. The range was adjusted for a best overall fit for each model. The spherical cosine model of Chapter 3, Subsection 3.6.2, Equation (3.14), with sill = 0.674 and range $r = 3.07$ was selected for best overall fit. Figure 6-5 was computed using the R package CircSpatial function CosinePlots (Appendix J, Section J.4).

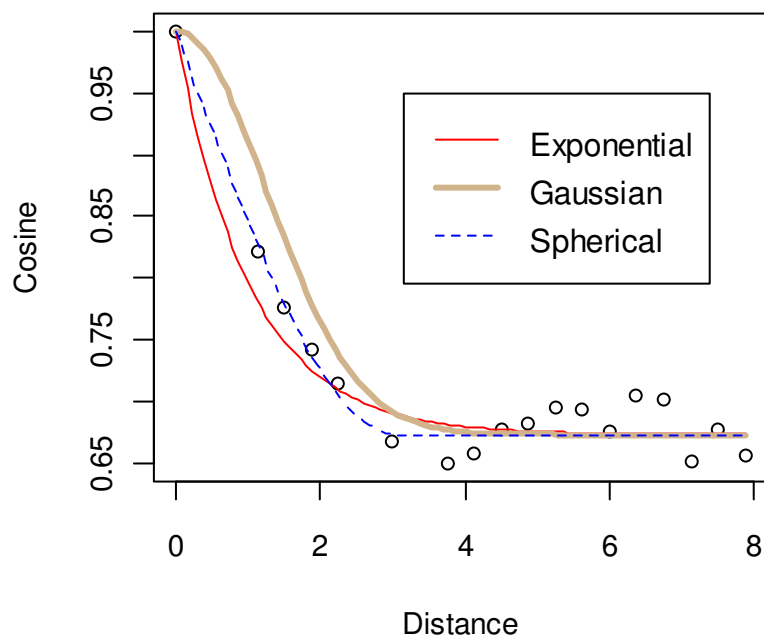


Figure 6-5. Comprehensive Example - Points of the Cosineogram, and the Exponential, Gaussian, and Spherical Cosine Models of Circular-Spatial Correlation. The spherical cosine model with range 3.07 and sill 0.674 was selected for best overall fit to the cosineogram points.

6.6 Kriging the Residuals

The estimates of the random components of direction were computed using the solution of Chapter 4, Subsection 4.2.7, Equation (4.16) with spherical cosine model, range $r = 3.07$, and sill = 0.674 as determined in Section 6.5. In Figure 6-6 corresponding to the green highlighted area in Figures 6-1 to 6-3, the kriging estimates are plotted as light grey arrows, and the residuals are plotted as red arrows. The red arrows match the direction of the residuals in Figure 6-4. The residuals coincide with the kriging estimates at measurement locations. This was proven in Chapter 4, Section 4.5, and is called “exact interpolation” in linear kriging by the spatial statistics community. Figure 6-6 was computed using the R package CircSpatial function KrigCRF (Appendix J, Section J.5).

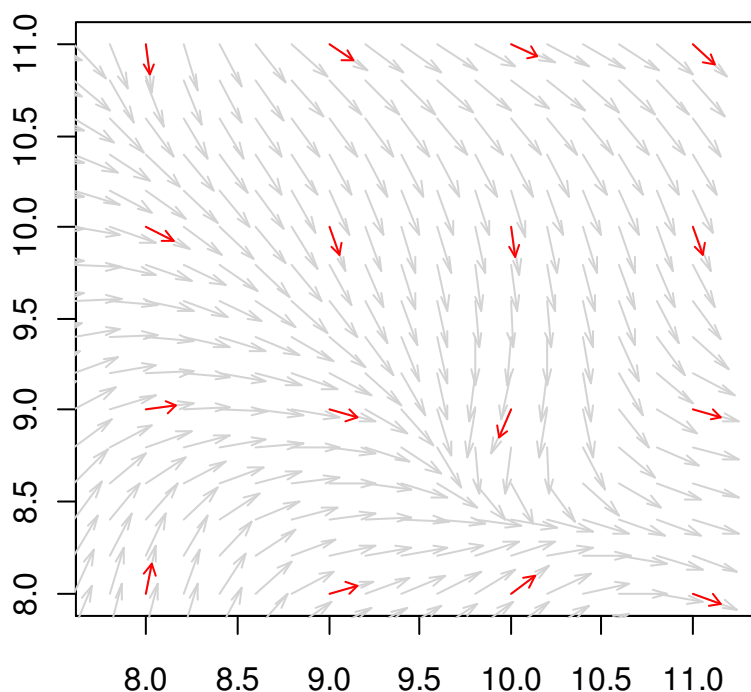


Figure 6-6. Comprehensive Example - Enlarged View of the Kriging (Light Grey) and the Residual Rotations (Red) Corresponding to the Green Highlighted Area in Figures 6-1 to 6-3. The solution of Chapter 4 produces “exact interpolation”, i.e., the kriging equals the residual where the kriging location equals the measurement location.

6.7 Interpolation of the Trend Estimate

To avoid cross over (Chapter 2, Section 2.3), interpolation of the trend estimate is obtained by separately interpolating the cosines and sines of the directions of the trend estimate. A plane is fitted to the three cosine values of the triangular partition of the grid cell of the trend estimate in which an interpolation location occurs. The interpolated cosine is the elevation of the plane at the interpolation location. The sine is interpolated by the same method. The interpolated direction is obtained by applying the quadrant specific inverse tangent of Chapter 3, Subsection 3.3.1, Equation (3.1) to the interpolated sines and cosines. In Figure 6-7, corresponding to the green highlighted area in Figures 6-1 to 6-3, the interpolated direction (purple) matches the direction of the trend estimate (tan) at a measurement location. The tan arrows match the tan arrows in Figures 6-3 and 6-4. Figure 6-7 was computed using the R package CircSpatial function `InterpDirection` (Appendix J, Section J.6).

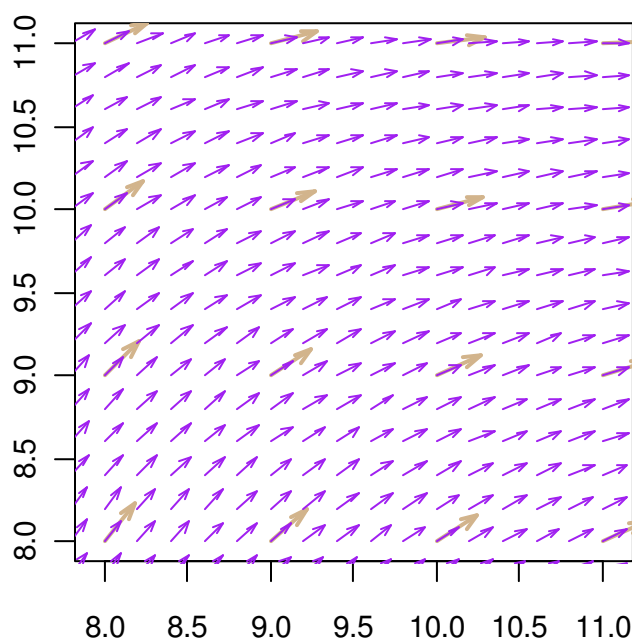


Figure 6-7. Comprehensive Example – Enlarged View of the Interpolation (Purple) of the Trend Estimate (Tan) Corresponding to the Green Highlighted Area in Figures 6-1 to 6-3. The interpolation coincides with the trend estimate at a measurement location.

6.8 Computing The Circular-Spatial Estimate

Figure 6-8, which was constructed using the R code in Appendix L, Section L.9, and corresponds to the green highlighted area in Figures 6-1 to 6-3, shows that the data (black arrows) coincide exactly with the circular-spatial estimate (gold arrows) at sample locations. This is a result of exact interpolation of both the kriging estimate and the spatial trend model. The black arrows match the black arrows in Figures 6-2 and 6-4. The opposing data at (10, 8) and (10, 9) cause the estimates of direction to collide around (10, 8.5).

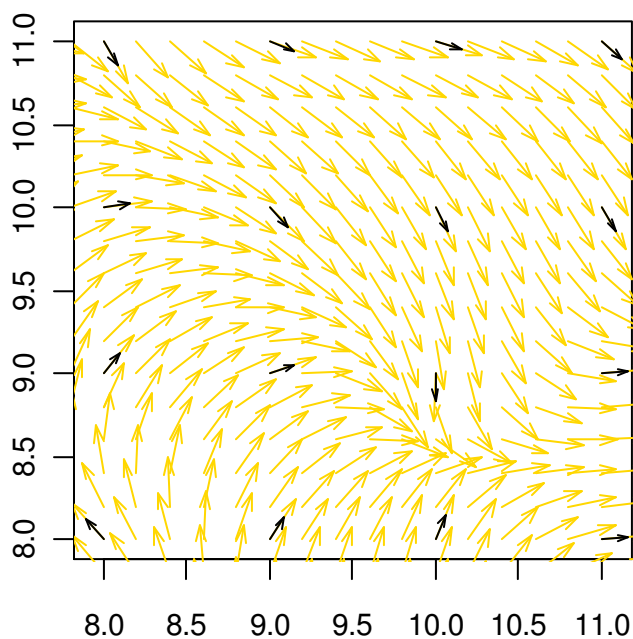


Figure 6-8. Comprehensive Example – Enlarged View of the Circular Spatial Data Estimate (Gold) and the Sample (Black) Corresponding to the Green Highlighted Area in Figures 6-1 to 6-3. At a sample location, the estimate equals the observed direction.

6.9 Imaging the Circular-Spatial Estimate

Figure 6-9 plots the circular-spatial estimate as the circular dataimage of Chapter 2. Figure 6-9 was computed using the R package CircSpatial function CircDataimage (Appendix J, Section J.10) with color wheel rotation = -105° , arrow length multiplier = 0.8, and arrow spacing of one arrow per 3 pixels horizontally and vertically starting at the lower left corner. The black arrows coincide with the black arrows of Figures 6-2, 6-4, and 6-8 where an arrow location coincides with a sample location, e.g., at (8,8).

6.10 Computing the Circular Kriging Variance

The circular kriging variance σ_{CK}^2 provides a measure of imprecision of the circular-spatial estimate. $0 \leq \sigma_{CK}^2 < 4$ (Chapter 4, Section 4.6). $\hat{\sigma}_{CK}^2$, the first order approximation of σ_{CK}^2 , was given in Equation (4.26).

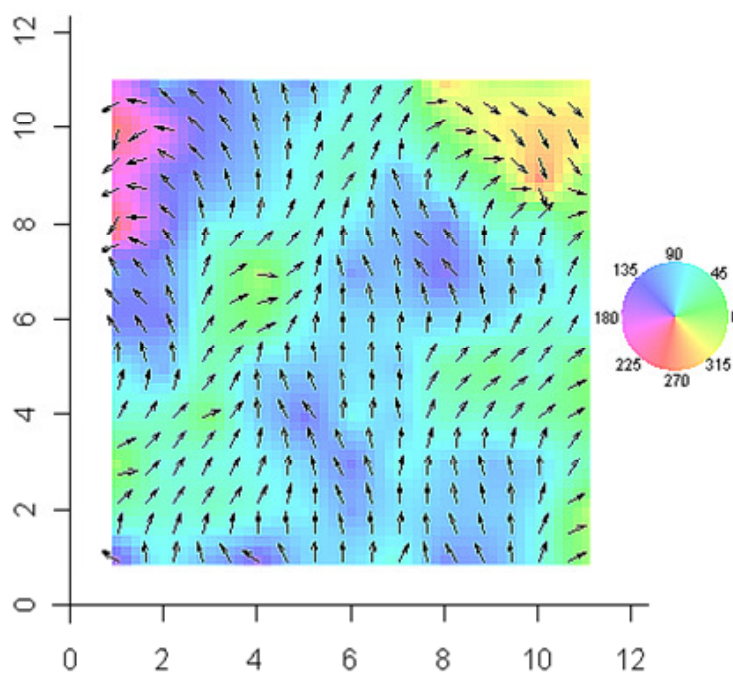


Figure 6-9. Comprehensive Example – Circular Dataimage (Left) of the Circular Spatial Data Estimate with HSV Color Wheel (Right) of Direction.

Figure 6-10 shows $\hat{\sigma}_{CK}^2$ for the estimated circular-spatial data shown in Figure 6-9 corresponding to the measurements on a regular grid shown in Figure 6-2. The measurement locations are indicated by black dots at the center of the green areas. The variability, which is indicated by the linear color scale, decreases as distance to measurement locations decreases. At a measurement location, the estimate equals the data (exact interpolation). Hence, the circular kriging variance is 0 at a measurement location. Figure 6-10 was computed using the R package CircSpatial function KrigCRF (Appendix J, Section J.5). An example of a plot of circular kriging variance with random locations is given in Appendix J, Subsection J.5.4, Figure J-14.

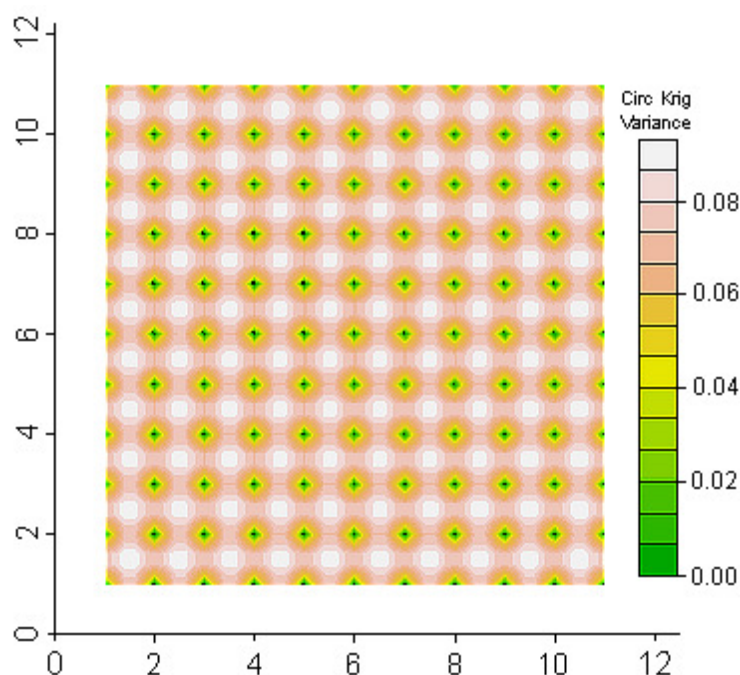


Figure 6-10. Comprehensive Example - Circular Kriging Variance. The circular kriging variance is highly structured with observation locations (black dots) on a regular grid, and decreases to zero as the distance to any measurement location decreases to zero due to “exact interpolation.”

CHAPTER 7

SUMMARY

This chapter summarizes new methods and theory for circular random fields (CRF) and circular-spatial data: the circular dataimage for visualization, the empirical cosineogram for extraction of spatial correlation, the fitted cosineogram model to provide a positive definite estimate of the circular-spatial correlation, a circular kriging solution with variance estimate, and possibly the first method to simulate circular random fields.

Chapter 1, which is the foundation for subsequent chapters, introduced the circular random variable (CRV) and the CRF. The CRF was defined as a space containing spatially correlated CRVs. With Θ the circular RV and \mathbf{x} the location in 2 dimensional space, the CRF is the set $\{\Theta(\mathbf{x}), \mathbf{x} \in \mathbb{R}^2\}$. A CRV takes random directions with the total probability of all possible directions distributed on the circular support $[0, 2\pi)$ or $[-\pi, \pi)$. Spatial correlation increases as distance between measurement locations decreases, i.e., the random components of direction tend to be more similar. An isotropic CRF was defined as a CRF in which spatial correlation is the same in all directions in space. Circular-spatial methods were summarized in a flow chart.

Chapter 2 introduced the circular dataimage. Traditional plots of circular-spatial data become less intelligible as random variation, missing data, and data density increase. These issues were resolved by the circular dataimage. The circular dataimage was defined by coding direction as the color at the same angle on a color wheel, with the color wheel defined as a sequence of three or more two-color gradients with the same color between connecting gradients. This eliminated color discontinuity at the cross over point 0 and 2π (or $-\pi$ and π) resulting in a high resolution continuous image of circular-spatial data in which fine detail on a small scale and large-scale

structure on a global scale can be simultaneously recognized. Various suitable color wheels were shown and compared to motivate experimentation, the objective being to effectively contrast and highlight interesting circular-spatial structure. The discrete color wheel was constructed from a continuous color wheel by holding color in an angular interval to the start color of the interval of the continuous color wheel. The advantages of various color schemes were summarized. Circular data image examples included:

- 1) Global and zoomed views of average wind direction
- 2) Internal flow of the Space Shuttle solid rocket motor nozzle
- 3) Families of circular time series of rocket nozzle vectoring direction angle vs. time
- 4) Direction of the Earth main magnetic horizontal (H) field
- 5) Deuteranopic (red-green color impairment) simulations
- 6) Highlighting a narrow band of directions (focus plot)
- 7) Overlay of magnitude as contour curves on circular data images
- 8) 3D polar plots of Earth main magnetic H field with magnitude as radius, and direction coded as color in a color wheel, and magnitude and direction depending on longitude and latitude.

Chapter 3 defined the empirical cosineogram. The cosineogram expresses the spatial correlation in circular-spatial data in a form consistent with the circular kriging solution of Chapter 4. The circular kriging solution requires the mean cosine of the angles between the random components of direction as a function of the distance between observation locations d . In the presence of a spatial trend, the random component equals the observed direction minus the mean direction at the observation location. In the absence of a spatial trend, the random component equals the observed direction. With $\hat{\zeta}(d)$ the mean cosine, $\|\mathbf{x}_j - \mathbf{x}_i\|$ the linear distance between observations i and j , and $N(d)$ the number of pairs of observations of direction

separated by a distance within a tolerance ε of d , the cosineogram is the plot of

$$\hat{\zeta}(d) = \frac{1}{N(d)} \sum_{\|\mathbf{x}_j - \mathbf{x}_i\| - d < \varepsilon} \cos(\theta_j - \theta_i) \quad (7.1)$$

vs. d . For a example, a cosineogram was computed from homogeneous ocean wind data in a south polar region.

The cosine model fitted to the cosineogram characterizes the spatial correlation as a smooth, continuous, and positive definite function with

- The mean cosine equals 1 at zero distance
- A reduction in the mean cosine at distance close to 0, which is called the nugget effect
- The range (scale parameter, which is also the distance CRV are uncorrelated when the input spatial covariance function is spherical)
- The sill (mean cosine at distances where CRV are uncorrelated).

The theoretical sill was derived as the square of the resultant vector mean length parameter of the circular probability distribution underlying the circular-spatial data. For the circular probability distributions uniform ($\rho = 0$), cardioid, triangular, von Mises, and wrapped Cauchy, it was determined that the resultant vector mean length equals the parameter ρ of circular probability distributions. The theoretical sill was verified by simulation.

Introductory cosine models for fitting to the empirical cosineogram were adapted from covariance functions for linear kriging by shifting and scaling. With $c(d)$ the mean cosine of the angle between random components of direction a distance d apart, ρ the resultant vector mean length of the circular probability distribution, $0 \leq \rho < 1$, n_g the nugget, $0 \leq n_g \leq 1 - \rho^2$, and $c(d)$ the covariance function with a maximum of 1,

the general form of the cosine model is

$$\zeta(d) = \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - \eta_g - \rho^2)c(d), & d > 0. \end{cases} \quad (7.2)$$

The general cosine model was proved to be positive definite for optimum circular kriging.

Chapter 4 developed a circular kriging solution. With \mathbf{w} a computed vector of weights based on the circular-spatial correlation, the estimated direction is the matrix of observed directions \mathbf{U} (each column is an observation of direction as a unit vector) post multiplied by \mathbf{w} . The approach avoided the first order Taylor series approximation of McNeill (1993), which results in a nonunit vector estimator. The solution was derived in full detail, and verified to produce a unit vector of maximum fit. With \mathbf{K} the positive definite matrix of cosines equal to the cosine model of the matrix of pairwise distances, and \mathbf{c} the vector of cosines between the estimation location and sample locations, the weight vector \mathbf{w} is

$$\mathbf{w} = \mathbf{K}^{-1}\mathbf{c} / \sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{K}^{-1} \mathbf{c}}. \quad (7.3)$$

A computationally efficient form of the estimator of direction was derived by omitting the denominator of (7.3). With h and v being the horizontal and vertical components of the vector $\mathbf{U} \mathbf{K}^{-1} \mathbf{c}$, respectively, the estimated direction in $[0, 2\pi)$ radians at location \mathbf{x}_0 is

$$\hat{\theta}_0 = \begin{cases} \tan^{-1}(v/h), & h > 0, v \geq 0 \\ \pi/2, & h = 0, v > 0 \\ \tan^{-1}(v/h) + \pi, & h < 0 \\ \frac{3}{2}\pi, & h = 0, v < 0 \\ \tan^{-1}(v/h) + 2\pi, & h > 0, v < 0 \\ \text{undefined,} & h = v = 0. \end{cases} \quad (7.4)$$

The estimated direction at a sampled location was proven to be the observed direction.

An estimate of the circular kriging variance $\hat{\sigma}_{CK}^2$ was defined as the mean squared length of the error vector between the estimator and the unobserved direction. $0 \leq \hat{\sigma}_{CK}^2 < 4$. It was approximated by a first order Taylor's series. The circular kriging variance approximation is

$$\hat{\sigma}_{CK}^2 = 2 - 2\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}. \quad (7.5)$$

McNeill's (1993) estimate $\hat{\sigma}_{CK}^2 = \sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}$ is actually proportional to concentration, which is in a sense opposite to variance, i.e., as concentration about the mean direction increases, variance about the mean direction decreases. The estimate at a sampled location is exact and has zero variance.

In Chapter 5, the CRF was defined as a set of (θ, \mathbf{x}) of where θ denotes direction and \mathbf{x} denotes the location of observation. In a CRF with spatial correlation, the mean cosine of the angle between random components of directions (nonrandom component removed) increases as the distance between observation locations decreases. The nonrandom component is removed so spatial correlation is not confused with a global or first order trend. The well known inverse cumulative distribution function (CDF) method was extended to the simulation of a CRF by applying the inverse CDF of a circular probability distribution to the cumulative probabilities of observations of the Gaussian random variables (GRV) of a Gaussian random field (GRF). The inverse CDF of a circular distribution is either a closed form expression, or interpolated from the CDF. The set of a CRV transformed from a GRV and the corresponding GRV observation location constitute a simulation of the CRF.

The mathematical properties of the simulated CRF were discussed:

- 1) The mean cosine at distance zero is defined as one.

- 2) The cosine at distances where GRV or CRV are uncorrelated is the square of the resultant vector mean length parameter of the CRV ρ as derived in Chapter 3.
- 3) At all other distances, correlation varies with distance. Spatially correlated observations of a GRF have spatially correlated cumulative probabilities because the CDF is monotonic increasing. Hence, observations which are close together have cumulative probabilities which are close together. Conversely, spatially correlated cumulative probabilities have spatially correlated CRV because the inverse CDF is also monotonic increasing. Depending on which circular distribution is being produced, this process involves 1 to 2 non closed form transformations which reshape the covariance function of the GRF. The resultant cosine curves were characterized as fitted positive definite functions adapted from the R package RandomFields (Schlather 2001) function CovarianceFct using the general form of the cosine model (7.2).

The effect of standardizing the observations of the GRF (center by subtracting the mean, and scale by dividing by the standard deviation) prior to evaluating the cumulative probabilities was considered. The effects of standardization include over fitting, bias of the spatial covariance function of the GRF, and inflated type 1 error rates in tests based on over fitted circular distributions. Standardization should not be used for analysis or development of tests. Qualitative evaluations with standardization demonstrated that a CRF was produced with very close and consistent distributional fit, and range consistent with the input specifications. The sill was consistent with both input specifications and expected value derived in Chapter 3.

Chapter 6 provides a comprehensive example combining the results of and connecting Chapters 2 – 5.

CITATIONS

- Adler, D. (2009), rgl, R package version 0.84, URL <http://rgl.neoscientists.org/about.shtml>.
- Bailey, T. C., and Gatrell, A. C. (1995), *Interactive Spatial Data Analysis*, San Francisco, CA: Prentice-Hall Inc.
- Batschelet, E. (1981), *Circular Statistics in Biology*, London: Academic Press.
- Besag, J. (1974), "Spatial Interaction and the Statistical Analysis of Lattice Systems," *Journal of the Royal Statistical Society, Series B (Methodological)*, 36(2), 192-236.
- Boogaart, K. G. v. d., and Schaeben, H. (2002a), "Kriging of Regionalized Directions, Axes, and Orientations I: Directions and Axes," *Mathematical Geology*, 34(5), 479-503.
- (2002b), "Kriging of Regionalized Directions, Axes, and Orientations II: Orientations," *Mathematical Geology*, 34(6), 671-677.
- Brewer, C. A. (1994), "Color Use Guidelines for Mapping and Visualization," in *Visualization in Modern Cartography*, eds. A. M. MacEachren and D. R. F. Taylor, Tarrytown, NY: Elsevier Science, 123-147.
- (1997), "Spectral Schemes: Controversial Color Use on Maps," *Cartography and Geographic Information Systems*, 24(4), 203-220.
- Brown, A. J. (2003), URL <http://www.aerospaceweb.org/question/spacecraft/q0127a.shtml>.
- Carr, D. B., Littlefield, R. J., Nicholson, W. L., and Littlefield, J. S. (1987), "Scatterplot Matrix Techniques for Large N," *Journal of the American Statistical Association*, 82(398), 424-436.
- CEI (2008), Ensignt, URL <http://www.ensight.com/visualization.html>.
- Edwards, C. H., and Penney, D. E. (1988), *Elementary Linear Algebra*, Englewood Cliffs, NJ: Prentice-Hall Inc.
- Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998), "Cluster Analysis and Display of Genome-Wide Expression Patterns," in *Proceedings of the National Academy of Sciences of the United States of America*, 95(25), 14863-14868.
- Fisher, N. I. (1993), *Statistical Analysis of Circular Data*, Cambridge: Cambridge University Press.
- FLUENT (2008), URL <http://www.fluent.com/software/fluent/index.htm>.

- Foley, J. D., van Dam, A., Feiner, S. K., and Huges, J. F. (1992), *Computer Graphics: Principles and Practice* (2nd ed.), Reading, MA: Addison-Wesley.
- Furrer, R., Nychka, D., and Sain, S. (2009), fields, R package version 5.02, URL <http://www.image.ucar.edu/GSP/Software/Fields/>.
- Gneiting, T., and Schlather, M. (2004), "Statistical modeling with covariance functions," In preparation.
- Grossman, S. I. (1988), *Calculus* (4th ed.), San Diego, CA: Harcourt Brace Jovanovich, 715-718.
- Ihaka, R. R. (2003), "Colour for Presentation Graphics," in *Proceedings of the 3rd International Workshop on Distributed Statistical Computing* (DSC 2003), March 20–22, Vienna, Austria, URL <http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/Ihaka.pdf>.
- Ihaka, R. R., and Gentleman, R. (1996), "R: A Language for Data Analysis and Graphics," *Journal of Computational and Graphical Statistics*, 3(5), 299-314.
- Ihaka, R. R., Murrell, P., Hornik, K., and Zeileis, A. (2009), colorspace, R package version 1.0-1.
- Intelligent Light (2008), FIELDVIEW, URL <http://www.ilight.com/products.php>.
- Jammalamadaka, S. R., and SenGupta, A. (2001), *Topics in Circular Statistics*, Singapore: World Scientific.
- Johnson, N. L., Kotz, S., and Balakrishnan, N. (1994), *Continuous Univariate Distributions* (Vol. 1, 2nd ed.), New York: John Wiley and Sons, 327-329.
- Jupp, P. E., and Mardia, K. V. (1989), "A Unified View of the Theory of Directional Statistics, 1975-1989," *International Statistical Review*, 57(3), 261-294.
- Kosara, K., Healey, C. G., Interrante, V., Laidlaw, D. H., and Ware, C. (2003), "Thoughts on User Studies: Why, How, and When," *IEEE Computer Graphics & Applications (CG&A), Visualization Viewpoints*, 23(4), 20–25, URL http://kosara.net/papers/Kosara_CGA_2003.pdf.
- Kovach Computing (2004), Oriana 2, URL <http://www.kovcomp.co.uk/oriana/oribroc.html>.
- Lantuejoul, C. H. (2002), *Geostatistical simulation*, New York: Springer.
- Lévy, P. (1939), "L'Addition des Variable Aléatoires Définies sur une Circonférence," *Bulletin de la Société Mathématique de France*, 67, 1-41.
- Lund, U. J., and Agostinelli, C. (2007), CircStats, R package version 0.2-3.
- Mardia, K. V. (1972), *Statistics of Directional Data*, London: Academic Press.

- Mardia, K. V., and Jupp, P. E. (2000), *Statistics of Directional Data*, New York: John Wiley & Sons.
- McNeill, L. (1993), "Interpolation and Smoothing of Mapped Circular Data," *South African Statistical Journal*, 27, 23-49.
- Minnotte, M. C., and West, R. W. (1998), "The Data Image: A Tool for Exploring High Dimensional Data Sets," in *Proceedings of the American Statistical Association Section on Statistical Graphics*, Alexandria, VA, 25-33, URL <http://www.math.usu.edu/~minnotte/research/pubs.html>.
- Pisces Conservation Ltd (2008), Axis, URL [http://www.pisces-conservation.com/index.html?softaxis.html\\$softwaremenu.html](http://www.pisces-conservation.com/index.html?softaxis.html$softwaremenu.html).
- Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1986), *Numerical Recipes, the Art of Scientific Computing*, Cambridge: Cambridge University Press.
- Quimby, W. F. (1986), "Selected Topics in Spatial Statistical Analysis: Non-Stationary Vector Kriging, Large Scale Conditional Simulation of Three Dimensional Gaussian Random Fields, and Hypothesis Testing in a Correlated Random Field," unpublished PhD Thesis, University of Wyoming, Laramie: Coe Library Microfilm.
- R Development Core Team (2008), R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, URL <http://www.r-project.org/>.
- Rencher, A. C. (1975), *Statistics 522, Introduction to Linear Models* (1st ed.), Provo, UT: Brigham Young University, 4-75 45 12336.
- Ribeiro Jr., P. J., and Diggle, P. J. (2001), "geoR: A Package for Geostatistical Analysis," *R News*, 1(2), 15-17, URL <http://www.est.ufpr.br/geoR>.
- Rice, J. A. (1995), *Mathematical Statistics and Data Analysis* (2nd ed.), Belmont, CA: Duxbury Press.
- Schaeben, H., Boogaart, K. G. v. d., and Apel, M. (2001), "Kriging of Surface Normal Vectors," CD-ROM in *Proceedings of 2001 Annual Conference of the International Association for Mathematical Geology*, September 6-12, Cancun, Mexico.
- Schlather, M. (1999), "Introduction to Positive Definite Functions and to Unconditional Simulation of Random Fields," Technical Report ST-99-10, Department of Mathematics and Statistics, Faculty of Applied Sciences, Lancaster University, UK.
- (2001), "Simulation and Analysis of Random Fields," *R News*, 1(2), 18-20.

- Scientific Software Group (2008), Surfer 8, URL http://www.ssg-surfer.com/html/surfer_details.html.
- Seber, G. A. F. (1977), *Linear Regression Analysis*, New York: John Wiley and Sons.
- Wand, M. P., and Jones, M. C. (1995), *Kernel Smoothing*, London: Chapman and Hall.
- Ware, C. (2004), *Information Visualization* (2nd ed.), San Francisco: Morgan Kaufmann.
- Watson, G. S., and Williams, E. J. (1956), "On the Construction of Significance Tests on the Circle and the Sphere," *Biometrika*, 43(3/4), 344-352.
- Weast, R. C. (1972), *Handbook of Chemistry and Physics* (53rd ed.), Cleveland, OH: CRC Press.
- Worsley, K. J. (2002), "Random Field, Gaussian," standard article in *Encyclopedia of Environmetrics*, eds. A.H. El-Shaarawi and W.W. Piegorsch, Chichester, NY: John Wiley & Sons, Ltd., 1674.
- Young, D. S. (1987), "Random Vectors and Spatial Analysis by Geostatistics for Geotechnical Applications," *Mathematical Geology*, 19(6), 467-479.
- Zeileis, A., Hornik, K., and Murrell, P. (2008), "Escaping RGBland: Selecting Colors for Statistical Graphics," *Computational Statistics & Data Analysis*, a preprint of an article accepted for publication at URL <http://statmath.wu-wien.ac.at/~zeileis/papers/Zeileis+Hornik+Murrell-2009.pdf>.
- Zippi, P. A. (2001), Vector Rose, URL <http://www.pazsoftware.com/VectorRose.html>.

APPENDICES

Appendix A

Notation

A nonbolded lower case letter indicates a scalar. For example, λ is the scalar Lagrange multiplier, θ_i is the i^{th} observed direction in radians, and θ_{ij} is the angle between the i^{th} and the j^{th} observed directions in radians.

A bolded lower case letter indicates a vector. For example, with superscript “T” indicating the transpose, $\mathbf{w} = [w_1 \ w_2 \ \dots \ w_n]^T$ is an n -component column vector containing real scalar weights w_i . $\mathbf{x}_i, i = 0, \dots, n$, are vectors of physical locations where direction is measured. Location will be used to determine the distance between measurement locations, which in turn, will be used to estimate the spatial correlation structure. $\mathbf{u}_i, i = 1, \dots, n$, are 2-component unit vectors of observed direction at location \mathbf{x}_i .

A bolded capital letter indicates a matrix.

$\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n] = \begin{bmatrix} \cos(\theta_1) & \cos(\theta_2) & \dots & \cos(\theta_n) \\ \sin(\theta_1) & \sin(\theta_2) & \dots & \sin(\theta_n) \end{bmatrix}$ is a sample as a matrix whose columns are directions as unit vectors. The first row is a vector of the horizontal components. The second row is a vector of the vertical components.

Equation numbers are denoted by $(m.n)$ with m the chapter number or appendix letter, and n a sequential number.

Appendix B
Linear Algebra

B.1 Identities for Vectors

Let \mathbf{u}_i and \mathbf{u}_j be unit vectors in R^2 with directions θ_i and θ_j in radians, and θ_{ij} be the angle in radians between \mathbf{u}_i and \mathbf{u}_j .

$$\begin{aligned} \cos(\theta_i - \theta_j) &\stackrel{\text{Trig Identity}}{=} \cos(\theta_i)\cos(\theta_j) + \sin(\theta_i)\sin(\theta_j) \stackrel{\text{Unit Vectors}}{=} \\ u_{i1}u_{j1} + u_{i2}u_{j2} &\stackrel{*}{=} \mathbf{u}_i^T \mathbf{u}_j \stackrel{\text{Unit Vectors}}{=} \frac{\mathbf{u}_i^T \mathbf{u}_j}{\|\mathbf{u}_i\| \|\mathbf{u}_j\|} \stackrel{**}{=} \cos(\theta_{ij}) \end{aligned} \quad (\text{B.1})$$

* (Edwards and Penney 1988, p. 211, eq. 1), ** (Edwards and Penney 1988, p. 142, eq. 6)

$$\|\mathbf{v}\| = \sqrt{\mathbf{v}^T \mathbf{v}} \quad (\text{B.2})$$

B.2 Some Properties of the Positive Definite Matrix \mathbf{K}

- A matrix \mathbf{K} is positive definite (P. D.) if and only if there exists an invertible matrix \mathbf{P} such that $\mathbf{K} = \mathbf{P}\mathbf{P}^T$.
- $\mathbf{K}^T = (\mathbf{P}\mathbf{P}^T)^T = \mathbf{P}\mathbf{P}^T = \mathbf{K} \Rightarrow \mathbf{K}$ symmetric.
- $\mathbf{c}^T \mathbf{K} \mathbf{c} = \mathbf{c}^T \mathbf{P}\mathbf{P}^T \mathbf{c} \equiv \mathbf{d}^T \mathbf{d} = \sum_i d_i^2 \stackrel{P \text{ non-singular, } \mathbf{c} \neq \mathbf{0}}{>} 0 \Rightarrow \mathbf{K} \text{ P.D.} \Rightarrow \mathbf{c}^T \mathbf{K} \mathbf{c} > 0, \forall \mathbf{c} \neq \mathbf{0}$
- Let \mathbf{Q} and $\mathbf{\Lambda}$ be the eigenvector and diagonal eigenvalue matrices of \mathbf{K} .

$$\begin{aligned} 0 &< \mathbf{x}^T \mathbf{K} \mathbf{x} \quad \forall \mathbf{x} \neq \mathbf{0} \\ &= \mathbf{x}^T \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T \mathbf{x} \\ &\stackrel{\mathbf{y} = \mathbf{Q}^T \mathbf{x}}{=} \mathbf{y}^T \mathbf{\Lambda} \mathbf{y} \\ &\stackrel{\mathbf{\Lambda} \text{ Diag}}{=} \sum_i y_i^2 \lambda_i \end{aligned}$$

$$\begin{aligned}
& \text{Let } \mathbf{y} = \mathbf{e}_j, \\
& \text{elementary vector} \\
& \Rightarrow \lambda_j > 0 \Rightarrow \\
& \mathbf{K P.D.} \Rightarrow \lambda_i \text{ of } \mathbf{\Lambda} > 0
\end{aligned}$$

Reversing the order of the proof, leads to

$$\lambda_i \text{ of } \mathbf{\Lambda} > 0 \Leftrightarrow \mathbf{K P.D.} \quad (\text{B.3})$$

B.3 Theorem: The P. D. Matrix Has an Inverse

$$\begin{aligned}
& \stackrel{\text{B.2}}{\mathbf{K}} = \mathbf{P P}^T \Rightarrow \\
& \mathbf{K}^{-1} = (\mathbf{P P}^T)^{-1} \\
& \stackrel{\text{P invertible}}{=} (\mathbf{P}^T)^{-1} \mathbf{P}^{-1}. \\
& \mathbf{P}^T (\mathbf{P}^{-1})^T = (\mathbf{P}^{-1} \mathbf{P})^T \\
& \quad = \mathbf{I}^T \\
& \quad = \mathbf{I} \Rightarrow \\
& (\mathbf{P}^{-1})^T = (\mathbf{P}^T)^{-1} \Rightarrow \\
& \mathbf{K}^{-1} = (\mathbf{P}^T)^{-1} \mathbf{P}^{-1} \\
& \quad = (\mathbf{P}^{-1})^T \mathbf{P}^{-1} \Rightarrow \\
& \mathbf{K}^{-1} = (\mathbf{P}^{-1})^T \mathbf{P}^{-1} \quad (\text{B.4})
\end{aligned}$$

B.4 Theorem: The Inverse of P. D. Matrix Is Symmetric

$$\begin{aligned}
& \mathbf{K}^{-1} \mathbf{K} = \mathbf{I} \\
& \quad = \mathbf{I}^T \\
& \quad = (\mathbf{K K}^{-1})^T \\
& \quad = (\mathbf{K}^{-1})^T \mathbf{K}^T \\
& \stackrel{\text{K=PP}^T \text{ symmetric}}{=} (\mathbf{K}^{-1})^T \mathbf{K} \Rightarrow \\
& \mathbf{K}^{-1} \mathbf{K K}^{-1} = (\mathbf{K}^{-1})^T \mathbf{K K}^{-1} \Rightarrow \\
& \mathbf{K}^{-1} = (\mathbf{K}^{-1})^T \Rightarrow \\
& \mathbf{K}^{-1} = (\mathbf{K}^{-1})^T \quad (\text{B.5})
\end{aligned}$$

B.5 Theorem: The Inverse of P. D. Matrix Is P. D.

$$\begin{aligned}
 \mathbf{K}^{-1} &= (\mathbf{P}\mathbf{P}^T)^{-1} \\
 &= (\mathbf{P}^T)^{-1}\mathbf{P}^{-1} \\
 &\stackrel{B.3: (\mathbf{P}^{-1})^T = (\mathbf{P}^T)^{-1}}{=} (\mathbf{P}^{-1})^T \mathbf{P}^{-1} \\
 &\stackrel{\mathbf{Q}^T = \mathbf{P}^{-1}}{=} \mathbf{Q}\mathbf{Q}^T. \\
 \mathbf{Q}^T = \mathbf{P}^{-1} &\Rightarrow \mathbf{Q} = (\mathbf{P}^{-1})^T. \\
 \mathbf{P}^T(\mathbf{P}^{-1})^T &= (\mathbf{P}^{-1}\mathbf{P})^T = \mathbf{I} \Rightarrow \\
 \mathbf{Q} \text{ invertible} &\Rightarrow \mathbf{K}^{-1} = \mathbf{Q}\mathbf{Q}^T \text{ P. D.}
 \end{aligned}$$

$$\mathbf{K} \text{ P. D.} \Rightarrow \mathbf{K}^{-1} \text{ P. D.} \quad (\text{B.6})$$

B.6 Some Properties of the Negative Definite Matrix

- A matrix \mathbf{K} is negative definite (N. D.) if $\mathbf{c}^T \mathbf{K} \mathbf{c} < 0 \forall \mathbf{c} \neq \mathbf{0}$.
- \mathbf{K} N. D. $\Rightarrow -\mathbf{K} \equiv \mathbf{M}$ P. D. $\Rightarrow \mathbf{K}^T = (-\mathbf{M})^T = -\mathbf{M} = \mathbf{K} \Rightarrow \mathbf{K}$ symmetric.
- This Th. will prove that the eigenvalues of a N. D. matrix are negative. Let \mathbf{Q} and $\mathbf{\Lambda}$ be the eigenvector and diagonal eigenvalue matrices of \mathbf{K} , respectively.

\mathbf{K} N. D. \Rightarrow

$$\begin{aligned}
 \forall \mathbf{x} \neq \mathbf{0} \\
 0 &> \mathbf{x}^T \mathbf{K} \mathbf{x} \\
 &= \mathbf{x}^T \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \mathbf{x} \\
 \mathbf{y} = \mathbf{Q}^T \mathbf{x} \\
 &= \mathbf{y}^T \mathbf{\Lambda} \mathbf{y} \\
 &= \sum_i y_i^2 \lambda_i \\
 \text{Let } \mathbf{y} = \mathbf{e}_j, \\
 \text{elementary vector} \\
 &\Rightarrow 0 > \lambda_j \Rightarrow
 \end{aligned}$$

\mathbf{K} N. D. $\Rightarrow \lambda_i$ of $\mathbf{\Lambda} < 0$

Reversing the order of the proof, leads to

$$\lambda_i \text{ of } \mathbf{A} < 0 \Leftrightarrow \mathbf{K} \text{ N.D.} \quad (\text{B.7})$$

B.7 Derivatives Required for Kriging

Let $\mathbf{c} = [c_1 \ \dots \ c_n]^T$ and $\mathbf{w} = [w_1 \ \dots \ w_n]^T \Rightarrow \mathbf{w}^T \mathbf{c} = w_1 c_1 + \dots + w_n c_n$.

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^T \mathbf{c}) &\equiv \begin{bmatrix} \frac{\partial}{\partial w_1} (\mathbf{w}^T \mathbf{c}) \\ \frac{\partial}{\partial w_2} (\mathbf{w}^T \mathbf{c}) \\ \vdots \\ \frac{\partial}{\partial w_n} (\mathbf{w}^T \mathbf{c}) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial}{\partial w_1} (w_1 c_1 + \dots + w_n c_n) \\ \vdots \\ \frac{\partial}{\partial w_n} (w_1 c_1 + \dots + w_n c_n) \end{bmatrix} \\ &= \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = \mathbf{c} \Rightarrow \end{aligned}$$

$$\frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^T \mathbf{c}) = \mathbf{c} \quad (\text{B.8})$$

Let $\mathbf{K} = \begin{bmatrix} k_{11} & \dots & k_{1n} \\ \vdots & \ddots & \vdots \\ k_{1n} & \dots & k_{nn} \end{bmatrix}$ be symmetric and

$$\mathbf{w}^T \mathbf{K} \mathbf{w} = k_{11} w_1^2 + 2k_{12} w_1 w_2 + \dots + 2k_{1n} w_1 w_n + \dots + k_{nn} w_n^2 \Rightarrow$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^T \mathbf{K} \mathbf{w}) &\equiv \begin{bmatrix} \frac{\partial}{\partial w_1} (\mathbf{w}^T \mathbf{K} \mathbf{w}) \\ \vdots \\ \frac{\partial}{\partial w_n} (\mathbf{w}^T \mathbf{K} \mathbf{w}) \end{bmatrix} \\ &= \begin{bmatrix} 2k_{11} w_1 + \dots + 2k_{1n} w_n \\ \vdots \\ 2k_{1n} w_1 + \dots + 2k_{nn} w_n \end{bmatrix} \end{aligned}$$

$$= 2 \begin{bmatrix} k_{11} & \cdots & k_{1n} \\ \vdots & \ddots & \vdots \\ k_{1n} & \cdots & k_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$$

$$= 2\mathbf{K}\mathbf{w} \Rightarrow$$

$$\frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^T \mathbf{K} \mathbf{w}) = 2\mathbf{K}\mathbf{w} \quad (\text{B.9})$$

The following derivative is required by (B.11). Let $\mathbf{y} = \mathbf{K}\mathbf{w} \Rightarrow y_i = \sum_j K_{ij} w_j$.

$$\frac{\partial \mathbf{K}\mathbf{w}}{\partial \mathbf{w}} = \left(\frac{\partial}{\partial \mathbf{w}} \right) \mathbf{y}^T$$

$$= \begin{pmatrix} \frac{\partial}{\partial w_1} \\ \vdots \\ \frac{\partial}{\partial w_n} \end{pmatrix} (y_1 \cdots y_n)$$

$$= \begin{pmatrix} \frac{\partial y_1}{\partial w_1} & \cdots & \frac{\partial y_n}{\partial w_1} \\ \vdots & & \vdots \\ \frac{\partial y_1}{\partial w_n} & \cdots & \frac{\partial y_n}{\partial w_n} \end{pmatrix}$$

$$= \begin{pmatrix} K_{11} & \cdots & K_{n1} \\ \vdots & & \vdots \\ K_{1n} & \cdots & K_{nn} \end{pmatrix}$$

$$\stackrel{\text{B.2}}{=} \begin{pmatrix} K_{11} & \cdots & K_{1n} \\ \vdots & & \vdots \\ K_{n1} & \cdots & K_{nn} \end{pmatrix}$$

$$= \mathbf{K} \Rightarrow$$

$$\frac{\partial \mathbf{K}\mathbf{w}}{\partial \mathbf{w}} = \mathbf{K} \quad (\text{B.10})$$

B.8 The Requirements for Maximization

Theorems 1.51 and 1.52 (a) from Rencher (1975) are required to show that the kriging solution is a maximum.

B.8.1 Theorem 1.51

“If $u = f(w_1, w_2, \dots, w_n)$ is such that all the first and second partial derivatives are continuous, and if \mathbf{B} is the matrix whose $(i, j)^{th}$ element is $\frac{\partial^2 u}{\partial w_i \partial w_j}$, then at the point

where $\frac{\partial u}{\partial \mathbf{w}} = \mathbf{0}$, u has a minimum if \mathbf{B} is positive definite and a maximum if \mathbf{B} is negative

definite.” In one-dimensional calculus, the first derivative of a function is decreasing where the second derivative is negative. Hence, at the point where the first derivative is zero and the second derivative is negative, the function is at a maximum. In multi-dimensional calculus, the corresponding second derivative is a matrix of second partial derivatives called the Hessian. Where the first derivatives are zero and the Hessian is negative definite, the function is at a maximum.

B.8.2 Theorem 1.52

“If $u = f(w_1, w_2, \dots, w_n) = \mathbf{w}^T \mathbf{K} \mathbf{w} + \mathbf{w}^T \mathbf{c} + d$, where \mathbf{K} is positive definite and d is a scalar, then the matrix $\mathbf{B} = \left(\frac{\partial^2 u}{\partial w_i \partial w_j} \right) = 2\mathbf{K}$.”

$$\begin{aligned}
\left(\frac{\partial^2 u}{\partial \mathbf{w}_i \partial \mathbf{w}_j}\right) &= \left(\frac{\partial}{\partial \mathbf{w}_j} \frac{\partial u}{\partial \mathbf{w}_i}\right) \\
&= \left(\frac{\partial}{\partial \mathbf{w}_j} \frac{\partial (\mathbf{w}^T \mathbf{K} \mathbf{w} + \mathbf{w}^T \mathbf{c} + d)}{\partial \mathbf{w}_i}\right) \\
&\stackrel{(B.8), (B.9)}{=} \left(\frac{\partial (2\mathbf{K} \mathbf{w} + \mathbf{c})}{\partial \mathbf{w}_j}\right) \\
&= \frac{\partial (2\mathbf{K} \mathbf{w} + \mathbf{c})}{\partial \mathbf{w}} \\
&= 2 \frac{\partial (\mathbf{K} \mathbf{w})}{\partial \mathbf{w}} \\
&\stackrel{(B.10)}{=} 2\mathbf{K} \Rightarrow
\end{aligned}
\tag{B.11}$$

B.9 Expectation

With \mathbf{Z} a random matrix of elements Z_{ij} , or $\mathbf{Z} = [(Z_{ij})]$, and E the expectation operator, Seber (1977, p. 8) defined the expectation of a matrix as

$$E\{\mathbf{Z}\} = E\{[(Z_{ij})]\} = [(E\{Z_{ij}\})] \tag{B.12}$$

Appendix C

Qualitative Evaluations of Other CRFs with Standardization

Appendix C extends the example of Chapter 5, Section 5.5, Figure 5-9, to other circular distributions: cardioid, triangular, uniform, and wrapped Cauchy. The purpose of these figures is to show that the method produces the desired distributional and spatial properties of the CRF. Figures C-1 to C-4 show the distributional fits on the left as QQ plots and the spatial properties of the samples on the right. Details of the interpretation are given in Chapter 5, Subsection 5.5.3. These figures were constructed as described in Chapter 5, Subsection 5.5.2 using the R code in Appendix L, Section L.8. To facilitate verification of the desired spatial properties, the spherical covariance model was chosen for the gaussian random field (GRF). With the spherical covariance, the sill (the plateau formed by the mean cosine at distances where the CRV are uncorrelated) and the range (the distance at which the mean cosine forms the sill) are easily recognized. The realizations of the GRF were standardized according to Chapter 5, Section 5.3, step 1) (Subsections 5.4.3 and 5.4.4). The mean resultant length parameter ρ was set to $\frac{1}{2}$ of the maximum, which depends on the distribution (Table 5-1).

Generally, the resulting QQ plots were highly linear, indicating a high degree of fit. The spatial properties of the cosineogram on the bottom right are a mirror of the variogram, approximately. When sampling variation resulted in a GRF realization with less than ideal spatial properties, these properties were mirrored from the variogram plot of linear-spatial properties of the GRF to the cosineogram plot of the circular-spatial properties of the CRF. An assessment of each figure is given in the figure caption.

The examples in Appendix C were selected for close fit to the desired distribution and spatial properties. In Appendix D, the circular distribution parameters were set to

extremes, the figures were generated sequentially with random seeds, both nonstandardized and standardized results were computed, and the distributional and spatial properties were scored.

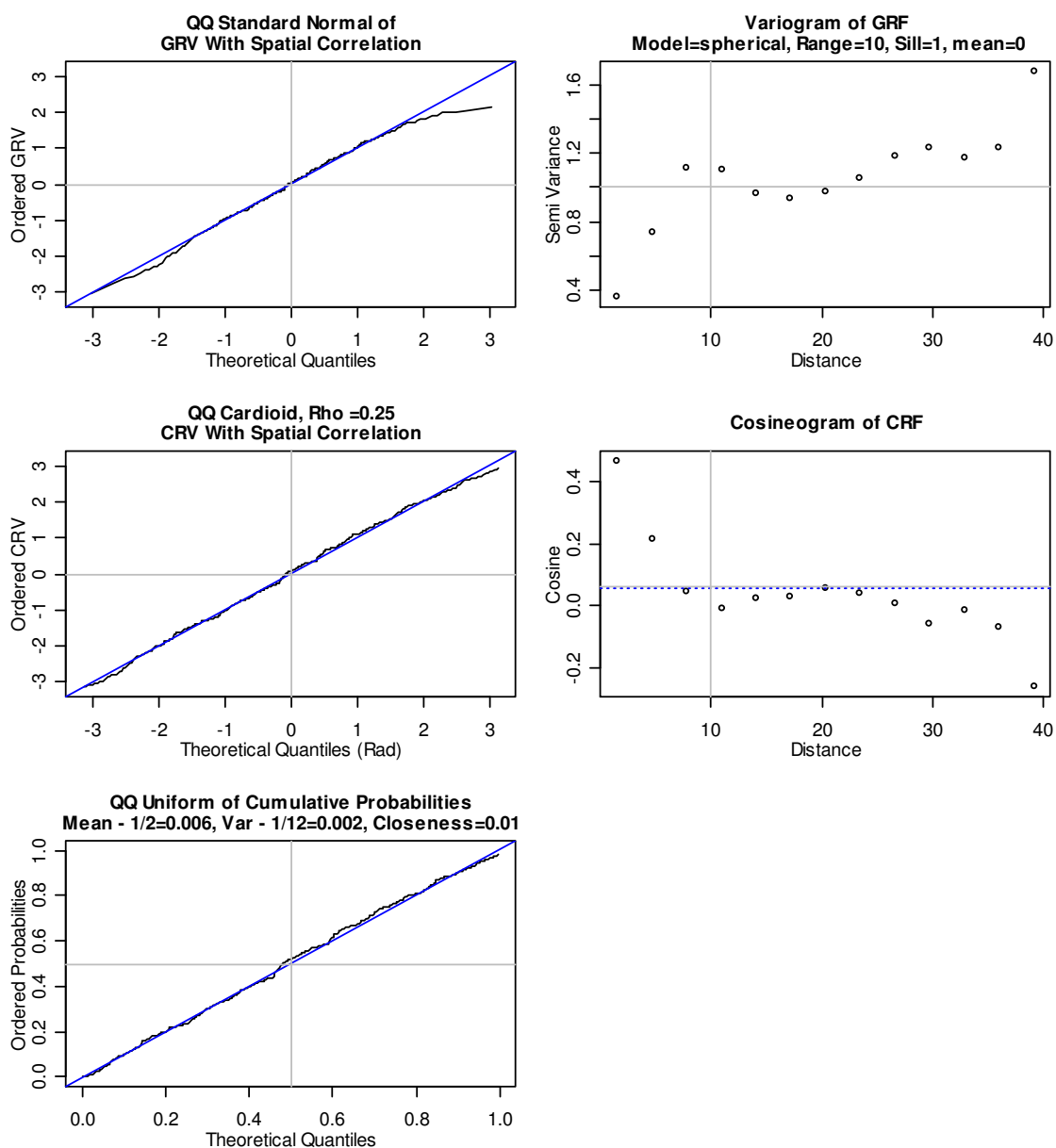


Figure C-1. Evaluation of a Cardioid CRF, $\rho = 0.25$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The spatial plots on the right show agreement with the desired spatial properties (range $r = 10$, sill $\approx \rho^2$).

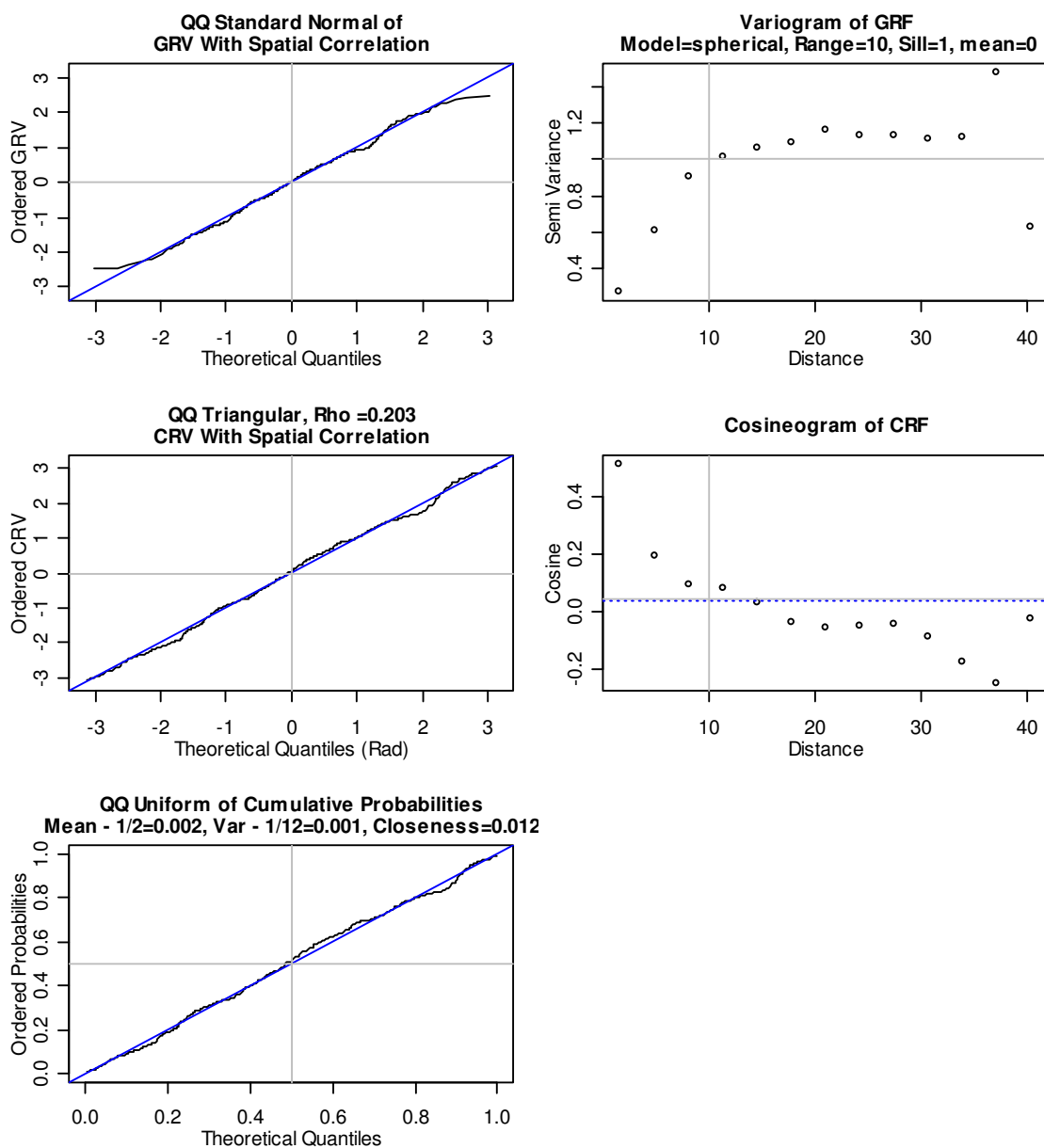


Figure C-2. Evaluation of a Triangular CRF, $\rho = 0.203$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The sill is not well defined in the spatial plots on the right. The range is between 10 and 20.

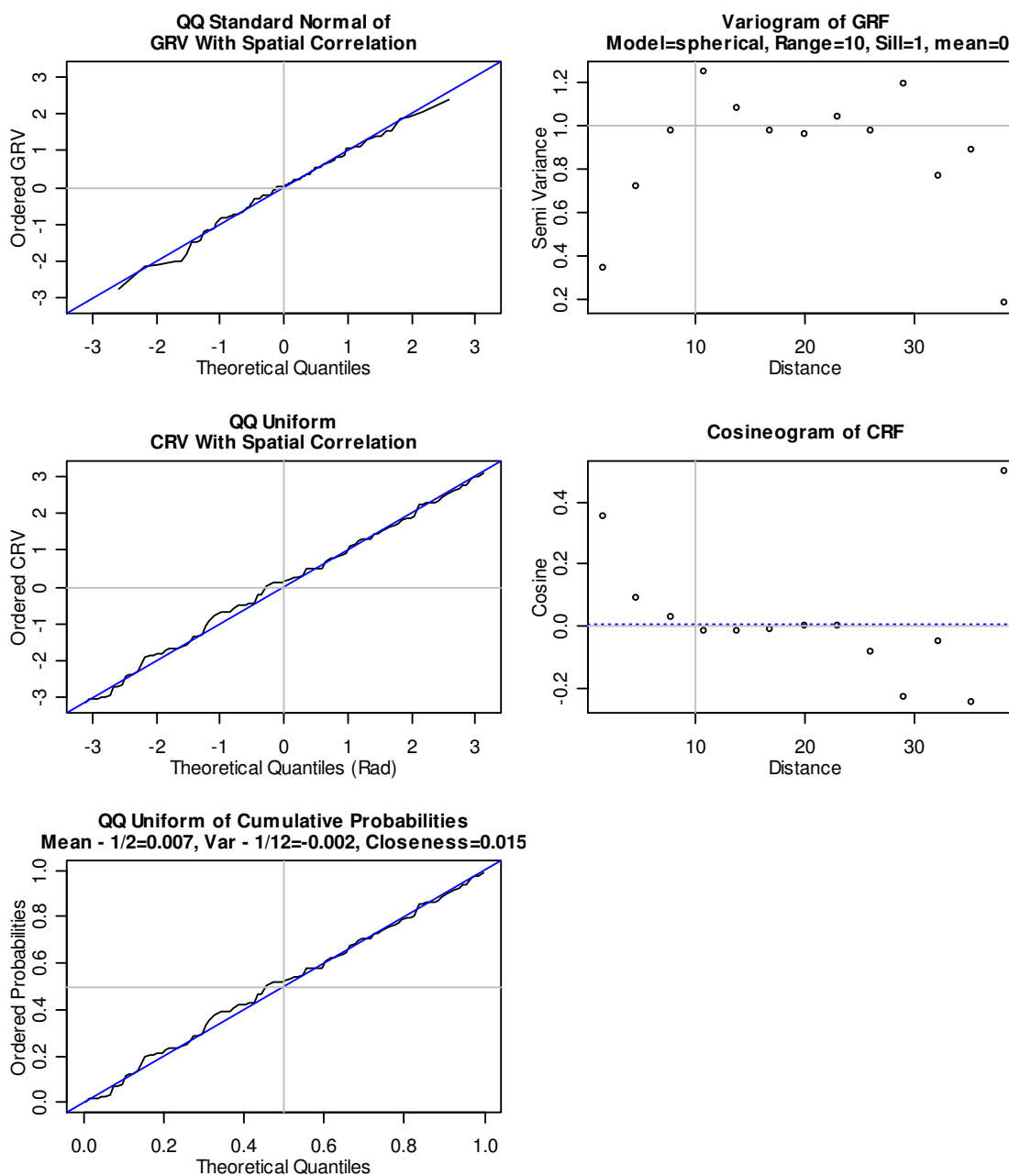


Figure C-3. Evaluation of a Uniform CRF, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The spatial plots on the right show agreement with the desired spatial properties (range $r = 10$, sill=0).

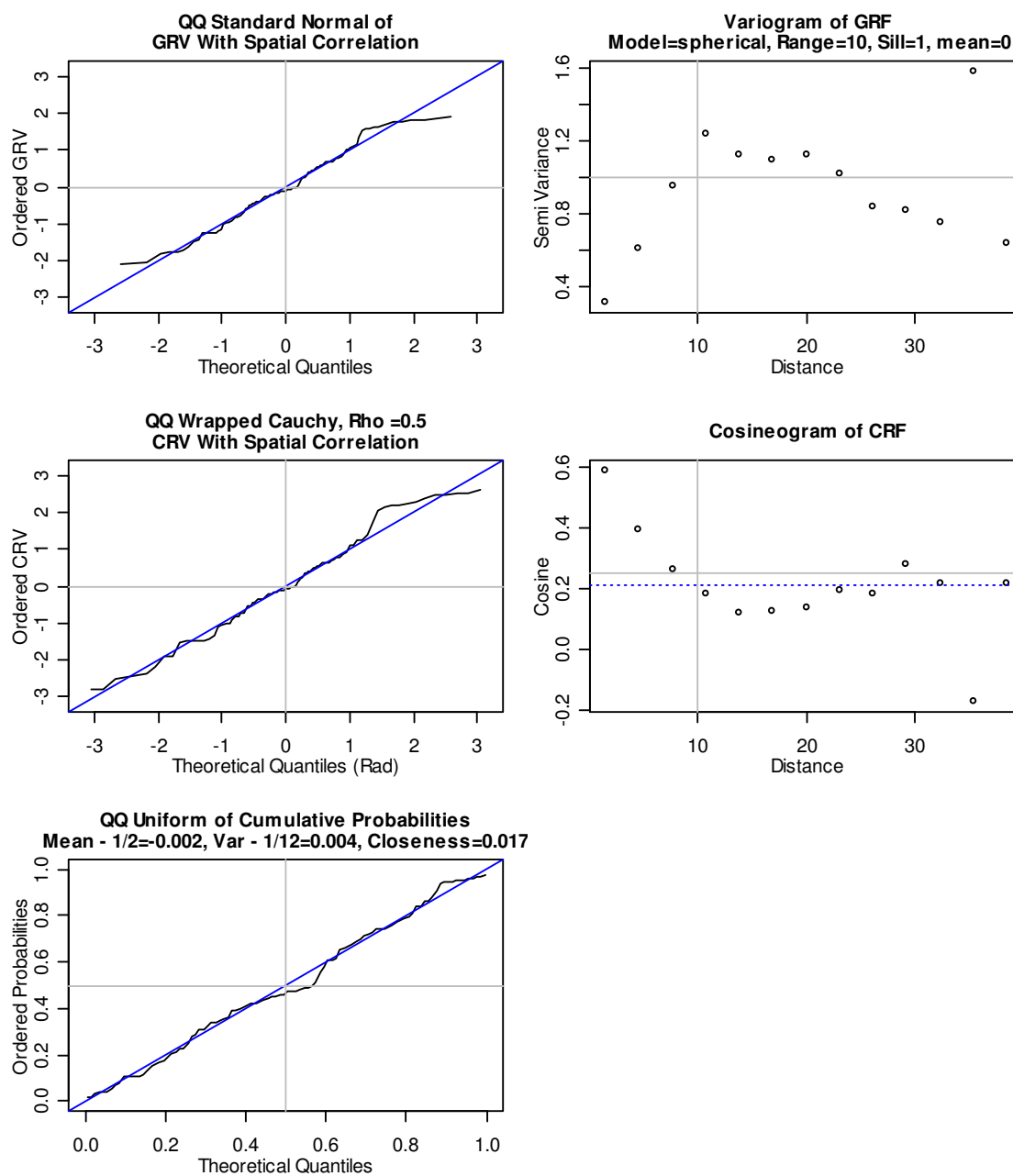


Figure C-4. Evaluation of a Wrapped Cauchy CRF, $\rho = 0.5$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The sill is not well defined in the spatial plots on the right. The sill of the variogram is high and the sill of the cosineogram is low. The range is between 10 and 15.

Appendix D

Qualitative Evaluations of CRFs Near Parameter Extremes

Appendix D extends the examples of Chapter 5, Section 5.5, Figure 5-8, and Appendix C to the selected circular distributions with mean resultant length parameter ρ at extremes of 0.05 and 95% of the maximum (Table 5-1). The purpose of these figures is to show that the method produces the desired distributional and spatial properties of the CRF. Standardized and nonstandardized realizations of the GRF with the same parameters were computed using the same random seed. Figures D-1 to D-16 show the distributional fits on the left as QQ plots and the spatial properties of the samples on the right. The interpretation is given in Chapter Subsection 5.5.3. These figures were constructed as described in Subsection 5.5.2 using the R code in Appendix L, Section L.8. To facilitate verification of the desired spatial properties, the spherical covariance model was chosen for the gaussian random field (GRF) because the spatial properties are easily recognized.

Generally, the QQ plots with standardization were highly linear, indicating a high degree of fit, but the QQ plots based on nonstandardized realizations showed typical sampling variation. The QQ wrapped Cauchy plot at high ρ showed a significant lack of distribution fit regardless of standardization. An assessment of each figure is given in the figure caption.

The cosineogram plot of circular-spatial properties mirrored the variogram plot of linear-spatial properties. When sampling variation resulted in a realization of the GRF with less than ideal spatial properties, these properties were mirrored in the cosineogram plot of the circular-spatial properties. The spatial assessments in the figure captions are summarized at the end of Appendix D. Standardization of the GRF had no apparent

effect on agreement of the spatial properties of a simulation with the desired spatial properties.

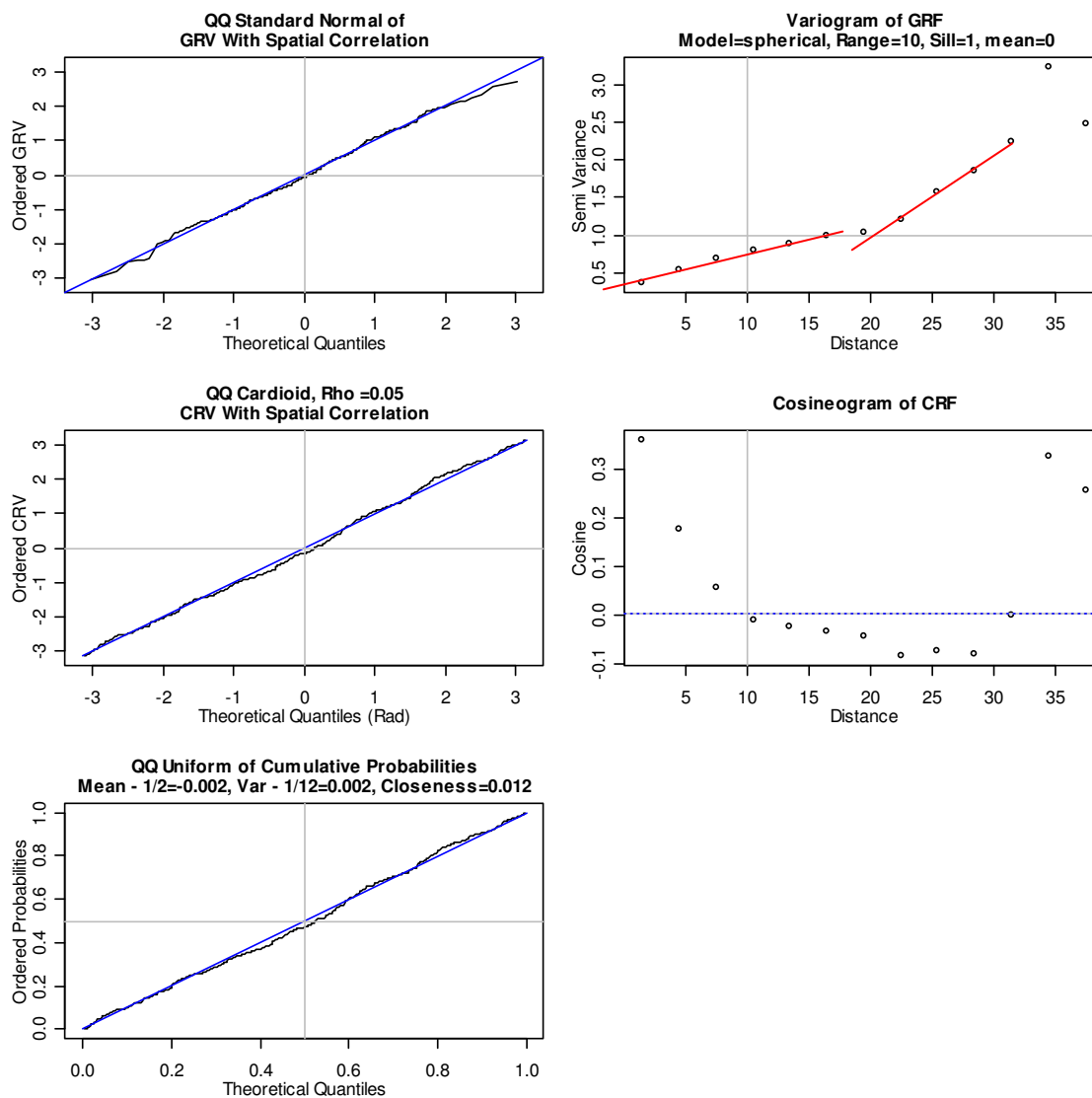


Figure D-1. Evaluation of a Cardioid CRF, $\rho = 0.05$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The sill is not well defined in the spatial plots on the right. The sill of the cosineogram is 0. Hence, the range is about 10.

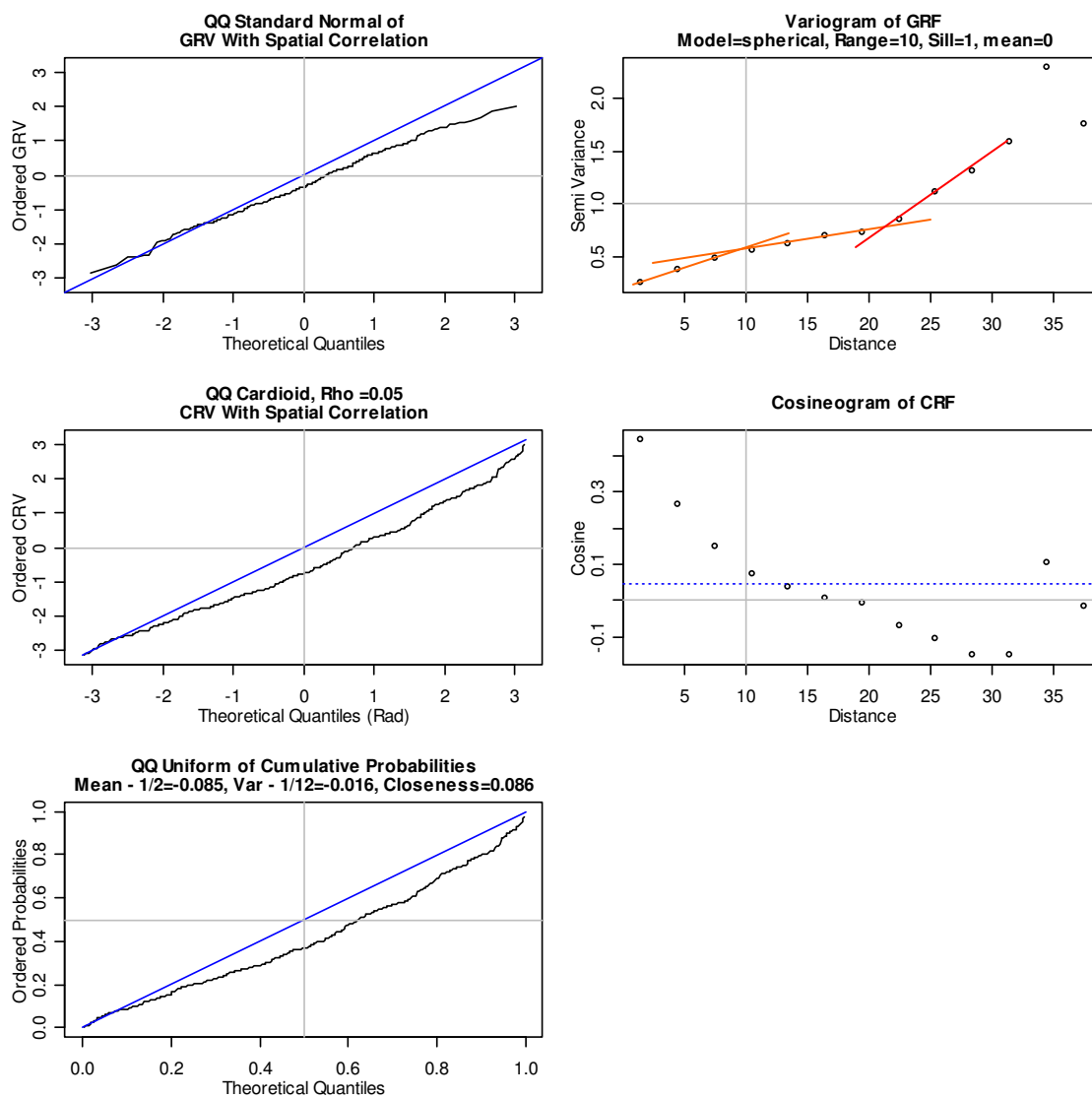


Figure D-2. Evaluation of a Cardioid CRF, $\rho = 0.05$, Range $r = 10$. The lack of linearity of the QQ plots is due to typical sampling variation. The sill is not well defined in the spatial plots on the right. The middle red line in the variogram may be the sill. Then, the range is about 10.

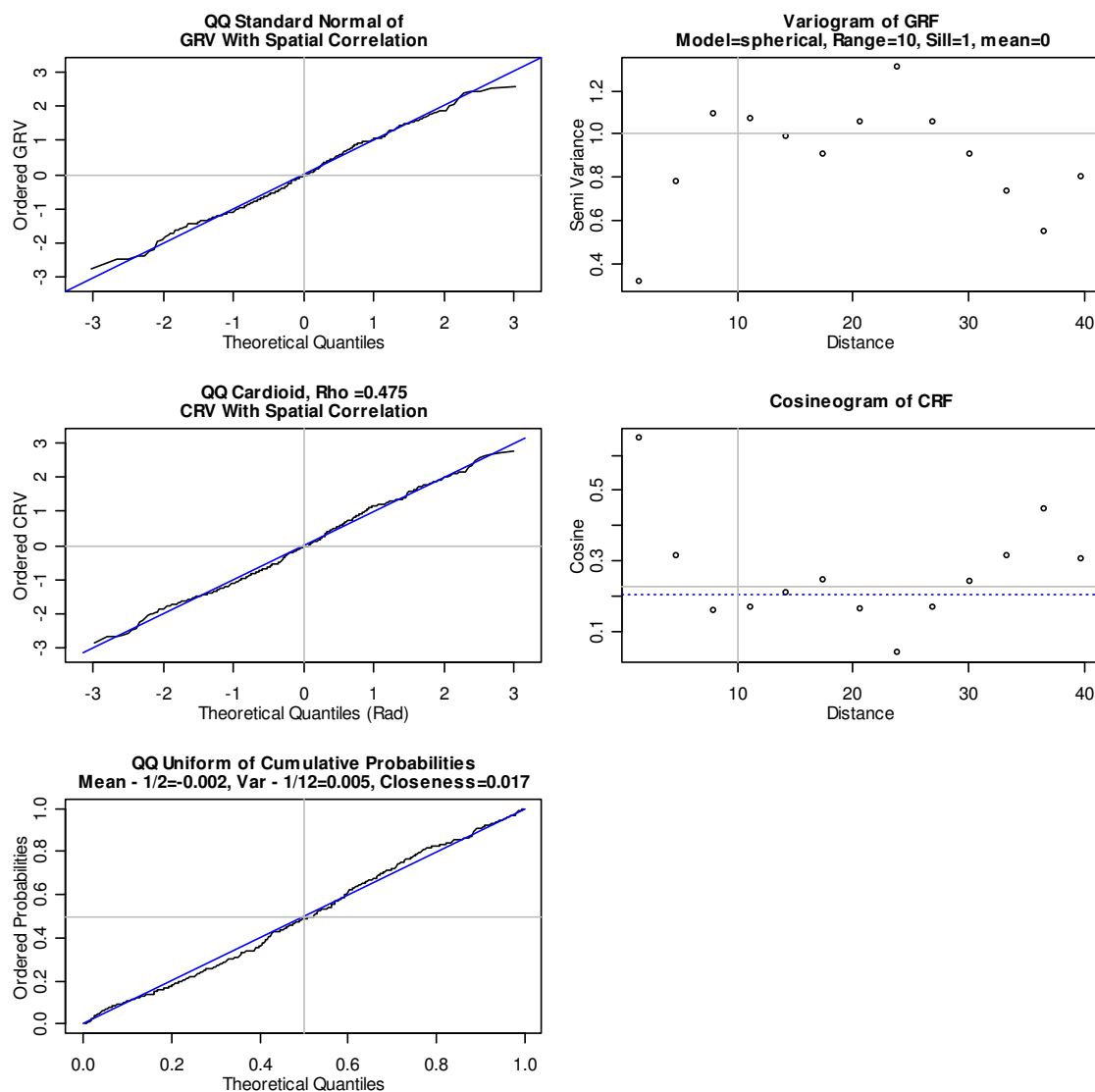


Figure D-3. Evaluation of a Cardioid CRF, $\rho = 0.475$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The spatial plots on the right indicate that the range is around 10 and the sill of the cosineogram is $\approx \rho^2$.

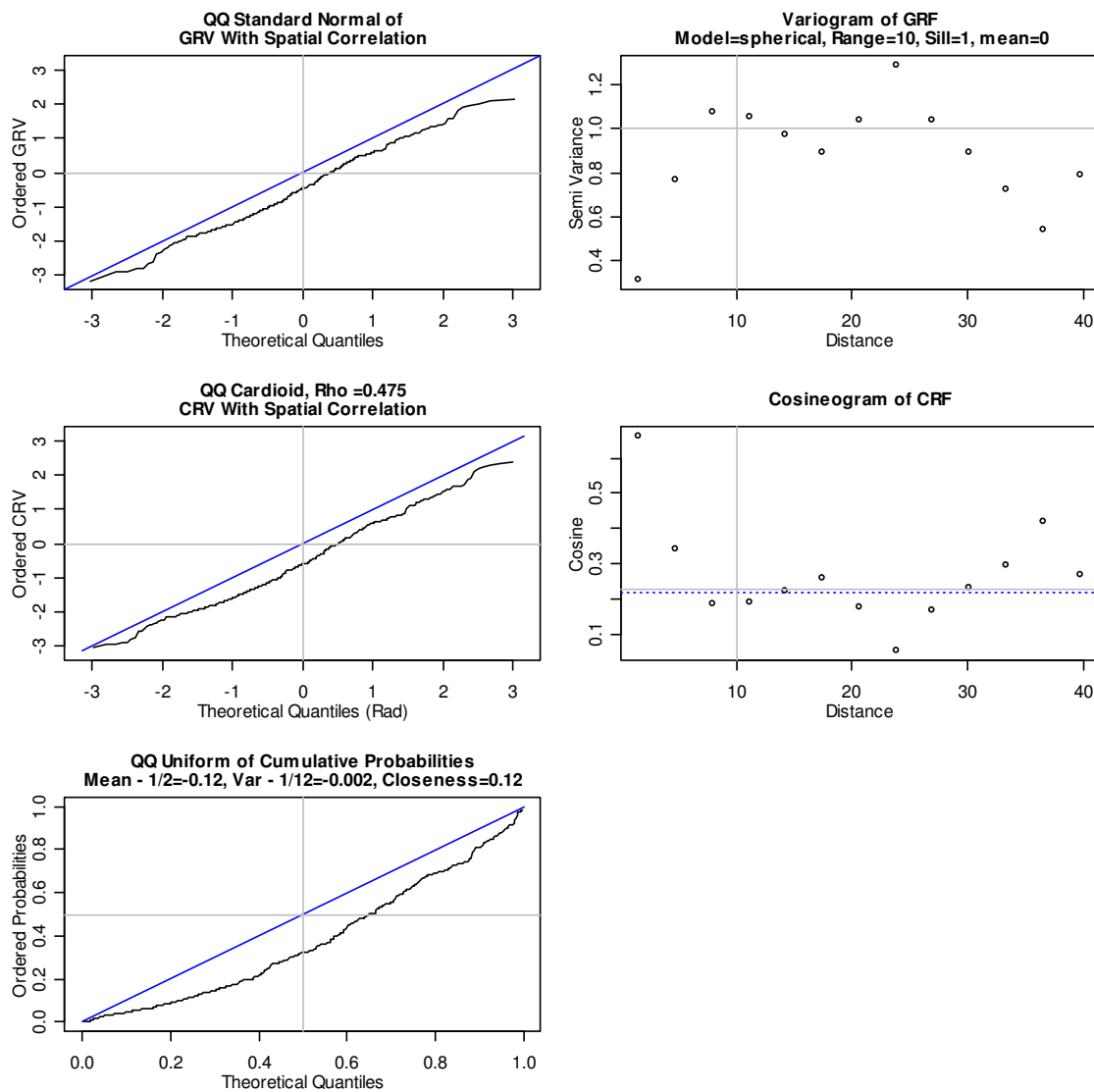


Figure D-4. Evaluation of a Cardioid CRF, $\rho = 0.475$, Range $r = 10$. The lack of linearity of the QQ plots is due to typical sampling variation. The spatial plots on the right indicate that the range is around 10 and the sill of the cosineogram is about ρ^2 .

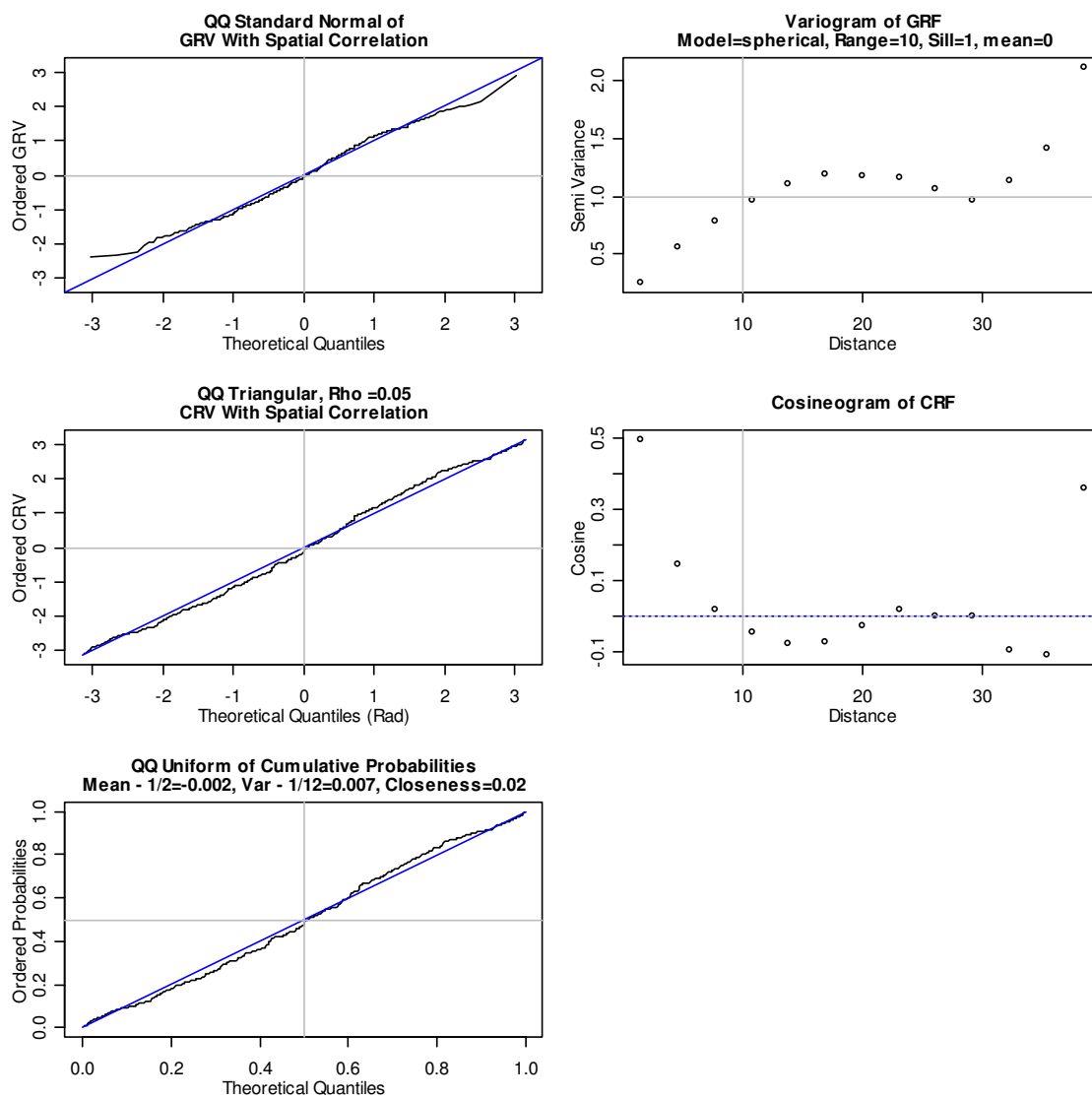


Figure D-5. Evaluation of a Triangular CRF, $\rho = 0.05$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The spatial plots on the right indicate that the range is about 10 and the sill of the cosineogram is 0.

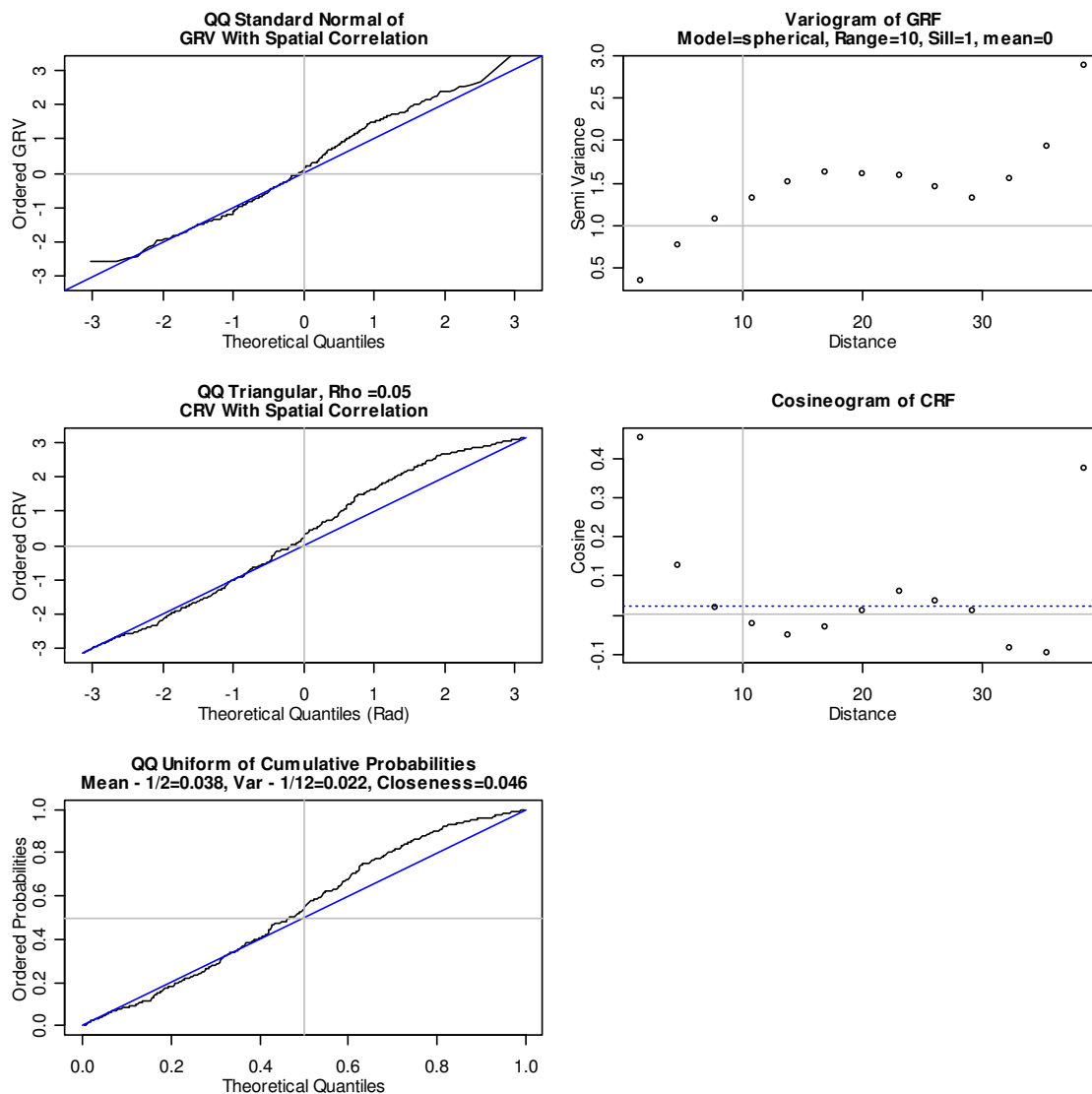


Figure D-6. Evaluation of a Triangular CRF, $\rho = 0.05$, Range $r = 10$. The lack of linearity of the QQ plots is due to typical sampling variation. The spatial plots on the right indicate that the range is around 10 and the sill is about 0.

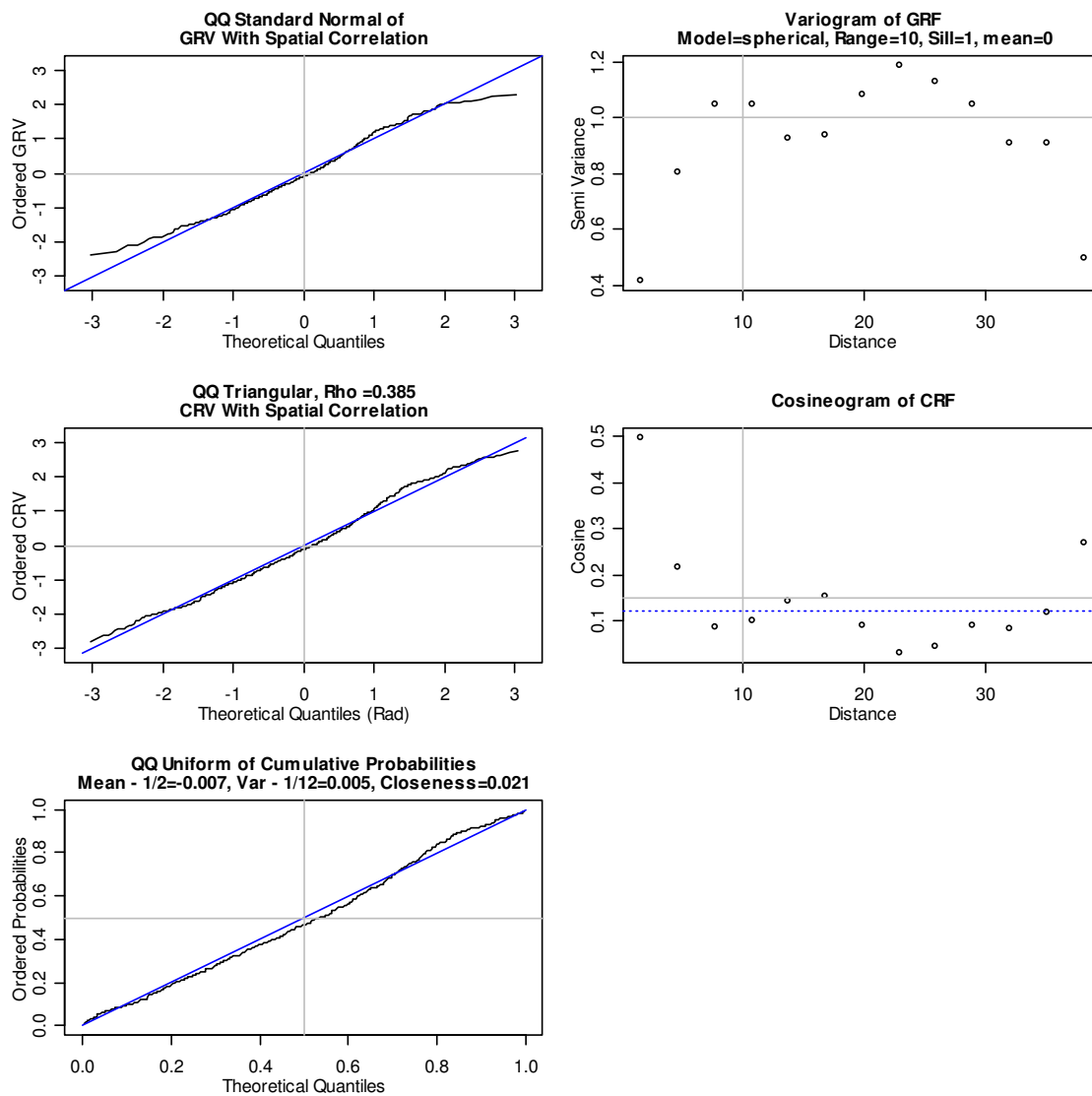


Figure D-7. Evaluation of a Triangular CRF, $\rho = 0.385$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The right plots indicate a range is about 10 and a sill of about $0.12 \approx \rho^2$.

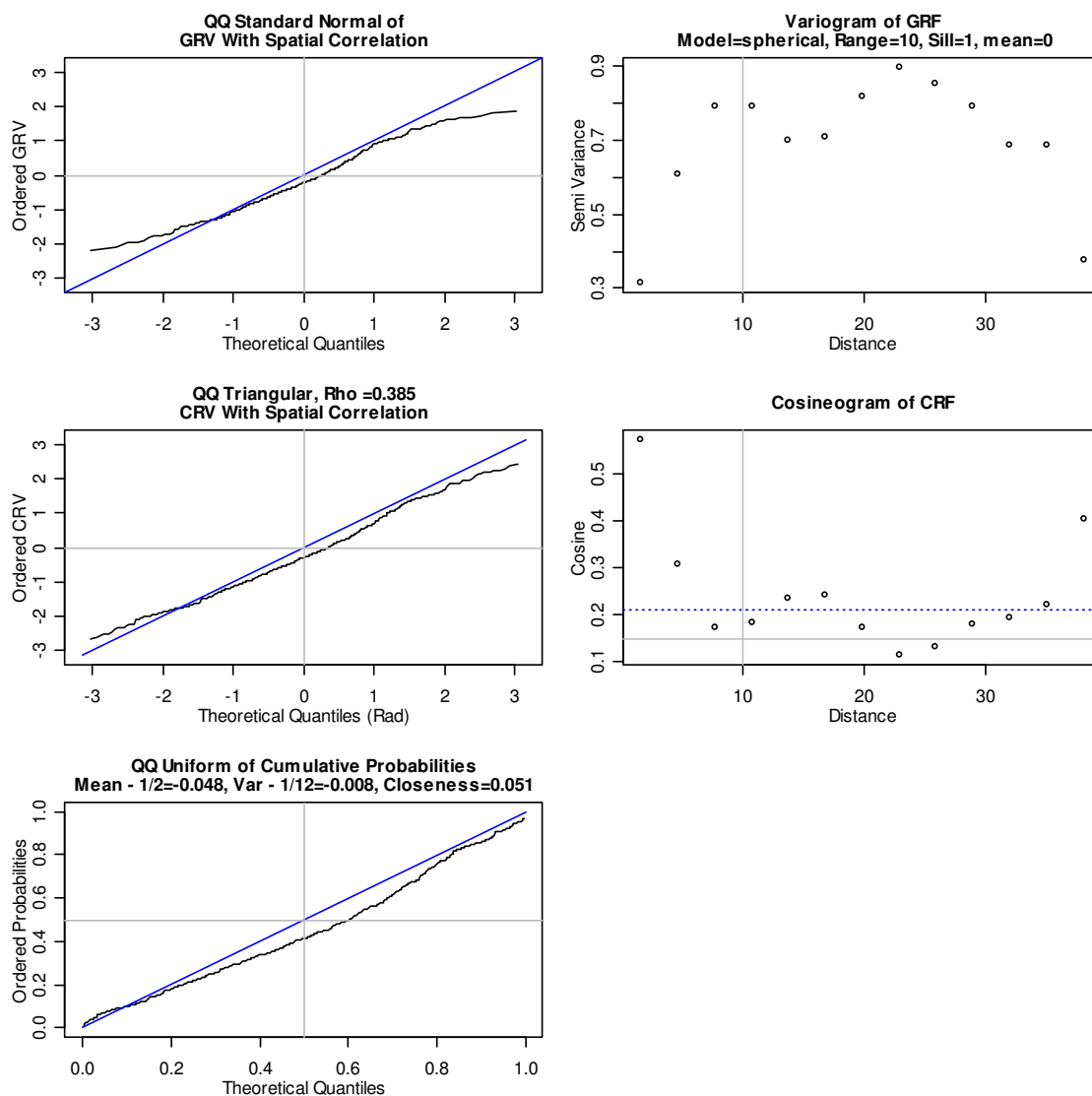


Figure D-8. Evaluation of a Triangular CRF, $\rho = 0.385$, Range $r = 10$. The lack of linearity of the QQ plots is due to typical sampling variation. The spatial plots on the right indicate that the range is about 10 and the sill is about 0.2 .

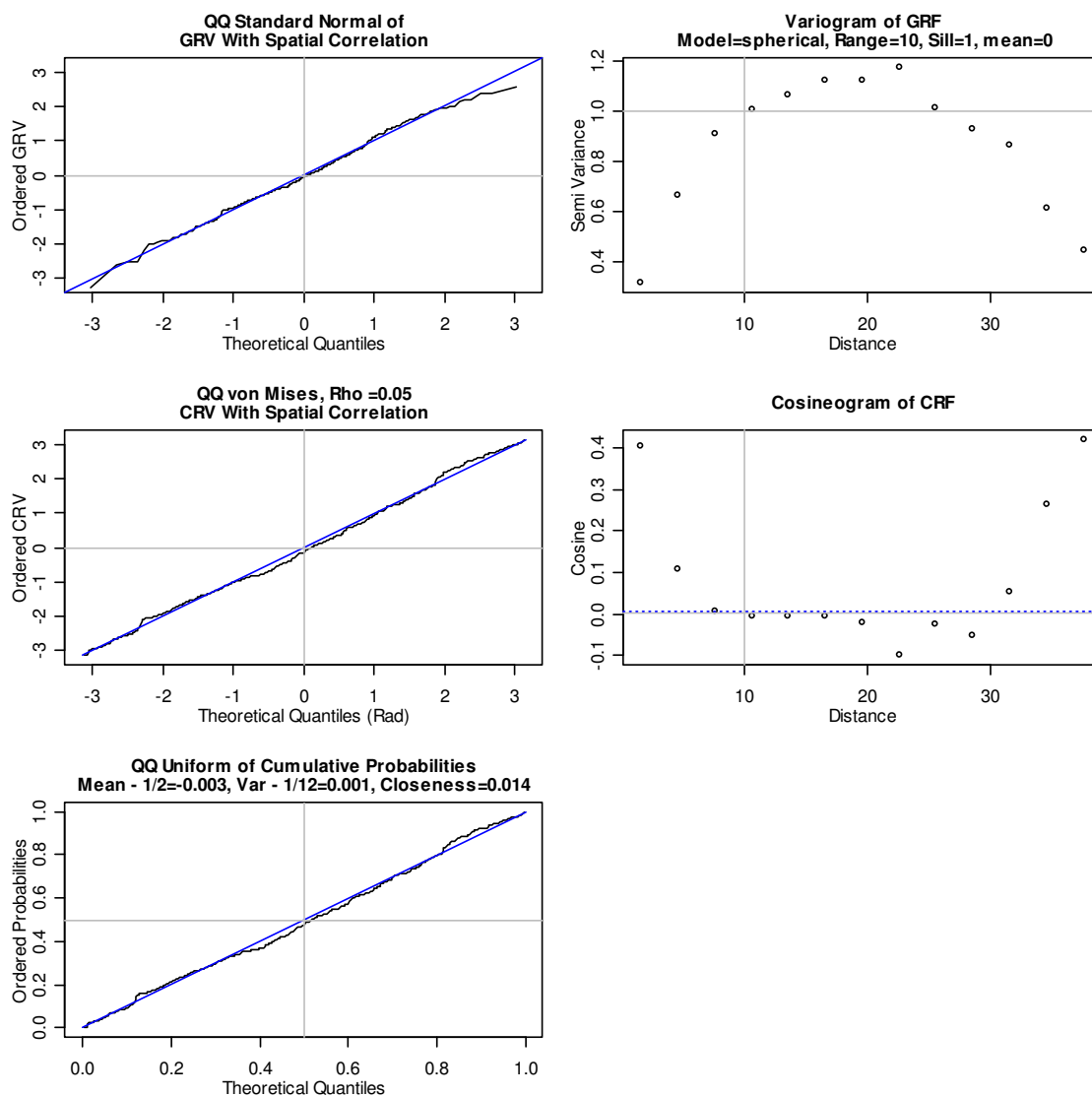


Figure D-9. Evaluation of a von Mises CRF, $\rho = 0.05$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The cosineogram plot shows agreement with the desired spatial characteristics (range $r = 10$, sill $\approx \rho^2$).

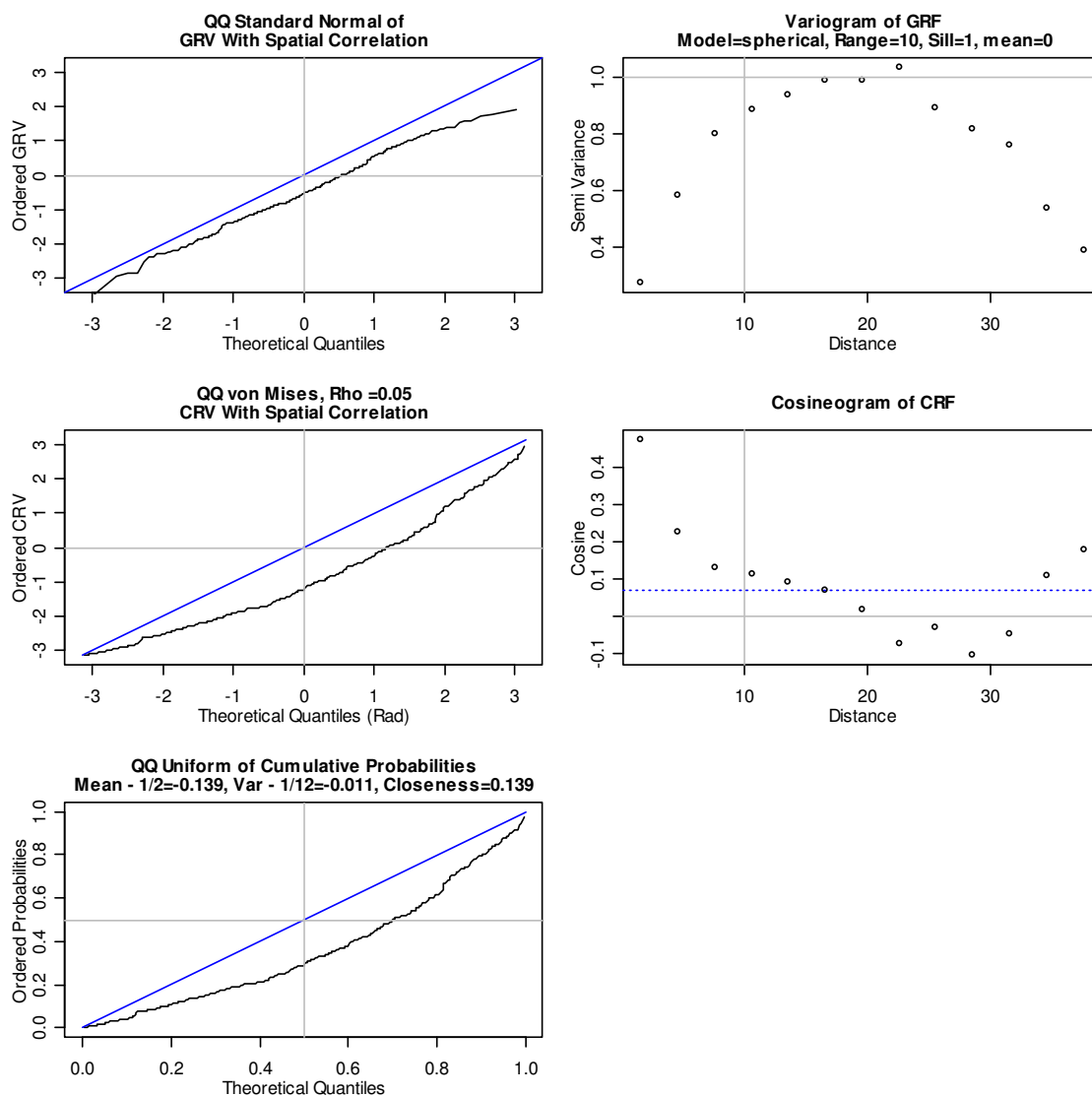


Figure D-10. Evaluation of a von Mises CRF, $\rho = 0.05$, Range $r = 10$. The lack of linearity of the QQ plots is due to typical sampling variation. The sill is not well defined in the right plots. The variogram suggest that the range is around 15.

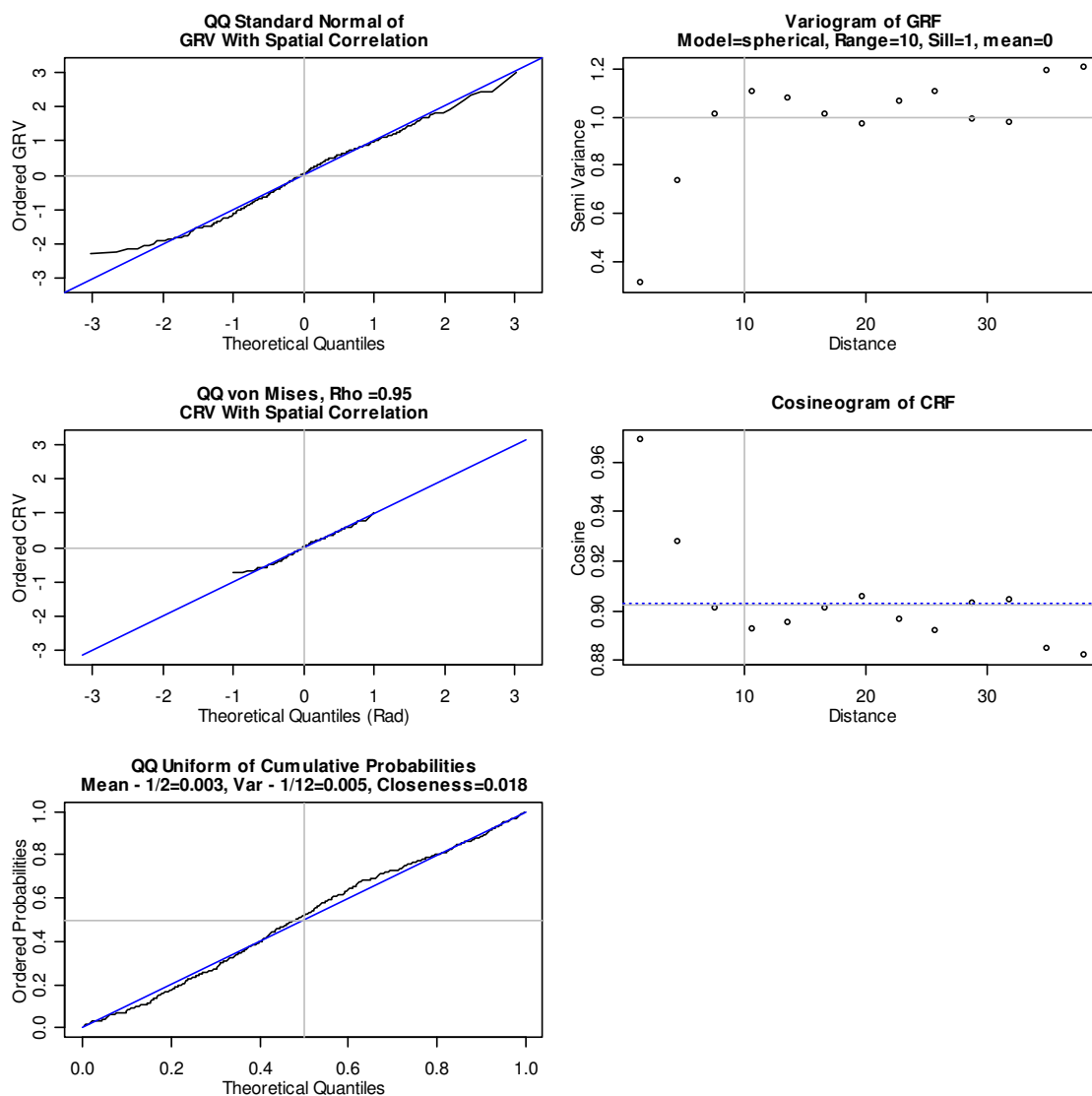


Figure D-11. Evaluation of a von Mises CRF, $\rho = 0.95$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The spatial plots on the right show agreement with the desired spatial characteristics (range $r = 10$, sill $\approx \rho^2$).

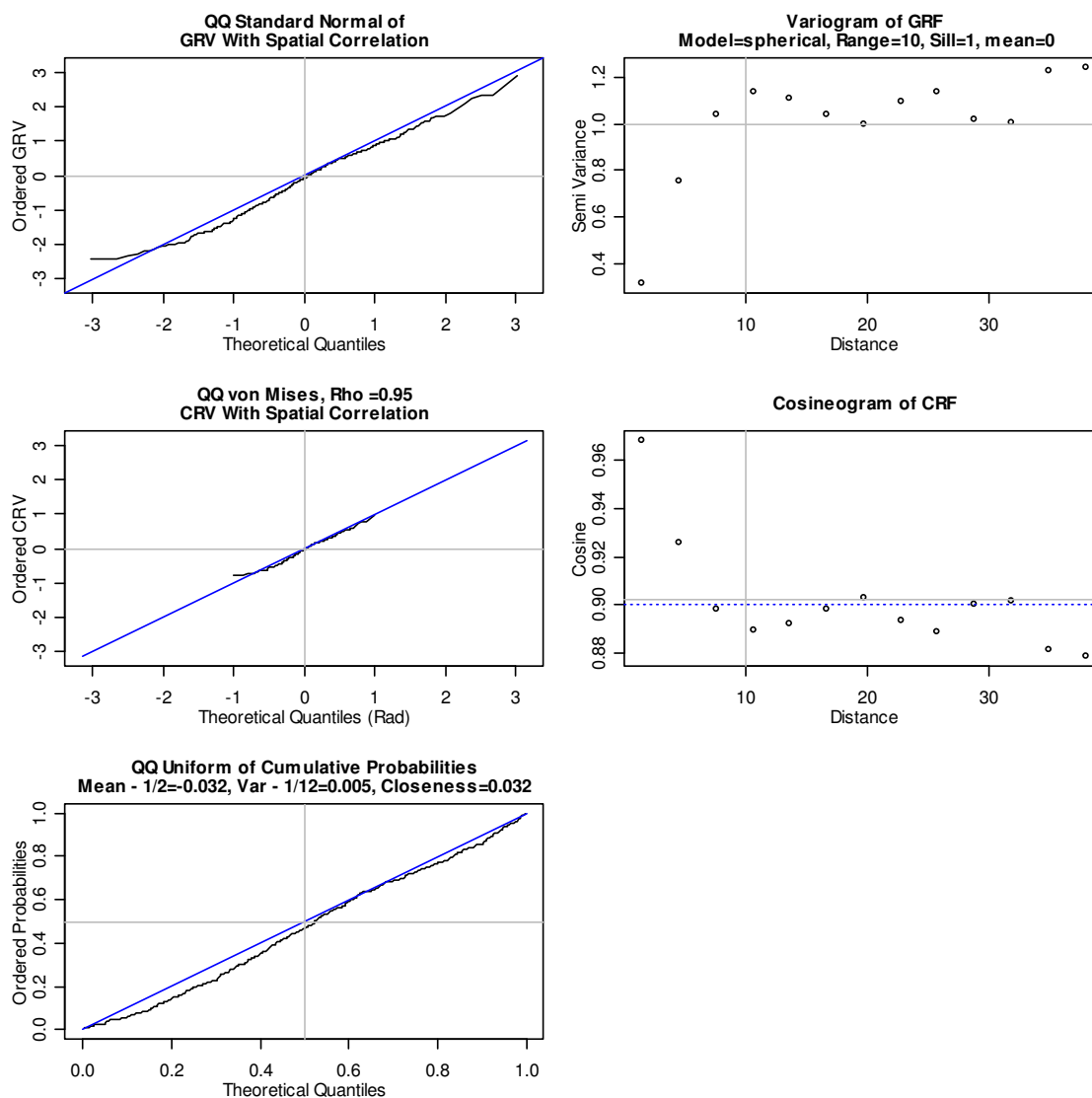


Figure D-12. Evaluation of a von Mises CRF, $\rho = 0.95$, Range $r = 10$. The lack of linearity of the QQ plots is due to typical sampling variation. The spatial plots on the right show agreement with the desired spatial characteristics (range $r = 10$, sill = $0.90 \approx \rho^2$).

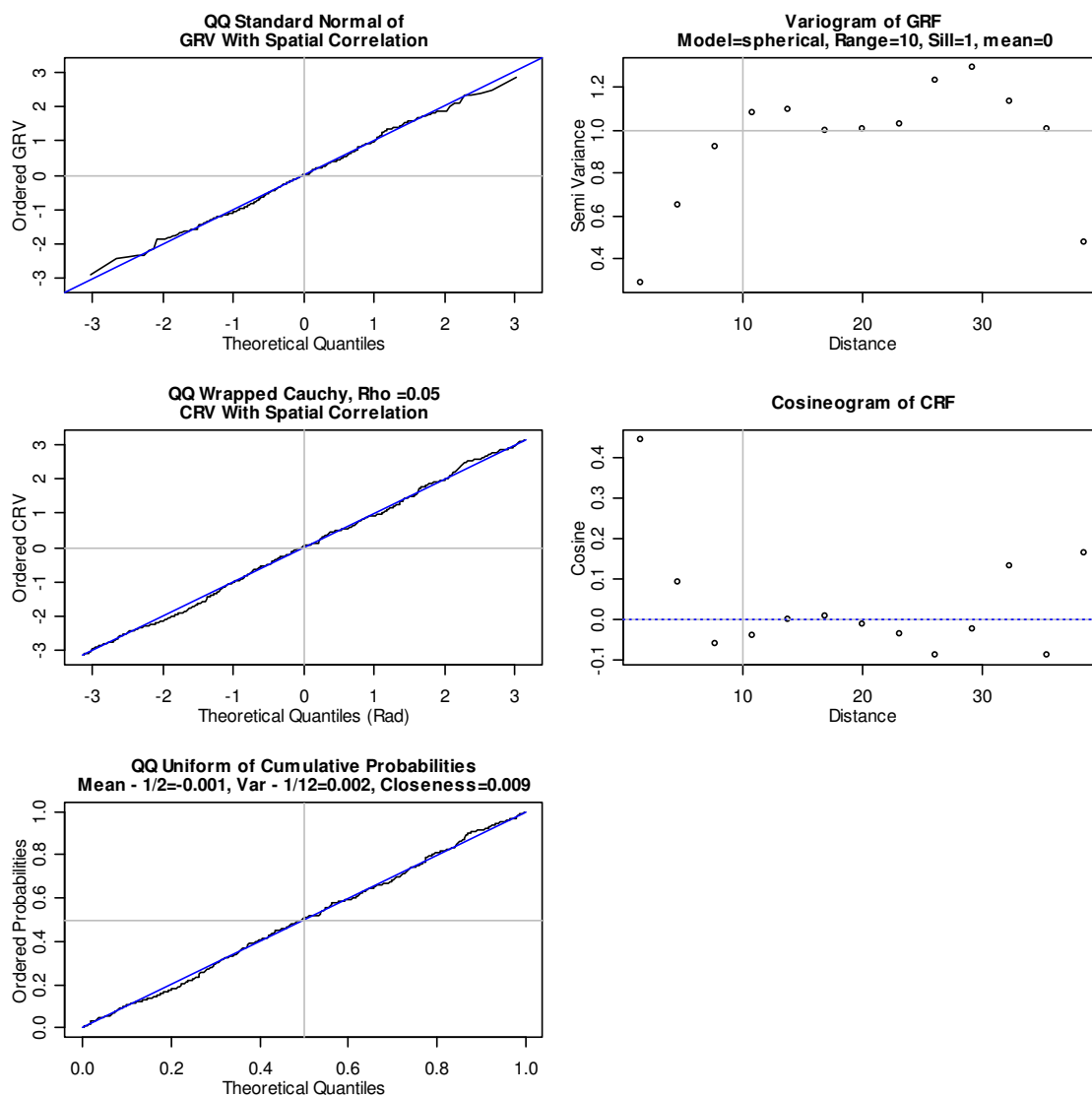


Figure D-13. Evaluation of a Wrapped Cauchy CRF, $\rho = 0.05$, Overfit, Range $r = 10$. The linearity of the QQ plots indicates that the distributional fits are close. The spatial plots on the right show agreement with the desired spatial characteristics (range $r = 10$, sill=0).

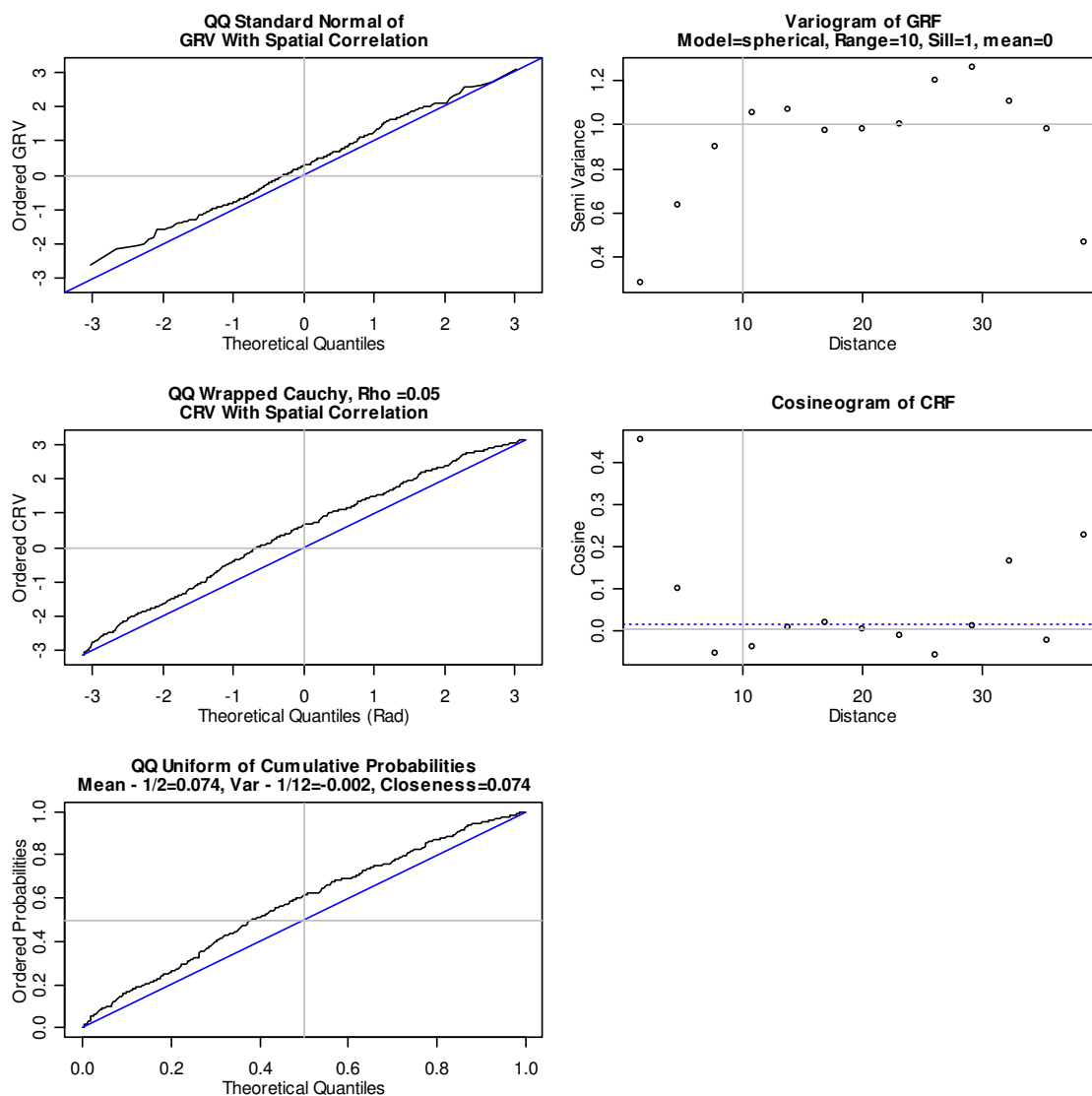


Figure D-14. Evaluation of a Wrapped Cauchy CRF, $\rho = 0.05$, Range $r = 10$. The lack of linearity of the QQ plots is due to typical sampling variation. The spatial plots on the right show agreement with the desired spatial characteristics (range $r = 10$, sill=0).

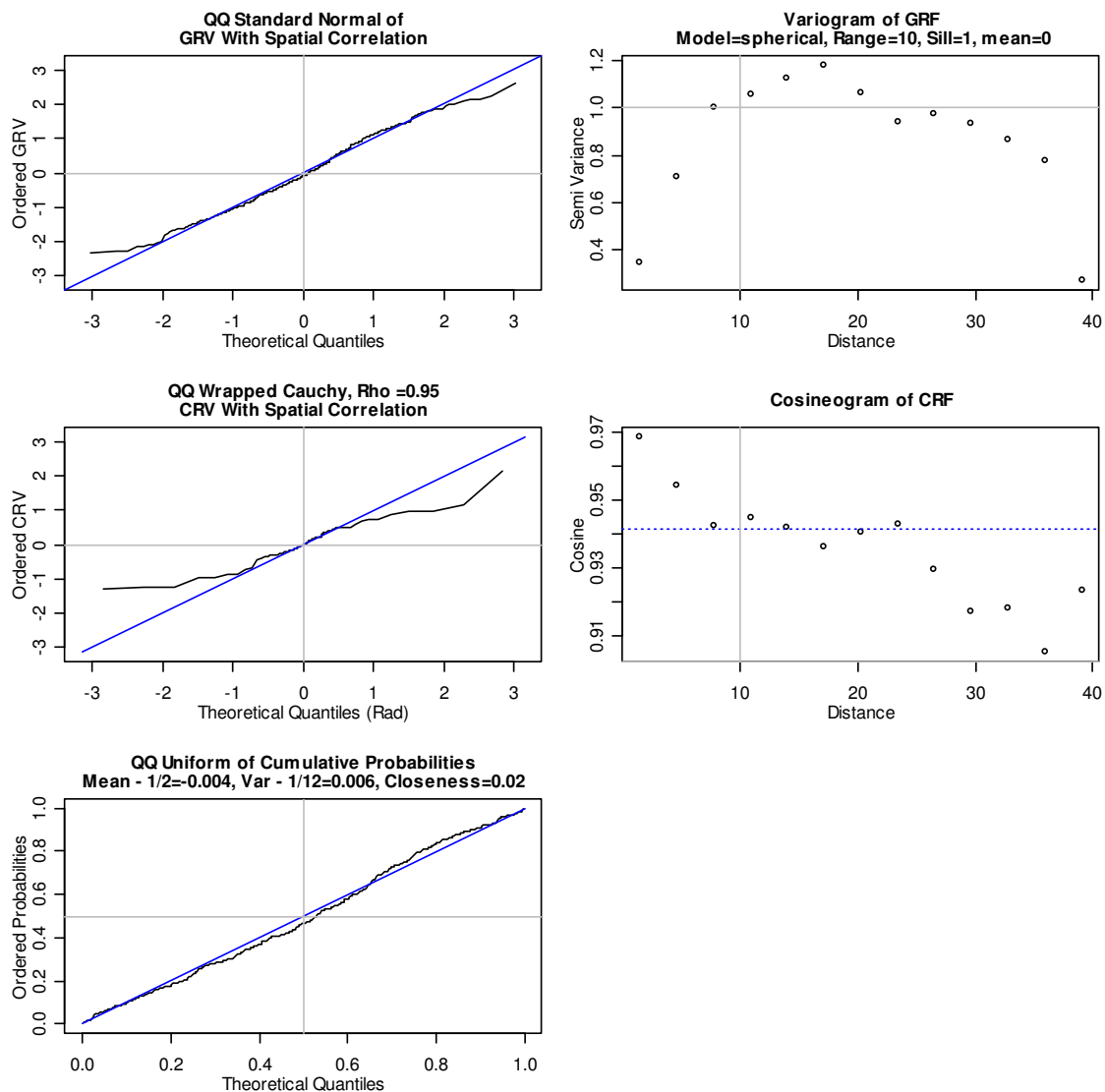


Figure D-15. Evaluation of a Wrapped Cauchy CRF, $\rho = 0.95$, Overfit, Range $r = 10$. The QQ plots show that the GRV and uniform samples (cumulative probabilities) were a close fit to the corresponding distributions, but the wrapped Cauchy circular sample at high ρ had less fit than the other circular samples. Since the linear Cauchy distribution gives samples with poor fit, it is not surprising that sampling from the linear Cauchy wrapped on a circle, or the wrapped Cauchy distribution, also gives samples with poor fit. The cosineogram on the right indicates a range of about 10 and a sill of about 0.94, which is slightly higher than ρ^2 .

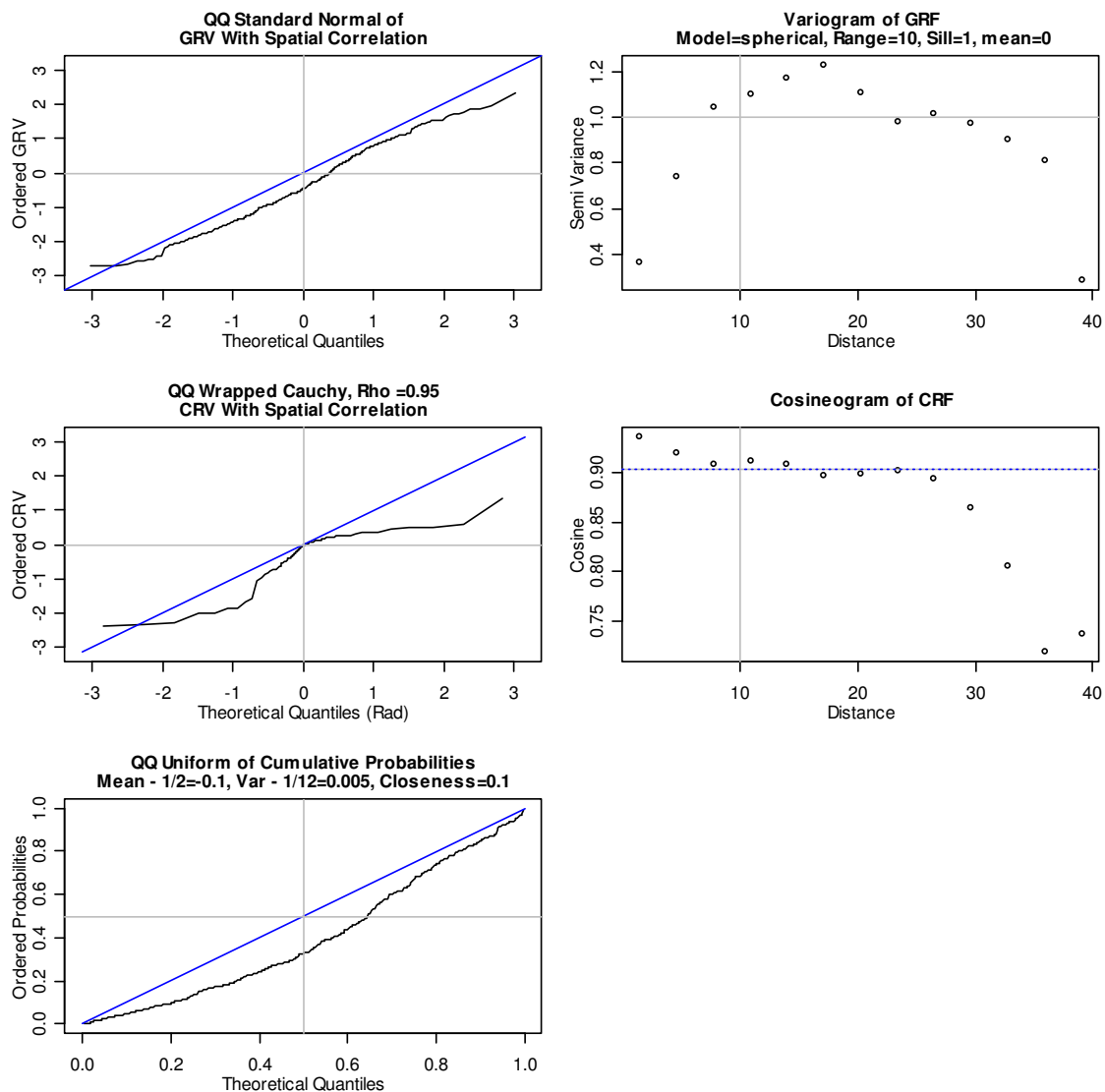


Figure D-16. Evaluation of a Wrapped Cauchy CRF, $\rho = 0.95$, Range $r = 10$. The lack of linearity of the QQ plots is due to typical sampling variation. The wrapped Cauchy circular sample at high ρ had poor fit similar to Figure D-15. The cosineogram on the right shows agreement with the desired spatial characteristics (range $r = 10$, sill = $0.9 = \rho^2$).

The spatial properties of Figures D-1 to D-16 were scored and tabulated in

Table D-1. The figures were scored based on the assessments in the figure captions:

- Good = Figure caption contains “agreement with the desired spatial characteristics”
- OK = Range and sill quantified
- Poor = Figure caption contains “sill is not well defined”.

Table D-1. Spatial Property Scores of Figures D-1 to D-16

Overfit	Good	OK	Poor
Yes	3	4	1
No	3	3	2

Appendix E

Derivations of the CDF Formulae for Support $[0, 2\pi)$

In Chapter 5, a Gaussian random field (GRF) is mapped to a circular random field (CRF) via the CDFs. The CDFs are derived for four circular distributions with $G_{\Theta}(\theta)$ the cumulative probability distribution function of Θ , ρ the mean resultant length parameter and κ the concentration parameter for the von Mises distribution, and $0 \leq \theta_1 < \theta_2 < 2\pi$. To simplify calculation, the mean direction is assumed to be 0. With exception to the circular uniform distribution, the mean direction of the CRF may be set to an arbitrary direction by adding the direction minus the sample mean direction to the direction of each of the observations. Probability density functions were obtained from Mardia (1972), Fisher (1993), and Jammalamadaka and Sengupta (2001). The derived cumulative distribution function (CDF) formulae of Appendix E will be verified by integration over the support $[0, 2\pi)$ in Appendix F, and mapped to support $[-\pi, +\pi)$ in Appendices G and H. . The CDFs will be denoted $G_U(\theta)$ for the uniform, $G_T(\theta; \rho)$ for the triangular, $G_C(\theta; \rho)$ for the cardioid, $G_{VM}(\theta; \rho)$ for the von Mises, and $G_{WC}(\theta; \rho)$ for the wrapped Cauchy distributions.

E.1 Cardioid

The probability density function (PDF) is given in Mardia (1972, p. 51, eq. 3.4.11) and the CDF is given in Fisher (1993, p. 45, eq. 3.22). With $0 < \rho \leq \frac{1}{2}$,

$$\begin{aligned}
 G_C(\theta_2; \rho) - G_C(\theta_1; \rho) &= \int_{\theta_1}^{\theta_2} \frac{1}{2\pi} (1 + 2\rho \cos(\theta)) d\theta \\
 &= \frac{1}{2\pi} \int_{\theta_1}^{\theta_2} d\theta + \frac{1}{2\pi} \int_{\theta_1}^{\theta_2} 2\rho \cos(\theta) d\theta \\
 &= \frac{\theta_2 - \theta_1}{2\pi} + \frac{2\rho}{2\pi} \sin(\theta) \Big|_{\theta_1}^{\theta_2} \\
 &= \frac{\theta_2 - \theta_1 + 2\rho (\sin(\theta_2) - \sin(\theta_1))}{2\pi} \Rightarrow \\
 G_C(\theta_2; \rho) - G_C(\theta_1; \rho) &= \frac{\theta_2 - \theta_1 + 2\rho (\sin(\theta_2) - \sin(\theta_1))}{2\pi}. \tag{E.1}
 \end{aligned}$$

E.2 Triangular

The PDF was obtained from Mardia (1972, p. 51, eq. (3.4.13)). The CDF and derivation were not found in the cited texts, and may be a new result. With $0 < \rho \leq \frac{4}{\pi^2}$,

$$\begin{aligned}
 G_T(\theta_2; \rho) - G_T(\theta_1; \rho) &= \int_{\theta_1}^{\theta_2} \frac{1}{8\pi} (4 - \pi^2 \rho + 2\pi\rho |\pi - \theta|) d\theta \\
 &= \frac{1}{8\pi} \int_{\theta_1}^{\theta_2} (4 - \pi^2 \rho) d\theta + \frac{1}{8\pi} \int_{\theta_1}^{\theta_2} 2\pi\rho |\pi - \theta| d\theta \\
 &= \frac{(4 - \pi^2 \rho)}{8\pi} (\theta_2 - \theta_1) + \frac{\rho}{4} \int_{\theta_1}^{\theta_2} |\pi - \theta| d\theta \Rightarrow
 \end{aligned}$$

Case 1, $0 \leq \theta_1 < \theta_2 \leq \pi$

$$\begin{aligned}
&= \frac{(4 - \pi^2 \rho)}{8\pi} (\theta_2 - \theta_1) + \frac{\rho}{4} \int_{\theta_1}^{\theta_2} (\pi - \theta) d\theta \\
&= \frac{(4 - \pi^2 \rho)}{8\pi} (\theta_2 - \theta_1) + \frac{\rho}{4} \left(\pi\theta - \frac{1}{2} \theta^2 \right) \Big|_{\theta_1}^{\theta_2} \\
&= \frac{(4 - \pi^2 \rho)}{8\pi} (\theta_2 - \theta_1) + \frac{\pi\rho}{4} \theta \Big|_{\theta_1}^{\theta_2} - \frac{\rho}{8} \theta^2 \Big|_{\theta_1}^{\theta_2} \\
&= \frac{(4 - \pi^2 \rho)}{8\pi} (\theta_2 - \theta_1) + \frac{\pi\rho}{4} (\theta_2 - \theta_1) - \frac{\rho}{8} (\theta_2^2 - \theta_1^2) \\
&= \left(\frac{(4 - \pi^2 \rho)}{8\pi} + \frac{2\pi^2 \rho}{8\pi} - \frac{\pi\rho}{8\pi} (\theta_2 + \theta_1) \right) (\theta_2 - \theta_1) \\
&= \left(\frac{4 + \pi^2 \rho - \pi\rho (\theta_2 + \theta_1)}{8\pi} \right) (\theta_2 - \theta_1) \Rightarrow
\end{aligned}$$

$$G_T(\theta_2; \rho) - G_T(\theta_1; \rho) = \left(\frac{4 + \pi^2 \rho - \pi\rho (\theta_2 + \theta_1)}{8\pi} \right) (\theta_2 - \theta_1) \quad (\text{E.2})$$

Case 2, $\pi \leq \theta_1 < \theta_2 \leq 2\pi$

$$\begin{aligned}
&= \frac{(4 - \pi^2 \rho)}{8\pi} (\theta_2 - \theta_1) + \frac{\rho}{4} \int_{\theta_1}^{\theta_2} |\pi - \theta| d\theta \\
&= \frac{4 - \pi^2 \rho}{8\pi} (\theta_2 - \theta_1) + \frac{\rho}{4} \int_{\theta_1}^{\theta_2} (\theta - \pi) d\theta \\
&= \frac{4 - \pi^2 \rho}{8\pi} (\theta_2 - \theta_1) + \frac{\rho}{4} \frac{1}{2} \theta^2 \Big|_{\theta_1}^{\theta_2} - \frac{\pi\rho}{4} \theta \Big|_{\theta_1}^{\theta_2} \\
&= \frac{4 - \pi^2 \rho}{8\pi} (\theta_2 - \theta_1) + \frac{\rho}{8} (\theta_2^2 - \theta_1^2) - \frac{\pi\rho}{4} (\theta_2 - \theta_1) \\
&= \left\{ \frac{4 - \pi^2 \rho}{8\pi} - \frac{2\pi^2 \rho}{8\pi} \right\} (\theta_2 - \theta_1) + \frac{\rho}{8} (\theta_2 + \theta_1) (\theta_2 - \theta_1) \\
&= \frac{4 - 3\pi^2 \rho + \pi\rho (\theta_2 + \theta_1)}{8\pi} (\theta_2 - \theta_1) \Rightarrow
\end{aligned}$$

$$G_T(\theta_2; \rho) - G_T(\theta_1; \rho) = \frac{4 - 3\pi^2 \rho + \pi\rho (\theta_2 + \theta_1)}{8\pi} (\theta_2 - \theta_1) \quad (\text{E.3})$$

Case 3, $0 \leq \theta_1 < \pi \leq \theta_2 \leq 2\pi$

$$\begin{aligned}
&= \left(\frac{4 - \pi^2 \rho}{8\pi} \right) (\theta_2 - \theta_1) + \frac{\rho}{4} \int_{\theta_1}^{\pi} (\pi - \theta) d\theta + \frac{\rho}{4} \int_{\pi}^{\theta_2} (\theta - \pi) d\theta \\
&= \left(\frac{4 - \pi^2 \rho}{8\pi} \right) (\theta_2 - \theta_1) + \frac{\rho}{4} \left(\pi\theta - \frac{1}{2}\theta^2 \right) \Big|_{\theta=\theta_1}^{\theta=\pi} + \frac{\rho}{4} \left(\frac{1}{2}\theta^2 - \pi\theta \right) \Big|_{\theta=\pi}^{\theta=\theta_2} \\
&= \left(\frac{4 - \pi^2 \rho}{8\pi} \right) (\theta_2 - \theta_1) + \frac{\rho}{4} \left(\pi\pi - \frac{1}{2}\pi^2 \right) - \frac{\rho}{4} \left(\pi\theta_1 - \frac{1}{2}\theta_1^2 \right) + \frac{\rho}{4} \left(\frac{1}{2}\theta_2^2 - \pi\theta_2 \right) - \frac{\rho}{4} \left(\frac{1}{2}\pi^2 - \pi\pi \right) \\
&= \left(\frac{4 - \pi^2 \rho}{8\pi} \right) (\theta_2 - \theta_1) + \frac{\rho}{4} \frac{1}{2}\pi^2 - \frac{\rho}{4} \left(\pi - \frac{1}{2}\theta_1 \right) \theta_1 + \frac{\rho}{4} \left(\frac{1}{2}\theta_2 - \pi \right) \theta_2 + \frac{\rho}{4} \frac{1}{2}\pi^2 \\
&= \left(\frac{4 - \pi^2 \rho}{8\pi} \right) (\theta_2 - \theta_1) - \frac{\pi\rho\theta_1}{4} + \frac{\rho}{4} \frac{1}{2}\theta_1^2 + \frac{\rho}{4} \frac{1}{2}\theta_2^2 - \frac{\pi\rho}{4} \theta_2 + \pi^2 \frac{\rho}{4} \\
&= \left(\frac{4 - \pi^2 \rho}{8\pi} \right) (\theta_2 - \theta_1) - \frac{2\pi^2 \rho (\theta_1 + \theta_2)}{8\pi} + \frac{\pi\rho}{8\pi} (\theta_1^2 + \theta_2^2) + \pi^2 \frac{\rho}{4} \\
&= \frac{(4 - \pi^2 \rho)(\theta_2 - \theta_1) - 2\pi^2 \rho (\theta_1 + \theta_2) + \pi\rho (\theta_1^2 + \theta_2^2)}{8\pi} + \pi^2 \frac{\rho}{4} \Rightarrow \\
&G_T(\theta_2; \rho) - G_T(\theta_1; \rho) = \frac{(4 - \pi^2 \rho)(\theta_2 - \theta_1) - 2\pi^2 \rho (\theta_1 + \theta_2) + \pi\rho (\theta_1^2 + \theta_2^2)}{8\pi} + \frac{\pi^2 \rho}{4}. \quad (\text{E.4})
\end{aligned}$$

E.3 Uniform

$$\begin{aligned}
G_U(\theta_2) - G_U(\theta_1) &= \int_{\theta_1}^{\theta_2} \frac{1}{2\pi} d\theta \\
&= \frac{1}{2\pi} \theta \Big|_{\theta_1}^{\theta_2} \\
&= (\theta_2 - \theta_1) / (2\pi) \Rightarrow
\end{aligned}$$

$$G_U(\theta_2) - G_U(\theta_1) = (\theta_2 - \theta_1) / (2\pi) \quad (\text{E.5})$$

E.4 von Mises

The CDF for the von Mises distribution is not derived because CircStats provides the function pvm for the von Mises CDF.

E.5 Wrapped Cauchy

The PDF was obtained from Mardia (1972, p. 56, eq. 3.4.33). The following form of the CDF and its derivation were not found in the referenced texts, and may be a new result. With $0 < \rho < 1$,

$$\begin{aligned}
G_{WC}(\theta_2; \rho) - G_{WC}(\theta_1; \rho) &= \int_{\theta_1}^{\theta_2} \frac{1}{2\pi} \left[1 + 2 \sum_{k=1}^{\infty} \rho^k \cos(k\theta) \right] d\theta \\
&= \int_{\theta_1}^{\theta_2} \left[\frac{1}{2\pi} + \frac{1}{\pi} \rho \cos(\theta) + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \cos(k\theta) \right] d\theta \\
&= \frac{1}{2\pi} \int_{\theta_1}^{\theta_2} d\theta + \frac{\rho}{\pi} \int_{\theta_1}^{\theta_2} \cos(\theta) d\theta + \frac{1}{\pi} \int_{\theta_1}^{\theta_2} \left[\sum_{k=2}^{\infty} \rho^k \cos(k\theta) \right] d\theta \\
&= \frac{\theta_2 - \theta_1}{2\pi} + \frac{\rho}{\pi} (\sin(\theta_2) - \sin(\theta_1)) + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \frac{1}{k} \int_{\theta_1}^{\theta_2} \cos(k\theta) k d\theta \\
&= \frac{\theta_2 - \theta_1 + 2\rho(\sin(\theta_2) - \sin(\theta_1))}{2\pi} + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \frac{1}{k} (\sin(k\theta_2) - \sin(k\theta_1)) \\
&\Rightarrow
\end{aligned}$$

$$\begin{aligned}
G_{WC}(\theta_2; \rho) - G_{WC}(\theta_1; \rho) &= \frac{\theta_2 - \theta_1 + 2\rho(\sin(\theta_2) - \sin(\theta_1))}{2\pi} \\
&\quad + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \frac{1}{k} (\sin(k\theta_2) - \sin(k\theta_1)). \tag{E.6}
\end{aligned}$$

Appendix F

Verification by Evaluation of the CDF Formulae with Support $[0, 2\pi)$

F.1 Cardioid

$$\begin{aligned}
G_C(\theta = 2\pi; \rho) - G_C(\theta = 0; \rho) &= \left. \left\{ \frac{\theta_2 - \theta_1 + 2\rho(\sin(\theta_2) - \sin(\theta_1))}{2\pi} \right\} \right|_{\theta_1=0}^{\theta_2=2\pi} \\
&= \frac{2\pi - 0 + 2\rho(\sin(2\pi) - \sin(0))}{2\pi} \\
&= \frac{2\pi + 2\rho(0 - 0)}{2\pi} \\
&= 1 \Rightarrow
\end{aligned}$$

$$G_C(\theta = 2\pi; \rho) - G_C(\theta = 0; \rho) = 1$$

F.2 Triangular

Case 1, $0 \leq \theta_1 < \theta_2 \leq \pi$

$$\begin{aligned}
G_T(\theta = \pi; \rho) - G_T(\theta = 0; \rho) &= \left. \left\{ \left(\frac{4 + \pi^2 \rho - \pi\rho(\theta_2 + \theta_1)}{8\pi} \right) (\theta_2 - \theta_1) \right\} \right|_{\theta_1=0}^{\theta_2=\pi} \\
&= \left(\frac{4 + \pi^2 \rho - \pi\rho(\pi + 0)}{8\pi} \right) (\pi - 0) \\
&= \left(\frac{4 + \pi^2 \rho - \pi\rho(\pi)}{8\pi} \right) \pi \\
&= 0.5 \Rightarrow
\end{aligned}$$

$$G_T(\theta = \pi; \rho) - G_T(\theta = 0; \rho) = 0.5$$

Case 2, $\pi \leq \theta_1 < \theta_2 \leq 2\pi$

$$\begin{aligned}
G_T(\theta = 2\pi; \rho) - G_T(\theta = \pi; \rho) &= \left. \left\{ \frac{4 - 3\pi^2 \rho + \pi\rho(\theta_2 + \theta_1)}{8\pi} (\theta_2 - \theta_1) \right\} \right|_{\theta_1=\pi}^{\theta_2=2\pi} \\
&= \frac{4 - 3\pi^2 \rho + \pi\rho(2\pi + \pi)}{8\pi} (2\pi - \pi) \\
&= \frac{4 - 3\pi^2 \rho + \pi\rho(3\pi)}{8\pi} (\pi) \\
&= 0.5 \Rightarrow
\end{aligned}$$

$$G_T(\theta = 2\pi; \rho) - G_T(\theta = \pi; \rho) = 0.5$$

Case 3, $0 \leq \theta_1 < \pi \leq \theta_2 < 2\pi$

$$\begin{aligned} G_T(\theta = 2\pi; \rho) - G_T(\theta = 0; \rho) &= \left. \left(\frac{(4 - \pi^2 \rho)(\theta_2 - \theta_1) - 2\pi^2 \rho(\theta_1 + \theta_2) + \pi \rho(\theta_1^2 + \theta_2^2)}{8\pi} + \frac{\pi^2 \rho}{4} \right) \right|_{\theta_1=0}^{\theta_2=2\pi} \\ &= \frac{(4 - \pi^2 \rho)(2\pi - 0) - 2\pi^2 \rho(0 + 2\pi) + \pi \rho(0^2 + 4\pi^2)}{8\pi} + \frac{\pi^2 \rho}{4} \\ &= \frac{(8\pi - 2\pi^3 \rho) - 4\pi^3 \rho + 4\pi^3 \rho}{8\pi} + \frac{2\pi^3 \rho}{8\pi} \\ &= \frac{8\pi - 2\pi^3 \rho + 2\pi^3 \rho}{8\pi} \\ &= 1 \Rightarrow \end{aligned}$$

$$G_T(\theta = 2\pi; \rho) - G_T(\theta = 0; \rho) = 1$$

F.3 Uniform

$$\begin{aligned} G_U(\theta = 2\pi) - G_U(\theta = 0) &= \left. \frac{\theta_2 - \theta_1}{2\pi} \right|_{\theta_1=0}^{\theta_2=2\pi} \\ &= \frac{2\pi - 0}{2\pi} \\ &= 1 \Rightarrow \end{aligned}$$

$$G_U(\theta = 2\pi) - G_U(\theta = 0) = 1$$

F.4 Von Mises

See E.4.

F.5 Wrapped Cauchy

$$\begin{aligned}
G_{WC}(\theta = 2\pi; \rho) - G_{WC}(\theta = 0; \rho) & \stackrel{(E.6)}{=} \left\{ \frac{\theta_2 - \theta_1 + 2\rho(\sin(\theta_2) - \sin(\theta_1))}{2\pi} \right. \\
& \quad \left. + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \frac{1}{k} (\sin(k\theta_2) - \sin(k\theta_1)) \right\} \Bigg|_{\theta_1=0}^{\theta_2=2\pi} \\
& \stackrel{F.2}{=} 1 + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \frac{1}{k} (\sin(k2\pi) - \sin(k0)) \\
& = 1 + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \frac{1}{k} (0 - 0) \\
& = 1 \\
& \quad G_{WC}(\theta = 2\pi; \rho) - G_{WC}(\theta = 0; \rho) = 1
\end{aligned}$$

Appendix G

Modification of the PDF and CDF Formulae for Rotated Support $[-\pi, +\pi)$

In Appendices G and H, the CDF formulae of Appendix E will be modified to rotate the support from $[0, 2\pi)$ to $[-\pi, +\pi)$ radians on the unit circle. In Appendix G, the PDF formulae will also be modified to rotate the support from $[0, 2\pi)$ to $[-\pi, +\pi)$.

Rotation of the CDF support is required to map a GRV to a CRV with mean direction 0 (Chapter 5, Section 5.3). Unlike the CDF of a linear random variable (RV), the circular CDF does not have a single origin. Let the support be $[-\pi, +\pi)$. Then, by means of the CDF – Inverse CDF transformation (Figure 5-2), the most negative values of the standard GRV map to the most negative values of a CRV, the modes coincide, and the most positive values of the GRV map to the most positive values of the CRV. Thus, the circular CDF has $G_{\theta}(\theta = -\pi) = 0$, $G_{\theta}(\theta = 0) = 0.5$, and $G_{\theta}(\theta = \pi) = 1.0$.

The uniform PDF is $g(\theta) = 1/2\pi$ for all angles. Hence, it is independent of the choice of support, and the same for both supports. The cardioid, von Mises, and wrapped Cauchy distributions are functions of $\cos(\theta)$. To transform the part of the support $\pi \leq \theta < 2\pi$ to the corresponding part $-\pi \leq \phi < 0$, let $\theta = \phi + 2\pi$. Then, $\cos(\theta) = \cos(\phi + 2\pi) = \cos(\phi)\underbrace{\cos(2\pi)}_1 - \sin(\phi)\underbrace{\sin(2\pi)}_0 = \cos(\phi) \Rightarrow g(\theta) = g(\phi)$. Thus, the circular PDF for the uniform, cardioid, von Mises, and wrapped Cauchy distributions do not change with rotated support. For $0 \leq \theta < \pi$, the PDF of the triangular distribution is $g(\theta) = \frac{1}{8\pi}(4 - \pi^2\rho + 2\pi\rho(\pi - \theta))$. For the part of the rotated support $-\pi \leq \phi < 0$, $g(\theta) = \frac{1}{8\pi}(4 - \pi^2\rho + 2\pi\rho(\theta - \pi)) = \frac{1}{8\pi}(4 - \pi^2\rho + 2\pi\rho(\phi + 2\pi - \pi)) = \frac{1}{8\pi}(4 - \pi^2\rho + 2\pi\rho(\phi + \pi))$.

This is given in Table 5-1 with $\delta = \pi + \theta, -\pi \leq \theta < 0$, and $\delta = \pi - \theta, 0 \leq \theta < \pi$.

G.1 Cardioid

$$G_C(\theta_2; \rho) - G_C(\theta_1; \rho) \stackrel{(E.1)}{=} \frac{\theta_2 - \theta_1 + 2\rho(\sin(\theta_2) - \sin(\theta_1))}{2\pi}$$

$$\stackrel{\text{By Inspection}}{\Rightarrow} G_C(\theta; \rho) = \begin{cases} \frac{\theta + \pi + 2\rho \sin(\theta + 2\pi)}{2\pi}, & -\pi \leq \theta < 0 \\ \frac{\theta + \pi + 2\rho \sin(\theta)}{2\pi}, & 0 \leq \theta \leq \pi \end{cases}$$

$$= \frac{\theta + \pi + 2\rho \sin(\theta)}{2\pi}, \quad -\pi \leq \theta \leq \pi$$

$$G_C(\theta; \rho) = \frac{\theta + \pi + 2\rho \sin(\theta)}{2\pi}, \quad -\pi \leq \theta \leq \pi \quad (\text{G.1})$$

Figure G-1, which verifies the cardioid CDF with support $[-\pi, +\pi)$, was plotted

using the following R code:

```
par(mai=c(.8,.75,.1,.1)); r<-0.3
theta <- seq(-pi, pi, length=201)
GC <- (theta + pi + 2*r*sin(theta))/(2*pi)
plot(theta, GC, ty="l")
```

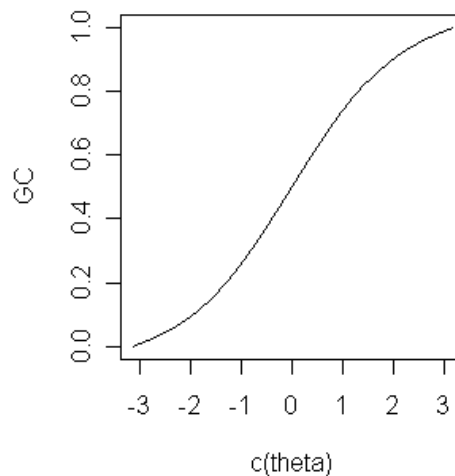


Figure G-1. Visual Verification of Cardioid CDF, $\rho = 0.30$, Support $[-\pi, +\pi)$ Radians. Slope is changing as expected with maximum slope occurring at zero radians.

G.2 Triangular

$$G_T(\theta_2; \rho) - G_T(\theta_1; \rho) \stackrel{(E.2)(E.3)}{=} \begin{cases} \left(\frac{4 + \pi^2 \rho - \pi \rho (\theta_2 + \theta_1)}{8\pi} \right) (\theta_2 - \theta_1), & 0 \leq \theta_1 < \theta_2 \leq \pi \\ \frac{4 - 3\pi^2 \rho + \pi \rho (\theta_2 + \theta_1)}{8\pi} (\theta_2 - \theta_1), & \pi \leq \theta_1 < \theta_2 < 2\pi \end{cases}$$

inspection
⇒

$$G_T(\theta; \rho) = \begin{cases} \frac{4 - 3\pi^2 \rho + \pi \rho (\theta + 3\pi)}{8\pi} (\theta + \pi), & -\pi \leq \theta < 0 \\ 0.5 + \left[\frac{4 + \pi^2 \rho - \pi \rho \theta}{8\pi} \right] \theta, & 0 \leq \theta \leq \pi \end{cases} \quad (G.2)$$

Figure G-2, which verifies the triangular CDF with support $[-\pi, +\pi)$, was plotted

using the following R code:

```
par(mai=c(.8,.75,.1,.1)); r<-0.3
theta1 <- seq(-pi,0,length=100); theta1 <- theta1[-100]; theta2 <- seq(0,pi,length=100)
GT1 <- (4-3*pi^2*r + pi*r*(theta1+ 3*pi))*(theta1+pi)/(8*pi)
GT2 <- 0.5 + (4 + pi^2*r - pi*r*theta2)*theta2/(8*pi)
plot(c(theta1,theta2), c(GT1, GT2), ty="l")
```

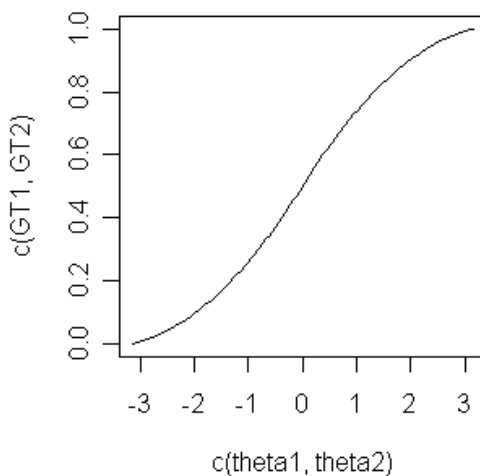


Figure G-2. Visual Verification of Triangular CDF, $\rho = 0.30$, Support $[-\pi, +\pi)$ Radians. Slope is changing as expected with maximum slope occurring at zero radians.

G.3 Uniform

$$G_U(\theta_2) - G_U(\theta_1) \stackrel{(E.5) \theta_2 - \theta_1 \text{ inspection}}{=} \frac{\theta_2 - \theta_1}{2\pi} \Rightarrow$$

$$G_U(\theta) = \frac{\theta + \pi}{2\pi}, \quad -\pi \leq \theta \leq \pi \quad (\text{G.3})$$

G.4 von Mises

The von Mises cumulative probabilities computed via function `pvm` in R package

`CircStats` (Lund and Agostinelli 2006):

```
> pvm(theta = -pi, mu=0, kappa=1)
[1] 0.5
> pvm(theta = -.0000001, mu=0, kappa=1)
[1] 1
> pvm(theta = 0, mu=0, kappa=1)
[1] 0
> pvm(theta = pi, mu=0, kappa=1)
[1] 0.5
> pvm(theta = 2*pi, mu=0, kappa=1)
[1] 0

> pvm(theta = 2*pi -.0000001, mu = 0, kappa = 1)
[1] 1
inspection
⇒
```

$$G_{vM}(\theta; \kappa) = \begin{cases} pvm(\theta, \mu = 0, \kappa = k) - 0.5, & -\pi \leq \theta < 0 \\ pvm(\theta, \mu = 0, \kappa = k) + 0.5, & 0 \leq \theta \leq \pi \end{cases} \quad (\text{G.4})$$

Figure G-3, which verifies the von Mises CDF with support $[-\pi, +\pi)$, was plotted

using the following R code:

```
require(CircStats)
r<-0.3; k<-A1inv(r)
par(mai=c(.8,.75,.1,.1))
theta1 <- seq(-pi,0,length=100); theta1 <- theta1[-100]; theta2 <- seq(0,pi,length=100)
GvM1 <- pvm(theta1, mu=0, kappa=k) - 0.5
GvM2 <- pvm(theta2, mu=0, kappa=k) + 0.5
plot(c(theta1,theta2), c(GvM1, GvM2), ty="l")
```

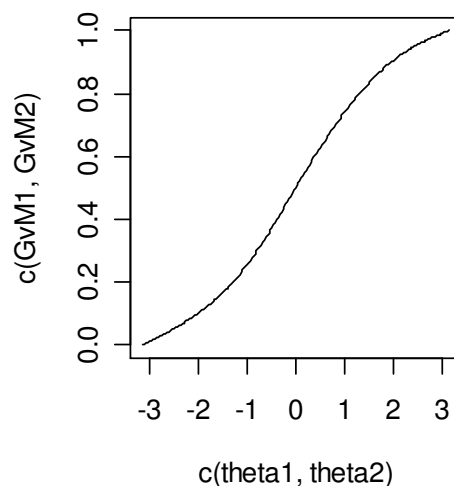


Figure G-3. Visual Verification of von Mises CDF, $\rho = 0.30$, Support $[-\pi, +\pi)$ Radians. Slope is changing as expected with maximum slope occurring at zero radians.

G.5 Wrapped Cauchy

Additional forms of the wrapped Cauchy CDF were discovered after the completion of Chapter 3. One of these forms is incorrect. Due to the complexity of the issues arising from the multiple forms of the wrapped Cauchy CDF, these forms will be treated in Appendix H.

Appendix H

Wrapped Cauchy CDF

H.1 Additional Forms of the CDF

H.1.1 Incorrect CDF

This distribution was introduced by Lévy (1939). (H.1) was taken from Mardia (1972, p. 57, eq. 3.4.36), and repeated in Fisher (1993, p. 46, 3.27). With $\mu = 0$, it is

$$G_{WC}(\theta; \rho) = \frac{1}{2\pi} \cos^{-1} \left(\frac{(1 + \rho^2) \cos(\theta) - 2\rho}{1 + \rho^2 - 2\rho \cos(\theta)} \right), 0 \leq \theta < 2\pi, \mu = 0. \quad (\text{H.1})$$

(H.1) is plotted in Figure H-1 with the following code:

```
theta <- seq(0, 2*pi, length = 197); r <- 0.75 # rho
G <- acos(((1+r^2)*cos(theta)-2*r)/(1+r^2-2*r*cos(theta)))/(2*pi)
plot(theta, G, ty = "l", cex.lab=1.6, cex.axis=1.2)
```

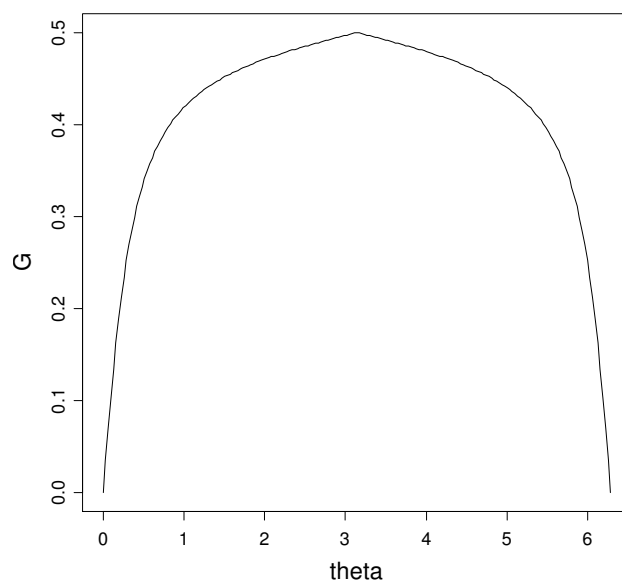


Figure H-1. Incorrect Wrapped Cauchy CDF, $\rho = 0.75$, Support $[0, 2\pi)$ Radians. A correct CDF is monotonic increasing.

H.1.2 WCACDF

WCACDF was obtained from National Institute of Standards and Technology (NIST), Statistical Engineering Division, Dataplot, at <http://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/wcacdf.pdf>, eq. Aux-326.

With $\mu = 0$, it is

$$G_{WC}(\theta; \rho) = \begin{cases} \frac{\tan^{-1}\left(\frac{-\sin(\frac{\theta}{2}) - \rho \sin(\frac{\theta}{2})}{-\cos(\frac{\theta}{2}) + \rho \cos(\frac{\theta}{2})}\right) - \tan^{-1}\left(\frac{\sin(\frac{\theta}{2}) + \rho \sin(\frac{\theta}{2})}{-\cos(\frac{\theta}{2}) + \rho \cos(\frac{\theta}{2})}\right)}{2\pi} \equiv G_{WC1}, & 0 \leq \theta < \pi \\ 1 - G_{WC1}(2\pi - \theta; \rho), & \pi < \theta < 2\pi. \end{cases} \quad (\text{H.2})$$

They cite Johnson, Kotz, and Balakrishnan (1994), which is a principle reference for looking up details on distributions. (H.2), as shown in Figure H-2, was plotted with the following code:

```
theta1 <- seq(pi, 2*pi, length = 100); theta2 <- seq(0, pi, length = 100)
theta3 <- seq(-pi, 0, length = 100); r <- 0.75
G <- function(theta) {(atan((-1-r)*sin(.5*theta)/((-1+r)*cos(.5*theta)) )
  - atan( (1+r)*sin(.5*theta)/((r-1)*cos(.5*theta)))/(2*pi)}
plot(c(theta2, theta1), c(G(theta2), 1-G(2*pi-theta1)), ty = "l", cex.lab=1.6, cex.axis=1.2)
```

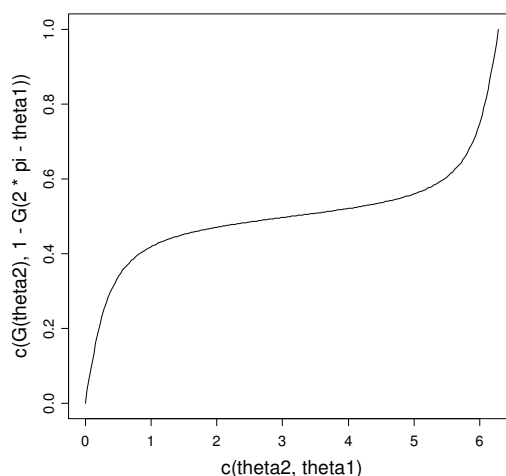


Figure H-2. Dataplot WCACDF of Wrapped Cauchy CDF, $\rho = 0.75$, Support $[0, 2\pi)$ Radians. The slope changes with maximum rate at zero radians as expected.

H.2 Alternate Forms with Support $[0, 2\pi)$

This section will modify the CDF formulae of Section H.1 and Appendix E, Section E.5 to facilitate evaluation. These circular CDFs, going counterclockwise, will have $G_{WC}(\theta = \pi; \rho) = 0$, $G_{WC}(\theta = 2\pi; \rho) = 0.5$, $G_{WC}(\theta = 0; \rho) = 0.5$, and $G_{WC}(\theta = \pi; \rho) = 1.0$.

H.2.1 Iterated

$$G_{WC}(\theta; \rho) \stackrel{(E.6)}{=} \begin{cases} 0.5 + \frac{\theta + 2\rho \sin(\theta)}{2\pi} + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \frac{1}{k} \sin(k\theta), & 0 \leq \theta < \pi \\ -0.5 + \frac{\theta + 2\rho \sin(\theta)}{2\pi} + \frac{1}{\pi} \sum_{k=2}^{\infty} \rho^k \frac{1}{k} \sin(k\theta), & \pi \leq \theta < 2\pi \end{cases} \quad (H.3)$$

H.2.2 Revised (H.1)

$$G_{WC}(\theta; \rho) \stackrel{(H.1)}{=} \begin{cases} 0.5 + \frac{1}{2\pi} \cos^{-1} \left(\frac{(1 + \rho^2) \cos(\theta) - 2\rho}{1 + \rho^2 - 2\rho \cos(\theta)} \right), & 0 \leq \theta < \pi \\ 0.5 - \frac{1}{2\pi} \cos^{-1} \left(\frac{(1 + \rho^2) \cos(\theta) - 2\rho}{1 + \rho^2 - 2\rho \cos(\theta)} \right), & \pi \leq \theta < 2\pi \end{cases} \quad (H.4)$$

H.2.3 WCACDF

$$F(\theta; \rho) \stackrel{(H.2)}{=} \frac{\tan^{-1} \left(\frac{-\sin(\frac{\theta}{2}) - \rho \sin(\frac{\theta}{2})}{-\cos(\frac{\theta}{2}) + \rho \cos(\frac{\theta}{2})} \right) - \tan^{-1} \left(\frac{\sin(\frac{\theta}{2}) + \rho \sin(\frac{\theta}{2})}{-\cos(\frac{\theta}{2}) + \rho \cos(\frac{\theta}{2})} \right)}{2\pi}$$

$$G_{WC}(\theta; \rho) = \begin{cases} 0.5 + F(\theta; \rho), & 0 \leq \theta < \pi \\ 0.5 - F(2\pi - \theta; \rho), & \pi \leq \theta < 2\pi \end{cases} \quad (H.5)$$

H.3 Evaluation of Alternate Forms

H.3.1 Forms of Wrapped Cauchy CDF Visually Indistinguishable

In Figure H-3, the circular CDFs are computed over $[\pi, 2\pi) \cup [0, \pi)$ and plotted over the equivalent support $[-\pi, +\pi)$ using the following R code:

```
theta1 <- seq(pi, 2*pi, length = 100); theta1 <- theta1[-100]
theta2 <- seq(0, pi, length = 100); theta2 <- theta2[-100]
theta3 <- seq(-pi,0, length = 100); theta3 <- theta3[-100]; r <- 0.75
# Iterated CDF with total 15 iterations
GI1 <- (theta1-pi+2*r*sin(theta1))/(2*pi); GI2 <- 0.5 + (theta2+2*r*sin(theta2))/(2*pi)
GI <- function(theta)
{ sum.iter <- 0; for (k in 2:15) { sum.iter <- sum.iter + (1/pi)*(r^k)*(1/k)*sin(k*theta) }
  return(sum.iter) }
GI1 <- GI1 + GI(theta1); GI2 <- GI2 + GI(theta2)
par(mai=c(.65,.6,.1,.1), mgp=c(2,1,0),cex.axis=.7, cex.lab=.8)
plot(c(theta3, theta2), c(GI1, GI2), ty = "l")
# Corrected wrapped Cauchy CDF
GM1 <- .5 - acos(((1+r^2)*cos(theta1)-2*r)/(1+r^2-2*r*cos(theta1)))/(2*pi)
GM2 <- .5 + acos(((1+r^2)*cos(theta2)-2*r)/(1+r^2-2*r*cos(theta2)))/(2*pi)
plot(c(theta3, theta2), c(GM1, GM2), ty = "l")
F <- function(theta) {(atan((-1-r)*sin(.5*theta)/((-1+r)*cos(.5*theta)) )
  - atan( (1+r)*sin(.5*theta)/((r-1)*cos(.5*theta)))/(2*pi)}
GD1 <- .5 - F(2*pi-theta1); GD2 <- F(theta2) + 0.5
plot(c(theta3, theta2), c(GD1, GD2), ty = "l")
```

In Figure H-3, the revised wrapped Cauchy CDF, WCACDF, and the iterative formula with 15 iterations are visually indistinguishable.

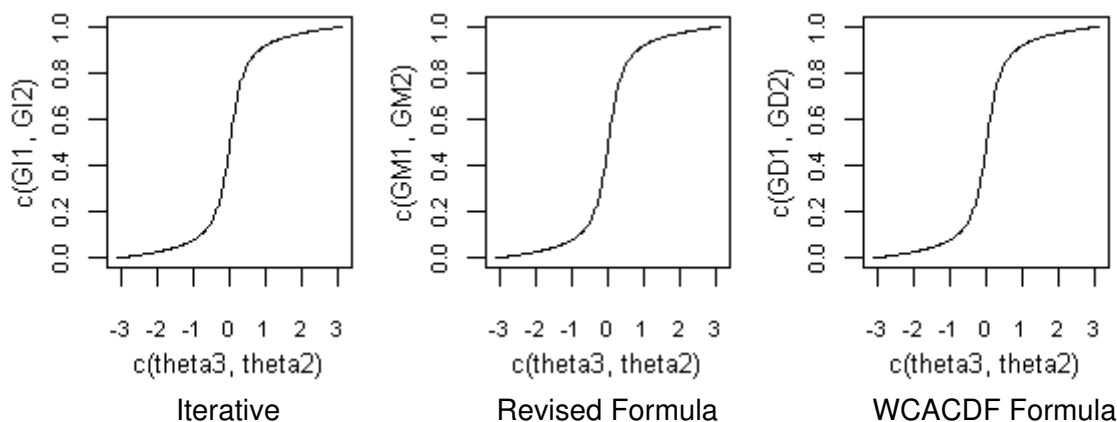


Figure H-3. Three Forms of the Wrapped Cauchy CDF, $\rho = 0.75$, Support $[-\pi, \pi)$ Radians. Visually, the three forms are indistinguishable.

H.3.2 Proximity of Alternative CDF Formulae

With 198 points, the revised CDF and WCACDF total absolute difference is about e^{-14} . The iterated CDF achieves the similar accuracy when the number of iterations is increased to about 160 iterations.

```
# Compare iterative to WCACDF
sum(abs(c(GI1, GI2) - c(GD1, GD2)))
[1] 0.02960684
```

```
# Compare iterative to revised CDF
sum(abs(c(GI1, GI2) - c(GM1, GM2)))
[1] 0.02960684
```

```
# Compare WCACDF to revised CDF
sum(abs(c(GD1, GD2) - c(GM1, GM2)))
[1] 3.375078e-14
```

The accuracy of the iterated wrapped Cauchy CDF (H.3) depends on ρ and the number of iterations. As ρ increases, the number of iterations must increase to maintain accuracy. The need for additional iterations at high ρ (0.95) is demonstrated in Figure H-4, which was plotted with the following R code:

```
r <- 0.95; GI1 <- (theta1-pi+2*r*sin(theta1))/(2*pi); GI2 <- 0.5 + (theta2+2*r*sin(theta2))/(2*pi)
GI1 <- GI1 + GI(theta1); GI2 <- GI2 + GI(theta2); plot(c(theta3, theta2), c(GI1, GI2), ty = "l")
```

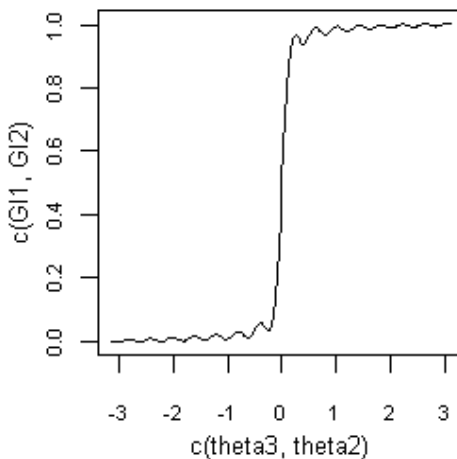


Figure H-4. Iterated Wrapped Cauchy CDF, $\rho = 0.95$, Support $[-\pi, \pi)$ Radians, 15 Iterations. The iterative form gets rough as $\rho \rightarrow 1$ and computation time increases.

H.4 Selected Form for Rotated Support $[-\pi, +\pi)$

The revised CDF (H.4) is more simple than the WCACDF (H.5), and does not have the inaccuracy of the iterative form (H.3) at high rho. Hence, the selected wrapped Cauchy CDF for support $[-\pi, +\pi)$ is:

$$G_{WC}(\theta; \rho) \stackrel{(H.4)}{=} \begin{cases} .5 - \frac{1}{2\pi} \cos^{-1} \left(\frac{(1 + \rho^2) \cos(\theta) - 2\rho}{1 + \rho^2 - 2\rho \cos(\theta)} \right), & -\pi \leq \theta < 0 \\ .5 + \frac{1}{2\pi} \cos^{-1} \left(\frac{(1 + \rho^2) \cos(\theta) - 2\rho}{1 + \rho^2 - 2\rho \cos(\theta)} \right), & 0 \leq \theta \leq \pi. \end{cases} \quad (H.6)$$

(H.6), as shown in Figure H-5, was plotted with the following code:

```
par(mai=c(.8,.75,.1,.1))
r <- 0.75
theta1 <- seq(-pi,0, length = 100); theta1 <- theta1[-100]
theta2 <- seq(0, pi, length = 100)
GM1 <- .5 - acos(((1+r^2)*cos(theta1)-2*r)/(1+r^2-2*r*cos(theta1)))/(2*pi)
GM2 <- .5 + acos(((1+r^2)*cos(theta2)-2*r)/(1+r^2-2*r*cos(theta2)))/(2*pi)
plot(c(theta1, theta2), c(GM1, GM2), ty = "l")
```

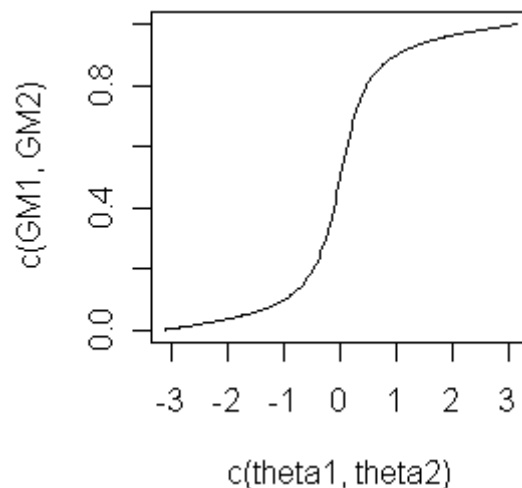


Figure H-5. Visual Verification of Wrapped Cauchy CDF, $\rho = 0.75$, Support $[-\pi, \pi)$ Radians. The slope changes with maximum rate at zero radians as expected.

Appendix I

Triangular Inverse CDF

The inverse CDF is computed in order to map a Gaussian random field to a circular random field according to the method of Chapter 5, Section 5.3. For the cumulative probability $u = F_z(z(\mathbf{x}))$, and the triangular CDF $G_T(\theta; \rho)$, $u \equiv G_T(\theta; \rho)$.

$$G_T(\theta; \rho) \stackrel{(G.2)}{=} \begin{cases} \frac{4 - 3\pi^2 \rho + \pi \rho (\theta + 3\pi)}{8\pi} (\theta + \pi), & -\pi \leq \theta < 0 \\ 0.5 + \left[\frac{4 + \pi^2 \rho - \pi \rho \theta}{8\pi} \right] \theta, & 0 \leq \theta \leq \pi \end{cases} \Rightarrow$$

$$u \equiv \begin{cases} +\frac{\rho}{8} \theta^2 + \frac{4 + \pi^2 \rho}{8\pi} \theta + \frac{1}{2}, & -\pi \leq \theta < 0 \\ -\frac{\rho}{8} \theta^2 + \frac{4 + \pi^2 \rho}{8\pi} \theta + \frac{1}{2}, & 0 \leq \theta \leq \pi \end{cases} \quad (1.1)$$

From (1.1), let
$$a = \begin{cases} +\frac{\rho}{8}, & -\pi \leq \theta < 0 \\ -\frac{\rho}{8}, & 0 \leq \theta \leq \pi \end{cases}, \quad b = \frac{4 + \pi^2 \rho}{8\pi}, \quad c = \frac{1}{2} - u. \quad (1.2)$$

Applying the quadratic solution of Press, Flannery, Teukolsky, and Vetterling (1986) for an accurate solution when a , c , or both are small (when $\rho \approx 0$, $u \approx 0.5$, the familiar quadratic solution $\theta = (-b \pm \sqrt{b^2 - 4ac}) / (2a)$ does not work),

$$q \stackrel{(1.2)}{\equiv} -\frac{1}{2} \left(b + \operatorname{sgn}(b) \sqrt{b^2 - 4ac} \right) = -\frac{1}{2} \left(b + \sqrt{b^2 - 4ac} \right) \Rightarrow$$

$$q = -\frac{1}{2} \left(b + \sqrt{b^2 - 4ac} \right) \quad (1.3)$$

$$\theta = \frac{c}{q} \quad (1.4)$$

Figure I-1, which was plotted with the following R code, verifies this result.

```
rho=.95*4/pi^2
u1 <-seq(0,.5,length=20)
a <- rho/8
b <- (4+pi^2*rho)/(8*pi)
c <- 0.5 - u1
q <- -.5*(b+sqrt(b^2-4*a*c))
x1 <- c/q

u2 <-seq(.5,1,length=20)
a <- -1*rho/8
b <- (4+pi^2*rho)/(8*pi)
c <- 0.5 - u2
q <- -.5*(b+sqrt(b^2-4*a*c))
x2 <- c/q

plot(c(u1,u2), c(x1,x2), ty="l")
```

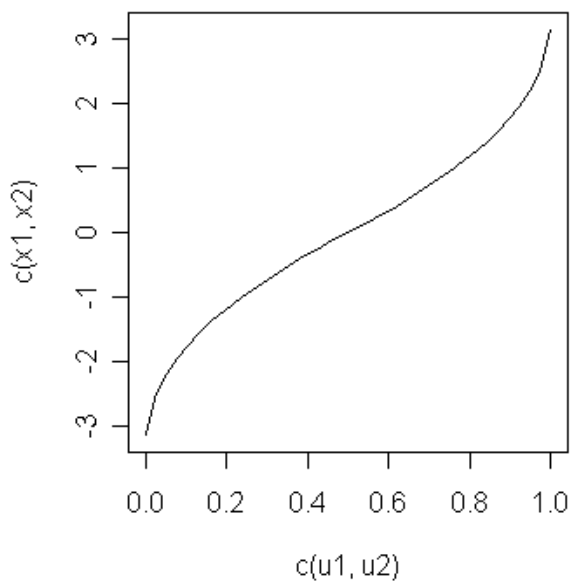


Figure I-1. Visual Verification of Triangular Inverse CDF, $\rho = 0.95 * 4 / \pi^2$, Support $[-\pi, \pi)$ Radians.

Appendix J

R Package Documentation

J.1 Introduction and Installation

J.1.1 Introduction

Vectors and periodic phenomena (e.g., traffic concentration vs. time) have direction on a circular scale of 360° . Circular-spatial data consists of location and direction. Circular random fields generate circular-spatial data. A circular RF (CRF) is defined as a space containing circular random variables (CRV) with spatial correlation. In 2 dimensional space, the CRF is the set $\{\Theta(\mathbf{x}), \mathbf{x} \in \mathbb{R}^2\}$ with Θ the circular RV and \mathbf{x} the location. A CRV takes random directions on a unit circle with the total probability of all possible directions distributed on the unit circle with support $[0, 2\pi)$ or $[-\pi, \pi)$. The starting point of the support is the same direction as the ending point. A CRV is expressed as either a scalar in units of radians or degrees ($^\circ$), or as a unit vector. Spatial correlation increases as distance between measurement locations decreases, i.e., rotations from the mean direction tend to be more similar. In the form required by the circular kriging solution of chapter 4, spatial correlation is defined as the mean cosine of the angle between random components of directions (nonrandom component removed) vs. distance between measurement locations. An isotropic CRF is defined as a CRF in which spatial correlation is the same in all directions in space. A geometric anisotropic CRF is defined as a CRF in which spatial correlation varies with direction in space.

The main functions of package CircSpatial include:

- SimulateCRF: Simulate a circular random field (CRF). (Appendix J.2)

- CircResidual: Compute rotational residuals (deviations) from the mean direction in radians. (Appendix J.3)
- CosinePlots: Plot the empirical and fitted cosineograms of the spatial correlation of circular-spatial data. (Appendix J.4)
- KrigGRF: Krig and smooth the circular-spatial residuals. (Appendix J.5)
- InterpDirection: Interpolate the model of mean direction at each kriging location. (Appendix J.6)
- CircDataimage: Generate a graphical user interface (GUI) for interactive circular dataimages. (Appendix J.10)
- PlotVectors: Traditional plots of vector-spatial data. (Appendix J.11)

J.1.2 PC Windows Installation

An R Package provides a mechanism for loading code and documentation. To install the R package CircSpatial from zip files:

- 1) Download and unzip the SupportingPackages.zip file under Software at <http://home.comcast.net/~morphwj/site/?/page/Dissertation/> (username=dissertation and password=JuergenS). Copy the extracted folders to the R library folder, e.g. in C:\Program Files\R\R-2.8.0\library (update path as necessary). CircSpatial depends on R packages CircStats, fields, geoR, RandomFields, sp, and spam.
- 2) Download and unzip the CircSpatial_1.0.zip. Copy the extracted CircSpatial folder to the R library. If unable to open the Help CircSpatial.chm file, open the file

properties. If you see

Security:	This file came from another computer and might be blocked to help protect this computer.
-----------	--

, click Unblock, Apply,

and OK.

- 3) Install the current version of Active State Tcl from <http://downloads.activestate.com/ActiveTcl/Windows/> to the root directory C:\

(recommended). This installation is required to access the `img` function to add graphics to the `CircDataimage` GUI. The `img` function is not included in the R distribution of Tcl.

- 4) The first time R is started after copying the dependent packages to the R library, update these packages from the R menu bar by selecting “Packages” and “Update package(s)...”
- 5) To load package `CircSpatial` from the R prompt, enter `require(CircSpatial)`.
- 6) Before using `CircDataimage` tell R where to find the special Tcl function `img`.

Update the highlighted input and enter the following commands from the R prompt:

```
require(tcltk, quietly=TRUE, warn.conflicts=TRUE)
Sys.setenv("TCL_LIBRARY"="C:/Tcl/lib/tcl8.5")
Sys.setenv("MY_TCLTK"="Yes")
addTclPath(path = "C:/Tcl/lib/teapot/package/win32-ix86/lib")
tclRequire("img::jpeg")
```

These commands were not embedded in the build of `CircSpatial` to make the build independent of the user installation path and version of Active State Tcl .

J.2 SimulateCRF

Generate an isotropic (depends on distance only) or geometric anisotropic (depends on distance and direction) CRF in a plane. An isotropic CRF is a space of circular random variables (CRV) with spatial correlation only dependent on distance. CRFs are implemented for the circular distributions uniform (U), triangular (Tri), cardioid (Card), von Mises (vM), or wrapped Cauchy (WrC), with mean resultant direction (μ) of zero (default), and specified mean resultant length (ρ) and range (distance at which CRV are uncorrelated) of spatial correlation. Spatial correlation means that as the distance between measurement locations decreases, random rotations of direction from the mean direction tend to be more similar.

J.2.1 Principle

As shown in Figure J-1, the CRF is obtained by transforming a Gaussian random field (GRF), which is generated by package geoR or Random Fields (if $N > 500$), via the inverse cumulative distribution function (CDF) of the CRV operating on the CDF of the GRV, which in turn, operates on observations of the GRV in the GRF. Because spatially dependent random variables tend to be similar at short distances and the CDF is monotonic increasing, their cumulative probabilities also tend to be similar. Thus, spatial correlation is transformed from the GRF to the normal cumulative probabilities as shown in Figure J-2. In turn, similarity of cumulative probabilities transforms to similarity of CRV via the monotonic inverse CDF of the CRV. Details of the GRF are available in the Help files of packages geoR and RandomFields. The direction of the inverse circular CDF is obtained via interpolation of the circular CDF on a fine scale. For additional information, see Chapter 5, Simulation Of A Circular Random Field.

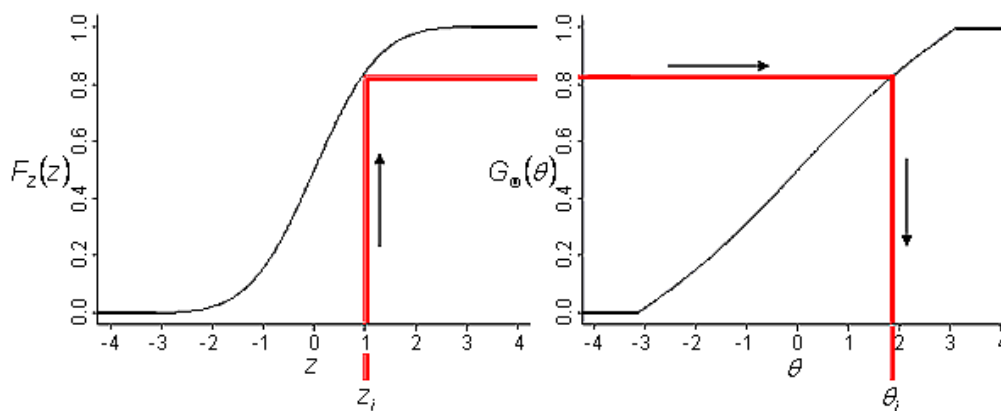


Figure J-1. Mapping a GRF to a CRF via CDFs. $\theta_i = G_{\Theta}^{-1}\{F_Z(z_i); \rho\}$. Direction of Θ is expressed in radian units.

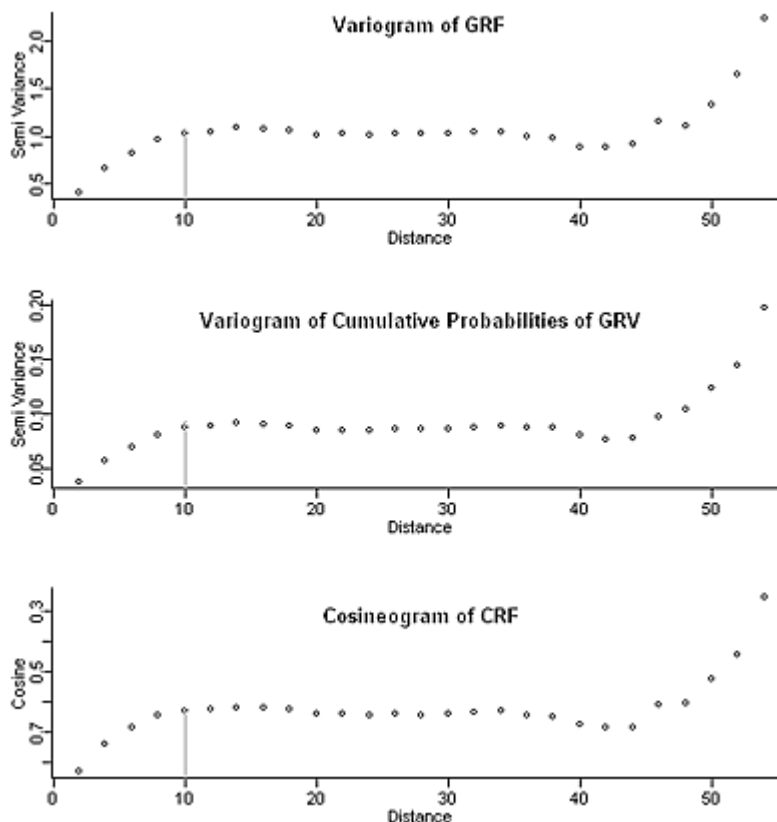


Figure J-2. Shapes of Variograms and Inverted Cosineogram Show Spatial Correlation Transformed from the GRF with Spherical Covariance and Range $r = 10$.

J.2.2 Usage and Input Arguments

SimulateCRF(N=100, CircDistr, Rho, Mu=0, Range, Ext=1, CovModel, Grid=NULL, Anisotropy=NULL, OverFit=FALSE, Resolution=.01)

N: The number of observations to simulate.

CircDistr: Name of circular distribution “Card”, “Tri”, “U”, “vM”, or “WrC”. Circular random fields are implemented for the uniform, von Mises, cardioid, triangular, and wrapped Cauchy distributions.

Rho: Mean resultant length parameter ρ of the circular distribution: Wrapped Cauchy, $0 < \rho < 1$; von Mises, $0 < \rho < 1$; cardioid, $0 < \rho \leq 0.5$; triangular, $0 < \rho \leq 4/\pi^2$; uniform, $\rho = 0$ (all directions have equal density)

Mu: Mean resultant direction of circular distribution (rad). $|\mu| \leq \pi$.

Range: Distance at which CRV are not correlated.

Ext: $\text{Ext} \geq 1$. Range \times Ext (Extension) is the horizontal and vertical width of square sample space. Simulation at distances beyond the range (distance at which CRV are no longer spatially dependent) reduces edge effect at the range resulting in a more accurate representation of the sill (mean cosine of independent CRV) near the range.

CovModel: Name of a spatial correlation function from package geoR function `cov.spatial`, e.g., “exponential”.

Grid: An $n \times 2$ matrix of regular or irregular location coordinates of the simulated data. Grid overrides N and Ext.

Anisotropy: Vector of geometric anisotropy (angle, ratio). Angle in radians, ratio ≥ 1 . See R Help for package geoR function `grf`.

OverFit: If TRUE, realizations of the GRV are standardized (centered to mean 0 and scaled to standard deviation 1) prior to the transformation to a CRF. Standardization stabilizes realizations of the GRV, enhancing the fit of the output CRF to the specified circular distribution. Standardization is suitable for demonstration with closer fit, visualization, and illustrations. Undesirable effects include loss of independence of the marginal GRVs, biased GRF covariance, and biased testing. If FALSE (default), the realizations are not standardized. Non standardization includes expected variation from transformation of variation in mean and standard deviation of the realization of the GRV of the GRF. OverFit=FALSE is recommended for the purposes of simulation, analysis, and testing.

Resolution: For nonclosed form inverse CDF, circular quantiles are interpolated at resolution Resolution. $0.001 \leq \text{Resolution} \leq 0.01$ recommended.

J.2.3 Output List

x: Vector of x coordinates of simulated observations.

y: Vector of y coordinates of simulated observations.

direction: Vector of direction of simulated observations in radians. To change mean direction, add a constant.

Z: Vector of simulated observations of the GRV of the GRF.

J.2.4 Example

```
## Compute isotropic vM CRF of 121 observations, Rho=sqrt(0.5) so sill about 0.5,
## from GRF (Range=4, spherical covariance) with OverFit=TRUE for demonstration.
```

```
require(CircSpatial)
x1<- 1:11; y1 <- 1:11; y1 <- rep(y1, 11); x1 <- rep(x1, each=11)
set.seed(666)
```



```
crf1<- SimulateCRF(CircDistr="vM", Rho=sqrt(0.5), Range=4, CovModel="spherical",
  Grid=cbind(x1, y1))
plot(crf1$x, crf1$y, type="n", xlab="", ylab="", asp=1)
arrow.plot(a1=crf1$x, a2=crf1$y, u=cos(crf1$direction), v=sin(crf1$direction), arrow.ex=0.1,
  xpd=TRUE, true.angle=TRUE, length=.1)
```

J.2.5 Additional Examples

```
## Compute isotropic Cardioid CRF of 200 observations, Rho=0.4 so sill about 0.16,
## from GRF(exponential covariance, Range=5)
```

```
crf2 <- SimulateCRF(N=200, CircDistr="Card", Rho=0.4, Range=5, Ext=3,
  CovModel="exponential")
```

```
## Compute isotropic uniform CRF of 100 observations, sill about 0,
```

```
## from GRF(Gaussian covariance, Range=8)
```

```
crf3 <- SimulateCRF(CircDistr="U", Range=8, Ext=3, CovModel="gaussian")
```

```
## Compute isotropic triangular CRF of 100 observations, sill about 0.04,
```

```
## from GRF(spherical covariance, Range=8)
```

```
crf4 <- SimulateCRF(CircDistr="Tri", Rho=0.5*4/pi^2, Range=8, Ext=3,
  CovModel="spherical")
```

```
## Compute isotropic wrapped Cauchy CRF of 100 observations, sill about 0.8,
```

```
## from GRF(exponential covariance, Range=8)
```

```
crf5 <- SimulateCRF(CircDistr="WrC", Rho=sqrt(0.8), Range=8, Ext=3,
  CovModel="exponential")
```

```
## Compute anisotropic wrapped Cauchy CRF of 400 observations, sill about 0.95,
```

```
## from GRF(spherical covariance, Range=8) with anisotropy angle pi/4 and ratio 3
```

```
crfaniso <- SimulateCRF(N=400, CircDistr="WrC", Rho=sqrt(0.95), Range=8, Ext=3,
  CovModel="spherical", Anisotropy=c(pi/4, 3))
```

J.3 CircResidual

The first order trend, if any, must be removed from the data via an appropriate fitted model. Separately fit the cosine and sine components of direction to functions of the spatial coordinates to avoid the cross over problem (direction of 0° equals direction of 360°). Then, the fitted direction is obtained using R function `atan2(fitted sines, fitted cosines)`. Spatial correlation is encoded in the residual rotations = the rotation in radians from the fitted model direction to the data direction. A positive residual rotation indicates that counterclockwise (CCW) rotation is required to rotate the fitted model direction to the data direction. A negative residual rotation indicates that CW rotation is required. `CircResidual` returns the residuals in radians, or plots data, model, and residuals with black, thick tan, and dashed red arrows, respectively. Figures J-3 (a) to (e) show the succession of model, CRF, sample, fitted model, and residual directions.

J.3.1 Usage and Input Arguments

`CircResidual(X, Y, Raw, Trend, Plot=FALSE, AdjArrowLength=1, ...)`

X: Vector of horizontal coordinates of observation and trend locations.

Y: Vector of vertical coordinates of observation and trend locations.

Raw: Vector of direction of observations in radians.

Trend: Vector of fitted direction in radians. NAs not allowed.

Plot: If FALSE return output list. If TRUE, plot data (black), model (tan), and residuals (dashed black) with `asp=1`.

AdjArrowLength: Multiplies length of arrows in plots.

... : Additional plot parameters.

J.3.2 Output List

x: Vector of horizontal coordinates of residuals.

y: Vector of vertical coordinates of residuals.

direction: Vector of direction residuals in radians.

J.3.3 Examples

```
require(CircSpatial)
## Construct Trend Model of 121 locations
x1<- 1:11; y1 <- 1:11; y1 <- rep(y1, 11); x1 <- rep(x1, each=11)
model.direction1 <- matrix(data=c(
  157, 141, 126, 113, 101, 90, 79, 67, 54, 40, 25, 152, 137, 123, 111, 100, 90, 80, 69, 57, 44, 30,
  147, 133, 120, 109, 99, 90, 81, 71, 60, 48, 35, 142, 129, 117, 107, 98, 90, 82, 73, 63, 52, 40,
  137, 125, 114, 105, 97, 90, 83, 75, 66, 56, 45, 132, 121, 111, 103, 96, 90, 84, 77, 69, 60, 50,
  127, 117, 108, 101, 95, 90, 85, 79, 72, 64, 55, 122, 113, 105, 99, 94, 90, 86, 81, 75, 68, 60,
  117, 109, 102, 97, 93, 90, 87, 83, 78, 72, 65, 112, 105, 99, 95, 92, 90, 88, 85, 81, 76, 70,
  107, 101, 96, 93, 91, 90, 89, 87, 84, 80, 75), ncol=11, byrow=TRUE)
model.direction1 <- as.vector(model.direction1)*pi/180

## Plot Trend Model, See Figure J-3 (a)
plot(x1, y1, type="n", xlab="", ylab="", asp=1)
arrow.plot(x1, y1, u=cos(model.direction1), v=sin(model.direction1), arrow.ex=0.1, xpd=TRUE,
  true.angle=TRUE, length=.1)

## Compute vM CRF of 121 observations, Rho=sqrt(0.5) so sill about 0.5,
## from GRF (Range=4, spherical covariance).
set.seed(666)
crf1<- SimulateCRF(CircDistr="vM", Rho=sqrt(0.5), Range=4, CovModel="spherical",
  Grid=cbind(x1, y1), OverFit=TRUE)

## Plot CRF, See Figure J-3 (b)
par(mai=c(0.4, 0.35, .25, 0.25))
plot(crf1$x, crf1$y, type="n", xlab="", ylab="", asp=1)
arrow.plot(a1=crf1$x, a2=crf1$y, u=cos(crf1$direction), v=sin(crf1$direction), arrow.ex=0.1,
  xpd=TRUE, true.angle=TRUE, length=.1)

# Make sample
sample.direction1 <- model.direction1 + crf1$direction

## Plot Sample, See Figure J-3 (c)
sample.direction1 <- model.direction1 + crf1$direction
plot(x1, y1, type="n", asp=1)
arrow.plot(a1=x1, a2=y1, u=cos(sample.direction1), v=sin(sample.direction1), arrow.ex=0.125,
  xpd=TRUE, true.angle=TRUE, length=.1)

## Fit An Appropriate Model
## Code for median polish is contained in Appendix K, Section K.12
FitHoriz1 <- lm(cos(sample.direction1) ~ (x1 + y1))
FitVert1 <- lm(sin(sample.direction1) ~ (x1 + y1))
fitted.direction1 <- atan2(FitVert1$fitted.values, FitHoriz1$fitted.values)
```

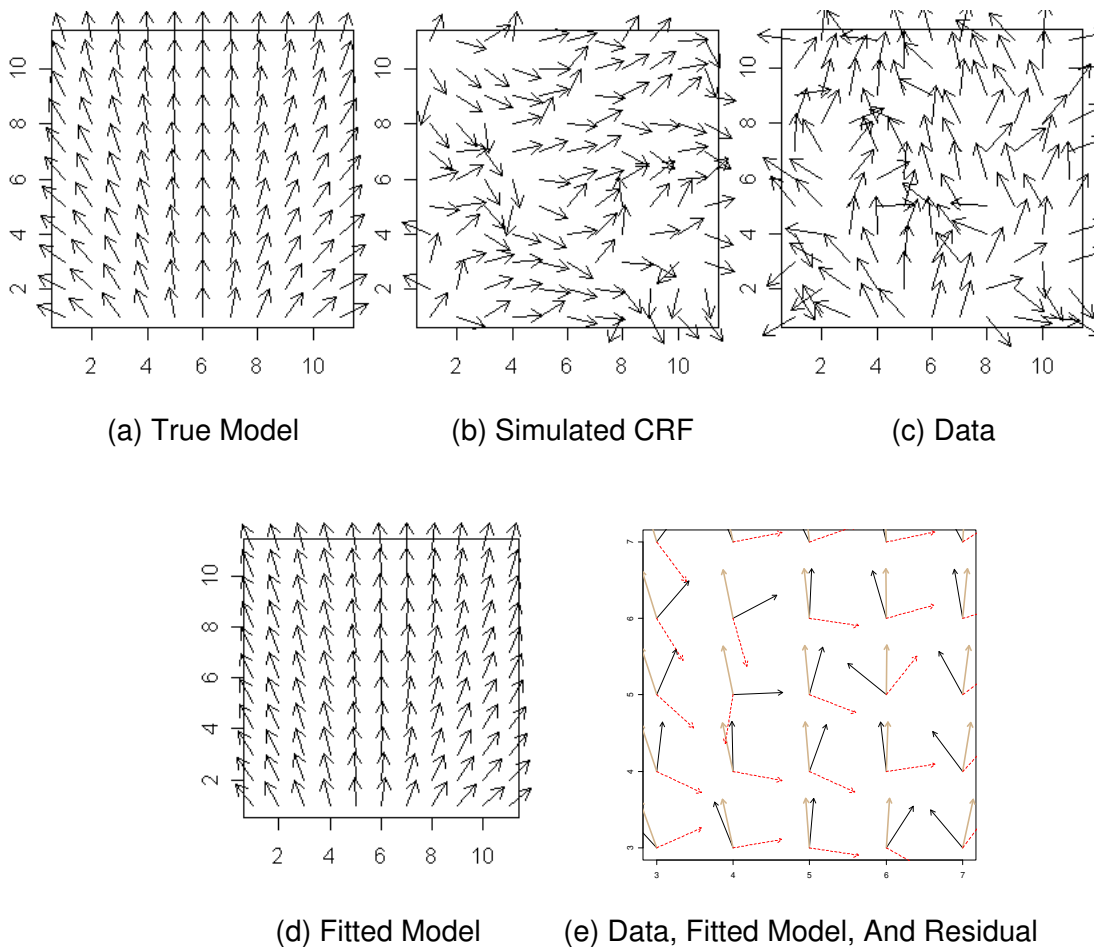


Figure J-3. Plots of True Model, Simulated CRF, Data, Fitted Model, and Residuals. In (e), the data black, the fitted model labeled is tan, and residuals are dashed red. A negative residual rotation = data – model is a clockwise rotation from the model to the data.

```
## Plot Fitted Model, See Figure J-3 (d)
plot(x1, y1, type="n", asp=1, xlab="", ylab="")
arrow.plot(x1, y1, u=cos(fitted.direction1), v=sin(fitted.direction1), arrow.ex=0.1, xpd=TRUE,
           true.angle=TRUE, length=.1)
## The estimated model in Figure J.3 (d) well approximates true model in Figure J.3 (a).

## Compute Residuals
resids1 <- CircResidual(X=x1, Y=y1, Raw=sample.direction1, Trend=fitted.direction1,
                        Plot=FALSE)

## Plot Sample, Fitted Model, and Residual Rotations, See Figure J-3 (e)
CircResidual(X=x1, Y=y1, Raw=sample.direction1, Trend=fitted.direction1, Plot=TRUE,
             xlim=c(3,7), ylim=c(3,7))
```

J.4 CosinePlots

The empirical omnidirectional cosineogram expresses the spatial correlation of an isotropic CRF (spatial correlation depends on distance not direction) as the mean cosine of the angular distances (Figure J-4) between random components of direction as a function of the linear distance between measurement locations.

J.4.1 Definitions

The nugget, range, sill, and cosine model, as shown in Figure J-5, are incorporated into the circular kriging solution. At the distance of zero, the mean cosine is 1. Close to 0 distance, the mean cosine is 1 minus the nugget. As distance increases spatial correlation decreases. This is reflected in a decreasing mean cosine. The range is a scale parameter. For the spherical covariance function, it is the distance at which rotations are no longer spatially dependent. The sill is the mean cosine at distances where CRV are not correlated. Theoretically, the sill is the square of the mean resultant length of the circular probability distribution (See Chapter 3, Section 3.3, for an extensive derivation of this result).

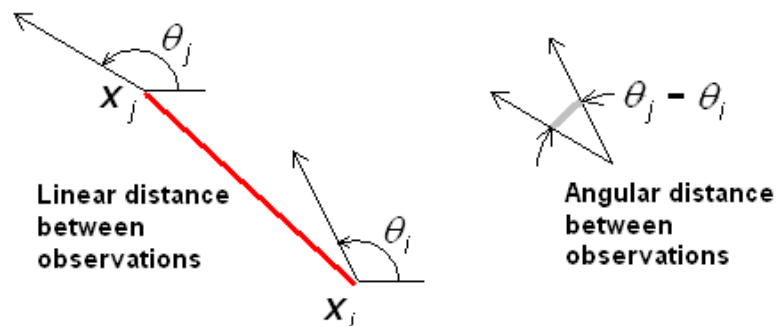


Figure J-4. Distance Between Locations (Red) vs. Angular Distance (Grey) Between Observations.

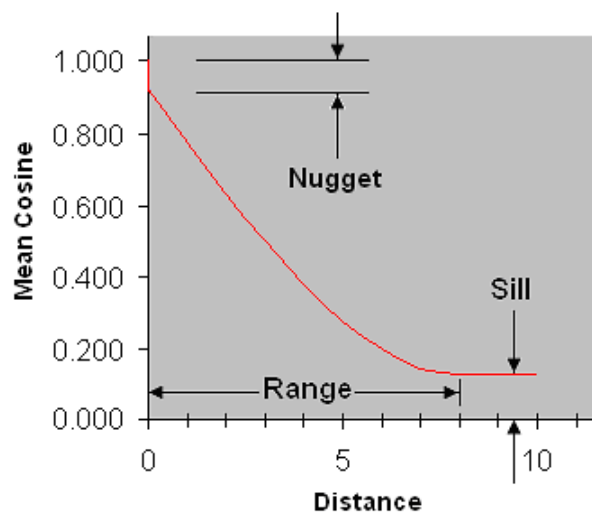


Figure J-5. Features of the Cosineogram Model. The cosineogram characterizes the correlation of random components of direction relative to distance between locations.

The cosineocloud plot, Figure J-6, which derives its name from the variocloud plot for linear kriging, may be useful to identify outliers that may be excluded from subsequent calculations. It shows all the cosine values computed from all pairs of directional observations.

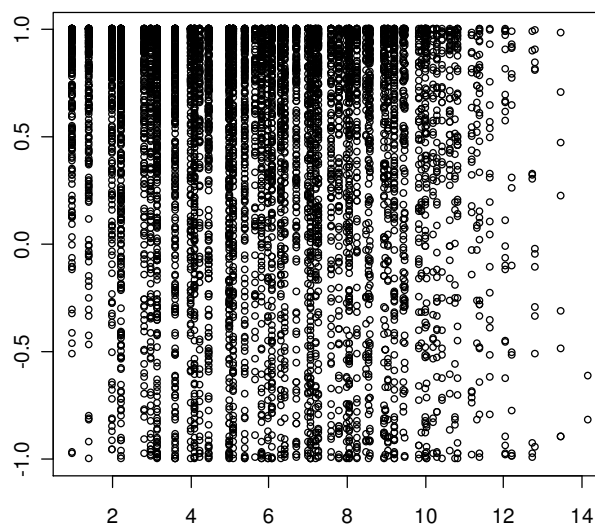


Figure J-6. Cosineocloud. The cosineocloud is a plot of cosines of pairs of random components of direction vs. distance between pairs. It is a diagnostic useful to identify outliers.

The empirical omnidirectional cosineogram captures spatial correlation of directional data or rotational residuals (when the first order trend is removed) as

$$\hat{\zeta}(\text{distance}) = 1/n(\text{distance}) \sum_{i=1}^{n(\text{distance})} \cos(\Delta_i)$$

with Δ_i the circular distance between the i^{th} pair of residual rotations. Typically, range and sill are determined visually. The nugget may be determined by linear regression of initial points of the empirical cosineogram.

Figure J-7 indicates a range of about 4 and a sill of about 0.5, which is consistent with the CRF (CircDistr="vM", Rho=sqrt(0.5), Range=4) with the sill = ρ^2 . The number of lag points and bin width of the cosineogram are described in detail in Subsection J.4.5.

J.4.2 Cosine Models

The empirical cosineogram may be overplotted with exponential, Gaussian, and spherical cosine models to help fit a model. The cosine model is a monotonic decreasing function of increasing distance between measurement locations up to the range, and produces a positive definite matrix of cosines, which is required for circular kriging, when applied to the pairwise distances between measurement locations.

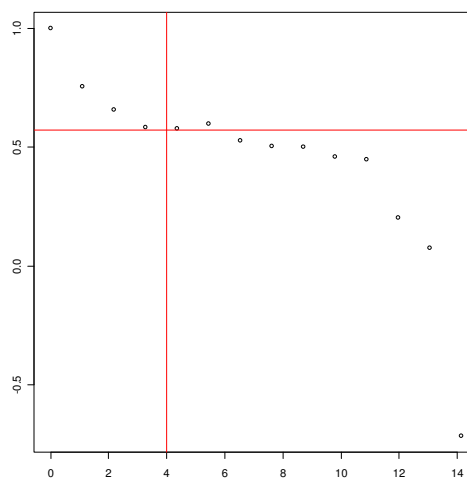


Figure J-7. Empirical Cosineogram. It reflects the circular-spatial correlation with range (distance at which cosine plateaus) of about 4 and sill (elevation of plateau) of about 0.5.

Cosineogram models, which were adapted from covariance models for linear kriging (Chapter 3, Section 3.6), have mean cosine = 1 at distance = 0, and sill = ρ^2 .

With $\zeta(d)$ = mean cosine function of d , d = linear distance between measurement locations, ρ the mean resultant length of the circular probability distribution, n_g the nugget ($0 \leq n_g \leq 1 - \rho^2$), and r = range, the implemented cosine models are:

Exponential (3.12):

$$\zeta(d) = \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2) \exp(-3d/r), & d > 0 \end{cases} \quad (\text{J.1})$$

Gaussian (3.13):

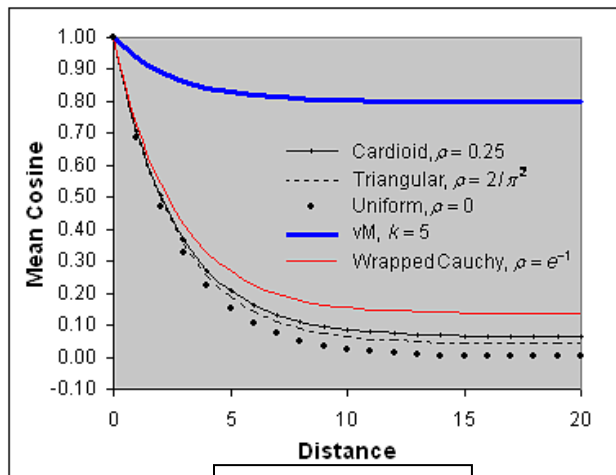
$$\zeta(d) = \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2) \exp(-3[d/r]^2), & d > 0 \end{cases} \quad (\text{J.2})$$

Spherical (3.14):

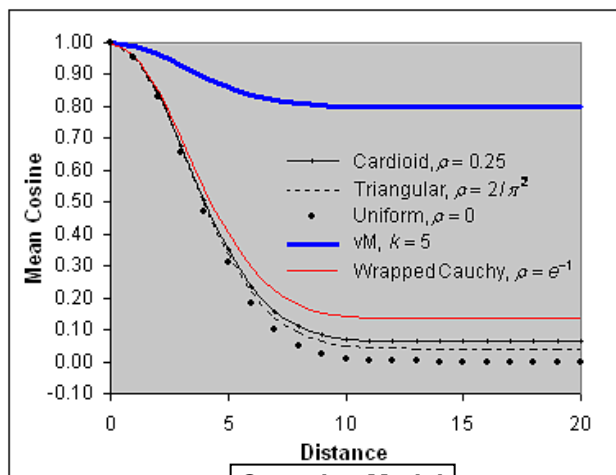
$$\zeta(d) = \begin{cases} 1, & d = 0 \\ 1 - n_g - (1 - n_g - \rho^2) \left(\frac{3}{2} [d/r] - \frac{1}{2} [d/r]^3 \right), & 0 < d \leq r \\ \rho^2, & d > r. \end{cases} \quad (\text{J.3})$$

Plots of the implemented cosine models are shown in Figure J-8. Additional suitable functions are given in Appendix M, Section M.5.

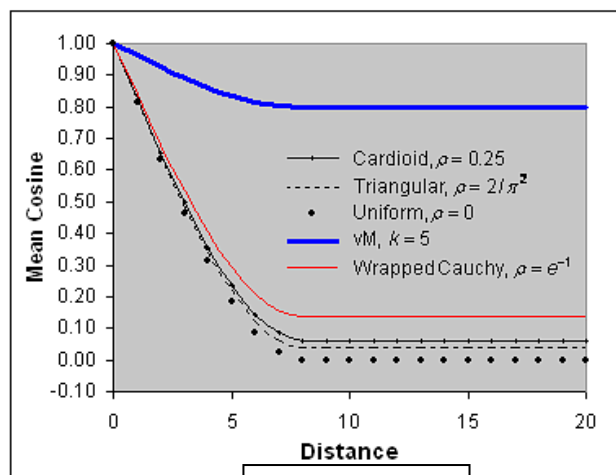
J.4.3 Cosine Model Plots



Exponential Model



Gaussian Model



Spherical Model

Figure J-8. Cosine Models for Circular-Spatial Data, Range $r = 8$.

J.4.4 Usage and Input Arguments

CosinePlots(x, y, directions, Lag=NULL, Lag.n.Adj=1, BinWAdj=1, Plot=TRUE, Cloud=FALSE, Model=FALSE, nugget=0, Range =NULL, sill =NULL, x.legend=0.6, y.legend=1.0, TrimMean=0.1, ...)

x: Vector of horizontal coordinates of observations.

y: Vector of vertical coordinates of observations.

directions: Vector of direction of observations or residual rotations in radians.

Lag: Vector of ascending distances, beginning with zero, where mean cosine is to be computed.

Lag.n.Adj: Multiplier (> 0) of the number of lag points. Value > 1 increases the number of points for more detail. Value < 1 decreases the number of points for less detail.

BinWAdj: Multiplier (≥ 1) of bin width. Value > 1 has a smoothing effect.

Plot: See Table J-1.

Cloud: See Table J-1.

Model: See Table J-1.

nugget: Model nugget or mean cosine near zero distance. $0 \leq \text{nugget} \leq 1$.

Range: Model range.

sill: Model sill.

x.legend: Model plot legend horizontal location as fraction of horizontal maximum coordinate.

y.legend: Model plot legend vertical coordinate.

TrimMean: Apply trimmed mean (0.0 to 0.5) in computing the mean cosine at a distance. See R Help for mean.

. . . : Additional plotting parameters.

J.4.5 Binning Details

The number of lag points (distances at which mean cosine is evaluated) and distance bin width are determined in a sequence:

- 1) Sturges rule determines nBins, the number of bins.
- 2) nBins and Lag.n.Adj determine Lag.n, the number of distances (lag points) to evaluate the mean cosine.
- 3) nBins is adjusted to Lag.n - 1 (To make bins narrower increase Lag.n.adj.).
- 4) nBins and BinWAdj determine bin width.

J.4.6 Output

Typically, range and sill are determined visually. When Plot=Model=TRUE, vary the range and sill parameters to fit. The nugget may be determined by linear regression of initial empirical cosineogram points.

Table J-1. Output of CosinePlots.

Plot	Cloud	Model	Value
FALSE	TRUE	FALSE	List of cosineocloud coordinates
FALSE	FALSE	FALSE	List of cosineogram coordinates
TRUE	TRUE	FALSE	Cosineocloud Plot
TRUE	FALSE	FALSE	Cosineogram Plot
TRUE	FALSE	TRUE	Cosineogram overplotted with cosine models

J.4.7 Examples

```

require(CircSpatial)
## Construct Trend Model of 121 locations
x1<- 1:11; y1 <- 1:11; y1 <- rep(y1, 11); x1 <- rep(x1, each=11)
model.direction1 <- matrix(data=c(
  157, 141, 126, 113, 101, 90, 79, 67, 54, 40, 25, 152, 137, 123, 111, 100, 90, 80, 69, 57, 44, 30,
  147, 133, 120, 109, 99, 90, 81, 71, 60, 48, 35, 142, 129, 117, 107, 98, 90, 82, 73, 63, 52, 40,
  137, 125, 114, 105, 97, 90, 83, 75, 66, 56, 45, 132, 121, 111, 103, 96, 90, 84, 77, 69, 60, 50,
  127, 117, 108, 101, 95, 90, 85, 79, 72, 64, 55, 122, 113, 105, 99, 94, 90, 86, 81, 75, 68, 60,
  117, 109, 102, 97, 93, 90, 87, 83, 78, 72, 65, 112, 105, 99, 95, 92, 90, 88, 85, 81, 76, 70,
  107, 101, 96, 93, 91, 90, 89, 87, 84, 80, 75), ncol=11, byrow=TRUE)
model.direction1 <- as.vector(model.direction1)*pi/180

## Compute vM CRF of 121 observations, Rho=sqrt(0.5) so sill about 0.5,
## from GRF (Range=4, spherical covariance).
set.seed(666)
crf1<- SimulateCRF(CircDistr="vM", Rho=sqrt(0.5), Range=4, CovModel="spherical",
  Grid=cbind(x1, y1), OverFit=TRUE)

# Make sample
sample.direction1 <- model.direction1 + crf1$direction

## Fit An Appropriate Model
## Code for median polish is contained in Appendix K, Section K.12
FitHoriz1 <- lm(cos(sample.direction1) ~ (x1 + y1))
FitVert1 <- lm(sin(sample.direction1) ~ (x1 + y1))
fitted.direction1 <- atan2(FitVert1$fitted.values, FitHoriz1$fitted.values)

## Compute Residuals
resids1 <- CircResidual(X=x1, Y=y1, Raw=sample.direction1, Trend=fitted.direction1,
  Plot=FALSE)

## Output list of cosineogram coordinates for fitting analytically
cosineogram.out <- CosinePlots(x=resids1$x, y=resids1$y, directions=resids1$direction,
  Lag.n.Adj=1, BinWAdj=1, Plot=FALSE, Cloud=FALSE, Model=FALSE)

## Cosineocloud, Figure J-6
CosinePlots(x=resids1$x, y=resids1$y, directions=resids1$direction, Lag.n.Adj=1, BinWAdj=1,
  Plot=TRUE, Cloud=TRUE)

## Cosineogram, Figure J-7.
CosinePlots(x=resids1$x, y=resids1$y, directions=resids1$direction, Lag.n.Adj=1, BinWAdj=1,
  Plot=TRUE, Cloud=FALSE, Model=FALSE)
abline(h=0.56, col=2); abline(v=4, col=2)

## Fit cosine Models, Figure J-9
CosinePlots(x=resids1$x, y=resids1$y, directions=resids1$direction, Lag.n.Adj=1, BinWAdj=1,
  Plot=TRUE, Cloud=FALSE, Model=TRUE, nugget=0, Range=4.0, sill=0.56, x.legend=.2,
  y.legend=0.3, xlim=c(0,8), ylim=c(0,1))

```

green highlighted
code same as
section J.3.3

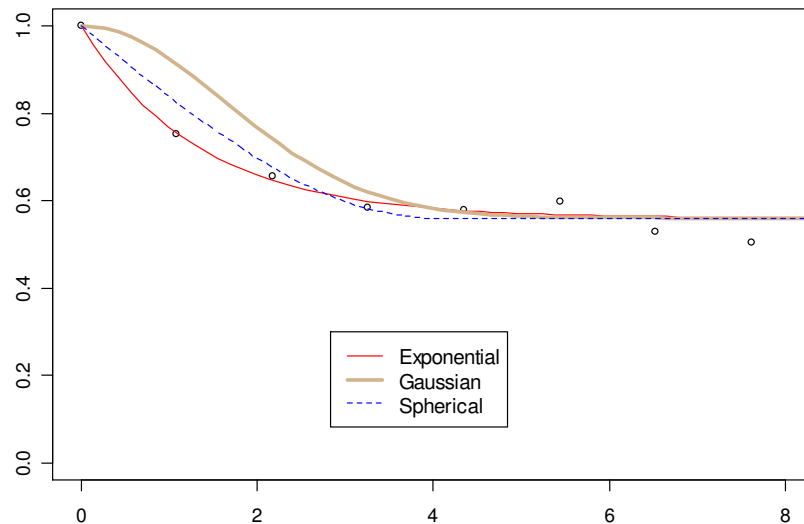


Figure J-9. Fitted Cosine Models. The exponential model with nugget 0.0, range 4.0 and a sill 0.56 has an adequate fit to the data at distances up to the range. The cosine may decrease or increase beyond the range.

J.5 KrigCRF

KrigCRF estimates circular-spatial data of an isotropic CRF with “exact interpolation” (estimate equals observed at measured locations). The solution is a linear combination of directional observations or residual rotations (when the first order trend is removed) that minimizes the squared length of the error vector. This is accomplished by incorporation of the spatial correlation (nugget, range, sill, and cosine model) as estimated by the cosineogram model fitted to the cosineogram. The circular kriging variance is an estimate of the mean squared length of the error vector. To avoid cross over, KrigCRF also separately applies `image.smooth` of R package fields to the horizontal and vertical components of kriged direction. The smoothed kriged direction is quadrant specific inverse tangent of the smoothed components. The circular kriging estimate can be combined with an interpolation of the fitted model. For additional information, see Chapter 4, Circular Kriging.

J.5.1 Solutions

With \mathbf{U} the matrix of unit vector observations, row 1 the cosines, row 2 the sines, and one observation per column, \mathbf{K}^{-1} the inverse of matrix of model cosines dependent on pairwise distances between measurement locations, and \mathbf{c} the vector of model cosines dependent on distance between estimation location 0 and measurement locations 1, 2, 3, ..., n, the solutions are:

$\hat{\mathbf{u}}_0$: Estimated direction as a unit vector, $\hat{\mathbf{u}}_0 = \text{atan2}((\mathbf{UK}^{-1}\mathbf{c})[2], (\mathbf{UK}^{-1}\mathbf{c})[1])$

$\hat{\sigma}_{CK}^2$: Circular kriging error variability, $\hat{\sigma}_{CK}^2 = 2 - 2\sqrt{\mathbf{c}^T \mathbf{K}^{-1} \mathbf{c}}$.

J.5.2 Usage and Input Arguments

KrigCRF(krig.x, krig.y, resid.x, resid.y, resid.direction, Model, Nugget=0, Range, sill, Smooth=FALSE, bandwidth, Plot=FALSE, PlotVar=FALSE, Xlim=NULL, Ylim=NULL, ...)

krig.x: Vector of horizontal coordinates of kriging locations.

krig.y: Vector of vertical coordinates of kriging locations corresponding to krig.x.

resid.x: Vector of horizontal coordinates of rotational residuals or data.

resid.y: Vector of vertical coordinates of rotational residuals or data.

resid.direction: Vector of direction in radians of rotational residuals or data.

Model: Covariance model of R package RandomFields function CovarianceFct best fitting the empirical cosineogram.

Nugget: 1 - mean cosine at distance close to zero due to measurement error, micro scale variation or sampling.

Range: Distance at which spatially correlated CRV are not correlated for the spherical model, or scale factor of other models.

sill: Mean cosine at the Range.

Plot: See Table J-2. Xlim and Ylim are the plot limits.

PlotVar: Plot circular kriging variance. See Table J-2.

Smooth: See Table J-2.

bandwidth: Kernel smoothing bandwidth (>0).

... : Additional model parameters.

J.5.3 Output

Table J-2. Output of KrigCRF.

Plot	PlotVar	Smooth	Value
FALSE	TRUE/FALSE	FALSE	Vectors of x, y, and kriged direction (rad)
FALSE	TRUE/FALSE	TRUE	Vectors of x, y, and smoothed direction
TRUE	FALSE	FALSE	Arrow plot of kriged direction
TRUE	FALSE	TRUE	Arrow plot of smoothed kriged direction
TRUE	TRUE	TRUE/FALSE	Filled contour plot of circular kriging error

J.5.4 Examples

```
require(CircSpatial)
## Using the residuals resid1 from Subsection J.4.7
x2 <- seq(1,11, by=0.2); y2 <- x2 ## Kriging locations

## Krig to residuals using range and sill estimate from Figure J-9
krig2 <- KrigCRF(krig.x=x2, krig.y=y2, resid.x=resid1$x, resid.y=resid1$y,
  resid.direction=resid1$direction, Model="exponential", Nugget=0.0, Range=4, sill=0.56,
  Plot=FALSE)

## Plot Kriging, Residuals Overplotted In Black, See Figure J-10
require(fields)
plot(krig2$x, krig2$y, ty="n", xlab="", ylab="", xlim=c(5, 8), ylim=c(5, 8), asp=1)
arrow.plot(krig2$x, krig2$y, u = cos(krig2$direction), v = sin(krig2$direction), arrow.ex = 0.06,
  xpd=FALSE, true.angle = TRUE, length=.05, col="tan")
arrow.plot(resid1$x, resid1$y, u = cos(resid1$direction), v = sin(resid1$direction), arrow.ex =
  0.09, xpd=FALSE, true.angle = TRUE, length=.05, col=1)
```

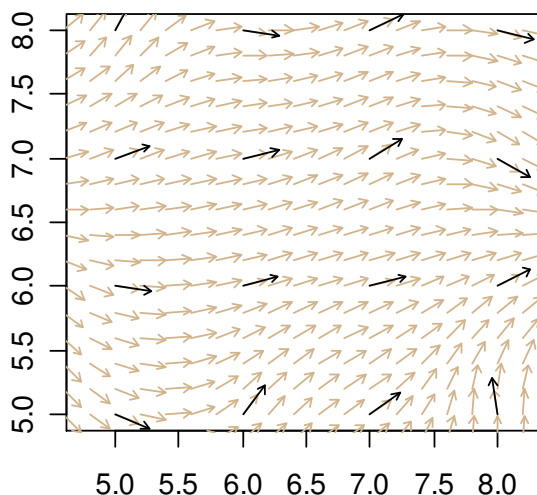


Figure J-10. Residual Rotations (Black) Overplotted on the Circular Kriging (Tan). The kriged direction appears to be equal to the residual direction at sample locations. This is called “exact interpolation”.

Figure J-11 shows that increasing the nugget smooths the kriging component of estimated direction at unsampled locations. The kriging estimate at sample locations is “exact”. At the maximum nugget, the data are uncorrelated, the unsampled locations are uncorrelated to the sample locations, and the kriging component of estimated direction at unsampled locations is zero. The kriging component is added to the fitted model direction. At uncorrelated locations, the estimated direction is the fitted direction.

```
KrigCRF(krig.x = x2, krig.y = y2, resid.x= resid1$x, resid.y= resid1$y, resid.direction=
  resid1$direction, Model="exponential", Nugget=0.0, Range=4, sill=0.56, Plot=TRUE,
  Xlim=c(7,10), Ylim=c(7,10))
# Repeat with Nugget = 0.15 and 0.3
KrigCRF(krig.x = x2, krig.y = y2, resid.x=resid1$x, resid.y=resid1$y, resid.direction=
  resid1$direction, Model="exponential", Nugget=0.44, Range=4, sill=0.56, Plot=TRUE,
  Xlim=c(7,10), Ylim=c(7,10))
arrow.plot(resid1$x, resid1$y, u =cos(resid1$direction), v = sin(resid1$direction), arrow.ex =
  0.09, xpd = F, true.angle = T, length=.05, col=2)
```

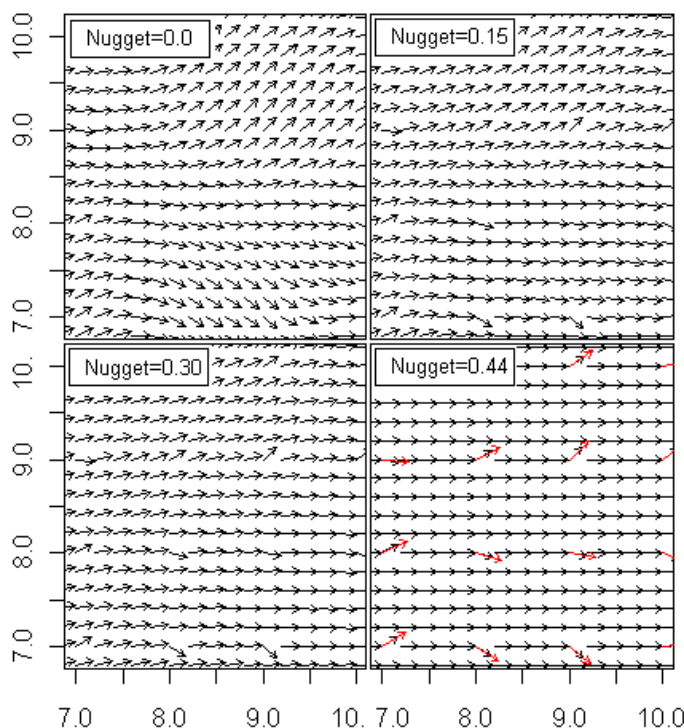


Figure J-11. Smoothing via the Nugget Not Effective at Data Locations. Red out of line arrows in the lower right panel are the residual rotations. This smoothing method does not affect estimates at data locations.

Smoothing applied to the kriging component of estimated direction smoothes

direction over all points. See Figure J-12.

```
KrigCRF(krig.x = x2, krig.y = y2, resid.x= resid1$x, resid.y= resid1$y, resid.direction=
  resid1$direction, Model="exponential", Nugget=0.0, Range=4, sill=0.56, Plot=TRUE,
  Xlim=c(7,10), Ylim=c(7,10), Smooth=TRUE, bandwidth=0.1)
KrigCRF(krig.x = x2, krig.y = y2, resid.x= resid1$x, resid.y= resid1$y, resid.direction=
  resid1$direction, Model="exponential", Nugget=0.0, Range=4, sill=0.56, Plot=TRUE,
  Xlim=c(7,10), Ylim=c(7,10), Smooth=TRUE, bandwidth=2)
KrigCRF(krig.x = x2, krig.y = y2, resid.x= resid1$x, resid.y= resid1$y, resid.direction=
  resid1$direction, Model="exponential", Nugget=0.0, Range=4, sill=0.56, Plot=TRUE,
  Xlim=c(7,10), Ylim=c(7,10), Smooth=TRUE, bandwidth=4)
KrigCRF(krig.x = x2, krig.y = y2, resid.x=resid1$x, resid.y=resid1$y, resid.direction=
  resid1$direction, Model="exponential", Nugget=0.0, Range=4, sill=0.56, Plot=TRUE,
  Xlim=c(7,10), Ylim=c(7,10), Smooth=TRUE, bandwidth=10)
```

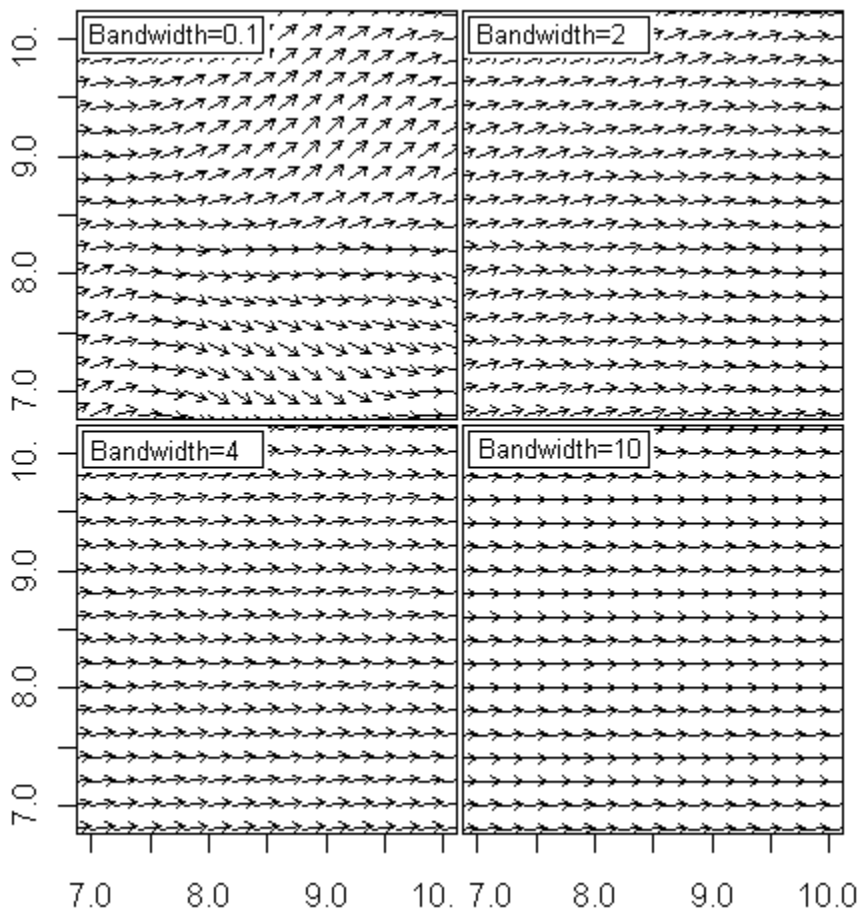


Figure J-12. Smoothing the Kriging Components is Effective at All Locations.

```

## Plot kriging estimate variability at sample locations on a regular grid, See Figure J-13
KrigCRF(krig.x = x2, krig.y = y2, resid.x= resid1$x, resid.y= resid1$y, resid.direction=
  resid1$direction, Model="exponential", Nugget=0.0, Range=4, sill=0.56, Plot=TRUE,
  Smooth=FALSE, PlotVar=TRUE)

## Plot kriging estimate variability with random sample locations, See Figure J-14.
set.seed(13)
crf6 <- SimulateCRF(N=400, CircDistr="Card", Rho= 0.4, Range=4, Ext=3,
  CovModel="spherical")
## Best fit is spherical with range=2.85 and sill=0.15
CosinePlots(x=crf6$x, y=crf6$y, directions=crf6$direction, Lag.n.Adj=1.5, BinWAdj=1,
  Plot=TRUE, Cloud=FALSE, Model=TRUE, nugget=0, Range=2.85, sill=0.15, x.legend=.14,
  y.legend=0.75, xlim=c(0,6), ylim=c(0,1))
x6 <- seq(4,7, by=0.02); y6 <- x6
# This may take a long time
KrigCRF(krig.x =x6, krig.y =y6, resid.x=crf6$x, resid.y=crf6$y, resid.direction=crf6$direction,
  Model="spherical", Nugget=0.0, Range=2.85, sill=0.15, Plot=TRUE, PlotVar=TRUE)

```

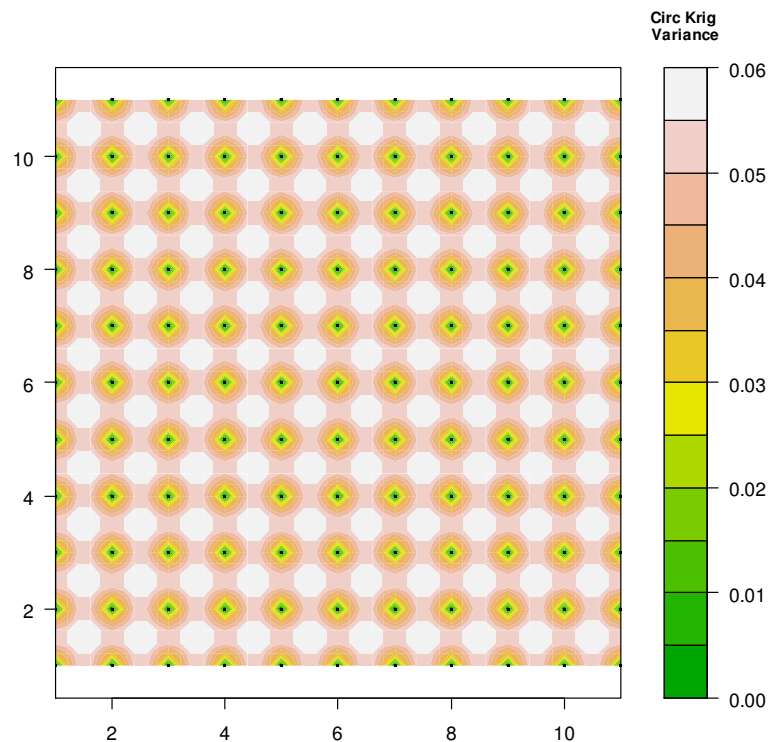


Figure J-13. Variability of the Circular Kriging Estimate with Locations on a Regular Grid. Measurement locations are indicated by black points. Estimate variability is zero at a sample location. $0 \leq \sigma_{CK}^2 < 4$ (Chapter 4, Section 4.6).

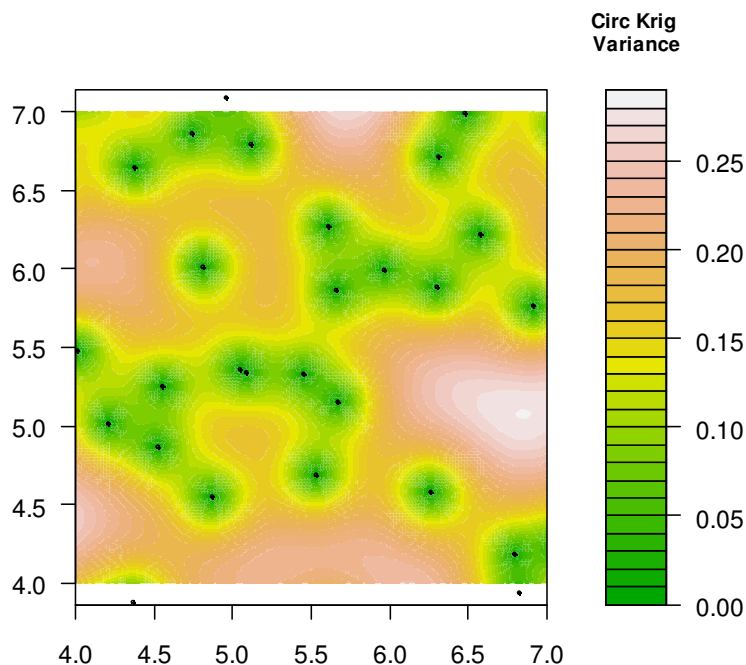


Figure J-14. Variability of the Circular Kriging Estimate with Random Locations. Measurement locations are indicated by black points. Estimate variability is zero at a sample location. $0 \leq \sigma_{CK}^2 < 4$ (Chapter 4, Section 4.6).

J.6 InterpDirection

The interpolated model direction is added to the kriged, or smoothed kriged residuals, to obtain the estimated direction. To avoid cross over, the cosine and the sine of the model direction are separately interpolated. The algorithm has 6 cases of interpolation location as indicated by the labeled red dots in Figure J-15. The corners of the grey rectangle are observation locations with observations indicated by the blue unit vectors. For example, assume the interpolation location falls in the lower triangle with label f. For the interpolation of the cosine, a plane is fitted to the three points $\{(x_1, y_1, \cos(\theta_1)), (x_2, y_2, \cos(\theta_2)), (x_3, y_3, \cos(\theta_3))\}$. The interpolated value is the elevation of the plane at the interpolation location. The inverse tangent is applied to the interpolated cosine and sine components. Figure J-16 shows the result of interpolating smoothed average wind direction. Interpolation outside the model gives an error.

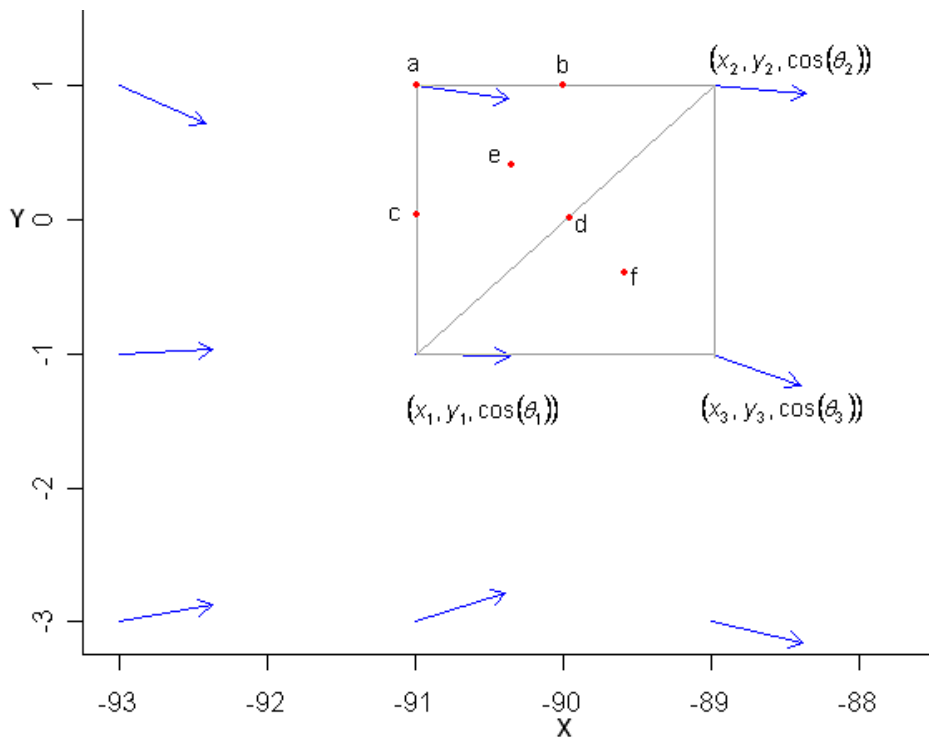


Figure J-15. Six Cases of Interpolation Location Indicated by Labeled Red Dots. The corners of the grey rectangle are observation locations. Planes are fitted to the triangular partitions. The interpolated component, cosine or sine, is the elevation of the plane at the location of interpolation.

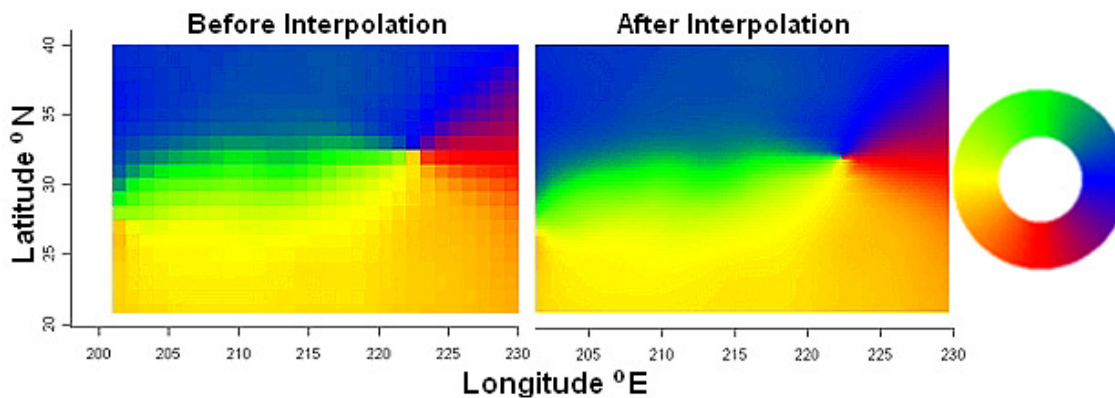


Figure J-16. Effect of Interpolation on Smoothed Average Wind Direction with BGYR Color Wheel.

J.6.1 Usage and Input Arguments

InterpDirection(in.x, in.y, in.direction, out.x, out.y)

in.x: Vector of input horizontal coordinates.

in.y: Vector of input vertical coordinates.

in.direction: Vector of input direction in radians.

out.x: Vector of output horizontal coordinates.

out.y: Vector of output vertical coordinates.

J.6.2 Output List

x: out.x.

y: out.y.

direction: Vector of interpolated direction in radians.

green highlighted
code matches
previous sections

J.6.3 Examples

```
require(CircSpatial)
## Construct Trend Model of 121 locations
x1<- 1:11; y1 <- 1:11; y1 <- rep(y1, 11); x1 <- rep(x1, each=11)
model.direction1 <- matrix(data=c(
  157, 141, 126, 113, 101, 90, 79, 67, 54, 40, 25, 152, 137, 123, 111, 100, 90, 80, 69, 57, 44, 30,
  147, 133, 120, 109, 99, 90, 81, 71, 60, 48, 35, 142, 129, 117, 107, 98, 90, 82, 73, 63, 52, 40,
  137, 125, 114, 105, 97, 90, 83, 75, 66, 56, 45, 132, 121, 111, 103, 96, 90, 84, 77, 69, 60, 50,
  127, 117, 108, 101, 95, 90, 85, 79, 72, 64, 55, 122, 113, 105, 99, 94, 90, 86, 81, 75, 68, 60,
  117, 109, 102, 97, 93, 90, 87, 83, 78, 72, 65, 112, 105, 99, 95, 92, 90, 88, 85, 81, 76, 70,
  107, 101, 96, 93, 91, 90, 89, 87, 84, 80, 75), ncol=11, byrow=TRUE)
model.direction1 <- as.vector(model.direction1)*pi/180

## Compute vM CRF of 121 observations, Rho=sqrt(0.5) so sill about 0.5,
## from GRF (Range=4, spherical covariance).
set.seed(666)
crf1<- SimulateCRF(CircDistr="vM", Rho=sqrt(0.5), Range=4, CovModel="spherical",
  Grid=cbind(x1, y1), OverFit=TRUE)

# Make sample
sample.direction1 <- model.direction1 + crf1$direction

## Fit An Appropriate Model
## Code for median polish is contained in Appendix K, Section K.12
FitHoriz1 <- lm(cos(sample.direction1) ~ (x1 + y1))
FitVert1 <- lm(sin(sample.direction1) ~ (x1 + y1))
fitted.direction1 <- atan2(FitVert1$fitted.values, FitHoriz1$fitted.values)
```

```

## Compute Residuals
resids1 <- CircResidual(X=x1, Y=y1, Raw=sample.direction1, Trend=fitted.direction1,
  Plot=FALSE)

## Fit cosine Models
CosinePlots(x=resids1$x, y=resids1$y, directions=resids1$direction, Lag.n.Adj=1, BinWAdj=1,
  Plot=TRUE, Cloud=FALSE, Model=TRUE, nugget=0, Range=4.0, sill=0.56, x.legend=0.2,
  y.legend=0.4, xlim=c(0,8), ylim=c(0,1))

## Krig to residuals using cosine Model (Figure J-9)
x2 <- seq(1,11, by=0.2); n <- length(x2); y2 <- x2; y2 <- rep(y2, n); x2 <- rep(x2, each=n); rm(n)
krig2 <- KrigCRF(krig.x=x2, krig.y=y2, resid.x=resids1$x, resid.y=resids1$y, resid.direction=
  resids1$direction, Model="exponential", Nugget=0.0, Range=4, sill=0.56, Plot=FALSE)

## Interpolate Fitted Model
interp2 <- InterpDirection(in.x=x1, in.y=y1, in.direction=fitted.direction1, out.x=krig2$x,
  out.y=krig2$y)

## Plot Interpolated Fitted Model and Overplot Fitted Model. See Figure J-17.
plot(interp2$x, interp2$y, type="n", asp=1, xlim=c(5,8), ylim=c(5,8), xlab="", ylab="")
arrow.plot(interp2$x, interp2$y, u=cos(interp2$direction), v=sin(interp2$direction), arrow.ex=0.09,
  xpd=FALSE, true.angle=TRUE, length=.1, col="tan")
arrow.plot(x1, y1, u =cos(fitted.direction1), v=sin(fitted.direction1), arrow.ex=0.06, xpd=FALSE,
  true.angle=TRUE, length=.1, col=1)

```

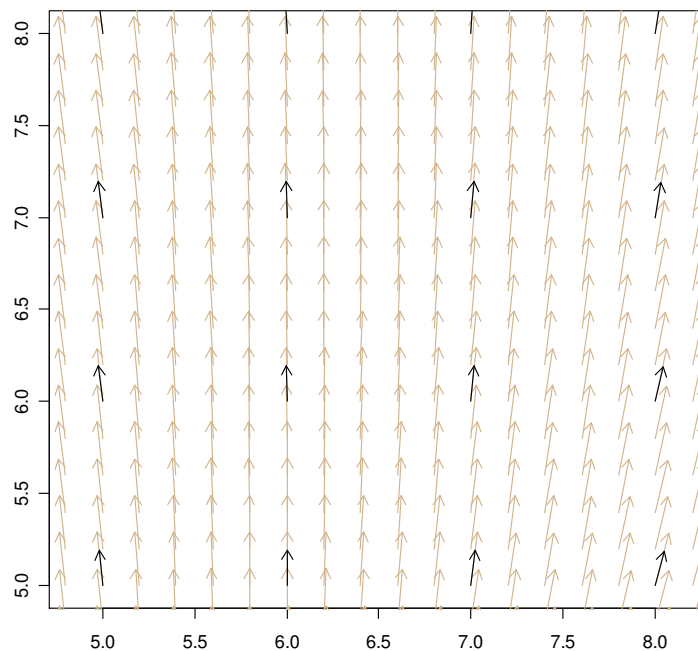


Figure J-17. Fitted Model (Black) Overplotted on the Fitted Model Interpolation (Tan). The fitted model is an enlargement of Figure J-3 (d). The interpolated direction is the same as the model direction at model locations, i.e. the interpolation is “exact”.

```

## Plot Estimate Of Direction And Overplot Sample. See Figure J-18.
estimate2=interp2$direction + krig2$direction
plot(interp2$x, interp2$y, type="n", xlab="", ylab="", asp=1)
arrow.plot(interp2$x, interp2$y, u=cos(estimate2), v= sin(estimate2), arrow.ex=0.05, xpd=FALSE,
  true.angle=TRUE, length=.05, col="tan")
arrow.plot(x1, y1, u=cos(sample.direction1), v=sin(sample.direction1), arrow.ex=0.05,
  xpd=FALSE, true.angle=TRUE, length=.05, col=1)

## Zoom. See Figure J-19.
plot(interp2$x, interp2$y, type="n", xlab="", ylab="", asp=1, xlim=c(3,6), ylim=c(3,6))
arrow.plot(interp2$x, interp2$y, u=cos(estimate2), v=sin(estimate2), arrow.ex=0.075,
  xpd=FALSE, true.angle=TRUE, length=.05, col="tan")
arrow.plot(x1, y1, u=cos(sample.direction1), v=sin(sample.direction1), arrow.ex=0.05,
  xpd=FALSE, true.angle=TRUE, length=.05, col=1)

```

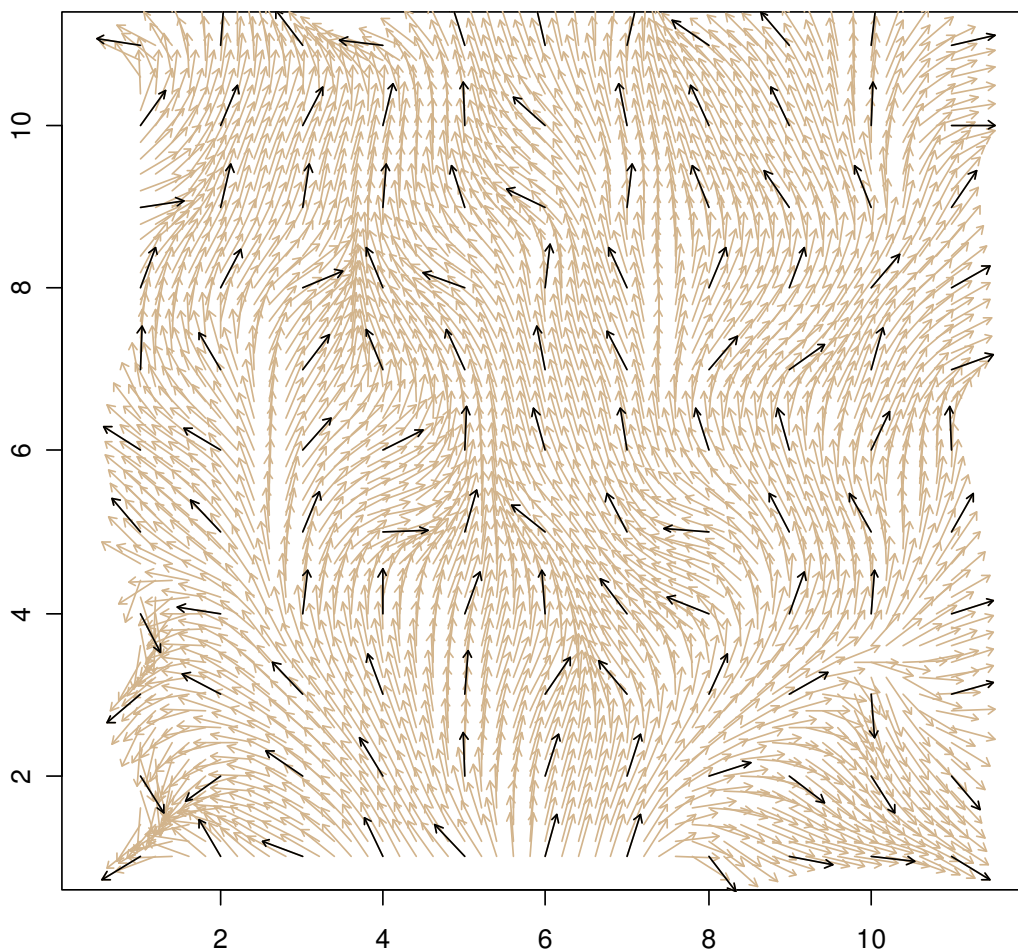


Figure J-18. Original Data (Black) Overplotted on the Estimates (Tan).

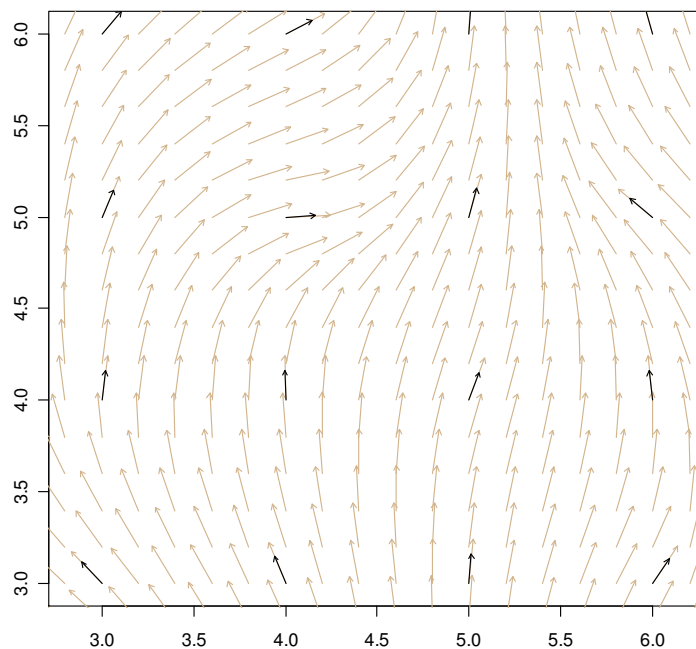


Figure J-19. Enlargement of Figure J-18. The estimate (\tan) equals the sample direction (black) at sample locations. This is called “exact interpolation”.

J.7 TestPattern

The function `TestPattern` (Appendix K, Section K.1) makes an intuitive simple test pattern to explore the function `CircDataimage`, which produces a GUI for interactive imaging of circular-spatial data. `TestPattern` computes direction such that the direction at any point is the angle between the line from origin to point and the horizon.

J.7.1 Usage

```
testpattern <- TestPattern()
```

J.7.2 Output Dataframe

x: Vector of horizontal location coordinates.

y: Vector of vertical location coordinates.

u: Vector of horizontal component of cosines of direction.

v: Vector of vertical component of sines of direction.

J.8 OceanWind

OceanWind provides a large dataset (495,688 observations) to further explore the function `CircDataimage`. The OceanWind data was freely obtained from ICOADS at <http://dss.ucar.edu/datasets/ds540.1/data/msga.form.html> for the El Nino years 1972, 1976, 1982, 1987, 1991, 1994, and 1997, January through April, in 1° increments for the area of longitude 0.5° E to +359.5° E by latitude -59.5° N to +60.5° N. See Chapter 2, Subsection 2.2.1.

J.8.1 Usage

```
data(OceanWind)
```

J.8.2 Dataframe

year: Vector of time of observation = year + month/12, month in [1972.083, 1997.333]

x: Vector of longitude.

y: Vector of latitude.

u: Vector of east component of wind (0.01 meters/second).

v: Vector of north component of wind (0.01 meters/second).

J.9 WorldMask

WorldMask is used by function `CircDataimage` to restore land contours to the circular dataimage of smoothed OceanWind. WorldMask was derived from the R package `fields` dataset `world.dat`.

J.9.1 Usage

```
data(WorldMask)
```

J.9.2 Value

WorldMask is a matrix of 360 rows (0.5° to 359.5° longitude) x 121 columns (-59.5° to $+60.5^\circ$ latitude) suited to OceanWind, with elements NA where wind data is not missing and 1 where wind data is missing. Figure J-20 is an image plot of WorldMask.

J.10 CircDataimage

CircDataimage generates a GUI for interaction with circular dataimages to facilitate the discovery of structure in circular-spatial data.

J.10.1 Introduction

The circular dataimage is useful for visualization of random, model, and kriged circular-spatial data. The circular dataimage is constructed by displaying direction as the color in a color wheel at the same angle. This implementation of color provides for simultaneous recognition of fine circular-spatial structure on a small scale and large-scale circular-spatial structure on a global scale (Figure J-21 b, d, f). Depending on data density and smoothness, this structure is lost traditional arrow plots (Figure J-21 a, c).



Figure J-20. Image Plot of WorldMask. Land masses are tan.

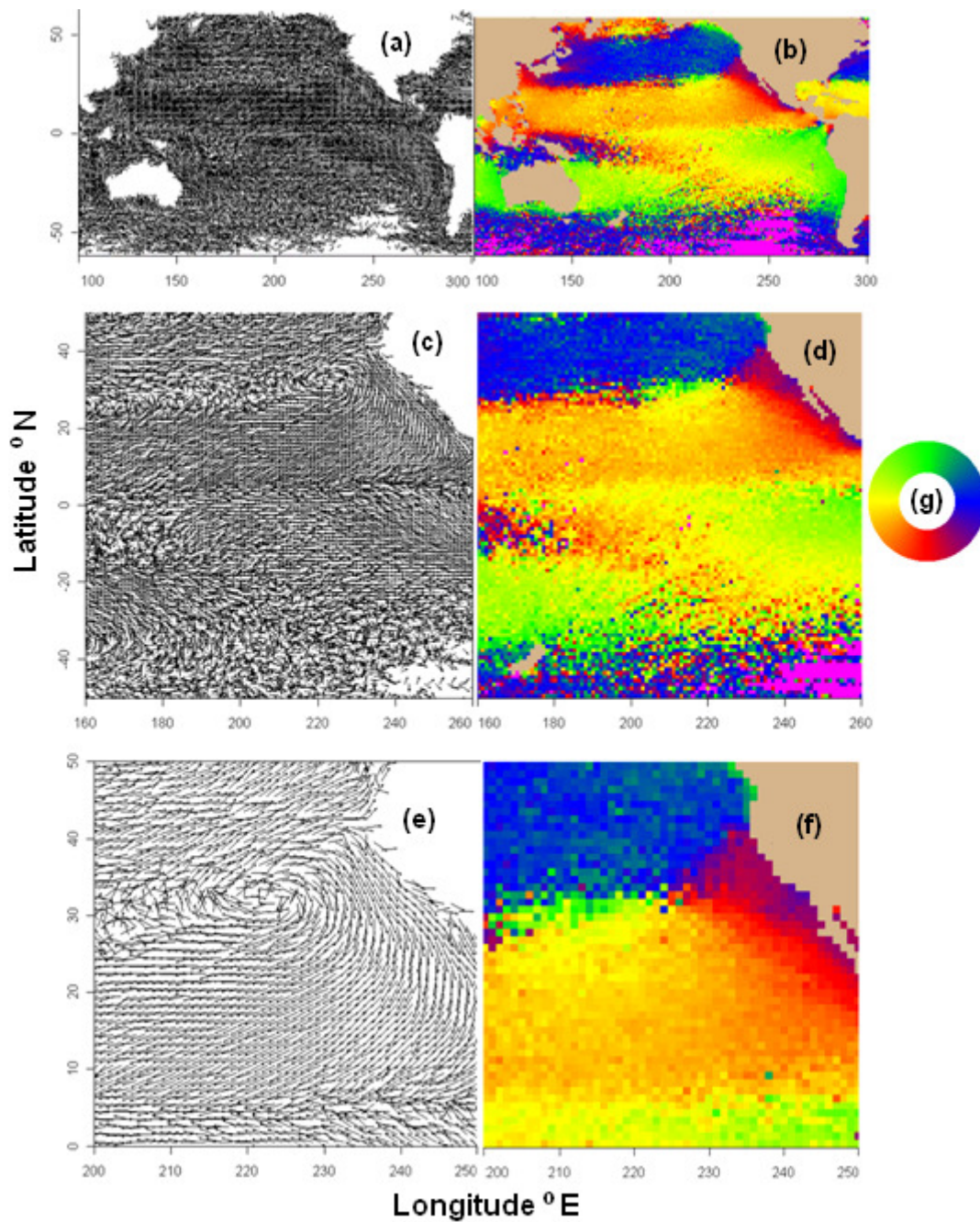


Figure J-21. Comparison of Arrow and Circular Dataimage Plots of Ocean Wind Average Direction. Plots (a), (b) cover 200° of longitude; (c), (d) cover 100° Of longitude; (e), (f) cover 50° of longitude; (g) BGYR Circular Color Wheel. Ability to recognize structure depends on plot type, smoothness and density of data and arrows, and distribution of missing data.

J.10.2 The Color Wheel

The color wheel evolved as shown in Figure J-22. In Figure J-22 (d) every point on the color wheel is color continuous, every direction is represented by a unique color, and the color change is linear. The resulting color wheel is named a Yellow-Red-Green-Blue (YRGB) Color Wheel. In general, the color wheel is composed of a sequence of color gradients with continuity between connecting color gradients such that the color coding the 0° direction is the same as the color coding the 360° direction. Thus, image discontinuity from using a single color gradient for visualization of circular-spatial data is eliminated. Package provides 7 continuous color wheels with 1 for red-green color impaired, 1 Hue Saturation Value (HSV), and 2 divergent, and 6 discrete color wheels with 3 divergent. For additional information, see Chapter 2, Circular Dataimage, A High Resolution Image Of Circular-Spatial Data.

J.10.3 Input Requirements

The input data.frame entered into the GUI Input Dataframe entry box contains:

x: Vector of measurement location horizontal coordinates on regular grid.

y: Vector of measurement location vertical coordinates on regular grid.

u: Vector of horizontal component of measured vector. NA is not allowed.

v: Vector of vertical component of measured vector. NA is not allowed.

Note: The input data are measured on a regular grid. The vertical and horizontal grid spacing may be different. Spacing is computed from the data as the second smallest location coordinate minus minimum location coordinate. Multiple observations at the same location will be replaced with the mean resultant = vector resultant/number of observations. Missing data are permitted, but a data.frame with rows of (x, y, NA, NA) are not allowed. u=v=0 OK.

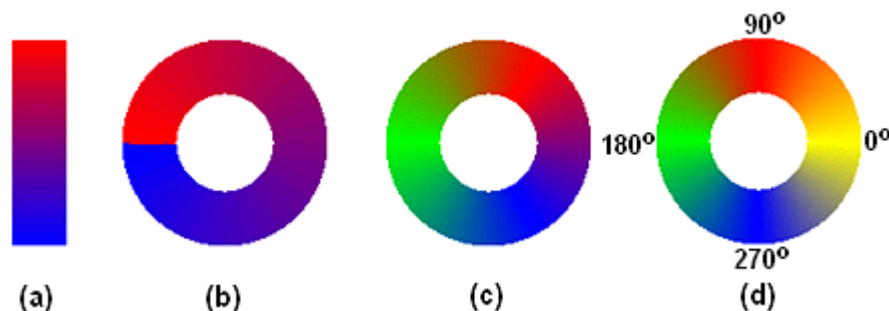


Figure J-22. Evolution of the YRGB Color Wheel. (a) Blue-red linear color scale, (b) Color scale (a) wrapped on circle, (c) red-green-blue linear color gradient inserted at 180°, and (d) blue-yellow-red gradient inserted at 0°. The YRGB color wheel (d) aligns the 4 main colors to the 4 main directions (0, 90, 180, 270°).

The mask matrix entered into the GUI Mask Matrix entry box is optional (See J.9 WorldMask). It must have rows and columns equal to the rows and columns of the measurement grid, respectively. Mask cell values are NA where pixel is not to be masked and 1 where a pixel is to be masked.

J.10.4 Output

The list object Globals, which contains the GUI inputs and results of computations, is written into the R workspace.

J.10.5 Startup Example

```
## Consider setting the R GUI preference to SDI
require(CircSpatial)
data(OceanWind)
data(WorldMask)
wind.subset <- OceanWind[, -1] # Using all the data, about 500k records
CircDataimage()
```

J.10.6 GUI Demonstration Using Ocean Wind Data

The initial display appears as in Figure J-23. “unknown” or an empty Input Dataframe (bolding indicates a GUI element in the referenced figure) entry box or Mask Matrix entry box indicate the absence of input.

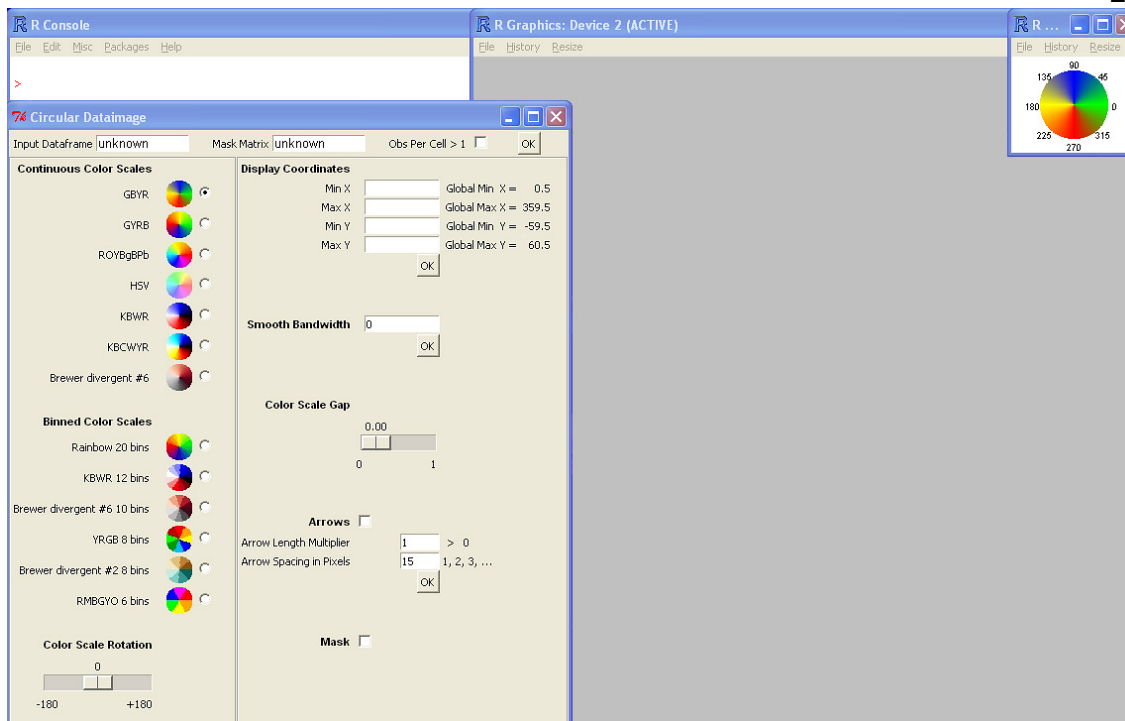


Figure J-23. Initial Display of the GUI, the Circular Dataimage Window (R Graphics Device 2), and the Color Wheel Window.

- 1 Enter the name of the data.frame wind.subset in the Input Dataframe entry box.
Enter the example optional mask matrix WorldMask in the Mask Matrix entry box.
N.B.: The input data.frame and optional mask must be input before other GUI controls will function correctly. After data is input, the controls may be operated in any order.
- 2 Check the Obs Per Cell > 1 checkbox because wind.subset has more than one observation per location. When the observations per cell ≤ 1 , the looping structure is replaced with vector-matrix expression to reduce computation time.

- 3 To accept the input data and optional mask, click the OK button at the right of the mask entry box. The OK button will remain depressed until processing is complete. If the Obs Per Cell > 1 checkbox is checked, the message “The computations necessarily may take significant time” is sent to the R GUI. The computation of average wind direction over the 500k observations of wind.subset is relatively slow. However, testpattern (J.2) is imaged instantaneously (Obs Per Cell > 1 checkbox is unchecked).
- 4 The display now appears as in Figure J-24. The circular dataimage appears in R Graphics: Device 2 and the current color wheel in R Graphics: Device 3. The Display Coordinates and reference global (extreme) coordinates to the right are updated. The Smooth Bandwidth is set to 0, and the Arrows and Mask check buttons are unchecked. Arrange the windows as necessary.

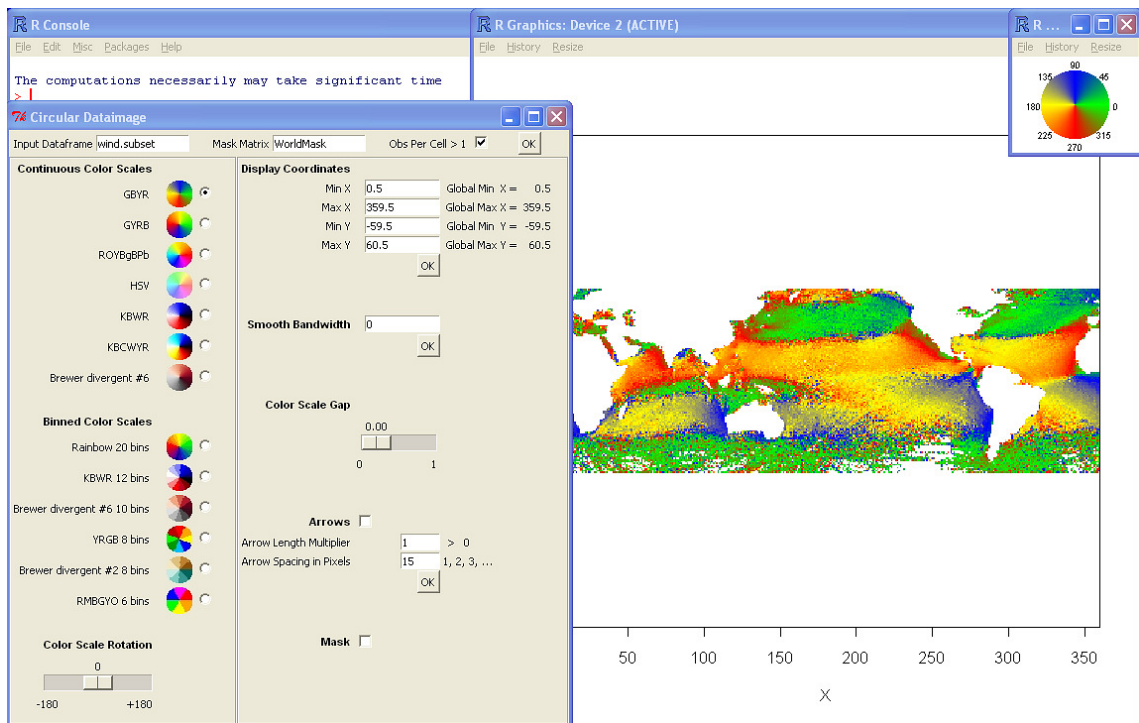


Figure J-24. Display with Circular Dataimage of Average Direction after Inputs Entered.

- 5 Select any color wheel scale by clicking the radio button next to the color wheel label and graphic, and the circular dataimage and color wheel (R Graphics: Device 3) are quickly updated.
- 6 Rotation is used to highlight structure in the displayed area of interest. The selected color wheel may be varied by moving the Color Scale Rotation slider. As the slider is moved, the label above the slider updates. When the mouse button (left) is released, the displayed color wheel rotation angle is processed with a positive value resulting in a counterclockwise rotation of the color wheel by the displayed value. The circular dataimage and color wheel are updated, but the data are unchanged.
- 7 Coordinates are entered to pan and zoom. Display coordinates reference the center of a pixel. Enter display coordinates into the Display Coordinates entry box. The user may enter any value, but entry box coordinates will snap to the coordinates of the nearest datum. Thus, coordinates outside the range of data are adjusted to the extremes of the data. A zoomed display may be reduced to as few as 2*2 pixels. 2*1, 1*1, and 1*2 pixel displays result in nonmatrix input to the image plot function and an image plot error message. Enter Min X=100 and Max X=275. Click OK to process the entered coordinates. The displayed coordinates in the entry boxes snap to Min X = 99.5 and Max X=274.5 because the input data.frame has longitude from 0.5° to 359.5° in 1° increments.
- 8 Smoothing is useful to reduce visual noise. Smoothing applies image.smooth of package fields separately to the horizontal and vertical components of the mean vector resultant to avoid “cross over”. The smoothed direction is obtained by applying the R function atan2 to the smoothed components. Enter a smooth bandwidth in the Smooth Bandwidth entry box and click OK. The OK button will remain depressed until processing is complete.

- 9 The result of selecting the Green-Yellow-Red-Blue (GYRB) color wheel with a $+90^\circ$ rotation (blue in the color wheel has rotated 90° CCW), and entering a smooth bandwidth=2.5 $^\circ$ longitude and latitude is shown in Figure J-25. Choice of bandwidth is determined by experimentation on the data.
- 10 Color scale gap increases the contrast at the boundaries of areas of similar direction (in the same color gradient component of the color wheel) by truncating the upper fraction of the gradient (stretching out the lower fraction or retarding the terminal color). For example, a 4 color-gradient, e.g., GBYR with Color Scale Gap=1 codes all directions in the range 0° to 90° as pure green. Move the slider of Color Scale Gap and the label above the slider updates. When the mouse button is released, the displayed color wheel gap is processed, and the color wheel and circular dataimage are updated.
- 11 Now select the HSV color wheel.

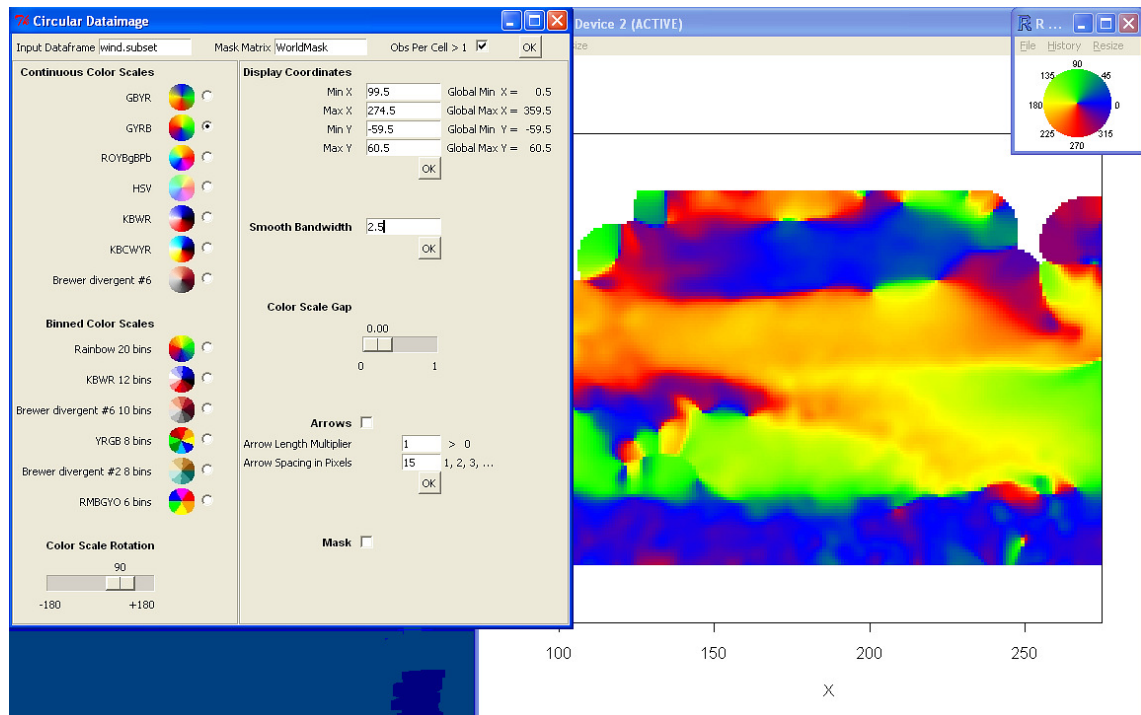


Figure J-25. GYRB Color Wheel Rotated 90° , Data Smoothed with Bandwidth 2.5, and Display Coordinates Changed (Zoomed).

- 12 To add arrows check the Arrows check box. An arrow is plotted if the value of direction at the arrow location is not NA, or the NA value is replaced by smoothing and the location is not a masked. Arrow Spacing in Pixels is the horizontal and vertical spacing between arrows. Valid values of Arrow Spacing in Pixels are indicated by the label to the right of the entry box, i.e., 1, 2, 3, The minimum value of Arrow Spacing in Pixels of 1 results in the maximum possible arrow density with one arrow per datum. At some “large” spacing and with missing data, no arrows may be displayed. If no arrows are displayed, the message “No arrows can be displayed at current spacing” is displayed in the R GUI. Increase arrow density by entering 5 in the Arrow Spacing in Pixels entry box and click OK. As arrow density is increased, arrow length can be decreased by decreasing the Arrow Length Multiplier to a value > 0 . Enter 0.5 in the Arrow Length Multiplier entry box and click OK. The current result should appear as in Figure J-26.

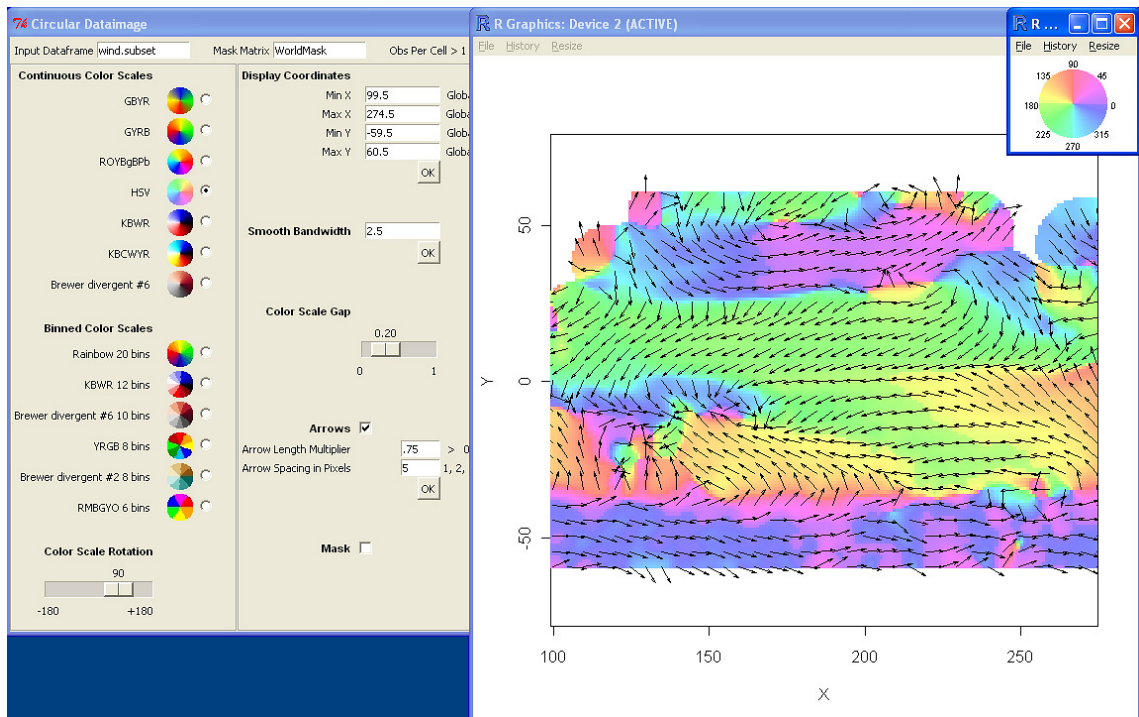


Figure J-26. HSV Color Wheel Rotated 90°, Data Smoothed with Bandwidth 2.5, Color Scale Gap 0.20, and Arrows on.

- 13 Smoothing replaces missing data (land in the wind.subset). The function of mask WorldMask is to restore the outlines of land masses. Pixels corresponding to matrix cells of 1 are overplotted in tan, and pixels corresponding to matrix cells of NA remain unchanged. If a mask matrix was not entered and the mask checkbox is clicked, nothing happens. Now check the Mask check box. The current result should appear as in Figure J-27.
- 14 Click the X in the upper right corners to exit.

J.11 PlotVectors

PlotVectors plots vector-spatial data as unit vectors, vectors, or triangle icons (Ware, 2004) with or without jittering. Triangle icons have area proportional to vector magnitude. Jittering can help clarify structure when vectors overlap. However, excessive jitter can obscure directional structure. The underlying fields function arrows will not plot any arrow of length less than 0.001 inch.

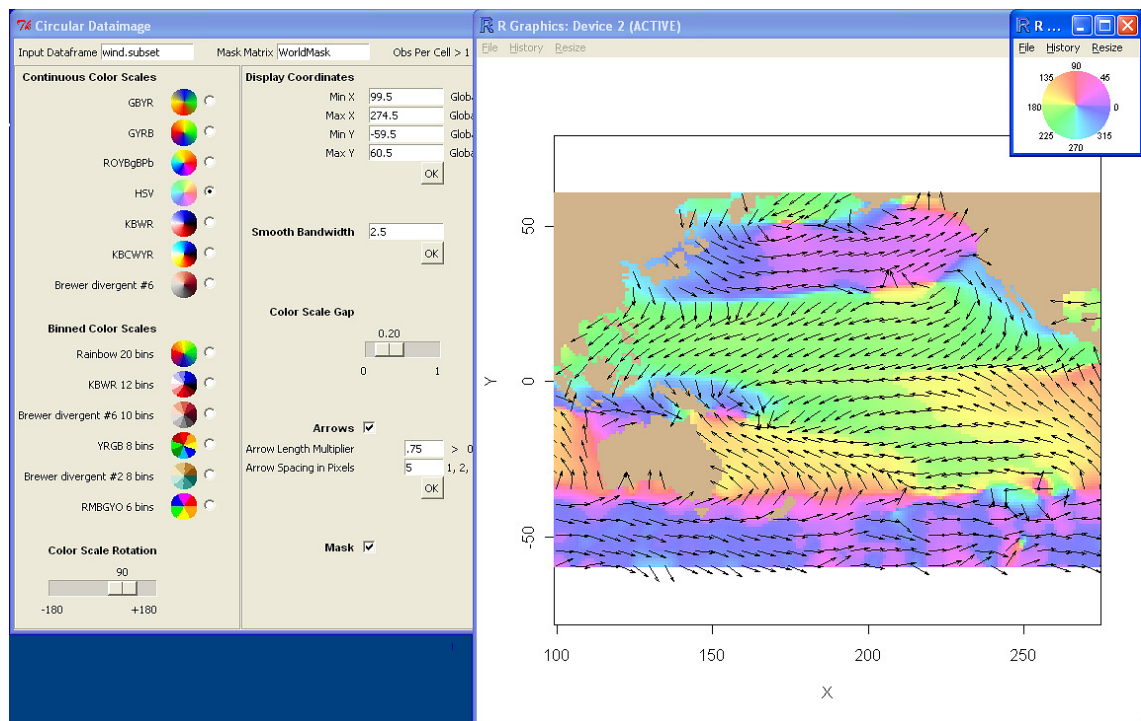


Figure J-27. Mask Restores Land Mass Shapes in Smoothed Data.

J.11.1 Usage and Input Arguments

PlotVectors(x, y, h, v, UnitVector=TRUE, Trilcon=FALSE, AdjArrowLength=1, AdjHeadLength=1, TrilconAdj=1, TriRatio=4, JitterPlot=FALSE, Jitter=1, \dots)

x: Vector of x coordinates.

y: Vector of y coordinates.

h: Vector of horizontal component. Missing values are permitted.

v: Vector of vertical component. Missing values are permitted.

UnitVector: TRUE or FALSE. See Table J-3.

Trilcon: TRUE or FALSE. See Table J-3.

AdjArrowLength: Arrow length multiplier.

AdjHeadLength: Arrow head length multiplier.

TrilconAdj: Multiplies size of icons.

TriRatio: Length to width ratio of triangle icon.

JitterPlot: If TRUE, add jitter to location coordinates.

Jitter: Amount of jitter = Jitter x runif value.

... : Additional plot parameters.

J.11.2 Output

Table J-3. Output of PlotVectors.

UnitVector	Trilcon	Plot Type
FALSE	FALSE	Direction and magnitude as variable length arrow
TRUE	FALSE	Direction as constant length arrow
FALSE	TRUE	Triangle icons with area proportional to magnitude
TRUE	TRUE	Direction as constant length arrow

J.11.3 Examples

```
require(CircSpatial)
data(OceanWind)
wind.1997.Jan <- OceanWind[OceanWind$year>1997 & OceanWind$year<1997.1,-1]
```

```
## Direction Only, Figure J-28
PlotVectors(x=wind.1997.Jan$x, y=wind.1997.Jan$y, h=wind.1997.Jan$u, v=wind.1997.Jan$v,
UnitVector=TRUE, AdjArrowLength=0.75, AdjHeadLength=0.75, xlim=c(320,350), ylim=c(0,30))
```

```
## Direction and Magnitude, Figure J-29
# fields function arrows omits arrowheads with a warning on
# any arrow of length less than 0.001 inch.
PlotVectors(x=wind.1997.Jan$x, y=wind.1997.Jan$y, h=wind.1997.Jan$u, v=wind.1997.Jan$v,
UnitVector=FALSE, Trilcon=FALSE, AdjArrowLength=3, AdjHeadLength=0.4, xlim=c(320,350),
ylim=c(0,30))
```

```
## Triangle Icons, Figure J-30
PlotVectors(x=wind.1997.Jan$x, y=wind.1997.Jan$y, h=wind.1997.Jan$u, v=wind.1997.Jan$v,
UnitVector=FALSE, Trilcon=TRUE, TrilconAdj=0.25, TriRatio=4, xlim=c(320,350),
ylim=c(0,30))
```

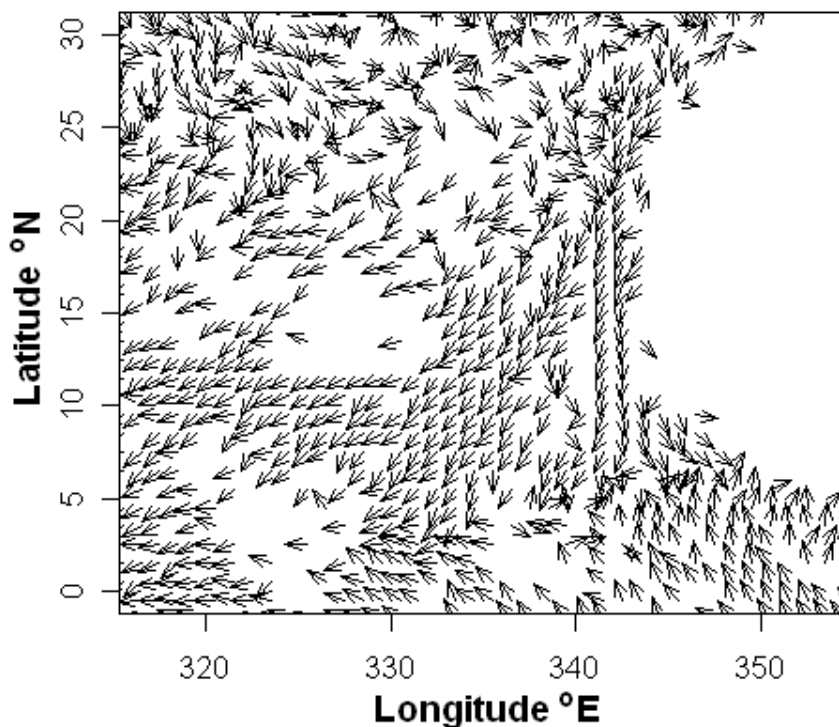


Figure J-28. Unit Vector Plot of Ocean Wind Data.

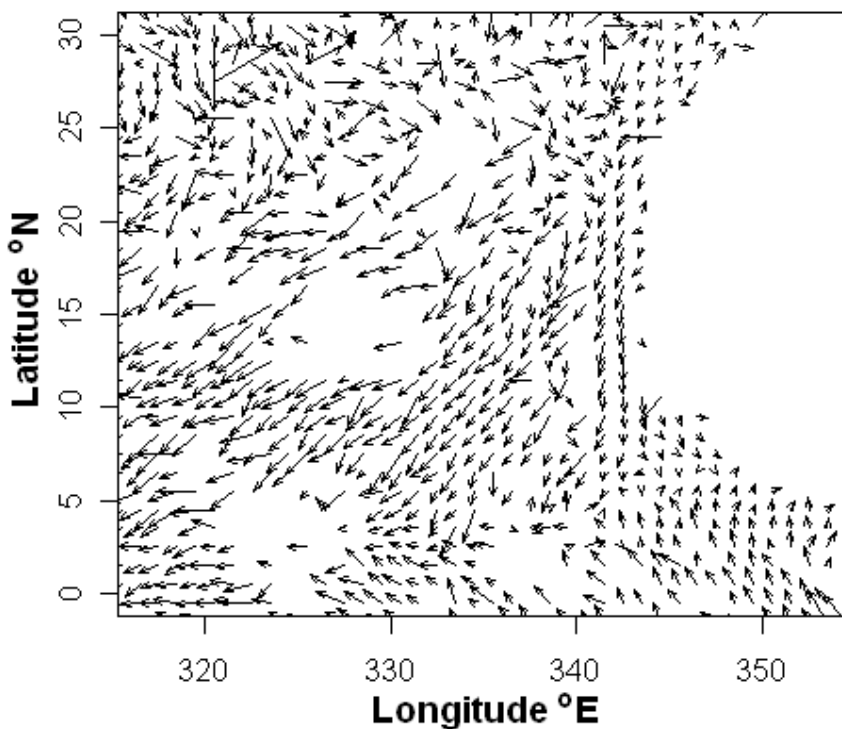


Figure J-29. Vector Plot of Ocean Wind Data.

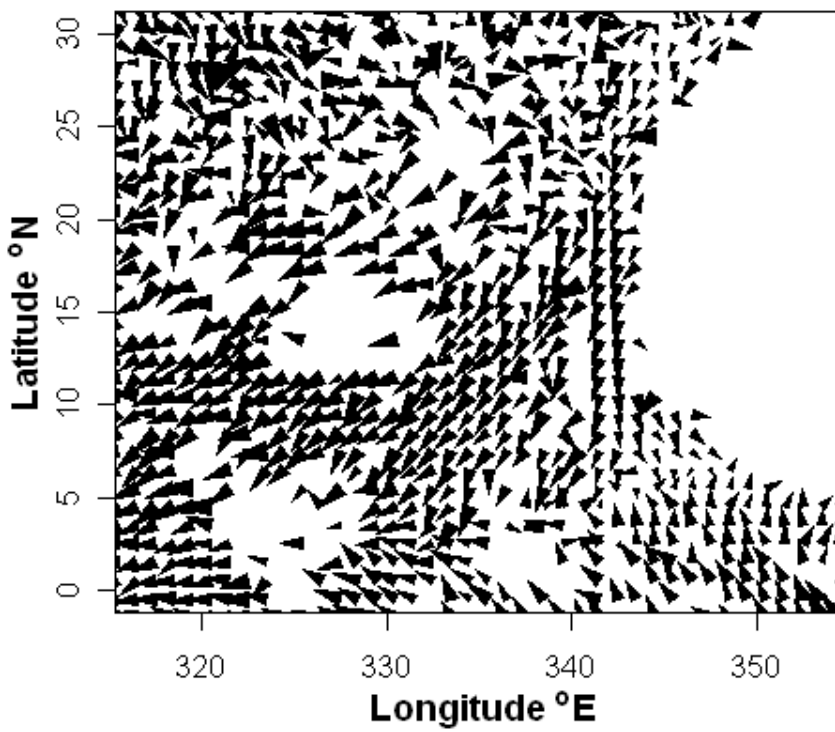


Figure J-30. Triangle Icon Plot of Ocean Wind Data.

Appendix K

R Function

K.1 TestPattern

In this appendix, some liberties have been taken with format. Titles match code names, which are case sensitive. To be readable, the code is structured. To fit, the structured code is in landscape orientation and single spaced. N.B.: The following code reflects the content of the R package. To run the code outside of the R package, paste the code into an R GUI taking care to correct double quotes “ as necessary.

Sometimes the double quotes in Word are misinterpreted in the R GUI, e.g.,

```
> plot(krig2$x, krig2$y, ty="n", xlab="", ylab="", xlim=c(5, 8), ylim=c(5, 8), asp=1)
Error: unexpected input in "plot(krig2$x, krig2$y, ty=""
```

```
TestPattern <- function()
{
  x <- seq(-1,1,length = 251)
  y <- x
  x <- rep(x, 251)
  y <- rep(y, ea = 251)
  dir <- atan2(y,x)
  dir[x == 0] <- NA
  dir[y == 0] <- NA
  u <- cos(dir)
  v <- sin(dir)
  return(as.data.frame(list(x = x, y = y, u = u, v = v)))
}
```

K.2 CircDataimage

The function `CircDataimage` embeds jpegs in the R package into the GUI. To paste this function into an R GUI, remove the **green highlighted/bolded code** on this page and on the 14th through 16th pages of this code. However, it is recommended that the R package `CircSpatial` be installed for examination. The Tcl path statements must match the user installation path of Active State Tcl from <http://downloads.activestate.com/ActiveTcl/Windows/>

```
CircDataimage <- function()
{
  # 2008-11-12.1919
  require(tcltk, quietly=TRUE, warn.conflicts=TRUE)
  require(fields, quietly=TRUE, warn.conflicts=TRUE)

  # Sys.setenv("TCL_LIBRARY"="C:/Tcl/lib/tcl8.5")
  # Sys.setenv("MY_TCLTK"="Yes")
  # addTclPath(path = "C:/Tcl/lib/teapot/package/win32-ix86/lib")
  # tclRequire("img::jpeg")

  # Make color wheel data
  x <- seq(-1,1,length=201) # x must be consistent with next image statement
  y <- x
  x2 <- rep(x, 201)
  y2 <- rep(y, ea=201)
  dir <- atan2(y2,x2)
  dir[dir<0] <- dir[dir<0] + 2*pi # Directions in [0, 2*pi)
  Dist <- sqrt(x2^2 + y2^2) # distance from origin
  filter <- Dist > 1
  dir[filter] <- NA
  wheel <- matrix(data=dir, nrow=201, ncol=201, byrow=FALSE)

  # Make FirstColorVector
  Angles1 <- 0:89
  Angles2 <- 90:179
  Angles3 <- 180:269
  Angles4 <- 270:359
  Dist1 = 255*Angles1/90
  Dist2 = 255*(Angles2-90)/90
```



```

Dist3 = 255*(Angles3-180)/90
Dist4 = 255*(Angles4-270)/90
Q1 <- rgb(0, 255-Dist1, Dist1, maxColorValue=255)
Q2 <- rgb(Dist2, Dist2, 255-Dist2, maxColorValue=255)
Q3 <- rgb(255, 255-Dist3, 0, maxColorValue=255)
Q4 <- rgb(255-Dist4, Dist4, 0, maxColorValue=255)
FirstColorVector <- c(Q1,Q2,Q3,Q4) # GBYR
if(is.null(dev.list())) dev.image=2 else dev.image= max(dev.list()) + 1
dev.wheel = dev.image + 1
windows() # device dev.image
windows(width = 1.15, height = 1, pointsize = 7) # device 3, width so menu bar on one row
# Current device is dev.wheel
par(plt=c(0.03,0.97,0.03,0.97)) # Applies to current device, min margin between labels and window
angles=seq(0, 315, by=45)
plot(x=1.2*cos(angles*pi/180), y=1.2*sin(angles*pi/180), type="n", asp=1, xaxt="n", yaxt="n", xlab="", ylab="",
      bty="n")
text(x=1.2*cos(angles*pi/180), y=1.2*sin(angles*pi/180), labels=as.character(angles))
image(x, y, z= wheel, col= FirstColorVector, add=TRUE)
Globals <- list() # This erases content of Globals if Globals exists from previous session
#####
R1.Prime <- function()
{
  #2007-08-20.1347
  # The following global variables are not dependent on data
  R1.WriteBinColorVectors() # 360 elements for each color vector
  Globals$ColorGap <- 0
  R1.WriteContColorVectors() # Must come after Globals$ColorGap

  Globals$ColorVector.g <- Globals$GBYR # Must come after R1.WriteContColorVectors
  Globals$ColorVector <- Globals$GBYR
  Globals$ColorVectorID <- 1
  Globals$ColorRotation <- 0
  Globals$PlotArrows <- FALSE
  Globals$ArrowAdj <- 1
  Globals$cpa <- 15
  Globals$Mask <- NULL
  Globals$PlotMask <- FALSE

```

```

}
#####
R1.WriteBinColorVectors <- function()
{
  # 2007-08-04.1103
  # Each vector has 360 elements for each of 360 degrees (°)
  Q1 <- rep(rgb( 0, 235, 35, maxColorValue=255), 18)
  Q2 <- rep(rgb( 0, 245, 0, maxColorValue=255), 18)
  Q3 <- rep(rgb(102, 250, 0, maxColorValue=255), 18)
  Q4 <- rep(rgb(153, 255, 0, maxColorValue=255), 18)
  Q5 <- rep(rgb(204, 255, 0, maxColorValue=255), 18)
  Q6 <- rep(rgb(255, 255, 0, maxColorValue=255), 18)
  Q7 <- rep(rgb(255, 204, 0, maxColorValue=255), 18)
  Q8 <- rep(rgb(255, 153, 0, maxColorValue=255), 18)
  Q9 <- rep(rgb(255, 102, 0, maxColorValue=255), 18)
  Q10 <- rep(rgb(255, 0, 0, maxColorValue=255), 18)
  Q11 <- rep(rgb(230, 0, 20, maxColorValue=255), 18)
  Q12 <- rep(rgb(195, 0, 51, maxColorValue=255), 18)
  Q13 <- rep(rgb(153, 0, 102, maxColorValue=255), 18)
  Q14 <- rep(rgb(102, 0, 153, maxColorValue=255), 18)
  Q15 <- rep(rgb( 0, 0, 150, maxColorValue=255), 18)
  Q16 <- rep(rgb( 51, 0, 175, maxColorValue=255), 18)
  Q17 <- rep(rgb( 0, 51, 204, maxColorValue=255), 18)
  Q18 <- rep(rgb( 0, 102, 153, maxColorValue=255), 18)
  Q19 <- rep(rgb( 0, 153, 102, maxColorValue=255), 18)
  Q20 <- rep(rgb( 0, 204, 51, maxColorValue=255), 18)
  Globals$Rainbow.20Bin <- c(Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, Q20)

  Q1 <- rep(rgb( 0, 0, 128, maxColorValue=255), 30)
  Q2 <- rep(rgb( 0, 0, 192, maxColorValue=255), 30)
  Q3 <- rep(rgb( 0, 0, 255, maxColorValue=255), 30)
  Q4 <- rep(rgb(128, 128, 255, maxColorValue=255), 30)
  Q5 <- rep(rgb(192, 192, 255, maxColorValue=255), 30)
  Q6 <- rep(rgb(255, 255, 255, maxColorValue=255), 30)
  Q7 <- rep(rgb(255, 219, 219, maxColorValue=255), 30)
  Q8 <- rep(rgb(255, 128, 128, maxColorValue=255), 30)
  Q9 <- rep(rgb(255, 0, 0, maxColorValue=255), 30)

```

```

Q10 <- rep(rgb(192, 0, 32, maxColorValue=255), 30)
Q11 <- rep(rgb(130, 0, 0, maxColorValue=255), 30)
Q12 <- rep(rgb( 0, 0, 0, maxColorValue=255), 30)
Globals$KBWR.12Bin <- c(Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12)

```

```

Q1 <- rep(rgb(103, 0, 31, maxColorValue=255), 36)
Q2 <- rep(rgb(178, 24, 43, maxColorValue=255), 36)
Q3 <- rep(rgb(214, 96, 77, maxColorValue=255), 36)
Q4 <- rep(rgb(244, 165, 130, maxColorValue=255), 36)
Q5 <- rep(rgb(253, 219, 199, maxColorValue=255), 36)
Q6 <- rep(rgb(224, 224, 224, maxColorValue=255), 36)
Q7 <- rep(rgb(186, 186, 186, maxColorValue=255), 36)
Q8 <- rep(rgb(135, 135, 135, maxColorValue=255), 36)
Q9 <- rep(rgb( 77, 77, 77, maxColorValue=255), 36)
Q10 <- rep(rgb( 64, 13, 28, maxColorValue=255), 36)
Globals$Brewer10Div6.10Bin <- c(Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10)

```

```

Q1 <- rep(rgb(255, 255, 0, maxColorValue=255),23) # yellow
Q2 <- rep(rgb(255, 255*0.65, 0, maxColorValue=255),45) # orange
Q3 <- rep(rgb(255, 0, 0, maxColorValue=255),45) # red
Q4 <- rep(rgb(255*0.75, 0, 0, maxColorValue=255),45) # dark red
Q5 <- rep(rgb(0, 255, 0, maxColorValue=255),45) # green
Q6 <- rep(rgb(0, 255*0.6, 0, maxColorValue=255),45) # dark green
Q7 <- rep(rgb(0, 255*.75, 255, maxColorValue=255),45) # blue
Q8 <- rep(rgb(0, 0, 255, maxColorValue=255),45) # dark blue
Q9 <- rep(rgb(255, 255, 0, maxColorValue=255),22) # yellow
Globals$YRGB.8Bin <- c(Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9)

```

```

Q1 <- rep(rgb(140, 81, 10, maxColorValue=255), 45)
Q2 <- rep(rgb(191, 129, 45, maxColorValue=255), 45)
Q3 <- rep(rgb(223, 194, 125, maxColorValue=255), 45)
Q4 <- rep(rgb(246, 232, 195, maxColorValue=255), 45)
Q5 <- rep(rgb(199, 234, 229, maxColorValue=255), 45)
Q6 <- rep(rgb(128, 205, 193, maxColorValue=255), 45)
Q7 <- rep(rgb( 53, 151, 143, maxColorValue=255), 45)
Q8 <- rep(rgb( 1, 102, 94, maxColorValue=255), 45)
Globals$Brewer8Div2.8Bin <- c(Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8)

```

```

Q1 <- rep(rgb(255, 0, 0, maxColorValue=255), 60) # red
Q2 <- rep(rgb(255, 0, 255, maxColorValue=255), 60) # magenta
Q3 <- rep(rgb(0, 0, 255, maxColorValue=255), 60) # blue
Q4 <- rep(rgb(0, 255, 0, maxColorValue=255), 60) # green
Q5 <- rep(rgb(255, 255, 0, maxColorValue=255), 60) # yellow
Q6 <- rep(rgb(255, 165, 0, maxColorValue=255), 60) # orange
Globals$RMBGYO.6Bin <- c(Q1, Q2, Q3, Q4, Q5, Q6)
}
#####
R1.WriteContColorVectors <- function()
{
  # 2007-09-11.1943
  # Each vector has 360 elements for each of 360°

  gap <- Globals$ColorGap
  Angles1 <- 0:89
  Angles2 <- 90:179
  Angles3 <- 180:269
  Angles4 <- 270:359
  Dist1 = (1-gap)*255*Angles1/90
  Dist2 = (1-gap)*255*(Angles2-90)/90
  Dist3 = (1-gap)*255*(Angles3-180)/90
  Dist4 = (1-gap)*255*(Angles4-270)/90
  Q1 <- rgb(0, 255-Dist1, Dist1, maxColorValue=255)
  Q2 <- rgb(Dist2, Dist2, 255-Dist2, maxColorValue=255)
  Q3 <- rgb(255, 255-Dist3, 0, maxColorValue=255)
  Q4 <- rgb(255-Dist4, Dist4, 0, maxColorValue=255)
  Globals$GBYR <- c(Q1,Q2,Q3,Q4)

  Q1 <- rgb(Dist1, 255, 0, maxColorValue=255)
  Q2 <- rgb(255, 255-Dist2, 0, maxColorValue=255)
  Q3 <- rgb(255-Dist3, 0, Dist3, maxColorValue=255)
  Q4 <- rgb(0, Dist4, 255-Dist4, maxColorValue=255)
  Globals$GYRB <- c(Q1,Q2,Q3,Q4)
  Q1 <- rgb(Dist1, 255, 0, maxColorValue=255)
  Q2 <- rgb(255-Dist2, 255-Dist2, Dist2, maxColorValue=255)

```

```
Q3 <- rgb(Dist3, 0, 255-Dist3, maxColorValue=255)
Q4 <- rgb(255-Dist4, Dist4, 0, maxColorValue=255)
Globals$GYBR <<- c(Q1,Q2,Q3,Q4)
```

```
Q1 <- rgb(0, 0, Dist1, maxColorValue=255)
Q2 <- rgb(Dist2, Dist2, 255, maxColorValue=255)
Q3 <- rgb(255, 255-Dist3, 255-Dist3, maxColorValue=255)
Q4 <- rgb(255-Dist4, 0, 0, maxColorValue=255)
Globals$KBWR <<- c(Q1,Q2,Q3,Q4)
```

```
Dist1 = (1-gap)*Angles1/360
Dist2 = 0.25 + (1-gap)*0.25*(Angles2 -90)/90
Dist3 = 0.50 + (1-gap)*0.25*(Angles3-180)/90
Dist4 = 0.75 + (1-gap)*0.25*(Angles4-270)/90
Q1 <- hsv(h=Dist1, s=0.5, v=1)
Q2 <- hsv(h=Dist2, s=0.5, v=1)
Q3 <- hsv(h=Dist3, s=0.5, v=1)
Q4 <- hsv(h=Dist4, s=0.5, v=1)
Globals$HSV <<- c(Q1,Q2,Q3,Q4)
```

```
Angles1 <- 0:59
Angles2 <- 60:119
Angles3 <- 120:179
Angles4 <- 180:239
Angles5 <- 240:299
Angles6 <- 300:359
```

```

Dist1 = (1-gap)*255*Angles1/60
Dist2 = (1-gap)*255*(Angles2-60)/60
Dist3 = (1-gap)*255*(Angles3-120)/60
Dist4 = (1-gap)*255*(Angles4-180)/60
Dist5 = (1-gap)*255*(Angles5-240)/60
Dist6 = (1-gap)*255*(Angles6-300)/60
Q1 <- rgb(0, 0, Dist1, maxColorValue=255)
Q2 <- rgb(0, Dist2, 255, maxColorValue=255)
Q3 <- rgb(Dist3, 255, 255, maxColorValue=255)
Q4 <- rgb(255, 255, 255-Dist4, maxColorValue=255)
Q5 <- rgb(255, 255-Dist5, 0, maxColorValue=255)
Q6 <- rgb(255-Dist6, 0, 0, maxColorValue=255)
Globals$KBCWYR <- c(Q1, Q2, Q3, Q4, Q5, Q6)

```

```

Angles1 <- 0:59
Angles2 <- 60:119
Angles3 <- 120:179
Angles4 <- 180:239
Angles5 <- 240:299
Angles6 <- 300:359
Dist1 = (1-gap)*(Angles1- 0)/60
Dist2 = (1-gap)*(Angles2- 60)/60
Dist3 = (1-gap)*(Angles3-120)/60
Dist4 = (1-gap)*(Angles4-180)/60
Dist5 = (1-gap)*(Angles5-240)/60
Dist6 = (1-gap)*(Angles6-300)/60

```

```

Q1 <- rgb(      255,      165*Dist1,      0, maxColorValue=255)
Q2 <- rgb(      255, 165+(255-165)*Dist2,      0, maxColorValue=255)
Q3 <- rgb(255*(1-Dist3),      255,      255*Dist3, maxColorValue=255)
Q4 <- rgb(      0,      255*(1-Dist4),      255, maxColorValue=255)
Q5 <- rgb(      255*Dist5,      0,      255, maxColorValue=255)
Q6 <- rgb(      255,      0, 255*(1-Dist6), maxColorValue=255)
Globals$ROYBgBPb <- c(Q1, Q2, Q3, Q4, Q5, Q6)

```

```

Angles1 <- 0:35
Angles2 <- 36:71
Angles3 <- 72:107
Angles4 <- 108:143
Angles5 <- 144:179
Angles6 <- 180:215
Angles7 <- 216:251
Angles8 <- 252:287
Angles9 <- 288:323
Angles10 <- 324:359
Dist1 = (1-gap)*Angles1/36
Dist2 = (1-gap)*(Angles2-36)/36
Dist3 = (1-gap)*(Angles3-72)/36
Dist4 = (1-gap)*(Angles4-108)/36
Dist5 = (1-gap)*(Angles5-144)/36
Dist6 = (1-gap)*(Angles6-180)/36
Dist7 = (1-gap)*(Angles7-216)/36
Dist8 = (1-gap)*(Angles8-252)/36
Dist9 = (1-gap)*(Angles9-288)/36
Dist10 = (1-gap)*(Angles10-324)/36
Q1 <- rgb(103+(178-103)*Dist1, 0 +(24-0)*Dist1, 31+(43-31)*Dist1, maxColorValue=255)
Q2 <- rgb(178+(214-178)*Dist2, 24 +(96-24)*Dist2, 43+(77-43)*Dist2, maxColorValue=255)
Q3 <- rgb(214+(244-214)*Dist3, 96+(165-96)*Dist3, 77+(130-77)*Dist3, maxColorValue=255)
Q4 <- rgb(244+(253-244)*Dist4, 165+(219-165)*Dist4, 130+(199-130)*Dist4, maxColorValue=255)
Q5 <- rgb(253+(224-253)*Dist5, 219+(224-219)*Dist5, 199+(224-199)*Dist5, maxColorValue=255)
Q6 <- rgb(224+(186-224)*Dist6, 224+(186-224)*Dist6, 224+(186-224)*Dist6, maxColorValue=255)
Q7 <- rgb(186+(135-186)*Dist7, 186+(135-186)*Dist7, 186+(135-186)*Dist7, maxColorValue=255)
Q8 <- rgb(135 +(77-135)*Dist8, 135 +(77-135)*Dist8, 135+(77-135)*Dist8, maxColorValue=255)
Q9 <- rgb(77 +(64-77)*Dist9, 77 +(13-77)*Dist9, 77 +(28-77)*Dist9, maxColorValue=255)
Q10 <- rgb(64+(103-64)*Dist10, 13 +(0-13)*Dist10, 28+(31-28)*Dist10, maxColorValue=255)
Globals$Brewer10Div6 <-<- c(Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10)
}
#####

```

```

R1.Initialize <- function(data.name2, mask.name2, nObs.cb.value2)
{
  # 2007-09-12.1930
  # Variable name suffix ".g" indicates variable is global, i.e. at limits of data
  # Variable name suffix ".d" indicates variable has been subset for display
  # Coordinates and direction will apply at center of pixel
  InputData <- as.matrix(eval(parse(file="", text=as.character(tclvalue(data.name2))))) # Must be matrix for loop below
  mask <- as.character(tclvalue(mask.name2))
  if(mask == "unknown" | mask == "") Globals$Mask <- NULL else Globals$Mask <- as.matrix(eval(parse(file="", text=mask)))

  Globals$Data <- InputData
  x <- sort(unique(InputData[,1])) # Ascending unique horizontal coordinates of sampling locations.
  y <- sort(unique(InputData[,2])) # Ascending unique vertical coordinates of sampling locations.
  Globals$MinX.g <- min(x) # Global minimum X.
  Globals$MaxX.g <- max(x) # Global maximum X.
  Globals$MinY.g <- min(y) # Global minimum Y.
  Globals$MaxY.g <- max(y) # Global maximum Y.
  # Measurement location horizontal spacing assumed to be constant in X.
  # Grid vertical spacing assumed to be constant in Y. Horiz and vert spacing do not have to be equal.
  Globals$DX <- x[2] - x[1] # Horizontal spacing of sampling grid.
  Globals$DY <- y[2] - y[1] # Vertical spacing of sampling grid.
  # Simple check
  DX2 <- x[3] - x[2]
  DY2 <- y[3] - y[2]
  if(DX2 != Globals$DX | DY2 != Globals$DY) stop("Measurement spacing not constant")
  Globals$nx.g <- round((Globals$MaxX.g-Globals$MinX.g)/Globals$DX + 1, digits = 0)
  Globals$ny.g <- round((Globals$MaxY.g-Globals$MinY.g)/Globals$DY + 1, digits = 0)
  Globals$x.g <- seq(from=Globals$MinX.g, to=Globals$MaxX.g,length=Globals$nx.g)
  Globals$y.g <- seq(from=Globals$MinY.g, to=Globals$MaxY.g,length=Globals$ny.g)

  # for display if Pan() not invoked
  Globals$MinX.d <- Globals$MinX.g
  Globals$MaxX.d <- Globals$MaxX.g
  Globals$MinY.d <- Globals$MinY.g
  Globals$MaxY.d <- Globals$MaxY.g
  # The number of rows of the matrix will be = Globals$nx.g = length(Globals$x.g)
  Globals$StartRow <- 1

```



```

Globals$EndRow <<- Globals$nx.g
Globals$StartCol <<- 1
Globals$EndCol <<- Globals$ny.g
if(as.character(tclvalue(nObs.cb.value2)) == "0")
{
  u.g <- matrix(data=NA,nrow=Globals$nx.g,ncol=Globals$ny.g) # u accumulator, because atan2(0,0)=0
  v.g <- u.g # v accumulator
  Rows <- round((InputData[, 1]- Globals$MinX.g)/Globals$DX + 1, digits = 0) # Indexing vector
  Columns <- round((InputData[, 2]- Globals$MinY.g)/Globals$DY + 1, digits = 0) # Indexing vector
  u.g[cbind(Rows, Columns)] <- InputData[, 3]
  v.g[cbind(Rows, Columns)] <- InputData[, 4]
} else
{
  cat("The initial computations necessarily may take significant time\n")
  u.g <- matrix(data=0, nrow=Globals$nx.g, ncol=Globals$ny.g) # u accumulator
  v.g <- u.g # v accumulator
  N.g <- u.g # Number of observations per cell
  Row <- round((InputData[, 1]- Globals$MinX.g)/Globals$DX + 1, digits = 0) # Indexing scalar
  Column <- round((InputData[, 2]- Globals$MinY.g)/Globals$DY + 1, digits = 0) # Indexing scalar
  for (i in 1:nrow(InputData))
  {
    u.g[Row[i], Column[i]] <- u.g[Row[i], Column[i]] + InputData[i, 3]
    v.g[Row[i], Column[i]] <- v.g[Row[i], Column[i]] + InputData[i, 4]
    N.g[Row[i], Column[i]] <- N.g[Row[i], Column[i]] + 1
  }
  # Averages
  filter1 <- N.g > 0
  u.g[filter1] <- u.g[filter1]/N.g[filter1]
  v.g[filter1] <- v.g[filter1]/N.g[filter1]
  # Replace 0's with NAs where there are no observations
  u.g[!filter1] <- NA
  v.g[!filter1] <- NA
}

```

```

Globals$u.g <- u.g # Cell contains average u or NA
Globals$v.g <- v.g
Globals$Direction.g <- R1.Standardize(atan2(v.g, u.g)) # atan2(NA,NA)=NA
Globals$Direction <- Globals$Direction.g
R1.SubsetColorScale(Globals$Direction[Globals$StartRow:Globals$EndRow, Globals$StartCol:Globals$EndCol])
Globals$PlotMask <- FALSE
Globals$PlotArrows <- FALSE
R1.PlotImage()
}
#####
R1.Standardize <- function(Input)
{
  # 2007-08-05.1218
  # Input and Output in radians
  filter <- !is.na(Input)
  temp <- Input[filter]
  temp[temp < 0] <- temp[temp < 0] + 2*pi
  temp[temp > 2*pi] <- temp[temp > 2*pi] - 2*pi
  Input[filter] <- temp
  return(Input)
}
#####
R1.SubsetColorScale <- function(Input)
{
  # 2007-08-20.1331
  filter <- !is.na(Input)
  Range <- floor(range(Input[filter]*180/pi))
  a <- which((0:359) == Range[1])
  b <- which((0:359) == Range[2])
  Globals$ColorFilter <- a:b
}
#####

```

```

R1.PlotImage <- function()
{
  # 2007-09-14.1720
  # Composite image = data overplotted with arrows overplotted with mask
  dev.set(which=dev.image)
  # The image color vector is subset based on range of data.
  image(x= Globals$x.g[Globals$StartRow:Globals$EndRow],
        y= Globals$y.g[Globals$StartCol:Globals$EndCol],
        z= Globals$Direction[Globals$StartRow:Globals$EndRow, Globals$StartCol:Globals$EndCol],
        col= Globals$ColorVector[Globals$ColorFilter], xlab="X", ylab="Y", asp=1)

  if(Globals$PlotMask) R1.PlotMask()
  if(Globals$PlotArrows) R1.PlotArrows()
}
#####
R1.PlotArrows <- function()
{
  # 2007-09-20.2244
  x <- Globals$x.g[Globals$StartRow:Globals$EndRow]
  y <- Globals$y.g[Globals$StartCol:Globals$EndCol]
  Directions <- Globals$Direction[Globals$StartRow:Globals$EndRow, Globals$StartCol:Globals$EndCol]
  nx=length(x)
  ny=length(y)
  x <- rep(x, ny)
  y <- rep(y, each=nx)
  Directions <- as.vector(Directions)
  filter1 <- rep(rep(Globals$cpa:1, length=nx), ny) == Globals$cpa
  filter2 <- as.vector(t(matrix(data=rep(rep(Globals$cpa:1, length=ny), nx), nrow=ny))) == Globals$cpa
  if(!Globals$PlotMask | is.null(Globals$Mask)) filter <- !is.na(Directions) & filter1 & filter2
  if(Globals$PlotMask & !is.null(Globals$Mask))
  {
    mask.boolean <- matrix(data=TRUE, nrow=Globals$nx.g, ncol=Globals$ny.g)
    mask.boolean[!is.na(Globals$Mask)] <- FALSE
    mask.boolean <- mask.boolean[Globals$StartRow:Globals$EndRow, Globals$StartCol:Globals$EndCol]
    filter <- !is.na(Directions) & filter1 & filter2 & as.vector(mask.boolean)
  }
  if(sum(filter) > 0)

```

```

{
  x <- x[filter]
  y <- y[filter]
  Directions <- Directions[filter]
  arrow.plot(x, y, u = cos(Directions), v = sin(Directions), arrow.ex = 0.05*Globals$ArrowAdj, xpd = FALSE,
    true.angle = TRUE, arrowfun=arrows, length=.05, angle=15, col=1)
} else cat("No arrows can be displayed at current spacing\n")
}
#####
R1.PlotMask <- function()
{
  #2007-08-09.2013
  image(x=Globals$x.g[Globals$StartRow:Globals$EndRow],
    y=Globals$y.g[Globals$StartCol:Globals$EndCol],
    z=Globals$Mask[Globals$StartRow:Globals$EndRow, Globals$StartCol:Globals$EndCol],
    col= "tan", add=TRUE)
}
#####
R1.PlotWheel <- function()
{
  dev.set(which=dev.wheel)
  # The image color vector is not subset based on range of data.
  image(x=seq(-1,1,length=201), y=seq(-1,1,length=201), z= wheel, col= Globals$ColorVector, add=TRUE)
}
#####
R1.Pan <- function()
{
  # 2007-09-11.2139
  Globals$StartRow <<- round((Globals$MinX.d - Globals$MinX.g)/Globals$DX + 1, digits=0)
  Globals$EndRow <<- round((Globals$MaxX.d - Globals$MinX.g)/Globals$DX + 1, digits=0)
  Globals$StartCol <<- round((Globals$MinY.d - Globals$MinY.g)/Globals$DY + 1, digits=0)
  Globals$EndCol <<- round((Globals$MaxY.d - Globals$MinY.g)/Globals$DY + 1, digits=0)
  R1.SubsetColorScale(Globals$Direction[Globals$StartRow:Globals$EndRow, Globals$StartCol:Globals$EndCol])
  R1.PlotImage()
}
#####
R1.ChangeColorWheel <- function()

```

```

{
# 2007-08-20.1903
ID <- Globals$ColorVectorID
if(ID == "1") Globals$ColorVector.g <<- Globals$GBYR
if(ID == "2") Globals$ColorVector.g <<- Globals$GYRB
if(ID == "3") Globals$ColorVector.g <<- Globals$ROYBgBPb
if(ID == "4") Globals$ColorVector.g <<- Globals$HSV
if(ID == "5") Globals$ColorVector.g <<- Globals$KBWR
if(ID == "6") Globals$ColorVector.g <<- Globals$KBCWYR
if(ID == "7") Globals$ColorVector.g <<- Globals$Brewer10Div6
if(ID == "8") Globals$ColorVector.g <<- Globals$Rainbow.20Bin
if(ID == "9") Globals$ColorVector.g <<- Globals$KBWR.12Bin
if(ID == "10") Globals$ColorVector.g <<- Globals$Brewer10Div6.10Bin
if(ID == "11") Globals$ColorVector.g <<- Globals$YRGB.8Bin
if(ID == "12") Globals$ColorVector.g <<- Globals$Brewer8Div2.8Bin
if(ID == "13") Globals$ColorVector.g <<- Globals$RMBGYO.6Bin
R1.AutoRotateColorWheel()
R1.PlotImage()
R1.PlotWheel()
}
#####
R1.AutoRotateColorWheel <- function()
{
# 2007-08-06.1603
Rotation <- Globals$ColorRotation
if(Rotation > 0)
{
a <- (360-Rotation+1):360
Globals$ColorVector <<- c(Globals$ColorVector.g[a], Globals$ColorVector.g[-a])
} else
if(Rotation == 0) {Globals$ColorVector <<- Globals$ColorVector.g} else
{Globals$ColorVector <<- c(Globals$ColorVector.g[-1:Rotation], Globals$ColorVector.g[1:-Rotation])}
}
#####
R1.RotateColorWheel <- function()
{
# 2007-08-20.1933

```

```

# To return to unrotated color wheel, enter zero for rotation.
Rotation <- Globals$ColorRotation
if(Rotation > 0)
{
  a <- (360-Rotation+1):360
  Globals$ColorVector <<- c(Globals$ColorVector.g[a], Globals$ColorVector.g[-a])
} else
if(Rotation == 0) {Globals$ColorVector <<- Globals$ColorVector.g} else
{Globals$ColorVector <<- c(Globals$ColorVector.g[-1:Rotation], Globals$ColorVector.g[1:-Rotation])}
R1.PlotImage()
R1.PlotWheel()
}
#####
R1.ChangeColorGap <- function()
{
  # 2007-08-11.1223
  R1.WriteContColorVectors() # Recompute with gap
  if(Globals$ColorVectorID == 1) Globals$ColorVector.g <<- Globals$GBYR
  if(Globals$ColorVectorID == 2) Globals$ColorVector.g <<- Globals$GYRB
  if(Globals$ColorVectorID == 3) Globals$ColorVector.g <<- Globals$ROYBgBPb
  if(Globals$ColorVectorID == 4) Globals$ColorVector.g <<- Globals$HSV
  if(Globals$ColorVectorID == 5) Globals$ColorVector.g <<- Globals$KBWR
  if(Globals$ColorVectorID == 6) Globals$ColorVector.g <<- Globals$KBCWYR
  if(Globals$ColorVectorID == 7) Globals$ColorVector.g <<- Globals$Brewer10Div6
  R1.AutoRotateColorWheel()
  R1.PlotImage()
  R1.PlotWheel()
}
#####

```

```

R1.Prime()

Top <- tkoplevel()
tkwm.geometry(Top,"565x600")
tkwm.title(Top,"Circular Dataimage")
FontHeading <- tkfont.create(family="arial", size=8, weight="bold")
FrameTop <- tkframe(Top, relief="flat", borderwidth=2)
data.name <- tclVar("unknown")
data.name.entry <- tkentry(FrameTop, width="15", textvariable=data.name)
mask.name <- tclVar("unknown")
mask.name.entry <- tkentry(FrameTop, width="15",textvariable=mask.name)
nObs.cb.value <- tclVar("0")
nObs.cb <- tkcheckboxbutton(FrameTop); tkconfigure(nObs.cb,variable=nObs.cb.value)
Input.but <- tkbutton(FrameTop, text="OK", command=function(){
  R1.Initialize(data.name, mask.name, nObs.cb.value)
  tclvalue(MinX.g) <- as.character(Globals$MinX.g)
  tclvalue(MaxX.g) <- as.character(Globals$MaxX.g)
  tclvalue(MinY.g) <- as.character(Globals$MinY.g)
  tclvalue(MaxY.g) <- as.character(Globals$MaxY.g)
  tclvalue(MinX) <- as.character(Globals$MinX.g)
  tclvalue(MaxX) <- as.character(Globals$MaxX.g)
  tclvalue(MinY) <- as.character(Globals$MinY.g)
  tclvalue(MaxY) <- as.character(Globals$MaxY.g)
  tclvalue(Smooth) <- "0"
  tclvalue(arrow.cb.value) <- "0"
  tclvalue(mask.cb.value) <- "0"
})
tkgrid(tklabel(FrameTop, text="Input Dataframe"), data.name.entry, tklabel(FrameTop,text="  "),
tklabel(FrameTop,text="Mask Matrix"), mask.name.entry, tklabel(FrameTop,text="  "),
tklabel(FrameTop,text="Obs Per Cell > 1"), nObs.cb, tklabel(FrameTop,text="  "),
Input.but, sticky="w")

```

```

FrameLeft <- tkframe(Top, relief="groove", borderwidth=2)
tkgrid(tklabel(FrameLeft,text="Continuous Color Scales", font=FontHeading), sticky="e")
image1 <- tclVar(); image2 <- tclVar(); image3 <- tclVar(); image4 <- tclVar(); image5 <- tclVar(); image6 <- tclVar()
image7 <- tclVar(); image8 <- tclVar(); image9 <- tclVar(); image10 <- tclVar(); image11 <- tclVar(); image12 <- tclVar()
image13 <- tclVar()
tcl("image", "create", "photo", image1, file=system.file("graphics", "GBYR.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image2, file=system.file("graphics", "GYRB.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image3, file=system.file("graphics", "ROYBgBPb.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image4, file=system.file("graphics", "HSV.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image5, file=system.file("graphics", "KBWR.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image6, file=system.file("graphics", "KBCWYR.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image7, file=system.file("graphics", "BREWER10D6.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image8, file=system.file("graphics", "Rainbow.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image9, file=system.file("graphics", "KBWR.12.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image10, file=system.file("graphics", "Brewer10D6.10.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image11, file=system.file("graphics", "YRGB.8.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image12, file=system.file("graphics", "Brewer8D2.8.jpeg", package="CircSpatial"))
tcl("image", "create", "photo", image13, file=system.file("graphics", "RMBGYO.6.jpeg", package="CircSpatial"))
wheel1 <- tklabel(FrameLeft, image=image1) # Image as label
wheel2 <- tklabel(FrameLeft, image=image2)
wheel3 <- tklabel(FrameLeft, image=image3)
wheel4 <- tklabel(FrameLeft, image=image4)
wheel5 <- tklabel(FrameLeft, image=image5)
wheel6 <- tklabel(FrameLeft, image=image6)
wheel7 <- tklabel(FrameLeft, image=image7)
wheel8 <- tklabel(FrameLeft, image=image8)
wheel9 <- tklabel(FrameLeft, image=image9)
wheel10 <- tklabel(FrameLeft, image=image10)
wheel11 <- tklabel(FrameLeft, image=image11)
wheel12 <- tklabel(FrameLeft, image=image12)
wheel13 <- tklabel(FrameLeft, image=image13)
rb1 <- tkradiobutton(FrameLeft)
rb2 <- tkradiobutton(FrameLeft)
rb3 <- tkradiobutton(FrameLeft)
rb4 <- tkradiobutton(FrameLeft)
rb5 <- tkradiobutton(FrameLeft)
rb6 <- tkradiobutton(FrameLeft)

```



```

rb7 <- tkradiobutton(FrameLeft)
rb8 <- tkradiobutton(FrameLeft)
rb9 <- tkradiobutton(FrameLeft)
rb10 <- tkradiobutton(FrameLeft)
rb11 <- tkradiobutton(FrameLeft)
rb12 <- tkradiobutton(FrameLeft)
rb13 <- tkradiobutton(FrameLeft)
rbValue <- tclVar("1")
ChangeColor <- function() {Globals$ColorVectorID <- as.character(tclvalue(rbValue)); R1.ChangeColorWheel();}
tkconfigure(rb1, variable=rbValue,value="1", command=ChangeColor)
tkconfigure(rb2, variable=rbValue,value="2", command=ChangeColor)
tkconfigure(rb3, variable=rbValue,value="3", command=ChangeColor)
tkconfigure(rb4, variable=rbValue,value="4", command=ChangeColor)
tkconfigure(rb5, variable=rbValue,value="5", command=ChangeColor)
tkconfigure(rb6, variable=rbValue,value="6", command=ChangeColor)
tkconfigure(rb7, variable=rbValue,value="7", command=ChangeColor)
tkconfigure(rb8, variable=rbValue,value="8", command=ChangeColor)
tkconfigure(rb9, variable=rbValue,value="9", command=ChangeColor)
tkconfigure(rb10,variable=rbValue,value="10", command=ChangeColor)
tkconfigure(rb11,variable=rbValue,value="11", command=ChangeColor)
tkconfigure(rb12,variable=rbValue,value="12", command=ChangeColor)
tkconfigure(rb13,variable=rbValue,value="13", command=ChangeColor)
tkgrid(tklabel(FrameLeft,text="GBYR "), wheel1, rb1, sticky="e")
tkgrid(tklabel(FrameLeft,text="GYRB "), wheel2, rb2, sticky="e")
tkgrid(tklabel(FrameLeft,text="ROYBgbPb "), wheel3, rb3, sticky="e")
tkgrid(tklabel(FrameLeft,text="HSV "), wheel4, rb4, sticky="e")
tkgrid(tklabel(FrameLeft,text="KBWR "), wheel5, rb5, sticky="e")
tkgrid(tklabel(FrameLeft,text="KBCWYR "), wheel6, rb6, sticky="e")
tkgrid(tklabel(FrameLeft,text="Brewer divergent #6 "), wheel7, rb7, sticky="e")
tkgrid(tklabel(FrameLeft,text=""), column=1)
tkgrid(tklabel(FrameLeft,text="Binned Color Scales", font=FontHeading), sticky="e")
tkgrid(tklabel(FrameLeft,text="Rainbow 20 bins "), wheel8, rb8, sticky="e")
tkgrid(tklabel(FrameLeft,text="KBWR 12 bins "), wheel9, rb9, sticky="e")
tkgrid(tklabel(FrameLeft,text="Brewer divergent #6 10 bins "), wheel10, rb10, sticky="e")
tkgrid(tklabel(FrameLeft,text="YRGB 8 bins "), wheel11, rb11, sticky="e")
tkgrid(tklabel(FrameLeft,text="Brewer divergent #2 8 bins "), wheel12, rb12, sticky="e")
tkgrid(tklabel(FrameLeft,text="RMBGYO 6 bins "), wheel13, rb13, sticky="e")

```

```

tkgrid(tklabel(FrameLeft,text="      "))
SliderValue1 <- tclVar("0")
SliderValueLabel1 <- tklabel(FrameLeft,text=as.character(tclvalue(SliderValue1)))
tkconfigure(SliderValueLabel1,textvariable=SliderValue1)
slider1 <- tkscale(FrameLeft, from=-180, to=180, showvalue=TRUE, variable=SliderValue1, resolution=5, orient="horizontal", length="1.15i")
tkbind(slider1,"<ButtonRelease-1>", function() {Globals$ColorRotation <<- as.numeric(tclvalue(SliderValue1)); R1.RotateColorWheel()})
tkgrid(tklabel(FrameLeft,text="Color Scale Rotation", font=FontHeading), column=0, sticky="e")
tkgrid(slider1, column=0, sticky="e")
tkgrid(tklabel(FrameLeft,text="-180      +180"), sticky="e")
FrameRight <- tkframe(Top, relief="groove", borderwidth=2)
tkgrid(tklabel(FrameRight, text="Display Coordinates", font=FontHeading))
MinX <- tclVar("")
MinX.entry <-tkentry(FrameRight, width="12",textvariable= MinX)
MinX.g <- tclVar("unknown")
MinX.g.label <- tklabel(FrameRight,text=tclvalue(MinX.g))
tkconfigure(MinX.g.label, textvariable=MinX.g)
tkgrid(tklabel(FrameRight,text="Min X"), MinX.entry, tklabel(FrameRight, text="Global Min X ="), MinX.g.label, sticky="e")
MaxX <- tclVar("")
MaxX.entry <-tkentry(FrameRight, width="12",textvariable= MaxX)
MaxX.g <- tclVar("unknown")
MaxX.g.label <- tklabel(FrameRight,text=tclvalue(MaxX.g))
tkconfigure(MaxX.g.label, textvariable=MaxX.g)
tkgrid(tklabel(FrameRight,text="Max X"), MaxX.entry, tklabel(FrameRight, text="Global Max X ="), MaxX.g.label, sticky="e")
MinY <- tclVar("")
MinY.entry <-tkentry(FrameRight, width="12",textvariable= MinY)
MinY.g <- tclVar("unknown")
MinY.g.label <- tklabel(FrameRight,text=tclvalue(MinY.g))
tkconfigure(MinY.g.label, textvariable=MinY.g)
tkgrid(tklabel(FrameRight,text="Min Y"), MinY.entry, tklabel(FrameRight, text="Global Min Y ="), MinY.g.label, sticky="e")
MaxY <- tclVar("")
MaxY.entry <-tkentry(FrameRight, width="12",textvariable= MaxY)
MaxY.g <- tclVar("unknown")
MaxY.g.label <- tklabel(FrameRight,text=tclvalue(MaxY.g))
tkconfigure(MaxY.g.label, textvariable=MaxY.g)
tkgrid(tklabel(FrameRight,text="Max Y"), MaxY.entry, tklabel(FrameRight, text="Global Max Y ="), MaxY.g.label, sticky="e")
Coord.but <- tkbutton(FrameRight,text="OK", command=function(){
    Globals$MinX.d <<- as.numeric(tclvalue(MinX))

```

```

Globals$MaxX.d <- as.numeric(tclvalue(MaxX))
Globals$MinY.d <- as.numeric(tclvalue(MinY))
Globals$MaxY.d <- as.numeric(tclvalue(MaxY))
indexClosest <- which.min(abs(Globals$x.g - Globals$MinX.d)); Globals$MinX.d <- Globals$x.g[indexClosest]
indexClosest <- which.min(abs(Globals$x.g - Globals$MaxX.d)); Globals$MaxX.d <- Globals$x.g[indexClosest]
indexClosest <- which.min(abs(Globals$y.g - Globals$MinY.d)); Globals$MinY.d <- Globals$y.g[indexClosest]
indexClosest <- which.min(abs(Globals$y.g - Globals$MaxY.d)); Globals$MaxY.d <- Globals$y.g[indexClosest]

tclvalue(MinX) <- as.character(Globals$MinX.d)
tclvalue(MaxX) <- as.character(Globals$MaxX.d)
tclvalue(MinY) <- as.character(Globals$MinY.d)
tclvalue(MaxY) <- as.character(Globals$MaxY.d)
R1.Pan()
}
)
tkgrid(Coord.but, column=1, sticky="e")
tkgrid(tklabel(FrameRight,text="    "))
tkgrid(tklabel(FrameRight,text="    "))
Smooth <- tclVar("0")
Smooth.function <- function()
{
  Bandwidth <- as.numeric(tclvalue(Smooth))
  if(Bandwidth > 0)
  {
    XVEC <- rep(Globals$x.g, Globals$ny.g)
    YVEC <- rep(Globals$y.g, ea=Globals$nx.g)
    ImageList.x <- as.image(as.vector(Globals$u.g), x=data.frame(lon=XVEC, lat=YVEC),
      nrow= Globals$nx.g, ncol= Globals$ny.g, boundary.grid=FALSE, na.rm=TRUE)
    u.g.Smooth <- image.smooth(ImageList.x, theta = Bandwidth)
    ImageList.y <- as.image(as.vector(Globals$v.g), x=data.frame(lon=XVEC, lat=YVEC),
      nrow= Globals$nx.g, ncol= Globals$ny.g, boundary.grid=FALSE, na.rm=TRUE)
    v.g.Smooth <- image.smooth(ImageList.y, theta = Bandwidth)
    Globals$Direction <- R1.Standardize(atan2(v.g.Smooth$z, u.g.Smooth$z))
  } else Globals$Direction <- Globals$Direction.g
  R1.SubsetColorScale(Globals$Direction[Globals$StartRow:Globals$EndRow, Globals$StartCol:Globals$EndCol])
  R1.PlotImage()
}
}

```

```

Smooth.entry <-tkentry(FrameRight, width="12", textvariable=Smooth)
Smooth.but <- tkbutton(FrameRight,text="OK", command=Smooth.function)
tkgrid(tklabel(FrameRight,text="Smooth Bandwidth", font=FontHeading), Smooth.entry, sticky="e")
tkgrid(Smooth.but, column=1, sticky="e")
tkgrid(tklabel(FrameRight,text="      "))
tkgrid(tklabel(FrameRight,text="      "))
SliderValue2 <- tclVar("0")
SliderValueLabel2 <- tklabel(FrameRight,text=as.character(tclvalue(SliderValue2)))
tkconfigure(SliderValueLabel2,textvariable=SliderValue2)
slider2 <- tkscale(FrameRight, from=0, to=1, showvalue=TRUE, variable=SliderValue2, resolution=.05, orient="horizontal", length=".8i")
tkbind(slider2,"<ButtonRelease-1>", function() {Globals$ColorGap <<- as.numeric(tclvalue(SliderValue2)); R1.ChangeColorGap()})
tkgrid(tklabel(FrameRight,text="Color Scale Gap", font=FontHeading), column=0, sticky="e")
tkgrid(slider2, column=1, sticky="e")
tkgrid(tklabel(FrameRight,text="0      1"), column=1)
tkgrid(tklabel(FrameRight,text=" "), sticky="e")
tkgrid(tklabel(FrameRight,text=" "), sticky="e")
arrow.cb.value <- tclVar("0")
arrow.cb.function <- function()
{
  cbVal <- as.character(tclvalue(arrow.cb.value))
  if (cbVal=="1") {Globals$PlotArrows <<- TRUE; R1.PlotImage()}
  if (cbVal=="0") {Globals$PlotArrows <<- FALSE; R1.PlotImage()}
}
arrow.cb <- tkcheckboxbutton(FrameRight, command=arrow.cb.function)
tkconfigure(arrow.cb, variable=arrow.cb.value)
tkgrid(tklabel(FrameRight,text="Arrows", font=FontHeading), sticky="e")
tkgrid(arrow.cb, row=17, column=1, sticky="w")
arrow.length <- tclVar("1")
arrow.density <- tclVar("15")
arrow.function <- function()
{
  Globals$ArrowAdj <<- as.numeric(tclvalue(arrow.length))
  Globals$cpa <<- as.numeric(tclvalue(arrow.density))
  R1.PlotImage()
}
arrow.length.entry <-tkentry(FrameRight, width="6",textvariable=arrow.length)
tkgrid(tklabel(FrameRight,text="Arrow Length Multiplier"), arrow.length.entry,

```

```

tklabel(FrameRight, text="> 0      "), sticky="e")
arrow.density.entry <-tkentry(FrameRight, width="6",textvariable=arrow.density)
tkgrid(tklabel(FrameRight,text="Arrow Spacing in Pixels"), arrow.density.entry,
tklabel(FrameRight, text="1, 2, 3, ...  "), sticky="e")
Arrow.but <- tkbutton(FrameRight,text="OK", command=arrow.function)
tkgrid(Arrow.but, column=1, sticky="e")
tkgrid(tklabel(FrameRight,text="      "))
tkgrid(tklabel(FrameRight,text="      "))
mask.cb.value <- tclVar("0")
mask.cb.function <- function()
{
  cbVal <- as.character(tclvalue(mask.cb.value))
  if (cbVal=="1") {if(!is.null(Globals$Mask)) {Globals$PlotMask <<- TRUE; R1.PlotImage()}}
  if (cbVal=="0") {Globals$PlotMask <<- FALSE; R1.PlotImage()}}
}
mask.cb <- tkcheckboxbutton(FrameRight, command=mask.cb.function)
tkconfigure(mask.cb, variable=mask.cb.value)

tkgrid(tklabel(FrameRight,text="Mask", font=FontHeading), sticky="e")
tkgrid(mask.cb, row=23, column=1, sticky="w")
tkgrid(tklabel(FrameRight,text="      "))
tkgrid(tklabel(FrameRight,text="      "))
tkpack(FrameTop, side="top", fill="x")
tkpack(FrameLeft, side="left", fill="both", expand=TRUE)
tkpack(FrameRight, side="right", fill="both", expand=TRUE)
}

```

K.3 SimulateSill

```
SimulateSill <- function()
{
  require(CircStats)

  CircDist <- function(alpha,beta)
  {
    alpha[alpha < 0] <- 2*pi + alpha[alpha < 0]
    beta[beta < 0] <- 2*pi + beta[beta < 0]
    theta <- abs(alpha - beta)
    theta[theta > pi] <- 2*pi - theta[theta > pi]
    return(theta)
  }

  VM <- c(); U <- c(); C <- c(); WC <- c(); T <- c()
  Cavg <- vector(mode="numeric", length=1000)
  Tavg <- vector(mode="numeric", length=1000)
  Uavg <- vector(mode="numeric", length=1000)
  VMavg <- vector(mode="numeric", length=1000)
  WCavg <- vector(mode="numeric", length=1000)

  filter <- upper.tri(matrix(data=NA, nrow=100, ncol=100), diag = F)
```

```

for (i in 1:1000)
{
  Sample <- rcard(n=100,mu=0,r=.25); cosines <- cos(outer(Sample, Sample, FUN="CircDist"))
  C <- c(C, cosines[filter])
  Cavg[i] <- mean(C)

  Sample <- rtri(n=100, r=.5*4/pi^2); cosines <- cos(outer(Sample, Sample, FUN="CircDist"))
  T <- c(T, cosines[filter])
  Tavg[i] <- mean(T)

  Sample <- 2*pi*runif(100); cosines <- cos(outer(Sample, Sample, FUN="CircDist"))
  U <- c(U, cosines[filter])
  Uavg[i] <- mean(U)

  Sample <- rvm(n=100, mean=0, k=5); cosines <- cos(outer(Sample, Sample, FUN="CircDist"))
  VM <- c(VM, cosines[filter])
  VMavg[i] <- mean(VM)

  Sample <- rwrpcauchy(n=100,location=0,rho=exp(-1)); cosines <- cos(outer(Sample, Sample, FUN="CircDist"))
  WC <- c(WC, cosines[filter])
  WCavg[i] <- mean(WC)
}
return(list(Cavg=Cavg, Tavg=Tavg, Uavg=Uavg, VMavg=VMavg, WCavg=WCavg))
}

```

K.4 CorrelationTransfer

```
CorrelationTransfer <- function(nPoints=50, CircDistr2="vM", Rho2=.75, Range2=10, Ext2=2, CovModel2="spherical", GRID=NULL,
OVERFIT=TRUE)
{
  # 2008-8-10.1356
  # Circular parameters: Triangular,  $0 < \text{Rho} \leq 4/\pi^2$ ; cardioid,  $0 < \text{Rho} \leq 0.5$ ; vM and wrapped Cauchy,  $0 < \text{Rho} < 1$ ; uniform,  $\text{Rho} = 0$ 
  # nPoints= number of points per simulation
  # OverFit=TRUE, or standardization (centering and rescaling realization of the GRV to mean 0 sd 1) results in closer fit
  # for qualitative evaluation of the CRV. Undesirable effects are loss of independence of the marginal GRVs, biased GRF
  # covariance, and biased testing. Standardization is suitable for demonstration with closer fit, visualization, and
  # illustrations. Do not standardize for purposes of simulation and testing. OverFit=FALSE, or non-standardization (default)
  # includes expected variation from transformation of variation in mean and sd of sample of GRV.

  if(is.null(GRID)) {output <- SimulateCRF(N=nPoints, CircDistr=CircDistr2, Rho=Rho2, Range=Range2, Ext=Ext2,
    CovModel=CovModel2, OverFit=OVERFIT)} else {output <- SimulateCRF(CircDistr=CircDistr2, Rho=Rho2, Range=Range2,
    CovModel= CovModel2, OverFit=OVERFIT, Grid=GRID)}
  par(mfrow=c(3,1), mgp=c(1.5,.5,0), mai=c(.4,.4,.3,.1))

  # GRF variogram
  vario.z <- variog(coords = cbind(output$x, output$y), data = output$Z, option = "bin", uvec=seq(2,54,by=2))
  plot(vario.z$u, vario.z$v, main="Variogram of GRF", cex.main=1.2, xlab="Distance", ylab="Semi Variance", ylim=c(0, 2))
  abline(v=Range2, col="grey")

  # Cumulative Probability variogram
  vario.p <- variog(coords = cbind(output$x, output$y), data =pnorm(output$Z, mean=0, sd=1, lower.tail = TRUE),
    option = "bin", uvec=seq(2,54,by=2))
  plot(vario.p$u, vario.p$v, main="Variogram of Cumulative Probabilities of GRV", cex.main=1.2, xlab="Distance", ylab="Semi Variance",
  ylim=c(0,0.2))
  abline(v=Range2, col="grey")

  # Cosineogram
  CosinePlots(x=output$x, y=output$y, directions=output$direction, Lag.n.Adj= 1, Lag=vario.p$u,
    main="Cosineogram of CRF", cex.main=1.2, ylim=c(0,1))
  abline(v=Range2, col="grey")
}
```


K.5 SimulateCRF

```
SimulateCRF <-function(N=100, CircDistr, Rho, Mu=0, Range, Ext=1, CovModel, Grid=NULL, Anisotropy=NULL, OverFit=FALSE,
Resolution=0.01)
{
  # 2008-11-10.2001
  # Simulate CRF ~ (Range, CircDistr, Rho, mu=0)

  # Input Arguments
  # N: Number of spatial locations to simulate
  # CircDistr: Circular distribution in {U, vM, WrC, Tri, Card},
  # Rho: Mean resultant length parameter
  #   For triangular,  $0 < \text{Rho} \leq 4/\pi^2$ 
  #   For cardioid,  $0 < \text{Rho} \leq 0.5$ 
  #   For vM and wrapped Cauchy,  $0 < \text{Rho} < 1$ , 1== degenerate
  #   For uniform,  $\text{Rho} = 0$ 
  # Range: Distance at which CRV independent
  # Ext: Range*Ext is horizontal and vertical length of sample space
  # CovModel: Name of spatial correlation function, see package geoR Help cov.spatial
  # Grid: Regular or irregular N x 2 matrix of simulation locations, overrides N and Ext
  # Anisotropy: Vector of geometric anisotropy angle in radians, ratio > 1.
  # OverFit=TRUE, or standardization (centering and rescaling realization of the GRV to mean 0 sd 1) results in closer fit
  # for qualitative evaluation of the CRV. Undesirable effects are loss of independence of the marginal GRVs, biased GRF
  # covariance, and biased testing. Standardization is suitable for demonstration with closer fit, visualization, and
  # illustrations. Do not standardize for the purposes of simulation and testing. OverFit=FALSE, or non-standardization (default)
  # includes expected variation from transformation of variation in mean and sd of sample of GRV.

  # Values
  # x,y: Vectors of location coordinates
  # direction: Vector of directions
  # Z: Vector of simulated observations of the associated GRV

  # Note:
  # At n > 500, geoR transfers processing the the package Random Fields because the option RF is set.

  if(CircDistr != "U" & CircDistr != "vM" & CircDistr != "WrC" & CircDistr != "Tri" & CircDistr != "Card")
    stop("CRF not implemented for input CircDistr")
}
```

```

if(CircDistr == "U") Rho = 0
if(abs(Mu) > pi) stop("abs(Mu) <= pi")

if(!is.null(Grid)) {
  if(class(Grid) != "matrix") stop("Grid not a matrix")
  if(dim(Grid)[2] != 2) stop("Grid not a N x 2 matrix")
  N <- dim(Grid)[1]}
if(!is.null(Anisotropy)) {
  if(length(Anisotropy) != 2) stop("Anisotropy is not a 2 element vector. See geoR Help")
}

if(N <= 0 | Rho < 0 | Range < 0 | Ext <= 0 | Resolution <= 0) stop("Improper numeric input")

direction <- vector(mode="numeric", length=N)

require(CircStats)
require(geoR)
# Standard normal GRF, see Help geoR grf
if(is.null(Grid)) {
  GRF <- grf(n=N, xlims=c(0, Range*Ext), ylims=c(0, Range*Ext), cov.model=CovModel,
  nugget=0, cov.pars=c(1, Range), aniso.pars=Anisotropy, RF=TRUE, messages=FALSE) } else {
  GRF <- grf(grid=Grid, cov.model=CovModel,
  nugget=0, cov.pars=c(1, Range), aniso.pars=Anisotropy, RF=TRUE, messages=FALSE)}

XY <- GRF$coords # N x 2 matrix
x <- XY[,1]; y <- XY[,2]
Z <- GRF$data # Vector of GRV
if(OverFit) {Z <- (Z - mean(Z))/sd(Z); GRF$data <- Z}
CumProbZ <- pnorm(Z, mean=0, sd=1, lower.tail = TRUE)

if(CircDistr=="U") {direction <- -pi + 2*pi*CumProbZ} else
if(CircDistr == "Tri")
{
  if(Rho==0 | Rho > 4/pi^2) stop("Tri: 0 < Rho <= 4/pi^2")
  filter <- CumProbZ < 0.5
  u1 <- CumProbZ[filter]
}

```

```

a <- Rho/8
b <- (4+pi^2*Rho)/(8*pi)
c <- 0.5 - u1
q <- -.5*(b+sqrt(b^2-4*a*c))
direction[filter] <- c/q

u2 <- CumProbZ[!filter]
a <- -Rho/8
b <- (4+pi^2*Rho)/(8*pi)
c <- 0.5 - u2
q <- -.5*(b+sqrt(b^2-4*a*c))
direction[!filter] <- c/q
} else
{
# For OTHER circular distributions compute table of circular CDF and interpolate
CircScale <- seq(-pi, pi, length=2*pi/Resolution)
# With resolution=.01, circular support from -pi to +pi has 629 elements, delta ~0.01000507, CircScale[315] = 0
n <- length(CircScale)
if(CircDistr == "vM")
{
if(Rho==0 | Rho >= 1) stop("vM: 0 < Rho < 1")
CircProb <- rep(-1, n)
Kappa=A1inv(Rho) # N. I Fisher, Statistical Analysis of Circular Data, 2000 p. 49
# As direction increases from -pi, pvm increases from .5
for(i in 1:length(CircScale)) CircProb[i] <- pvm(CircScale[i], mu=0, kappa=Kappa)
filter <- CircScale < 0
CircProb[filter] <- CircProb[filter] - 0.5
CircProb[!filter] <- CircProb[!filter] + 0.5
} else
if(CircDistr == "Card")
{
if(Rho==0 | Rho > 0.5) stop("Cardioid: 0 < Rho <= 0.5")
CircProb <- (CircScale + pi + 2*Rho*sin(CircScale))/(2*pi)
} else
if(CircDistr == "WrC")
{
if(Rho==0 | Rho >= 1) stop("Wrapped Cauchy: 0 < Rho < 1 ")

```

```

Angles1 <- CircScale[CircScale < 0]
Angles2 <- CircScale[CircScale >= 0]
prob1 <- 0.5 - acos(((1+Rho^2)*cos(Angles1) - 2*Rho)/(1 + Rho^2 - 2*Rho * cos(Angles1)))/(2*pi)
prob2 <- 0.5 + acos(((1+Rho^2)*cos(Angles2) - 2*Rho)/(1 + Rho^2 - 2*Rho * cos(Angles2)))/(2*pi)
CircProb <-c(prob1, prob2)
}
CircProb[1] <- 0; CircProb[n] <- 1

# Interpolation
DeltaTh <- CircScale[2] + pi
for(i in 1:N)
{
  p <- CumProbZ[i] # Cumulative prob of GRV
  a <- max((1:n)[CircProb <= p]) # Index
  if(a==n) {r <- 0} else
  {
    if(CircProb[a]==p) {r <- 0} else {r <- (p -CircProb[a])/( CircProb[a+1] -CircProb[a])}
  }
  direction[i] <- CircScale[a] + r*DeltaTh
}
}

direction <- direction + Mu
return(list(x=x, y=y, direction=direction, Z=Z))
}

```

K.6 AssessCRF

```
AssessCRF <- function(nPoints=100, CircDistr2="vM", Rho2=.75, Range2=10, Ext2=3, CovModel2="spherical", GRID=NULL, OVERFIT=TRUE)
{
  # 2008-2-15.0600
  # Generate a nPoints x nPoints and compare QQ Circ plots to QQ norm plots
  # range of parameters
  # For triangular,  $0 < \text{Rho} \leq 4/\pi^2$ 
  # For cardioid,  $0 < \text{Rho} \leq 0.5$ 
  # For vM and wrapped Cauchy,  $0 < \text{Rho} < 1$ , 1== degenerate
  # For uniform,  $\text{Rho} = 0$ 
  # nPoints= number of points
  # OverFit=TRUE, or standardization (centering and rescaling realization of the GRV to mean 0 sd 1) results in closer fit
  # for qualitative evaluation of the CRV. Undesirable effects are loss of independence of the marginal GRVs, biased GRF
  # covariance, and biased testing. Standardization is suitable for demonstration with closer fit, visualization, and
  # illustrations. Do not standardize for purposes of simulation and testing. OverFit=FALSE, or non-standardization (default)
  # includes expected variation from transformation of variation in mean and sd of sample of GRV.

  require(CircStats)
  if(is.null(GRID)) {output <- SimulateCRF(N=nPoints, CircDistr=CircDistr2, Rho=Rho2, Range=Range2, Ext=Ext2,
    CovModel=CovModel2, OverFit=OVERFIT)} else {output <- SimulateCRF(CircDistr=CircDistr2, Rho=Rho2, Range=Range2,
    CovModel= CovModel2, OverFit=OVERFIT, Grid=GRID); nPoints <- nrow(GRID) }

  Z <- output$Z
  Zsort <- sort(Z)
  a <- ifelse(nPoints <= 10, 3/8, 1/2)
  CumProb <- ((1:nPoints)- a)/(nPoints + 1 - 2*a) # Vector of symmetric cumulative probabilities for QQ plots
  ZQuantiles <- qnorm(CumProb, mean=0, sd=1, lower.tail = TRUE)

  Theta <- output$direction
  Thetasort <- sort(Theta)
  # Compute theta quantiles
  if(CircDistr2=="U") { ThetaQuantiles <- -pi + 2*pi*CumProb } else
  {
    # For non-uniform circular distributions use circular CDF to get ThetaQuantiles
    CircScale <- seq(-pi, pi, length=2*pi/.01) # Circular support from -pi to +pi, 629 elements, d~.01, CircScale[315] is zero
    n <- length(CircScale)
  }
}
```

```

if(CircDistr2=="vM")
{
  if(Rho2==0 | Rho2 >= 1) stop("vM: 0 < Rho < 1")
  CircProb <- rep(-1, n)
  Kappa=A1inv(Rho2) # N. I Fisher, Statistical Analysis of Circular Data, 2000 p. 49
  # As theta increases from -pi, pvm increases from .5
  for(i in 1:n) {CircProb[i] <- pvm(CircScale[i], mu=0, kappa=Kappa)}
  filter <- CircScale < 0
  CircProb[filter] <- CircProb[filter] - 0.5
  CircProb[!filter] <- CircProb[!filter] + 0.5
} else
if(CircDistr2=="Tri")
{
  if(Rho2==0 | Rho2 > 4/pi^2) stop("Tri: 0 < Rho <= 4/pi^2")
  Angles1 <- CircScale[CircScale < 0] + 2*pi
  Angles2 <- CircScale[CircScale >= 0]
  CircProb <- c( (4 - 3*pi^2*Rho2 + pi*Rho2*(Angles1 + pi))*(Angles1-pi)/(8*pi),
    .5 + (4 + pi^2* Rho2 - pi*Rho2 * Angles2) * Angles2/(8*pi) )
} else
if(CircDistr2=="Card")
{
  if(Rho2==0 | Rho2 > 0.5) stop("Cardioid: 0 < Rho <= 0.5")
  Angles1 <- CircScale[CircScale < 0] + 2*pi
  Angles2 <- CircScale[CircScale >= 0]
  CircProb <- c( (Angles1 - pi + 2*Rho2*sin(Angles1))/(2*pi), 0.5 + (Angles2 + 2*Rho2*sin(Angles2))/(2*pi) )
} else
if(CircDistr2=="WrC")
{
  if(Rho2==0 | Rho2 >= 1) stop("Wrapped Cauchy: 0 < Rho < 1")
  Angles1 <- CircScale[CircScale < 0] + 2*pi
  Angles2 <- CircScale[CircScale >= 0]
  CircProb <- c( 0.5 - acos(((1+Rho2^2)*cos(Angles1) - 2*Rho2)/(1 + Rho2^2 - 2*Rho2 *
    cos(Angles1)))/(2*pi), 0.5 + acos(((1+Rho2^2)*cos(Angles2) - 2*Rho2)/(1 + Rho2^2 - 2*Rho2 * cos(Angles2)))/(2*pi) )
}
CircProb[1] <- 0; CircProb[n] <- 1 # For any numerical imprecision

# Quantiles From Inverse Circular CDF

```

```

ThetaQuantiles <- vector(mode="numeric", length=nPoints)
DeltaTh <- CircScale[2] + pi
for(i in 1:nPoints)
{
  p <- CumProb[i]
  a <- max((1:n)[CircProb <= p]) # Left index
  if(a==n) { r <- 0} else { if(CircProb[a]==p) {r <- 0} else {r <- (p - CircProb[a])/(CircProb[a+1]-CircProb[a])} }
  ThetaQuantiles[i] <- CircScale[a] + r*DeltaTh
}
}
par(mfrow=c(3,2), mgp=c(2,1,0), mar=c(4.1,3.1,3.1,1.1))

# QQ std norm
plot(ZQuantiles, Zsort, main=paste("QQ Standard Normal of", "\nGRV With Spatial Correlation", sep=""), cex.main=1,
      xlab="Theoretical Quantiles", ylab="Ordered GRV", col=1, xlim=c(-pi,pi),ylim=c(-pi,pi), ty='l')
abline(0,1,col=4); abline(v=0, col="grey"); abline(h=0, col="grey")

# GRF variogram
vario.b <- variog(coords = cbind(output$x, output$y), data = Z, option = "bin")
plot(vario.b$u, vario.b$v, main=paste("Variogram of GRF", "\n Model=", CovModel2, ", Range=", Range2, ", Sill=1", ", mean=0", sep=""),
      cex.main=1, xlab="Distance", ylab="Semi Variance")
abline(v=Range2, col="grey")
abline(h=1, col="grey")

# QQ Circ probability law
if(CircDistr2=='U') {Distrib = "Uniform"} else
if(CircDistr2=='vM') {Distrib = "von Mises"} else
if(CircDistr2=='WrC') {Distrib = "Wrapped Cauchy"} else
if(CircDistr2=='Tri') {Distrib = "Triangular"} else
if(CircDistr2=='Card') {Distrib = "Cardioid"}

```

```

if(CircDistr2=="U") {title.rho=""} else {title.rho=paste(" Rho =", round(Rho2, digits=3), sep="")}
plot(ThetaQuantiles, Thetasort, main=paste( "QQ ", Distrib, title.rho, "\nCRV With Spatial Correlation",sep="" ), cex.main=1,
     xlab="Theoretical Quantiles (Rad)", ylab="Ordered CRV", col=1, xlim=c(-pi,pi),ylim=c(-pi,pi), ty='l')
lines(c(-pi,pi), c(-pi,pi), col=4); abline(v=0, col="grey"); abline(h=0, col="grey")

# Cosineogram
CosinePlots(x=output$x, y=output$y, directions=output$direction, Lag.n.Adj= 1, Lag=vario.b$u, main="Cosineogram of CRF")
if(CircDistr2!="U") abline(h=Rho2^2, col="grey") else abline(h=0, col="grey")
abline(h= est.rho(Theta)^2, col=4, lty=3) # Sample mean resultant length
abline(v=Range2, col="grey")

# Uniformity plot
probabilities <- pnorm(Z, mean=0, sd=1, lower.tail = TRUE)
uniformity <- mean(abs(CumProb - sort(probabilities))) # Mean absolute deviation
plot(CumProb, sort(probabilities) , main=paste( "QQ Uniform of Cumulative Probabilities",
      "\nMean - 1/2=", round(mean(probabilities)-.5, digits=3), ", Var - 1/12=", round((sd(probabilities))^2-1/12, digits=3),
      ", Closeness=", round(uniformity, digits=3), sep="" ), cex.main=1,
     xlab="Theoretical Quantiles", ylab="Ordered Probabilities", col=1, xlim=c(0,1),ylim=c(0,1), ty='l')
lines(c(0,1), c(0,1),col=4); abline(v=.5, col="grey"); abline(h=.5, col="grey")
}

```


K.7 PlotVectors

```
PlotVectors <- function(x, y, h, v, UnitVector=TRUE, Trilcon=FALSE, AdjArrowLength=1, AdjHeadLength=1, TrilconAdj=1,
  TriRatio=4, JitterPlot=FALSE, Jitter=1, ...)
{
  # 2008-11-11.1535
  # Arrows do not plot where data is missing.
  require(fields)

  if( (length(x) != length(y)) | (length(h) != length(v)) | (length(x) != length(h)) ) stop("lengths of vector inputs unequal")

  filter <- is.na(h) | is.na(v) | (h==0 & v==0)
  x <- x[!filter]; y <- y[!filter]; h <- h[!filter]; v <- v[!filter]
  # fields function arrows omits arrowheads with a warning on any arrow of length less than 1/1000 inch.

  Dir <- atan2(v, h)
  Dir[Dir<0] <- Dir[Dir<0]+2*pi
  if(JitterPlot==TRUE)
  {
    x <- x + Jitter*runif(length(x))
    y <- y + Jitter*runif(length(y))
  }
  plot(x, y, ty="n", asp=1, ...)

  if(UnitVector)
  { arrow.plot(x, y, cos(Dir), sin(Dir), true.angle=TRUE, arrow.ex=AdjArrowLength*0.05, length=AdjHeadLength*0.125,
    angle=20, xpd=FALSE)
  } else
  {
    if(Trilcon)
    {
      m <- sqrt(h^2 + v^2) # magnitude
      w = sqrt(m/TriRatio)
      n <- length(x)
      xa <- x + TrilconAdj* w*cos(Dir+pi/2)
      ya <- y + TrilconAdj* w*sin(Dir+pi/2)
      xb <- x + TrilconAdj*TriRatio*w*cos(Dir)
    }
  }
}
```

```
    yb <- y + TrilconAdj*TriRatio*w*sin(Dir)
    xc <- x + TrilconAdj*   w*cos(Dir-pi/2)
    yc <- y + TrilconAdj*   w*sin(Dir-pi/2)

    for(i in 1:n) polygon(x=c(xa[i], xb[i], xc[i]), y=c(ya[i], yb[i], yc[i]), density=-1, col=1)
  } else arrow.plot(x, y, h, v, true.angle=TRUE, arrow.ex=AdjArrowLength*0.05, length=AdjHeadLength*0.125,
    angle=20, xpd=FALSE)
}
```

K.8 CircResidual

```
CircResidual <- function(X, Y, Raw, Trend, Plot = FALSE, AdjArrowLength = 1, ...)
{
  # 2008-11-10.2053
  # Assumptions: Raw may have NAs, trend has no NAs. Trend locations and Raw locations are identical to compute residuals.
  require(fields)
  if((length(X) != length(Y)) | (length(X) != length(Raw)) | (length(X) != length(Trend)) | (length(Y) != length(Raw)) |
      (length(Y) != length(Trend)) | (length(Raw) != length(Trend))) stop("lengths of vector inputs unequal")
  if(AdjArrowLength <= 0) stop("AdjArrowLength invalid")
  if(sum(is.na(Trend)) > 0) stop("NAs not allowed in Trend")

  FilterNA <- is.na(Raw)
  x <- X[!FilterNA]; y <- Y[!FilterNA]; raw <- Raw[!FilterNA]; trend <- Trend[!FilterNA]
  raw[raw < 0] <- raw[raw < 0] + 2*pi # Like R1.Standardize in CircDataimage
  trend[trend < 0] <- trend[trend < 0] + 2*pi
  circdist <- abs(raw - trend) # Linear distance in radians with NAs where raw has NAs
  circdist[circdist > pi] <- 2*pi - circdist[circdist > pi] # Circular distance in radians
  resids <- circdist
  filter <- (trend>raw) & (trend-raw)<pi | (raw >trend) & (raw-trend)>pi
  resids[filter] <- -1* circdist[filter]
  if(Plot==TRUE)
  {
    plot(X, Y, type="n", xlab="", ylab="", asp=1, ...)
    arrow.plot(x, y, u=cos(raw), v=sin(raw), xpd=FALSE, true.angle=TRUE, arrow.ex=.15*AdjArrowLength, length=.1,col=1)
    arrow.plot(X, Y, u=cos(Trend), v=sin(Trend), xpd=FALSE, true.angle=TRUE, arrow.ex=.15*AdjArrowLength, length=.1,
               col="tan", lwd=3)
    arrow.plot(x, y, u=cos(resids), v=sin(resids), xpd=FALSE, true.angle=TRUE, arrow.ex=.15*AdjArrowLength, length=.1,
               col=2, lty=2)
  } else return(list(x=x, y=y, direction=resids))
}
```

K.9 CosinePlots

```
CosinePlots <- function(x, y, directions, Lag=NULL, Lag.n.Adj = 1, BinWAdj=1, Plot = TRUE,
  Cloud = FALSE, Model=FALSE, nugget=0, Range=NULL, sill=NULL, x.legend=0.6, y.legend=1.0, TrimMean=0.1, ...)
{
  # 2008-11-11.1050
  # Assumption: Isotropic circular random field
  # x, y are vectors of location coordinates, directions is a vector of directions in radians.
  # Lag is a vector of lag points. Lag.n.Adj > 0 multiplies the number of lag points.
  # BinWAdj >= 1 multiplies bin width (to make bins narrower increase Lag.n.adj). Sturges rule determines nBins.
  # nBins and Lag.n.Adj determine Lag.n. Lag.n adjusts nBins. nBins and BinWAdj determine bin width.
  # Plot = TRUE plot cosinecloud or cosineogram, else output list of points. Cloud = TRUE plots cosinecloud, else cosineogram.
  # Model = TRUE overplots exponential, gaussian, and spherical models with nugget, Range, and sill parameters.
  # x.legend and y.legend adjust legend location.
  # TrimMean = 0.1 applies trimmed mean in computing mean cosine.

  if( (length(x) != length(y)) | (length(x) != length(directions)) | (length(y) != length(directions)) )
    stop("lengths of vector inputs unequal")
  if(Lag.n.Adj <= 0) stop("Lag.n.Adj invalid")
  if( (nugget < 0) | (nugget > 1) ) stop("nugget invalid")
  if(!is.null(Range)) {if(Range <= 0) stop("Range negative") }
  if(!is.null(sill)) {
    if( (sill < 0) | (sill >=1) ) stop("sill invalid")
    if(1-nugget < sill) stop("1-nugget < sill")}

  # Repair Input and Remove missings
  if(BinWAdj < 1) BinWAdj <- 1 # points will fall out of bins if adjust < 1
  filter <- !is.na(directions)
  x <- x[filter]; y <- y[filter]; directions <- directions[filter]

  # Pairwise cosines
  # Subroutine to compute circular distances in radians
  CircDist <- function(alpha,beta)
  {
    alpha[alpha < 0] <- 2*pi + alpha[alpha < 0]
    beta[beta < 0] <- 2*pi + beta[beta < 0]
    theta <- abs(alpha - beta)
```

```

    theta[theta > pi] <- 2*pi - theta[theta > pi]
    return(theta)
}
Cosines <- cos(outer(directions, directions, FUN="CircDist"))
filter.tri <- upper.tri(Cosines)
Cosines <- Cosines[filter.tri]

# Pairwise distances
Distances <- as.matrix(dist(cbind(x, y))) # Diagonal of zero distances
X <- Distances[filter.tri] # vector of distances corresponding to vector of cosines Cosines

if(Cloud) {Y <- Cosines} else
{
  if(!is.null(Lag))
  {
    # Assumes equally spaced lags, except for first lag point
    HalfBinWidth <- BinWAdj * 0.5 * (Lag[2] - Lag[1])
    Lag.n <- length(Lag)
  } else
  {
    nBins <- trunc(log2(sum(filter.tri)) + 1) # Sturges rule
    Lag.n <- trunc(Lag.n.Adj*(nBins + 1))
    nBins <- Lag.n - 1
    distance.max <- max(X)
    HalfBinWidth <- BinWAdj * 0.5 * distance.max/nBins
    Lag <- seq(0, distance.max, length.out= Lag.n)
  }

  Y <- vector(mode = "numeric", length = Lag.n)
  if(Lag[1] == 0) {Y[1] <- 1; i1 <- 2} else i1 <- 1
  for (i in i1:Lag.n)
  {
    filter <- abs(X - Lag[i]) <= HalfBinWidth
    Y[i] <- mean(Cosines[filter], trim=TrimMean)
  }

  X <- Lag # Cloud=FALSE filtered distances replaced by lag vector

```

```

}
if(Plot)
{
  plot(X, Y, xlab = "Distance", ylab = "Cosine", cex.main=1, ...)
  if(Cloud == FALSE & Model == TRUE)
  {
    xx <- seq(min(X), max(X), length.out=101)
    c.e <- 1-nugget -(1-nugget-sill)*(1-exp(-3*xx/Range))
    c.g <- 1-nugget -(1-nugget-sill)*(1-exp(-3*(xx/Range)^2))
    X1 <- xx[xx <= Range]
    c.s <- 1-nugget -(1-nugget-sill)*(1.5*X1/Range-0.5*(X1/Range)^3)
    X2 <- xx[xx > Range]
    c.s <- c(c.s, rep(sill, length(X2)))
    lines(xx, c.e, col=2, lty=1, lwd=1)
    lines(xx, c.g, col="tan", lty=1, lwd=3)
    lines(xx, c.s, col=4, lty=2, lwd=1)

    legend(x=x.legend*max(X), y=y.legend, c("Exponential", "Gaussian", "Spherical"),
           lty = c(1, 1, 2), col=c(2, "tan", 4), lwd=c(1, 3, 1), cex=1.1)
  }
} else {return(list(distance = X, cosine = Y))}
}

```

K.10 KrigCRF

```
KrigCRF <- function(krig.x, krig.y, resid.x, resid.y, resid.direction, Model, Nugget=0, Range, sill,
  Smooth=FALSE, bandwidth, Plot=FALSE, Xlim=NULL, Ylim=NULL, PlotVar=FALSE, ...)
{
  # 2008-11-11.1213
  # select model from covariance models in R package Random Fields functin CovarianceFct
  # resid.x and resid.y have no NAs

  if( (length(krig.x) != length(krig.y)) | (length(resid.x) != length(resid.y)) | (length(resid.x) != length(resid.direction)) |
    (length(resid.y) != length(resid.direction)) ) stop("lengths of vector inputs unequal")
  if( (Nugget < 0) | (Nugget > 1) ) stop("Nugget invalid")

  # fix the order of the kriging coordinates
  xx <- sort(unique(krig.x)); yy <- sort(unique(krig.y))
  nx <- length(xx); ny <- length(yy) # rectangular or square grid
  krig.y <- rep(yy, nx); krig.x <- rep(xx, each=ny)

  require(fields)
  require(RandomFields)

  Distances <- as.matrix(dist(cbind(resid.x, resid.y)))
  Ncol <- ncol(Distances)
  K <- c()
  for (i in 1:Ncol)
  {
    K <- cbind(K, sill + (1 - Nugget - sill)*CovarianceFct(x=Distances[, i]/Range, model=Model,
      param=c(mean=0,variance=1,nugget=0,scale=1, ...), dim=1, fctcall="Covariance"))
  }
  diag(K) <- 1 # TRUE even if nugget > 0 for any model
  Kinv <- solve(K)

  U <- t(cbind(cos(resid.direction), sin(resid.direction)))
  # V <- t(U) %*% U

  n <- length(krig.x) # krig.x=krig.y for square or rect grid
  krig.direction <- vector(mode="numeric", length=n)
```

```

krig.variance <- krig.direction

for(i in 1:n)
{
  distances <- sqrt((krig.x[i]-resid.x)^2 + (krig.y[i]-resid.y)^2)
  c <- sill + (1 - Nugget - sill)*CovarianceFct(x=distances/Range, model=Model,
    param=c(mean=0,variance=1,nugget=0,scale=1, ...), dim=1, fctcall="Covariance")
  c[distances == 0] <- 1 # TRUE even if nugget > 0 for any model
  w <- Kinv %*% c

  # w <- (Kinv %*% c)/sqrt(as.numeric(t(c) %*% Kinv %*% V %*% Kinv %*% c)) # gives same directions
  u <- U %*% w
  krig.direction[i] <- atan2(u[2],u[1])
  krig.variance[i] <- 2 - 2*sqrt(as.numeric(t(c) %*% Kinv %*% c))
}

if(Smooth)
{
  xx.dx <- xx[2] - xx[1]; yy.dy <- yy[2] - yy[1]
  # as.image loads the matrix by row
  ImageList.x <- as.image(cos(krig.direction), x=data.frame(krig.x, krig.y), nrow=nx, ncol=ny, boundary.grid=FALSE)
  smooth.x <- image.smooth( ImageList.x, theta = bandwidth)
  ImageList.y <- as.image(sin(krig.direction), x=data.frame(krig.x, krig.y), nrow=nx, ncol=ny, boundary.grid=FALSE)
  smooth.y <- image.smooth( ImageList.y, theta = bandwidth)
  krig.direction <- as.vector(t(atan2(smooth.y$, smooth.x$)))
}

if(Plot)
{
  if(!PlotVar)
  {
    plot(krig.x, krig.y, ty="n", xlab="", ylab="", asp=1, xlim=Xlim, ylim=Ylim)
    arrow.plot(a1=krig.x, a2=krig.y, u=cos(krig.direction), v=sin(krig.direction), arrow.ex=0.06,
      xpd=FALSE, true.angle=TRUE, length=.05, col=1)
  } else
  {
    krig.variance.matrix <- matrix(data=krig.variance, nrow=nx, ncol=ny, byrow=TRUE)
  }
}

```



```
if(!is.null(Xlim))
{
  filled.contour(x = xx, y = yy, z = krig.variance.matrix, xlim=Xlim, ylim=Ylim,
  color = terrain.colors, key.title = title(main="Circ Krig \nVariance", cex.main=0.8),
  asp = 1, plot.axes={axis(1); axis(2); points(resid.x, resid.y, pch=20, cex=.65)})
} else {
  filled.contour(x = xx, y = yy, z = krig.variance.matrix,
  color = terrain.colors, key.title = title(main="Circ Krig \nVariance", cex.main=0.8),
  asp = 1, plot.axes={axis(1); axis(2); points(resid.x, resid.y, pch=20, cex=.65)})
}
}
} else return(list(x=krig.x, y=krig.y, direction=krig.direction, variance=krig.variance))
}
```

K.11 InterpDirection

```
InterpDirection <- function(in.x, in.y, in.direction, out.x, out.y)
{
  # 2008-11-11.1444
  # Interpolate models of direction cosines and sines, separately to avoid cross over. Fit plane to triangular half of cell
  # (rectangular element of regular grid of measurement locations) in which interpolation location occurs.
  # Assumptions - Locations to interpolate are within range of (in.x, in.y), inputs have no missing.

  # Arguments
  # in.x vector of input horizontal coordinates
  # in.y vector of input vertical coordinates
  # in.direction vector of input direction in radians

  # Value
  # out.x vector of interpolation output horizontal coordinates
  # out.y vector of interpolation output vertical coordinates
  # out.direction vector of interpolation output direction

  # Verify input
  minx.in <- min(in.x); maxx.in <- max(in.x); miny.in <- min(in.y); maxy.in <- max(in.y)
  minx.out <- min(out.x); maxx.out <- max(out.x); miny.out <- min(out.y); maxy.out <- max(out.y)
  if(minx.out < minx.in | maxx.out > maxx.in | miny.out < miny.in | maxy.out > maxy.in)
    stop("Interpolation range exceeds range of (in.x, in.y)")

  if( (length(in.x) != length(in.y)) | (length(in.x) != length(in.direction)) | (length(in.y) != length(in.direction)))
    stop("lengths of vector inputs unequal")

  if( length(out.x) != length(out.y) ) stop("lengths of vector outputs unequal")

  # Organize model data
  X <- sort(unique(in.x), decreasing = FALSE) # Increases left to right
  m <- length(X)
  Y <- sort(unique(in.y), decreasing = TRUE) # Decreases top to bottom
  n <- length(Y)
  # Col of matrix of directions reflects the horiz or west to east component location
  xmin <- min(X); deltax <- X[2] - X[1]; ymax <- max(Y); deltay <- Y[1] - Y[2]
```

```

Col <- 1 + (in.x -xmin)/deltax
Row <- 1 + (ymax - in.y)/deltay
directions <- matrix(data = NA, nrow=n, ncol=m)
directions[cbind(Row, Col)] <- in.direction # matrix of organized directions
U <- cos(directions) # matrix of organized cosines of directions
V <- sin(directions) # matrix of organized sines of directions

n <- length(out.x)

CosOut <- rep(NA, n) # for interpolated cosine
SinOut <- CosOut # for interpolated sin

p <- 1:length(X)
q <- 1:length(Y)

for(i in 1:n)
{
  xx <- out.x[i]
  yy <- out.y[i]
  Vert=FALSE; Horiz=FALSE
  if(sum(X==xx)==1) Vert=TRUE
  if(sum(Y==yy)==1) Horiz=TRUE

  if(Vert==FALSE & Horiz==FALSE)
  {
    west <- max(p[X <= xx])
    east <- west + 1
    south <- min(q[Y <= yy])
    north <- south - 1
    x.west <- X[west]
    x.east <- X[east]
    y.south <- Y[south]
    y.north <- Y[north]
    cos.nw <- U[north,west]
    cos.ne <- U[north,east]
    cos.sw <- U[south,west]
    cos.se <- U[south,east]
  }
}

```

```

sin.nw <- V[north,west]
sin.ne <- V[north,east]
sin.sw <- V[south,west]
sin.se <- V[south,east]

m <- (y.north-y.south)/(x.east-x.west) # 1 if vert res=horiz res
b <- y.north-m*x.east
ydiag <- m*xx+ b
if(yy <= ydiag) # On diagonal or in lower triangular
{
  # Fit plane to lower triangular
  AB <- c(x.east-x.west, 0, cos.se-cos.sw)
  AC <- c(x.east-x.west, y.north-y.south, cos.ne-cos.sw)
  # Coefficients of cross product AB X AC
  a <- AB[2]*AC[3]-AB[3]*AC[2]
  b <- AB[3]*AC[1]-AB[1]*AC[3]
  c <- AB[1]*AC[2]-AB[2]*AC[1]
  CosOut[i] <- cos.sw + (a*(x.west-xx) + b*(y.south-yy))/c
  # Fit plane to lower triangular
  AB <- c(x.east-x.west, 0, sin.se-sin.sw)
  AC <- c(x.east-x.west, y.north-y.south, sin.ne-sin.sw)
  # Coefficients of cross product AB X AC
  a <- AB[2]*AC[3]-AB[3]*AC[2]
  b <- AB[3]*AC[1]-AB[1]*AC[3]
  c <- AB[1]*AC[2]-AB[2]*AC[1]
  SinOut[i] <- sin.sw + (a*(x.west-xx) + b*(y.south-yy))/c
}
else
{
  # In upper triangular
  AC <- c(x.east-x.west, y.north-y.south, cos.ne-cos.sw)
  AD <- c(0,y.north-y.south, cos.nw-cos.sw)
  a <- AC[2]*AD[3]-AC[3]*AD[2]
  b <- AC[3]*AD[1]-AC[1]*AD[3]
  c <- AC[1]*AD[2]-AC[2]*AD[1]
  CosOut[i] <- cos.sw + (a*(x.west-xx) + b*(y.south-yy))/c
  AC <- c(x.east-x.west, y.north-y.south, sin.ne-sin.sw)
}

```

```

        AD <- c(0,y.north-y.south, sin.nw-sin.sw)
        a <- AC[2]*AD[3]-AC[3]*AD[2]
        b <- AC[3]*AD[1]-AC[1]*AD[3]
        c <- AC[1]*AD[2]-AC[2]*AD[1]
        SinOut[i] <- sin.sw + (a*(x.west-xx) + b*(y.south-yy))/c
    }
}
else if(Vert==TRUE & Horiz==FALSE)
{
    p1 <- p[X==xx] # Column of vert grid line
    q1 <- min(q[Y < yy]) # Even spacing is not assumed
    q2 <- q1 - 1
    cos1 <- U[q1, p1]
    cos2 <- U[q2, p1]
    CosOut[i] <- cos1 + (cos2-cos1)*(yy-Y[q1])/(Y[q2]-Y[q1])
    sin1 <- V[q1, p1]
    sin2 <- V[q2, p1]
    SinOut[i] <- sin1 + (sin2- sin1)*(yy-Y[q1])/(Y[q2]-Y[q1])
}
else if(Vert==FALSE & Horiz==TRUE)
{
    q1 <- q[Y==yy] # Row of horiz grid line
    p1 <- max(p[X < xx])
    p2 <- p1 + 1
    cos1 <- U[q1, p1]
    cos2 <- U[q1, p2]
    CosOut[i] <- cos1 + (cos2-cos1)*(xx-X[p1])/(X[p2]-X[p1])
    sin1 <- V[q1, p1]
    sin2 <- V[q1, p2]
    SinOut[i] <- sin1 + (sin2-sin1)*(xx-X[p1])/(X[p2]-X[p1])
}
else # Vert==TRUE & Horiz==TRUE
{
    CosOut[i] <- U[q[Y==yy], p[X==xx]]
    SinOut[i] <- V[q[Y==yy], p[X==xx]]
}
}
}

```

```
dir <- atan2(SinOut, CosOut)
return(list(x=out.x, y=out.y, direction=dir))
}
```

K.12 CircMedianPolish

```

CircMedianPolish <- function(x, y, h, v, delta, MaxIter=20, tol=0.01)
{
  # 2007-07-06
  # Inputs:
  # x = vector of longitudes (x coordinates); y=vector of latitudes (y coordinates).
  # h = vector of Horizontal components of vectors; v=vector of Vertical component of vectors.
  # delta is lattice horizontal and vertical spacing. Why should the horiz & vert spacings be different?
  # Assumptions:
  # Direction measured on regular lattice, but doesn't have a specific order. Outputs are ordered in the code.
  # Process:
  # 1) Vector h is organized into matrix H. Vector v is organized into matrix V.
  # 2) Median polish is performed on matrices H and V separately.
  # 3) Polished direction is determined by applying atan2 to (V, H).
  # Outputs:
  # X = vector of ordered longitudes
  # Y = vector of ordered latitudes
  # polished.dir = vector of ordered polished directions
  # raw.dir = vector of ordered raw directions, which may have NAs

  MinX <- min(x); MaxX <- max(x); MinY <- min(y); MaxY <- max(y)
  Cols <- 1 + (x -MinX)/delta # For indexing h and v into matrix format
  Rows <- 1 + (MaxY - y)/delta # For indexing h and v into matrix format
  H <- matrix(data = NA, nrow=1 + (MaxY-MinY)/delta, ncol=1+ (MaxX-MinX)/delta); V <- H; Dirs <- H
  # Organize the components and data
  H[cbind(Rows, Cols)] <- h
  V[cbind(Rows, Cols)] <- v
  Dirs[cbind(Rows, Cols)] <- atan2(v,h)

  polish.H <- medpolish(H, maxiter=MaxIter, trace.iter=TRUE, na.rm=T, eps=tol) # Convergence progress reported
  polish.V <- medpolish(V, maxiter=MaxIter, trace.iter=TRUE, na.rm=T, eps=tol)
  # polished.H1 <- H - polish.H$residuals; polished.V1 <- V - polish.V$residuals # NAs
  polished.H <- polish.H$overall + outer(polish.H$row, polish.H$col, FUN="+") # no NAs
  polished.V <- polish.V$overall + outer(polish.V$row, polish.V$col, FUN="+")
  polished.dir <- atan2(polished.V, polished.H)
  polished.dir <- as.vector(polished.dir)
}

```

```
polished.dir[polished.dir < 0] <- polished.dir[polished.dir < 0] + 2*pi  
polished.dir[polished.dir > 2*pi] <- polished.dir[polished.dir > 2*pi] - 2*pi
```

```
raw.dir <- as.vector(Dirs) # has NAs, is organized  
filterNA <- is.na(raw.dir)  
raw.dir2 <- raw.dir[!filterNA]  
raw.dir2[raw.dir2 < 0] <- raw.dir2[raw.dir2 < 0] + 2*pi  
raw.dir2[raw.dir2 > 2*pi] <- raw.dir2[raw.dir2 > 2*pi] - 2*pi  
raw.dir[!filterNA] <- raw.dir2
```

```
x <- rep(seq(MinX, MaxX, by=delta), each=1 + (MaxY-MinY)/delta)  
y <- rep(seq(MaxY, MinY, by=-delta), 1 + (MaxX-MinX)/delta)  
return(list(x=x, y=y, polished.dir=polished.dir))
```

```
}
```


K.13 AssessStandardization

```
AssessStandardization <- function(CircDistr2="vM", Rho2=.75, CovModel2="spherical", Range2=10, Ext2=3, nSim=30, nPoints=400,
  OVERFIT=TRUE, ZLevels, CircLevels)
{
  # 2008-2-15.0600
  # Generate a nPoints x nSim and compute bivkde of QQ plots Standard Norm, Circular, Unif
  # CircDistr2 = circular probability distribution in {"U", "vm", "Tri", "WrC", "Card"}
  # to visually compare mean and variability with and without standardization
  # Range of parameter Rho2
  # For triangular,  $0 < \text{Rho2} \leq 4/\pi^2$ 
  # For cardioid,  $0 < \text{Rho2} \leq 0.5$ 
  # For vM and wrapped Cauchy,  $0 < \text{Rho2} < 1$ , 1== degenerate
  # For uniform,  $\text{Rho2} = 0$ 
  # CovModel2 = Any covariance model valid in geoR or RandomFields
  # Range2 = Distance at which RV are not correlated
  # Ext2 = Multiplies Range2. Range2 x Ext2 is width and height of RF. Default protects sill against edge effects.
  # nSim = Number of simulations
  # ZLevels = Number of color bins for filled.contour plots of QQ norm density
  # CircLevels = Number of color bins for filled.contour plots of QQ circular density
  # nPoints = Number of points per simulation
  # OverFit=TRUE, or standardization (centering and rescaling realization of the GRV to mean 0 sd 1) results in closer fit
  # for qualitative evaluation of the CRV. Undesirable effects are loss of independence of the marginal GRVs, biased GRF
  # covariance, and biased testing. Standardization is suitable for demonstration with closer fit, visualization, and
  # illustrations. Do not standardize for purposes of simulation and testing. OverFit=FALSE, or non-standardization (default)
  # includes expected variation from transformation of variation in mean and sd of sample of GRV.

  a <- ifelse(nPoints <= 10, 3/8, 1/2)
  CumProb <- ((1:nPoints)- a)/(nPoints + 1 - 2*a) # Vector of symmetric cumulative probabilities for QQ plots
  ZQuantiles <- qnorm(CumProb, mean=0, sd=1, lower.tail = TRUE)
  ZQuantiles <- rep(ZQuantiles, nSim)
```

```

# Compute circular quantiles
ThetaQuantiles <- vector(mode="numeric", length=nPoints)
if(CircDistr2=="U") { ThetaQuantiles <- -pi + 2*pi*CumProb } else
if(CircDistr2 == "Tri")
{
  if(Rho2==0 | Rho2 > 4/pi^2) stop("Tri: 0 < Rho <= 4/pi^2")
  filter <- CumProb < 0.5
  u1 <- CumProb[filter]
  a <- Rho2/8
  b <- (4+pi^2*Rho2)/(8*pi)
  c <- 0.5 - u1
  q <- -.5*(b+sqrt(b^2-4*a*c))
  ThetaQuantiles[filter] <- c/q

  u2 <- CumProb[!filter]
  a <- -Rho2/8
  b <- (4+pi^2*Rho2)/(8*pi)
  c <- 0.5 - u2
  q <- -.5*(b+sqrt(b^2-4*a*c))
  ThetaQuantiles[!filter] <- c/q
} else
{
  # For non-uniform circular distributions first get circular CDF in order to get ThetaQuantiles
  CircScale <- seq(-pi, pi, length=2*pi/.01) # Circular support from -pi to +pi, 629 elements, d~.01, CircScale[315] is zero
  n <- length(CircScale)
  if(CircDistr2=="vM")
  {
    if(Rho2==0 | Rho2 >= 1) stop("vM: 0 < Rho < 1")
    require(CircStats)
    CircProb <- rep(-1, n)
    Kappa=A1inv(Rho2) # N. I Fisher, Statistical Analysis of Circular Data, 2000 p. 49
    # As theta increases from -pi, pvm increases from .5
    for(i in 1:n) {CircProb[i] <- pvm(CircScale[i], mu=0, kappa=Kappa)}
    filter <- CircScale < 0
    CircProb[filter] <- CircProb[filter] - 0.5
    CircProb[!filter] <- CircProb[!filter] + 0.5
  }
}

```

```

} else
if(CircDistr2=="Card")
{
  if(Rho2==0 | Rho2 > 0.5) stop("Cardioid: 0 < Rho <= 0.5")
  CircProb <- (CircScale + pi + 2*Rho2*sin(CircScale))/(2*pi)
} else
if(CircDistr2=="WrC")
{
  if(Rho2==0 | Rho2 >= 1) stop("Wrapped Cauchy: 0 < Rho < 1")
  Angles1 <- CircScale[CircScale < 0] + 2*pi
  Angles2 <- CircScale[CircScale >= 0]
  prob1 <- 0.5 - acos(((1+Rho^2)*cos(Angles1) - 2*Rho)/(1 + Rho^2 - 2*Rho * cos(Angles1)))/(2*pi)
  prob2 <- 0.5 + acos(((1+Rho^2)*cos(Angles2) - 2*Rho)/(1 + Rho^2 - 2*Rho * cos(Angles2)))/(2*pi)
  CircProb <-c(prob1, prob2)
}
CircProb[1] <- 0; CircProb[n] <- 1 # For any numerical imprecision

# Get Quantiles From Inverse Circular CDF for distributions not uniform
DeltaTh <- CircScale[2] - CircScale[1]
for(i in 1:nPoints)
{
  p <- CumProb[i]
  a <- max((1:n)[CircProb <= p]) # Left index
  if(CircProb[a]==p) {r <- 0} else {r <- (p - CircProb[a])/(CircProb[a+1]- CircProb[a])}
  ThetaQuantiles[i] <- CircScale[a] + r*DeltaTh
}
}

ThetaQuantiles <- rep(ThetaQuantiles, nSim)

```

```

Thetasort <- Zsort <- matrix(data = NA, nrow = nPoints, ncol = nSim)
for(i in 1:nSim)
{
  output <- SimulateCRF(N=nPoints, CircDistr=CircDistr2, Rho=Rho2, Range=Range2, Ext=Ext2, CovModel= CovModel2,
OverFit=OVERFIT)
  Z <- output$Z
  Zsort[,i] <- sort(Z)

  Theta <- output$direction
  Thetasort[,i] <- sort(Theta)
}

require(KernSmooth)
biv.Z <- bkde2D(x=cbind(ZQuantiles, as.vector(Zsort)), bandwidth=c(0.1,0.1), range.x=list(c(-3,3), c(-3,3)))
biv.C <- bkde2D(x=cbind(ThetaQuantiles, as.vector(Thetasort)), bandwidth=c(0.1,0.1), range.x=list(c(-pi,pi), c(-pi,pi)))

# QQ Standard Normmal
dev.set(which=2)
par(mgp=c(1.5,.5,0), mar=c(3.2,2.8,1.7,0.1), cex.main=.75, cex.lab=.75, cex.axis=.75)
filled.contour(biv.Z$x1, biv.Z$x2, biv.Z$fhat, color = terrain.colors, levels=ZLevels,
  plot.title = title(main = "Density of QQ Standard Normal Points", xlab = "Theoretical Quantile", ylab = "Sample Quantile"))

# QQ Circular
dev.set(which=3)
par(mgp=c(1.5,.5,0), mar=c(3.2,2.8,1.7,0.1), cex.main=.75, cex.lab=.75, cex.axis=.75)
filled.contour(biv.C$x1, biv.C$x2, biv.C$fhat, color = terrain.colors, xlim=c(-pi,pi), ylim=c(-pi,pi), levels=CircLevels,
  plot.title = title(main = "Density of QQ Circular Points", xlab = "Theoretical Quantile", ylab = "Sample Quantile"))
}

```

K.14 MakeCosineData

```
MakeCosineData <-function(nSim=400, N=400, model, Range, Ext=2, CircDistr, Rho, Resolution=0.01, ...)
{
  # 2008-8-14.0550
  # Make cosine data from transformation of standard normal GRF to fit cosine models

  # Input Arguments
  # nSim: Number of simulations
  # N: Number of spatial locations per simulation
  # model: Name of spatial correlation function, see package geoR Help cov.spatial
  # Range: The range parameter of the covariance model
  # Ext: Range*Ext is horizontal and vertical length of sample space
  # CircDistr: Circular distribution in {U, vM, WrC, Tri, Card},
  # Rho: Mean resultant length parameter
  #   For triangular,  $0 < \text{Rho} \leq 4/\pi^2$ 
  #   For cardioid,  $0 < \text{Rho} \leq 0.5$ 
  #   For vM and wrapped Cauchy,  $0 < \text{Rho} < 1$ , 1== degenerate
  #   For uniform,  $\text{Rho} = 0$ , not required
  # Resolution: For interpolation of theta on CDF for non-closed form inverse CDFs

  # Values
  # matrix of lag distances in column 1 and cosineogram ordinates in column 2

  require(CircStats)
  require(RandomFields)
  require(KernSmooth)
  x <- c()
  CosineoG <- c()
  mean <- 0
  variance <- 1
  nugget <- 0

  for (i in 1:nSim)
  {
    direction <- vector(mode="numeric", length=N)
    X= runif(N, max=Range*Ext)
```

```

Y= runif(N, max=Range*Ext)
GRV <- GaussRF(grid=FALSE, x=X, y=Y, model=model, param=c(mean, variance, nugget, scale=Range, ...))
CumProbZ <- pnorm(GRV, mean=0, sd=1, lower.tail = TRUE)

if(CircDistr=="U") {direction <- -pi + 2*pi*CumProbZ} else
if(CircDistr == "Tri")
{
  if(Rho==0 | Rho > 4/pi^2) stop("Tri: 0 < Rho <= 4/pi^2")
  filter <- CumProbZ < 0.5
  u1 <- CumProbZ[filter]
  a <- Rho/8; b <- (4+pi^2*Rho)/(8*pi); c <- 0.5 - u1
  q <- -.5*(b+sqrt(b^2-4*a*c))
  direction[filter] <- c/q

  u2 <- CumProbZ[!filter]
  a <- -Rho/8; b <- (4+pi^2*Rho)/(8*pi); c <- 0.5 - u2
  q <- -.5*(b+sqrt(b^2-4*a*c))
  direction[!filter]<- c/q
} else
{
  # For OTHER circular distributions compute table of circular CDF and interpolate
  CircScale <- seq(-pi, pi, length=2*pi/Resolution)
  # With resolution=.01, circular support from -pi to +pi has 629 elements, delta ~0.01000507, CircScale[315] = 0
  n <- length(CircScale)
  if(CircDistr == "vM")
  {
    if(Rho==0 | Rho >= 1) stop("vM: 0 < Rho < 1")
    CircProb <- rep(-1, n)
    Kappa=A1inv(Rho) # N. I Fisher, Statistical Analysis of Circular Data, 2000 p. 49
    # As direction increases from -pi, pvm increases from .5
    for(i in 1:length(CircScale)) CircProb[i] <- pvm(CircScale[i], mu=0, kappa=Kappa)
    filter <- CircScale < 0
    CircProb[filter] <- CircProb[filter] - 0.5
    CircProb[!filter] <- CircProb[!filter] + 0.5
  } else
  if(CircDistr == "Card")
  {

```

```

    if(Rho==0 | Rho > 0.5) stop("Cardioid: 0 < Rho <= 0.5")
    CircProb <- (CircScale + pi + 2*Rho*sin(CircScale))/(2*pi)
  } else
  if(CircDistr == "WrC")
  {
    if(Rho==0 | Rho >= 1) stop("Wrapped Cauchy: 0 < Rho < 1 ")
    Angles1 <- CircScale[CircScale < 0]
    Angles2 <- CircScale[CircScale >= 0]
    prob1 <- 0.5 - acos(((1+Rho^2)*cos(Angles1) - 2*Rho)/(1 + Rho^2 - 2*Rho * cos(Angles1)))/(2*pi)
    prob2 <- 0.5 + acos(((1+Rho^2)*cos(Angles2) - 2*Rho)/(1 + Rho^2 - 2*Rho * cos(Angles2)))/(2*pi)
    CircProb <- c(prob1, prob2)
  }
  CircProb[1] <- 0; CircProb[n] <- 1

# Interpolation
DeltaTh <- CircScale[2] + pi
for(i in 1:N)
{
  p <- CumProbZ[i] # Cumulative prob of GRV
  a <- max((1:n)[CircProb <= p]) # Index
  if(a==n) {r <- 0} else
  {
    if(CircProb[a]==p) {r <- 0} else {r <- (p -CircProb[a])/( CircProb[a+1] -CircProb[a])}
  }
  direction[i] <- CircScale[a] + r*DeltaTh
}
}
cosineog.out <- CosinePlots(x=X, y=Y, directions=direction, Lag.n.Adj=1, Lag=seq(0, Range, by=0.25), Plot=FALSE)
CosineoG <- c(CosineoG, cosineog.out$cosine)
}

fitCosineoG <- locpoly(rep(seq(0, Range, by=0.25),nSim), CosineoG, bandwidth = 0.15)
return(list(fitCosineoG=fitCosineoG, CosineoG=CosineoG ))
}

```

K.15 FitCosineData

```
FitCosineData <- function(output=U.s.5$fitCosineoG, output.distr="U", code="0.5", GRF.model="spherical", ...)
{
  # 2008-11-08.0759

  require(RandomFields)

  rhoMax <- 1 # WrC, vM
  if(output.distr=="card") rhoMax <- .5
  if(output.distr=="tri") rhoMax <- 4/pi^2
  if(output.distr=="U") rhoMax <- 0

  if(code == "0.5") {Fraction=0.00; Range=5}
  if(code == "0.10") {Fraction=0.00; Range=10}
  if(code == "05.5") {Fraction=0.05; Range=5}
  if(code == "05.10") {Fraction=0.05; Range=10}
  if(code == "95.5") {Fraction=0.95; Range=5}
  if(code == "95.10") {Fraction=0.95; Range=10}

  Xlim=c(0, Range); Ylim=c(0, 1)
  Rho <- Fraction*rhoMax

  u <- output$x/Range # Standardized distance required by CovarianceFct
  v <- Rho^2+(1-Rho^2)*CovarianceFct(x=u, model=GRF.model, param=c(mean=0,variance=1,nugget=0,scale=1, ...),
    dim=1, fctcall="Covariance")
  par(mfrow=c(2,1))
  plot(u,v, xlim=Xlim, ylim=Ylim, ty="l", col=2, lwd=2, xlab="Distance", ylab="Cosines")
  lines(output, col=1)
  abline(h=1, col="grey"); abline(h=Rho^2, col="grey"); abline(v=0, col="grey"); abline(v=Range, col="grey")
  plot(u, output$y-v, xlim=Xlim, ty="l", main=paste("Mean Abs Difference= ", round(mean(abs(output$y-v)),6)),
    xlab="Distance", ylab="Difference of Cosines")
  abline(h=0, col="grey")
}
```


K.16 FitOceanWind

For Figure 4-8

```
FitOceanWind <- function(input, Rho, Nugget, Range, GRF.model, ...)
{
  # 2008-11-08.0759
  require(RandomFields)
  delta <- seq(0, .95, by=.05)
  u <- c(delta, input$x)/Range
  v <- Rho^2+(1-Rho^2-Nugget)*CovarianceFct(x=u, model=GRF.model, param=c(mean=0,variance=1,nugget=0,scale=1, ...),
    dim=1, fctcall="Covariance")
  par(mfrow=c(2,1))
  plot(c(delta, input$x), v, ty="l", col=2,lwd=2, xlab="Distance", ylab="Cosines", xlim=c(0, Range), ylim=c(0,1))
  lines(input, col=1)
  abline(h=1, col="grey")
  abline(h=Rho^2, col="grey")
  abline(v=0, col="grey")
  abline(v=Range, col="grey")

  v2 <- v[-c(1:length(delta))]
  filter <- input$x <= Range
  MAD <- round(mean( abs( v2[filter]-input$y[filter] ) ), 6)
  plot(input$x, v2 - input$y, ty="l", main=paste("Mean Abs Difference= ", MAD),
    xlab="Distance", ylab="Difference of Cosines", xlim=c(0,Range))
  abline(h=0, col="grey")
}
```

K.17 3DPolarMainMagnetic

```
MainMagnetic <- function(Data="1900.dat")
{
  # 2008-06-23
  # The data must be sorted first by latitude decreasing and second by longitude increasing: north to south, and west to east.
  #
  # Variables:
  # id long lat direction magnitude x.north y.east z.vertical
  # id is a sequential integer.
  # long is longitude from -180° to +180° by 1°.
  # lat is latitude from -89° to +89° by 1°. Data are not defined at poles.
  # Direction is from 0 to 2pi radians.
  # Magnitude is the magnitude of the resultant of the east and north components.

  require(rgl)
  require(fields)
  data <- read.table(file=Data,header=T)
  long <- data[,2]*pi/180 # longitude in radians
  colat <- (90 - data[,3])*pi/180 # colatitude in radians
  Direction <- data[,4]*180/pi # Direction in degrees
  Magnitude <- data[,5]
  MaxMag <- max(Magnitude[!is.na(Magnitude)])

  # VERTICES
  x <- Magnitude*sin(colat)*cos(long)
  y <- Magnitude*sin(colat)*sin(long)
  z <- Magnitude*cos(colat)

  # Color of direction from GYRB color wheel
  c <- vector(mode="character", length=length(x))

  filter <- (Direction < 90); ZeroToOne <- Direction[filter]/90
  c[filter] <- rgb(255*ZeroToOne, 255, 0, maxColorValue=255)

  filter <- (Direction >= 90 & Direction < 180); ZeroToOne <- (Direction[filter]-90)/90
  c[filter] <- rgb(255, 255*(1 - ZeroToOne), 0, maxColorValue=255)
```

```

filter <- (Direction >= 180 & Direction < 270); ZeroToOne <- (Direction[filter] - 180)/90
c[filter] <- rgb(255*(1 - ZeroToOne), 0, 255*ZeroToOne, maxColorValue=255)

filter <- (Direction >= 270); ZeroToOne <- (Direction[filter] - 270)/90
c[filter] <- rgb(0, 255*ZeroToOne, 255*(1 - ZeroToOne), maxColorValue=255)

# Matrix rows are lat 89, 88, 87, ..., -88, -89
X <- matrix(data=x, byrow=T, ncol=361)# 179 rows x 361 cols
Y <- matrix(data=y, byrow=T, ncol=361)
Z <- matrix(data=z, byrow=T, ncol=361)
C <- matrix(data=c, byrow=T, ncol=361)

# Add continental profiles for reference
data(world.dat) # range of longitude (x) is -180, +180; range of latitude (y) is -78.5, 83.7
filter <- lis.na(world.dat$x)
xx <- world.dat$x[filter]
yy <- world.dat$y[filter]
C[cbind(90 - round(yy,0), 181 + round(xx,0))] <- rgb(0,0,0)

# q=quadrilateral primitive
Xq <- vector(mode="numeric", len=178*360*4); Yq <- Xq; Zq <- Xq
Cq <- vector(mode="character", len=178*360*4)
QuadFilter <- c( rep(c(1,2,3,4), 178*360) ) # Earth magnetic at 1° resolution + no poles has 64,080 quads.

Xq[QuadFilter == 1] <- X[-179,-361]
Yq[QuadFilter == 1] <- Y[-179,-361]
Zq[QuadFilter == 1] <- Z[-179,-361]
Cq[QuadFilter == 1] <- C[-179,-361]

Xq[QuadFilter == 2] <- X[-179,-1]
Yq[QuadFilter == 2] <- Y[-179,-1]
Zq[QuadFilter == 2] <- Z[-179,-1]
Cq[QuadFilter == 2] <- C[-179,-1]

Xq[QuadFilter == 3] <- X[-1,-1]
Yq[QuadFilter == 3] <- Y[-1,-1]

```

```
Zq[QuadFilter == 3] <- Z[-1,-1]
Cq[QuadFilter == 3] <- C[-1,-1]

Xq[QuadFilter == 4] <- X[-1,-361]
Yq[QuadFilter == 4] <- Y[-1,-361]
Zq[QuadFilter == 4] <- Z[-1,-361]
Cq[QuadFilter == 4] <- C[-1,-361]
rgl.quads(Xq,Yq,Zq,color=Cq,lit=FALSE)
L <- 1.33*MaxMag
rgl.lines(c(0,L), c(0,0), c(0,0), color=2, size=4)
rgl.lines(c(0,0), c(0,L), c(0,0), color=3, size=4)
rgl.lines(c(0,0), c(0,0), c(0,L), color=4, size=4)
rgl.lines(c(0,-L), c(0,0), c(0,0), color=2, size=1)
rgl.lines(c(0,0), c(0,-L), c(0,0), color=3, size=1)
rgl.lines(c(0,0), c(0,0), c(0,-L), color=4, size=1)

rgl.viewpoint(fov=1)
}
```

K.18 Circular Kriging Variance

```

PlotCKVar <- function(rho=0, ng=0, range=8, x.legend=1, y.legend=1)
{
  d01 <- seq(0,10,0.05); d02 <- sqrt(d01^2 + 1) # distance between observations=1
  c01.e <- rho^2 + (1-ng-rho^2)*exp(-3*d01/range)
  c02.e <- rho^2 + (1-ng-rho^2)*exp(-3*d02/range)
  c.e <- rbind(c01.e, c02.e)
  k.e <- rho^2 + (1-ng-rho^2)*exp(-3*1/range)
  K.e <- matrix(data=c(1,k.e,k.e,1), ncol=2, byrow=TRUE)
  K.e.inv <- solve(K.e)
  ckvar.e <- 2 - 2*sqrt( diag(t(c.e) %*% K.e.inv %*% c.e) )

  c01.g <- rho^2 + (1-ng-rho^2)*exp(-3*(d01/range)^2)
  c02.g <- rho^2 + (1-ng-rho^2)*exp(-3*(d02/range)^2)
  c.g <- rbind(c01.g, c02.g)
  k.g <- rho^2 + (1-ng-rho^2)*exp(-3*(1/range)^2)
  K.g <- matrix(data=c(1,k.g,k.g,1), ncol=2, byrow=TRUE)
  K.g.inv <- solve(K.g)
  ckvar.g <- 2 - 2*sqrt( diag(t(c.g) %*% K.g.inv %*% c.g) )

  c01.s <- 1 - ng - (1-ng-rho^2)*(1.5*d01/range - 0.5*(d01/range)^3); c01.s[d01 > range] <- rho^2
  c02.s <- 1 - ng - (1-ng-rho^2)*(1.5*d02/range - 0.5*(d02/range)^3); c02.s[d02 > range] <- rho^2
  c.s <- rbind(c01.s, c02.s)
  k.s <- 1 - ng - (1-ng-rho^2)*(1.5*1/range - 0.5*(1/range)^3)
  if(1 > range) k.s <- rho^2
  K.s <- matrix(data=c(1,k.s,k.s,1), ncol=2, byrow=TRUE)
  K.s.inv <- solve(K.s)
  ckvar.s <- 2 - 2*sqrt( diag(t(c.s) %*% K.s.inv %*% c.s) )

  plot(d01, ckvar.e, ty="l", col=1, ylim=c(0,2), xlab="Distance",
       ylab="Circular Kriging Variance")
  lines(d01, ckvar.g, col="tan", lwd=3); lines(d01, ckvar.s, col=2, lty=2)
  legend(x=x.legend, y=y.legend, c("Exponential", "Gaussian", "Spherical"),
        lty = c(1, 1, 2), col=c(1, "tan", 2), lwd=c(1, 3, 1), cex=1.1)
}

```

Appendix L

R Command Line Input

L.1 Figures 3-4 to 3-8

```
out <- SimulateSill()
```

```
plot(1:1000, out$Cavg, type="l", xlab="Number of Simulations", ylab="Mean Cosine", main="Cardioid", ylim=c(0.05,0.10))  
abline(h=0.25^2, lty=2)
```

```
plot(1:1000, out$Tavg, type="l", xlab="Number of Simulations", ylab="Mean Cosine", main="Triangular", ylim=c(0.02,0.07))  
abline(h=(.5*4/pi^2)^2, lty=2)
```

```
plot(1:1000, out$Uavg, type="l", xlab="Number of Simulations", ylab="Mean Cosine", main="Uniform", ylim=c(-0.01,0.04))  
abline(h=0, lty=2)
```

```
plot(1:1000, out$VMavg, type="l", xlab="Number of Simulations", ylab="Mean Cosine", main="Von Mises", ylim=c(0.77,0.82))  
abline(h=0.798, lty=2)
```

```
plot(1:1000, out$WCavg, type="l", xlab="Number of Simulations", ylab="Mean Cosine", main="Wrapped Cauchy", ylim=c(0.1,0.15))  
abline(h=exp(-2), lty=2)
```

L.2 Figure 3-13

```
years <- sort(unique(OceanWind[,1]))

# Smoothed average direction from CircDataimage
model.x <- Globals$x.g[Globals$StartRow:Globals$EndRow]
n.x <- length(model.x)
xmin <- min(model.x); xmax <- max(model.x)
model.y <- Globals$y.g[Globals$StartCol:Globals$EndCol]
n.y <- length(model.y)
ymin <- min(model.y); ymax <- max(model.y)
model.dir <- Globals$Direction[Globals$StartRow:Globals$EndRow, Globals$StartCol:Globals$EndCol] # in radians

## Slice matrix and stack into vectors
model.xv <- rep(model.x, n.y)
model.yv <- rep(model.y, each=n.x)
model.dirv <- as.vector(model.dir)

## subset data to longitude and latitude once
filter1 <- OceanWind[, 2]>=xmin & OceanWind[, 2]<= xmax & OceanWind[, 3]>=ymin & OceanWind[, 3]<=ymax
data1 <- OceanWind[filter1, ]

SouthPolarCorr <- function(trend)
{
  # Vectors of the sequence of ordinates of cosineograms are column binded
  Cosines <- c()
  for(year in years)
  {
    ## subset the data to year
    filter2 <- data1[,1] == year; data2 <- data1[filter2,-1] # year value omitted

    ## Organize data into a matrix using vector expressions
    u <- matrix(data=NA, nrow=n.x, ncol=n.y); v <- u
    Rows <- round((data2[, 1] - xmin) + 1, digits = 0) # Indexing vector
    Columns <- round((data2[, 2] - ymin) + 1, digits = 0) # Indexing vector
    ## test one observation per cell
    if (max(table(data2[,c(1,2)])) > 1) stop(paste("Duplicate observation in year ", year))
  }
}
```

```

u[cbind(Rows, Columns)] <- data2[, 3]; v[cbind(Rows, Columns)] <- data2[, 4]

## convert u and v to direction in 0 to 2pi, and slice and stack matrix to vector
data.dir <- atan2(v, u); data.dirv <- as.vector(data.dir)

residuals <- CircResidual(X=model.xv, Y=model.yv, Raw=data.dirv, Trend=trend, Plot=FALSE)

cosines <- CosinePlots(x=residuals$x, y=residuals$y, directions=residuals$direction, Lag=1:40, Lag.n.Adj = 1, BinWAdj=1,
Plot=FALSE)$cosine
Cosines <- cbind(Cosines, cosines)
}
return(Cosines)
}

Cosineograms <- SouthPolarCorr(trend=model.dirv)
Mean <- function(x) { mean(x, na.rm=TRUE, trim=0.2) }

par(mai=c(0.95, 0.75, 0.25, 0.25))
d <- 1:40
plot(rep(d,28), Cosineograms, xlab="Distance In Degrees Of Longitude And Latitude", ylab="Cosines", col="grey", xlim=c(0, 15))
lines(c(0,d), c(1, apply(Cosineograms, 1, Mean)), col=2, lw=3)

# Exponential model
d2 <- seq(0, 40, by=0.25)
r = 3.8 # range
rho = sqrt(0.45) # mean resultant length
lines(d2, rho^2 + (1 - rho^2)*exp(-3*d2/r), col=4, lty=2, lwd=3 )

```


L.3 Figure 5-1, Simulated CRF

```
ygrid <- xgrid <- seq(-20,20,length=21)
xgrid <- rep(xgrid, 21)
ygrid <- rep(ygrid, each=21)
set.seed(9) # So Figures 5-1, 5-3, 5-4, and 5-7 will be consistent
crf1 <- SimulateCRF(CircDistr="vM", Rho=0.8, Mu=0, Range=10, CovModel="spherical", Grid=cbind(xgrid, ygrid), OverFit=TRUE)
require(CircSpatial)
par(mgp=c(2,1,0), mar=c(3.1, 3.1, 0.5, 0.5), cex=.8)
PlotVectors(x=crf1$x, y=crf1$y, h=cos(crf1$direction), v=sin(crf1$direction), AdjArrowLength=1.25, AdjHeadLength=.7)
```

L.4 Figure 5-3, Image of GRF

```
ygrid <- xgrid <- seq(-20,20,length=21)
xgrid <- rep(xgrid, 21); ygrid <- rep(ygrid, each=21)
set.seed(9)
crf1 <- SimulateCRF(CircDistr="vM", Rho=0.8, Mu=0, Range=10, CovModel="spherical", Grid=cbind(xgrid, ygrid), OverFit=TRUE)
image(seq(-20,20,length=21), seq(-20,20,length=21), matrix(data=crf1$Z, nrow=21, ncol=21, byrow=FALSE), col = terrain.colors(12),
      xlab="x", ylab="y")
```

L.5 Figure 5-4, Variogram and Inverted Cosineogram Similar

```
ygrid <- xgrid <- seq(-20,20,length=21)
xgrid <- rep(xgrid, 21)
ygrid <- rep(ygrid, each=21)
set.seed(9); CorrelationTransfer(CircDistr2="vM", Rho2=0.8, Range2=10, CovModel2="spherical", GRID=cbind(xgrid, ygrid), OVERFIT=TRUE)
```

L.6 Figure 5-5, Standardization

Figure constructed from GRV and CRV windows

```
windows(); windows()
set.seed(10); AssessStandardization(CircDistr2="U", ZLevels=seq(0,.8,length=16), CircLevels=seq(0,0.34,length=16), OVERFIT=TRUE)
set.seed(10); AssessStandardization(CircDistr2="U", ZLevels=seq(0,.8,length=16), CircLevels=seq(0,0.34,length=16), OVERFIT=FALSE)
```

L.7 Figure 5-6, Variability vs. ρ

Figure constructed from CRV windows

```
windows(); windows()
set.seed(20); AssessStandardization(CircDistr2="Tri", Rho2=.95*4/pi^2, ZLevels=seq(0,.8,length=16), CircLevels=seq(0,0.4,length=16),
OVERFIT=TRUE)
set.seed(20); AssessStandardization(CircDistr2="Tri", Rho2=.50*4/pi^2, ZLevels=seq(0,.8,length=16), CircLevels=seq(0,0.4,length=16),
OVERFIT=TRUE)
set.seed(20); AssessStandardization(CircDistr2="Tri", Rho2=.25*4/pi^2, ZLevels=seq(0,.8,length=16), CircLevels=seq(0,0.4,length=16),
OVERFIT=TRUE)
set.seed(20); AssessStandardization(CircDistr2="Tri", Rho2=.25*4/pi^2, ZLevels=seq(0,.8,length=16), CircLevels=seq(0,0.4,length=16),
OVERFIT=FALSE)
```

L.8 Figure 5-8 and the Figures in Appendices C and D

```
ygrid <- xgrid <- seq(-20,20,length=21) # Source CosinePlots before running the following code
xgrid <- rep(xgrid, 21); ygrid <- rep(ygrid, each=21)
set.seed(9); AssessCRF(CircDistr2="vM", Rho2=0.8, Range2=10, CovModel2="spherical", GRID=cbind(xgrid, ygrid), OVERFIT=TRUE)
```

Appendix C

```
AssessCRF(nPoints=400, CircDistr2="Card", Rho2=0.5*0.5, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
AssessCRF(nPoints=400, CircDistr2="Tri", Rho2=0.5*4/pi^2, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
AssessCRF(nPoints=400, CircDistr2="U", Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
AssessCRF(nPoints=400, CircDistr2="Wrc", Rho2=0.5, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
```

Appendix D

```

S=1000*runif(1)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="Card", Rho2=0.05, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="Card", Rho2=0.05, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=FALSE)

S=1000*runif(1)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="Card", Rho2=0.95*0.5, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="Card", Rho2=0.95*0.5, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=FALSE)

S=1000*runif(1)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="Tri", Rho2=0.05, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="Tri", Rho2=0.05, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=FALSE)

S=1000*runif(1)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="Tri", Rho2=0.95*4/pi^2, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="Tri", Rho2=0.95*4/pi^2, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=FALSE)

S=1000*runif(1)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="vM", Rho2=0.05, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="vM", Rho2=0.05, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=FALSE)

S=1000*runif(1)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="vM", Rho2=0.95, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="vM", Rho2=0.95, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=FALSE)

S=1000*runif(1)
set.seed(S)

```

```

AssessCRF(nPoints=400, CircDistr2="WrC", Rho2=0.05, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="WrC", Rho2=0.05, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=FALSE)

S=1000*runif(1)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="WrC", Rho2=0.95, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=TRUE)
set.seed(S)
AssessCRF(nPoints=400, CircDistr2="WrC", Rho2=0.95, Range2=10, Ext2=3, CovModel2="spherical", OVERFIT=FALSE)

```

L.9 Figures 6-1 to 6-10

```
require(CircSpatial)
```

```
# Figure 6-1, Comprehensive Example - Global Trend Model of 121 Locations
```

```

model1.x<- 1:11; model1.y <- 11:1; model1.y <- rep(model1.y, 11); model1.x <- rep(model1.x, each=11)
model1.direction <- matrix(data=c(
  157, 141, 126, 113, 101, 90, 79, 67, 54, 40, 25, 152, 137, 123, 111, 100, 90, 80, 69, 57, 44, 30,
  147, 133, 120, 109, 99, 90, 81, 71, 60, 48, 35, 142, 129, 117, 107, 98, 90, 82, 73, 63, 52, 40,
  137, 125, 114, 105, 97, 90, 83, 75, 66, 56, 45, 132, 121, 111, 103, 96, 90, 84, 77, 69, 60, 50,
  127, 117, 108, 101, 95, 90, 85, 79, 72, 64, 55, 122, 113, 105, 99, 94, 90, 86, 81, 75, 68, 60,
  117, 109, 102, 97, 93, 90, 87, 83, 78, 72, 65, 112, 105, 99, 95, 92, 90, 88, 85, 81, 76, 70,
  107, 101, 96, 93, 91, 90, 89, 87, 84, 80, 75), ncol=11, byrow=TRUE)
model1.direction <- as.vector(model1.direction)*pi/180

par(mai=c(0.5, 0.5, 0.15, 0.15), ps=11)
plot( x=model1.x,y=model1.y, type='n', xlim=c(0,12), ylim=c(0,12), asp=1, xlab="", ylab=")
arrow.plot(a1=model1.x, a2=model1.y, u=cos(model1.direction),v=sin(model1.direction), arrow.ex=.06, length=.07, col='blue3', lwd=1, angle=18)

```

```

# Figure 6-2, Comprehensive Example - Simulated Sample of a Von Mises CRF,  $\rho = \sqrt{0.5}$ 

# Compute vM CRF of 121 observations, Rho=sqrt(0.5) so sill about 0.5, from GRF (Range=4, spherical covariance).
set.seed(666)
crf1<- SimulateCRF(CircDistr="vM", Rho=sqrt(0.5), Range=4, CovModel="spherical", Grid=cbind(model1.x, model1.y), OverFit=FALSE)
names(crf1) # [1] "x"      "y"      "direction" "Z"

# Make sample
sample1.direction <- model1.direction + crf1$direction
par(mai=c(0.5, 0.5, 0.15, 0.15), ps=11)
plot( x=crf1$x,y=crf1$y, type='n', xlim=c(0,12), ylim=c(0,12), asp=1, xlab="", ylab="")
arrow.plot(a1=crf1$x, a2=crf1$y, u=cos(sample1.direction),v=sin(sample1.direction), arrow.ex=.06, length=.07, col=1, lwd=1, angle=18)

# Figure 6-3, Comprehensive Example - Estimate of the Global Trend Model

FitHoriz1 <- lm(cos(sample1.direction) ~ (model1.x + model1.y)^2)
FitVert1 <- lm(sin(sample1.direction) ~ (model1.x + model1.y)^2)
fitted1.direction <- atan2(FitVert1$fitted.values, FitHoriz1$fitted.values)

plot( x=crf1$x,y=crf1$y, type='n', xlim=c(0,12), ylim=c(0,12), asp=1, xlab="", ylab="")
arrow.plot(a1=model1.x, a2=model1.y, u=cos(fitted1.direction),v=sin(fitted1.direction), arrow.ex=.06, length=.07, col="tan", lwd=3, angle=18)
arrow.plot(a1=model1.x, a2=model1.y, u=cos(model1.direction),v=sin(model1.direction), arrow.ex=.06, length=.07, col="blue3", lwd=1, angle=18)

# Figure 6-4, Comprehensive Example - Enlarged View of the Data, Model, and Residual Rotation

par(mai=c(0.45, 0.45, 0.1, 0.1), ps=11)
CircResidual(X=model1.x, Y=model1.y, Raw=sample1.direction, Trend=fitted1.direction, Plot=TRUE, AdjArrowLength=.9, xlim=c(8.0, 11.0),
  ylim=c(8, 11))

resids1 <- CircResidual(X=model1.x, Y=model1.y, Raw=sample1.direction, Trend=fitted1.direction, Plot=FALSE)

```

```
# Figure 6-5, Comprehensive Example - Points of the Cosineogram, and the Exponential, Gaussian,  
# and Spherical Cosine Models of Spatial Correlation
```

```
par(mai=c(0.85, 0.75, 0.15, 0.15), ps=11)  
CosinePlots(x=resids1$x, y=resids1$y, directions=resids1$direction, Plot=TRUE, Cloud=FALSE, Model=TRUE, nugget=0, Range=3.07,  
sill=0.674, x.legend=0.4, y.legend=0.95, Lag=seq(0, 8, by=.375), BinWAdj=1)
```

```
# Figure 6-6, Enlarged View of The Kriging and the Residual Rotations
```

```
x2 <- seq(1,11, by=0.2); n <- length(x2); y2 <- x2; y2 <- rep(y2, n); x2 <- rep(x2, each=n)  
krig1 <- KrigCRF(krig.x=x2, krig.y=y2, resid.x=resids1$x, resid.y=resids1$y, resid.direction= resid1$direction, Model="spherical", Nugget=0.0,  
Range=3.07, sill=0.674, Plot=FALSE)  
par(mai=c(0.45, 0.45, 0.1, 0.1), ps=11)  
plot(x=krig1$x, y=krig1$y, ty="n", xlim=c(8.0, 11.0), ylim=c(8.0, 11.0), asp=1)  
arrow.plot(a1=krig1$x, a2=krig1$y, u=cos(krig1$direction), v=sin(krig1$direction), col="light grey", arrow.ex=0.08, xpd=FALSE,  
length = 0.08, angle=15)  
arrow.plot(a1=resids1$x, a2=resids1$y, u=cos(resids1$direction), v=sin(resids1$direction), arrow.ex=0.05, xpd=FALSE, length= 0.06,  
col=2, lty=1, lwd=1)
```

```
# Figure 6-7, Comprehensive Example – Enlarged View of the Interpolation of the Global Trend Model
```

```
interp1 <- InterpDirection(in.x=model1.x, in.y=model1.y, in.direction=fitted1.direction, out.x=krig1$x, out.y=krig1$y)  
plot(x=interp1$x, y=interp1$y, ty="n", xlim=c(8, 11.0), ylim=c(8.0, 11.0), asp=1)  
arrow.plot(a1=model1.x, a2=model1.y, u=cos(fitted1.direction), v=sin(fitted1.direction), arrow.ex=0.09, xpd=FALSE, length = 0.09,  
col="tan", lwd=2, angle=17)  
arrow.plot(a1=interp1$x, a2=interp1$y, u=cos(interp1$direction), v=sin(interp1$direction), col="purple", arrow.ex=0.05, xpd=FALSE,  
length = 0.06, angle=17)
```

Figure 6-8, Comprehensive Example – Arrow Plot of the Circular Spatial Data Estimate, Enlarged View

```
estimate1.direction=interp1$direction + krig1$direction
par(mai=c(0.45, 0.45, 0.1, 0.1), ps=11)
plot(x=interp1$x, y=interp1$y, ty="n", xlim=c(8.0, 11.0), ylim=c(8.0, 11.0), asp=1)
arrow.plot(a1=interp1$x, a2=interp1$y, u=cos(estimate1.direction), v=sin(estimate1.direction), arrow.ex=0.09, xpd=FALSE,
  length = 0.09,col="gold", lwd=1, angle=17)
arrow.plot(a1=model1.x, a2=model1.y, u=cos(sample1.direction), v=sin(sample1.direction), col=1, arrow.ex=0.05, xpd=FALSE,
  length = 0.06, angle=17)
```

Figure 6-9, Comprehensive Example – Circular Dataimage of the Circular Spatial Data Estimate

```
output1 <- data.frame(x=interp1$x, y=interp1$y, u=cos(estimate1.direction), v=sin(estimate1.direction), check.rows=TRUE, check.names=TRUE)
CircDataimage() # Use HSV Color Wheel, -105 rotation, arrow length multiplier 0.8, arrow spacing in pixels 3
```

Figure 6-10, Comprehensive Example - Circular Kriging Variance with Observations on a Regular Grid

```
KrigCRF(krig.x=x2, krig.y=y2, resid.x=resids1$x, resid.y=resids1$y, resid.direction= resid1$direction, Model="spherical", Nugget=0.0,
  Range=3.07, sill=0.674, Plot=TRUE, PlotVar=TRUE)
```

L.10 Make Cosine Datasets

```
range <- 5
```

```
U.s.5 <- MakeCosineData(model="spherical", CircDistr="U", Range=range)
U.e.5 <- MakeCosineData(model="exponential", CircDistr="U", Range=range)
U.g.5 <- MakeCosineData(model="gauss", CircDistr="U", Range=range)
gc()
```

```
RhoMax <- 0.5
```

```
Distr <- "Card"
```

```
card.s.05.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
card.s.25.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
card.s.50.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
card.s.75.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
card.s.95.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
card.e.05.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
card.e.25.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
card.e.50.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
card.e.75.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
card.e.95.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
card.g.05.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
card.g.25.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
card.g.50.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
card.g.75.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
card.g.95.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
gc()
```

```
RhoMax <- 4/pi^2
```

```
Distr <- "Tri"
```

```
tri.s.05.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
tri.s.25.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
tri.s.50.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
tri.s.75.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
tri.s.95.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
tri.e.05.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
tri.e.25.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
```



```

tri.e.50.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
tri.e.75.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
tri.e.95.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
tri.g.05.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
tri.g.25.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
tri.g.50.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
tri.g.75.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
tri.g.95.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
gc()

```

```
RhoMax <- 1
```

```
Distr <- "vM"
```

```

vM.s.05.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
vM.s.25.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
vM.s.50.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
vM.s.75.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
vM.s.95.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
vM.e.05.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
vM.e.25.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
vM.e.50.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
vM.e.75.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
vM.e.95.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
vM.g.05.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
vM.g.25.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
vM.g.50.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
vM.g.75.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
vM.g.95.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
gc()

```

```
RhoMax <- 1
```

```
Distr <- "WrC"
```

```

WrC.s.05.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
WrC.s.25.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
WrC.s.50.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
WrC.s.75.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
WrC.s.95.5 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
WrC.e.05.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.05*RhoMax, Range=range)

```

```
WrC.e.25.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
WrC.e.50.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
WrC.e.75.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
WrC.e.95.5 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
WrC.g.05.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
WrC.g.25.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
WrC.g.50.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
WrC.g.75.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
WrC.g.95.5 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
gc()
```

```
range <- 10
```

```
U.s.10 <- MakeCosineData(model="spherical", CircDistr="U", Range=range)
U.e.10 <- MakeCosineData(model="exponential", CircDistr="U", Range=range)
U.g.10 <- MakeCosineData(model="gauss", CircDistr="U", Range=range)
gc()
```

```
RhoMax <- 0.5
Distr <- "Card"
```

```
card.s.05.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
card.s.25.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
card.s.50.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
card.s.75.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
card.s.95.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
card.e.05.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
card.e.25.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
card.e.50.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
card.e.75.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
card.e.95.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
card.g.05.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
card.g.25.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
card.g.50.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
card.g.75.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
card.g.95.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
gc()
```

```

RhoMax <- 4/pi^2
Distr <- "Tri"
tri.s.05.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
tri.s.25.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
tri.s.50.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
tri.s.75.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
tri.s.95.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
tri.e.05.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
tri.e.25.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
tri.e.50.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
tri.e.75.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
tri.e.95.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
tri.g.05.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
tri.g.25.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
tri.g.50.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
tri.g.75.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
tri.g.95.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
gc()

```

```

RhoMax <- 1
Distr <- "vM"
vM.s.05.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
vM.s.25.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
vM.s.50.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
vM.s.75.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
vM.s.95.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
vM.e.05.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
vM.e.25.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
vM.e.50.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
vM.e.75.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
vM.e.95.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
vM.g.05.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
vM.g.25.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
vM.g.50.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
vM.g.75.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
vM.g.95.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
gc()

```

```
RhoMax <- 1
Distr <- "WrC"
WrC.s.05.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
WrC.s.25.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
WrC.s.50.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
WrC.s.75.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
WrC.s.95.10 <- MakeCosineData(model="spherical", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
WrC.e.05.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
WrC.e.25.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
WrC.e.50.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
WrC.e.75.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
WrC.e.95.10 <- MakeCosineData(model="exponential", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
WrC.g.05.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.05*RhoMax, Range=range)
WrC.g.25.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.25*RhoMax, Range=range)
WrC.g.50.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.50*RhoMax, Range=range)
WrC.g.75.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.75*RhoMax, Range=range)
WrC.g.95.10 <- MakeCosineData(model="gauss", CircDistr=Distr, Rho=.95*RhoMax, Range=range)
gc()
```

L.11 Figure M-1, Fitted Covariogram an Unbiased Estimator

```
MakeVariogData <- function(nSim=400, nPoints=400, Range=10, Ext=3, CovModel="spherical", Grid=NULL)
{
  # 2008-7-26.1112
  # Generate a nPoints x nPoints, fit variograms

  require(geoR)
  require(KernSmooth)
  VarioG.x <- c()
  VarioG.y <- c()
  for (i in 1:nSim)
  {
    if(is.null(Grid))
    {
      GRF <- grf(n=nPoints, xlims=c(0, Range*Ext), ylims=c(0, Range*Ext), cov.model=CovModel,
        nugget=0, cov.pars=c(1, Range), aniso.pars=Anisotropy, RF=TRUE, messages=FALSE) } else {
      GRF <- grf(grid=Grid, cov.model=CovModel,
        nugget=0, cov.pars=c(1, Range), aniso.pars=Anisotropy, RF=TRUE, messages=FALSE)
    }
    # GRF variogram
    vario.i <- variog(coords = GRF$coords, data = GRF$data, option = "bin", breaks=seq(0,Range, by=.2))
    VarioG.x <- c(VarioG.x, vario.i$u)
    VarioG.y <- c(VarioG.y, vario.i$v)
  }
  return(list(x=VarioG.x,y=VarioG.y))
}
```

```

PlotFittedVaroG <- function(Data=fittedXY, Degree=1, BandW=.83, Range=10)
{
  # Transform MakeVariogData output to covariance
  x <- fittedXY$x
  y <- fittedXY$y
  fitVarioG <- locpoly(x, y, bandwidth = BandW, range.x=c(0,Range), degree=Degree)
  par(mfrow=c(2,1))
  plot(fitVarioG$x, 1-fitVarioG$y, ty="l", lwd=8, xlab="Distance Between Observations", ylab="Spherical Covariance")
  u <- fitVarioG$x
  v <- 1-(1.5*u/Range - 0.5*(u/Range)^3)
  lines(u,v, col=2)
  Difference <- (1-fitVarioG$y) - v
  plot(fitVarioG$x,Difference, ty="l", xlab="Distance Between Observations", ylab="Vertical Distance Between Curves",
       sub=paste("Average Vertical Distance between Curves", round(mean(abs(Difference)),4)))
  abline(h=0,col="grey80")
}

```

L.12 Plot Figures M-2, M-3, and M-4, Families of Curves

```

Plot1 <- function(x, Main, Pch, off1=0.3, off2=0.0)
{
  plot(x, ty="l", xlim=c(0,10.5), ylim=c(0,1), xlab="X", ylab="Cosine(X)", main=Main, col="tan")
  points(x$x[401] + off1, x$y[401] + off2, pch=Pch, col=2)
}

```

```

Plot2 <- function(x, Pch, off1=0.3, off2=0.0)
{
  lines(x, ty="l", col="tan")
  points(x$x[401] + off1, x$y[401] + off2, pch=Pch, col=2)
}

```

```
Plot3 <- function(x, Pch, off1=0.3, off2=0.0)
{
  lines(x, ty="l", col=1)
  points(x$x[401] + off1, x$y[401] + off2, pch=Pch, col=2)
}
```

Figure M-2

```
par(mfrow=c(3,2), bty="n", mgp=c(2,1,0), mar=c(4.0, 3.5, 2.0, 0.0))
Plot1(card.e.05.10$fitCosineoG, Main="Cardioid", Pch="1")
```

```
Plot2(card.e.25.10$fitCosineoG, Pch="2")
Plot2(card.e.50.10$fitCosineoG, Pch="3")
Plot2(card.e.75.10$fitCosineoG, Pch="4")
Plot2(card.e.95.10$fitCosineoG, Pch="5")
```

```
Plot3(card.e.05.5$fitCosineoG, Pch="1")
Plot3(card.e.25.5$fitCosineoG, Pch="2")
Plot3(card.e.50.5$fitCosineoG, Pch="3")
Plot3(card.e.75.5$fitCosineoG, Pch="4")
Plot3(card.e.95.5$fitCosineoG, Pch="5")
```

```
Plot1(tri.e.05.10$fitCosineoG, Main="Triangular", Pch="1")
```

```
Plot2(tri.e.25.10$fitCosineoG, Pch="2", 0.5, 0.01)
Plot2(tri.e.50.10$fitCosineoG, Pch="3")
Plot2(tri.e.75.10$fitCosineoG, Pch="4")
Plot2(tri.e.95.10$fitCosineoG, Pch="5")
```

```
Plot3(tri.e.05.5$fitCosineoG, Pch="1")
Plot3(tri.e.25.5$fitCosineoG, Pch="2", 0.5, 0.01)
Plot3(tri.e.50.5$fitCosineoG, Pch="3")
Plot3(tri.e.75.5$fitCosineoG, Pch="4")
Plot3(tri.e.95.5$fitCosineoG, Pch="5")
```

```
Plot1(vM.e.05.10$fitCosineoG, Main="von Mises", Pch="1")
```

```
Plot2(vM.e.25.10$fitCosineoG, Pch="2")
Plot2(vM.e.50.10$fitCosineoG, Pch="3")
Plot2(vM.e.75.10$fitCosineoG, Pch="4")
Plot2(vM.e.95.10$fitCosineoG, Pch="5")
```

```
Plot3(vM.e.05.5$fitCosineoG, Pch="1")
Plot3(vM.e.25.5$fitCosineoG, Pch="2")
Plot3(vM.e.50.5$fitCosineoG, Pch="3")
Plot3(vM.e.75.5$fitCosineoG, Pch="4")
Plot3(vM.e.95.5$fitCosineoG, Pch="5")
```

```
Plot1(WrC.e.05.10$fitCosineoG, Main="Wrapped Cauchy", Pch="1")
```

```
Plot2(WrC.e.25.10$fitCosineoG, Pch="2")
Plot2(WrC.e.50.10$fitCosineoG, Pch="3")
Plot2(WrC.e.75.10$fitCosineoG, Pch="4")
Plot2(WrC.e.95.10$fitCosineoG, Pch="5")
```

```
Plot3(WrC.e.05.5$fitCosineoG, Pch="1")
Plot3(WrC.e.25.5$fitCosineoG, Pch="2")
Plot3(WrC.e.50.5$fitCosineoG, Pch="3")
Plot3(WrC.e.75.5$fitCosineoG, Pch="4")
Plot3(WrC.e.95.5$fitCosineoG, Pch="5")
```

```
Plot1(U.e.10$fitCosineoG, Main="Uniform", Pch="1")
Plot3(U.e.5$fitCosineoG, Pch="1")
```

```
# Figure M-3
```

```
par(mfrow=c(3,2), bty="n", mgp=c(2,1,0), mar=c(4.0, 3.5, 2.0, 0.0))
Plot1(card.g.05.10$fitCosineoG, Main="Cardioid", Pch="1")
```

```
Plot2(card.g.25.10$fitCosineoG, Pch="2", 0.5)
Plot2(card.g.50.10$fitCosineoG, Pch="3")
Plot2(card.g.75.10$fitCosineoG, Pch="4")
Plot2(card.g.95.10$fitCosineoG, Pch="5")
```



```
Plot3(card.g.05.5$fitCosineoG, Pch="1")
Plot3(card.g.25.5$fitCosineoG, Pch="2", 0.5)
Plot3(card.g.50.5$fitCosineoG, Pch="3")
Plot3(card.g.75.5$fitCosineoG, Pch="4")
Plot3(card.g.95.5$fitCosineoG, Pch="5")
```

```
Plot1(tri.g.05.10$fitCosineoG, Main="Triangular", Pch="1")
```

```
Plot2(tri.g.25.10$fitCosineoG, Pch="2", 0.5, 0.02)
Plot2(tri.g.50.10$fitCosineoG, Pch="3")
Plot2(tri.g.75.10$fitCosineoG, Pch="4")
Plot2(tri.g.95.10$fitCosineoG, Pch="5")
```

```
Plot3(tri.g.05.5$fitCosineoG, Pch="1")
Plot3(tri.g.25.5$fitCosineoG, Pch="2", 0.5)
Plot3(tri.g.50.5$fitCosineoG, Pch="3")
Plot3(tri.g.75.5$fitCosineoG, Pch="4")
Plot3(tri.g.95.5$fitCosineoG, Pch="5")
```

```
Plot1(vM.g.05.10$fitCosineoG, Main="von Mises", Pch="1")
```

```
Plot2(vM.g.25.10$fitCosineoG, Pch="2")
Plot2(vM.g.50.10$fitCosineoG, Pch="3")
Plot2(vM.g.75.10$fitCosineoG, Pch="4")
Plot2(vM.g.95.10$fitCosineoG, Pch="5")
```

```
Plot3(vM.g.05.5$fitCosineoG, Pch="1")
Plot3(vM.g.25.5$fitCosineoG, Pch="2")
Plot3(vM.g.50.5$fitCosineoG, Pch="3")
Plot3(vM.g.75.5$fitCosineoG, Pch="4")
Plot3(vM.g.95.5$fitCosineoG, Pch="5")
```

```
Plot1(WrC.g.05.10$fitCosineoG, Main="Wrapped Cauchy", Pch="1")
```

```
Plot2(WrC.g.25.10$fitCosineoG, Pch="2")
Plot2(WrC.g.50.10$fitCosineoG, Pch="3")
Plot2(WrC.g.75.10$fitCosineoG, Pch="4")
```

```
Plot2(WrC.g.95.10$fitCosineoG, Pch="5")
```

```
Plot3(WrC.g.05.5$fitCosineoG, Pch="1")
```

```
Plot3(WrC.g.25.5$fitCosineoG, Pch="2")
```

```
Plot3(WrC.g.50.5$fitCosineoG, Pch="3")
```

```
Plot3(WrC.g.75.5$fitCosineoG, Pch="4")
```

```
Plot3(WrC.g.95.5$fitCosineoG, Pch="5")
```

```
Plot1(U.g.10$fitCosineoG, Main="Uniform", Pch="1")
```

```
Plot3(U.g.5$fitCosineoG, Pch="1")
```

```
# Figure M-4
```

```
par(mfrow=c(3,2), bty="n", mgp=c(2,1,0), mar=c(4.0, 3.5, 2.0, 0.0))
```

```
Plot1(card.s.05.10$fitCosineoG, Main="Cardioid", Pch="1")
```

```
Plot2(card.s.25.10$fitCosineoG, Pch="2", 0.5, 0.01)
```

```
Plot2(card.s.50.10$fitCosineoG, Pch="3")
```

```
Plot2(card.s.75.10$fitCosineoG, Pch="4")
```

```
Plot2(card.s.95.10$fitCosineoG, Pch="5")
```

```
Plot3(card.s.05.5$fitCosineoG, Pch="1")
```

```
Plot3(card.s.25.5$fitCosineoG, Pch="2", 0.5, 0.01)
```

```
Plot3(card.s.50.5$fitCosineoG, Pch="3")
```

```
Plot3(card.s.75.5$fitCosineoG, Pch="4")
```

```
Plot3(card.s.95.5$fitCosineoG, Pch="5")
```

```
Plot1(tri.s.05.10$fitCosineoG, Main="Triangular", Pch="1")
```

```
Plot2(tri.s.25.10$fitCosineoG, Pch="2", 0.5, 0.01)
```

```
Plot2(tri.s.50.10$fitCosineoG, Pch="3")
```

```
Plot2(tri.s.75.10$fitCosineoG, Pch="4")
```

```
Plot2(tri.s.95.10$fitCosineoG, Pch="5")
```

```
Plot3(tri.s.05.5$fitCosineoG, Pch="1")
```

```
Plot3(tri.s.25.5$fitCosineoG, Pch="2", 0.55, 0.01)
```

```
Plot3(tri.s.50.5$fitCosineoG, Pch="3")
```

Plot3(tri.s.75.5\$fitCosineoG, Pch="4")
Plot3(tri.s.95.5\$fitCosineoG, Pch="5")

Plot1(vM.s.05.10\$fitCosineoG, Main="von Mises", Pch="1")

Plot2(vM.s.25.10\$fitCosineoG, Pch="2")
Plot2(vM.s.50.10\$fitCosineoG, Pch="3")
Plot2(vM.s.75.10\$fitCosineoG, Pch="4")
Plot2(vM.s.95.10\$fitCosineoG, Pch="5")

Plot3(vM.s.05.5\$fitCosineoG, Pch="1")
Plot3(vM.s.25.5\$fitCosineoG, Pch="2")
Plot3(vM.s.50.5\$fitCosineoG, Pch="3")
Plot3(vM.s.75.5\$fitCosineoG, Pch="4")
Plot3(vM.s.95.5\$fitCosineoG, Pch="5")

Plot1(WrC.s.05.10\$fitCosineoG, Main="Wrapped Cauchy", Pch="1")

Plot2(WrC.s.25.10\$fitCosineoG, Pch="2")
Plot2(WrC.s.50.10\$fitCosineoG, Pch="3")
Plot2(WrC.s.75.10\$fitCosineoG, Pch="4")
Plot2(WrC.s.95.10\$fitCosineoG, Pch="5")

Plot3(WrC.s.05.5\$fitCosineoG, Pch="1")
Plot3(WrC.s.25.5\$fitCosineoG, Pch="2")
Plot3(WrC.s.50.5\$fitCosineoG, Pch="3")
Plot3(WrC.s.75.5\$fitCosineoG, Pch="4")
Plot3(WrC.s.95.5\$fitCosineoG, Pch="5")

Plot1(U.s.10\$fitCosineoG, Main="Uniform", Pch="1")
Plot3(U.s.5\$fitCosineoG, Pch="1")

L.13 Plot Figures M-6 to M-10

Figure M-6

```
range=5
x <- seq(0,range,length=101)/range
a1=0.5; a2=0.8; a3=2

PlotCosModels <- function(a)
{
  y <- 2^(1-a) * (gamma(a))^(-1) * x^a * besselK(x, a)
  plot(x, y, ty="l" , col=1, xlab=paste("a=",a), ylim=c(0,1))
}

par(mfrow=c(3,1), mai=c(0.625, 0.50, 0.15, 0.25))
PlotCosModels(a=a1); PlotCosModels(a=a2); PlotCosModels(a=a3)
```

Figure M-7

```
range=5
x <- seq(0,range,length=101)/range
a1=0.1; a2=1; a3=2
b1=0.1; b2=1.5; b3=3
c1=2; c2=4; c3=6

PlotCosModels <- function(a,b)
{
  y <- (1+(1-b/c1)*x^a)*(1+x^a)^(-1-b/a)
  plot(x, y, ty="l" , col=1, xlab=paste("a=",a), ylab=paste("b=", b))
  y <- (1+(1-b/c2)*x^a)*(1+x^a)^(-1-b/a)
  lines(x,y, col="tan")
  y <- (1+(1-b/c3)*x^a)*(1+x^a)^(-1-b/a)
  lines(x,y, col=1, lwd=2, lty=2)
}
```

```
par(mfrow=c(3,3))
PlotCosModels(a=a1,b=b3); PlotCosModels(a=a2,b=b3); PlotCosModels(a=a3,b=b3)
PlotCosModels(a=a1,b=b2); PlotCosModels(a=a2,b=b2); PlotCosModels(a=a3,b=b2)
PlotCosModels(a=a1,b=b1); PlotCosModels(a=a2,b=b1); PlotCosModels(a=a3,b=b1)
```

Figure M-8

```
range=5
x <- seq(0,range,length=101)/range
a1=0.1; a2=1; a3=2
b1=0.1; b2=1.5; b3=3
```

```
PlotCosModels <- function(a,b)
{
  y <- (1+x^a)^(-b/a)
  plot(x, y, ty="l" , col=1, xlab=paste("a=",a), ylab=paste("b=", b))
}
```

```
par(mfrow=c(3,3))
PlotCosModels(a=a1,b=b3); PlotCosModels(a=a2,b=b3); PlotCosModels(a=a3,b=b3)
PlotCosModels(a=a1,b=b2); PlotCosModels(a=a2,b=b2); PlotCosModels(a=a3,b=b2)
PlotCosModels(a=a1,b=b1); PlotCosModels(a=a2,b=b1); PlotCosModels(a=a3,b=b1)
```

```
# Figure M-9
```

```
range=5; x <- seq(0,range,length=101)/range  
a1=1; a2=5; a3=10  
b1=1; b2=5; b3=10  
c1=1; c2=2; c3=3
```

```
PlotCosModels <- function(a,b)  
{  
  y <- c1^(-b) * (besselK(a*c1, b))^(-1) * (c1^2 + x^2)^(0.5*b) * besselK(a*sqrt(c1^2 + x^2), b)  
  plot(x, y, ty="l", col=1, xlab=paste("a=",a), ylab=paste("b=", b), ylim=c(0,1))  
  y <- c2^(-b) * (besselK(a*c2, b))^(-1) * (c2^2 + x^2)^(0.5*b) * besselK(a*sqrt(c2^2 + x^2), b)  
  lines(x,y, col="tan")  
  y <- c3^(-b) * (besselK(a*c3, b))^(-1) * (c3^2 + x^2)^(0.5*b) * besselK(a*sqrt(c3^2 + x^2), b)  
  lines(x,y, col=1, lwd=2, lty=2)  
}
```

```
par(mfrow=c(3,3))  
PlotCosModels(a=a1,b=b3); PlotCosModels(a=a2,b=b3); PlotCosModels(a=a3,b=b3)  
PlotCosModels(a=a1,b=b2); PlotCosModels(a=a2,b=b2); PlotCosModels(a=a3,b=b2)  
PlotCosModels(a=a1,b=b1); PlotCosModels(a=a2,b=b1); PlotCosModels(a=a3,b=b1)
```

```
# Figure M-10
```

```
range=5; x <- seq(0,range,length=101)/range  
a1=0.5; a2=1; a3=2
```

```
PlotCosModels <- function(a)  
{  
  y <- exp(-x^a)  
  plot(x, y, ty="l", col=1, xlab=paste("a=",a))  
}
```

```
par(mfrow=c(3,1), mai=c(0.625, 0.50, 0.15, 0.25))  
PlotCosModels(a=a1); PlotCosModels(a=a2); PlotCosModels(a=a3)
```

L.14 Plot Figures 4-3 and 4-4

```
# Figure 4-3, Cosine Model Behavior around the Observation Location 0
```

```
sample4.x <- c(-20, -10, 0, 10, 20); sample4.y <- c(0,0,0,0,0); sample4.direction <- c(pi/2, 0, pi/2, 0, pi/2)
x4 <- seq(-20, 20, length.out=201); y4 <- rep(0, 201)
```

```
krig4.e <- KrigCRF(krig.x=x4, krig.y=y4, resid.x=sample4.x, resid.y=sample4.y, resid.direction= sample4.direction, Model="exponential",
  Nugget=0.0, Range=10.0, sill=0, Plot=FALSE)
krig4.g <- KrigCRF(krig.x=x4, krig.y=y4, resid.x=sample4.x, resid.y=sample4.y, resid.direction= sample4.direction, Model="gauss",
  Nugget=0.0, Range=10.0, sill=0, Plot=FALSE)
krig4.s <- KrigCRF(krig.x=x4, krig.y=y4, resid.x=sample4.x, resid.y=sample4.y, resid.direction= sample4.direction, Model="spherical",
  Nugget=0.0, Range=10.0, sill=0, Plot=FALSE)
```

```
par(mai=c(0.65, 0.6, 0.05, 0.05), lab=c(5,10,7), mgp=c(2,1,0))
plot( krig4.g$x, krig4.g$direction*180/pi, ty="l", col="tan", lwd=4, xlim=c(-10,1), ylim=c(0, 100),
  xlab="Location", ylab="Kriging Estimate in Degrees")
lines(krig4.s$x, krig4.s$direction*180/pi, col=2, lty=2)
lines( krig4.e$x, krig4.e$direction*180/pi, col=1)
```

```
krig5.e <- KrigCRF(krig.x=x4, krig.y=y4, resid.x=sample4.x, resid.y=sample4.y, resid.direction= sample4.direction, Model="exponential",
  Nugget=0.1, Range=10.0, sill=0, Plot=FALSE)
krig5.g <- KrigCRF(krig.x=x4, krig.y=y4, resid.x=sample4.x, resid.y=sample4.y, resid.direction= sample4.direction, Model="gauss",
  Nugget=0.1, Range=10.0, sill=0, Plot=FALSE)
krig5.s <- KrigCRF(krig.x=x4, krig.y=y4, resid.x=sample4.x, resid.y=sample4.y, resid.direction= sample4.direction, Model="spherical",
  Nugget=0.1, Range=10.0, sill=0, Plot=FALSE)
```

```
par(mai=c(0.65, 0.05, 0.05, 0.05), yaxt="n")
plot( krig5.g$x, krig5.g$direction*180/pi, ty="l", col="tan", lwd=4, xlim=c(-10,1), ylim=c(0, 100), xlab="Location", ylab="")
lines(krig5.s$x, krig5.s$direction*180/pi, col=2, lty=2)
lines( krig5.e$x, krig5.e$direction*180/pi, col=1)
```

Figure 4-4

```
PlotCKVar(rho=0, ng=0, range=4, x.legend=4.0, y.legend=1.5)  
PlotCKVar(rho=0.5, ng=0, range=4, x.legend=3.5, y.legend=1.0)  
PlotCKVar(rho=0, ng=0.2, range=4, x.legend=3.5, y.legend=1.25)  
PlotCKVar(rho=0, ng=0, range=8, x.legend=4.5, y.legend=1.0)
```


Appendix M

Cosine Curves of Simulated Circular Random Fields (CRF)

M.1 Review

This appendix characterizes the circular-spatial correlation produced by the method of simulating CRFs of Chapter 5. This form of circular-spatial correlation is expressed as the mean cosine of the angle between random components of directions vs. distance between observation locations, which is required to solve the kriging of circular RV of Chapter 4. Since the method involves transformations of the nonclosed form of the normal CDF and a circular inverse CDFs, some of which do not have closed form expression, exact expressions for the cosine curves were not derived. Instead, the cosine curves produced by the method were characterized by fitted models adapted from the covariance models used for kriging of linear RV. The covariance models with variance 1 were adapted by Equation (5.2) in Chapter 5, Subsection 5.4.2.2 to the behavior of the mean cosine of a CRF.

M.2 Generation of Cosine vs. Distance Curves

For characterization of the mean cosine vs. distance of the simulated CRF, cosine vs. distance curves were generated using the R code in Appendices K.14 and L.10. The commands in L.10 specify 126 different sets of simulation inputs based on:

- The cardioid, triangular, uniform, von Mises, and wrapped Cauchy circular probability distributions as described in Table 5-1 in Chapter 5, Section 5.3,
- 5 values of the parameter ρ of a circular probability distribution (See Table M-1),
- Range r , the scale parameter of the covariance model of the Gaussian random field (GRF), at 5 and 10, and

Table M-1. Mean Resultant Vector Length ρ of Circular Distributions for Figures M-2, M-3, and M-4.

Index	% of ρ max	Cardioid	Triangular	Uniform	von Mises	Wrapped Cauchy
1	5%	0.025	0.020	0.000	0.050	0.050
2	25%	0.125	0.101		0.250	0.250
3	50%	0.25	0.203		0.500	0.500
4	75%	0.375	0.304		0.750	0.750
5	95%	0.475	0.385		0.950	0.950
	Range of ρ	$0 < \rho \leq 0.5$	$0 < \rho \leq 4/\pi^2$	$\rho = 0^*$	$0 < \rho < 1$	$0 < \rho < 1$

* All directions of the uniform circular distribution have equal probability density resulting in a mean resultant length ρ of zero.

- Exponential, Gaussian, spherical covariance models $c(d)$ of the GRF as listed in Chapter 5, Subsection 5.2.2, step 4).

For each of the 126 sets of inputs, 400 simulations of the CRF were computed without standardization of the GRF (Chapter 5, Section 5.3, step 1). For each simulation, 400 observations of direction at random sample locations were computed, and a cosineogram (Chapter 3) was evaluated.

For each of the 126 sets of inputs, the cosineograms points were collected into a datasets. Each dataset was reduced to a curve of cosine vs. distance by local polynomial regression of degree 1 (Wand and Jones 1995). These curves will be shown in the next section.

Do these curves represent the underlying spatial dependence produced by the method of simulation? Figure M-1 was constructed using the R code in Appendix L, Section L.11. 400 variograms of simulations of a GRF with spherical covariance, range=10, and variance=1, were reduced by local polynomial regression of degree 1 to a curve of variance vs. distance. The variance vs. distance curve was transformed to covariance vs. distance (Bailey and Gatrell 1995, p. 163).

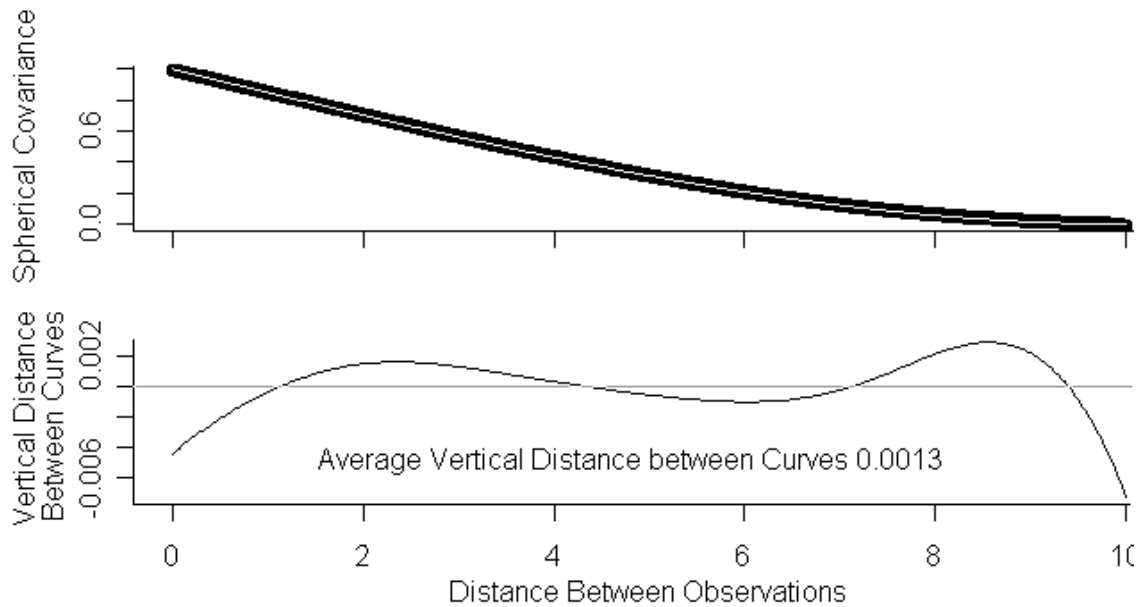


Figure M-1, Fitted Covariogram an Unbiased Estimator of Spherical Covariance

In the top plot, the theoretical covariance is over plotted in white on the covariance curve from the simulations (black). Visually, the curves coincide indicating that the fitted curve is unbiased. The bottom plot shows the difference of the covariance curve minus the theoretical covariance. The average vertical distance between curves is 0.001. Thus, the proximity of the curves provides verification by simulation that the curves in the next section are unbiased estimates of the spatial dependence produced by the method of simulation, and justifies the approach of the next section.

M.3 Families of Cosine Curves

Figures M-2 to M-4, which were constructed using the R code in Appendix L, Section L.12, show 126 curves of the mean cosine vs. distance derived from the data sets. The combinations of ρ and range provide a full spectrum of cosine curves for each input covariance model of the Gaussian random field (GRF). The black curves have range=5 and the tan curves have range=10. ρ increases from (1) to (5) as specified by the values in Table M-1 because the maximum value of ρ depends on the circular probability distribution. Examination of these figures indicates that each GRF covariance model (exponential, Gaussian, spherical) produces a distinct family of CRF cosine curves of similar shape. The exponential covariance model in Figure M-2 produces a cosine curve that approaches a horizontal asymptote as distance increases similar to the exponential covariance model. The Gaussian covariance model in Figure M-3 produces a cosine curve that has an “S” shape similar to the Gaussian covariance model. The spherical covariance model in Figure M-4 produces a cosine curve that has a horizontal tangent at distance=range similar to the spherical covariance model.

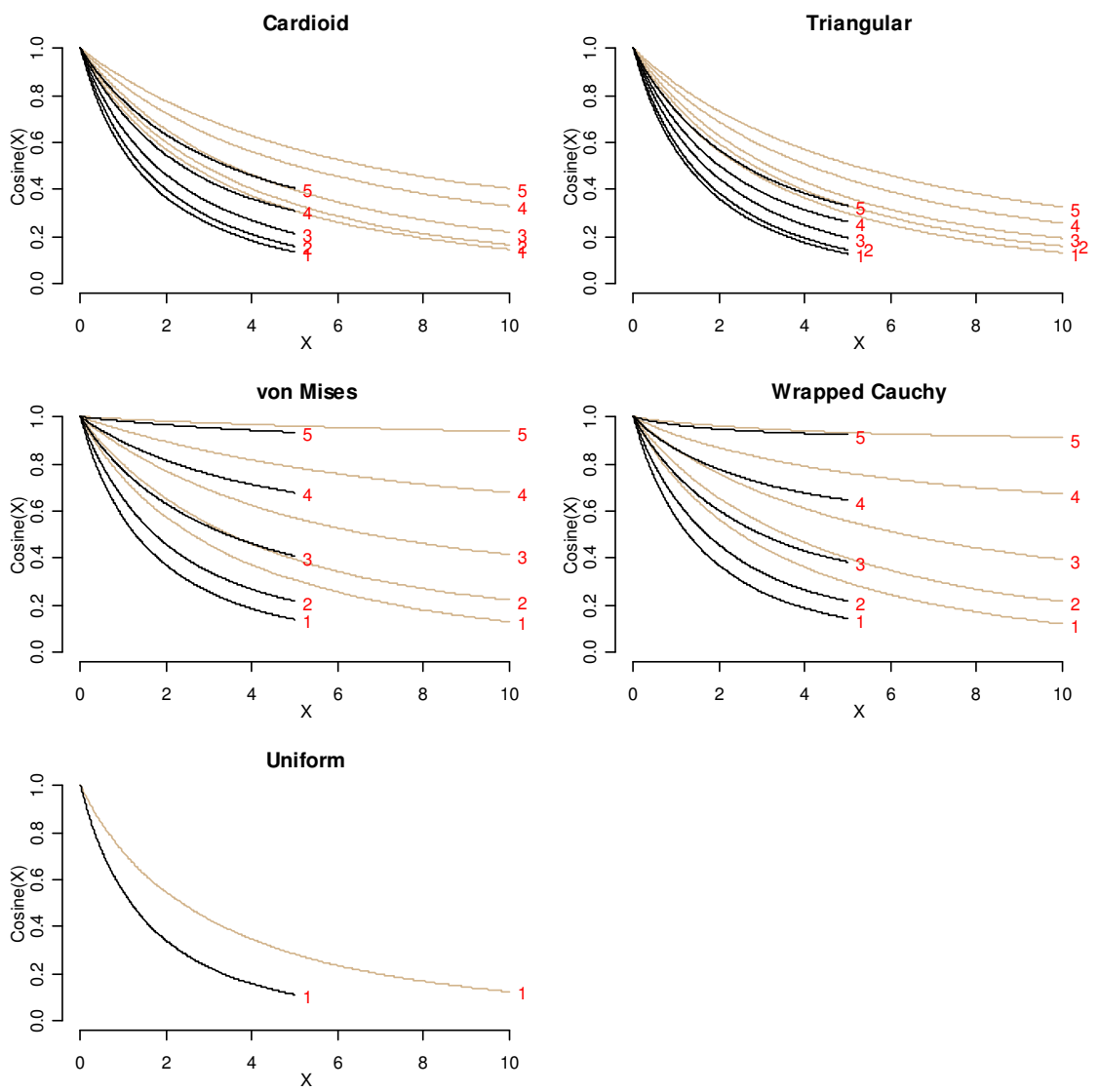


Figure M-2. Family of Cosine vs. Distance Curves from the GRF with Exponential Covariance. The black curves have range = 5 and the tan curves have range = 10. Within the classes of circular probability distributions and range value, the parameter ρ increases from bottom to top (Cross reference red numbers to Index values in Table M-1). The exponential covariance produces a curve that approaches a horizontal asymptote as distance increases similar to the exponential covariance model.

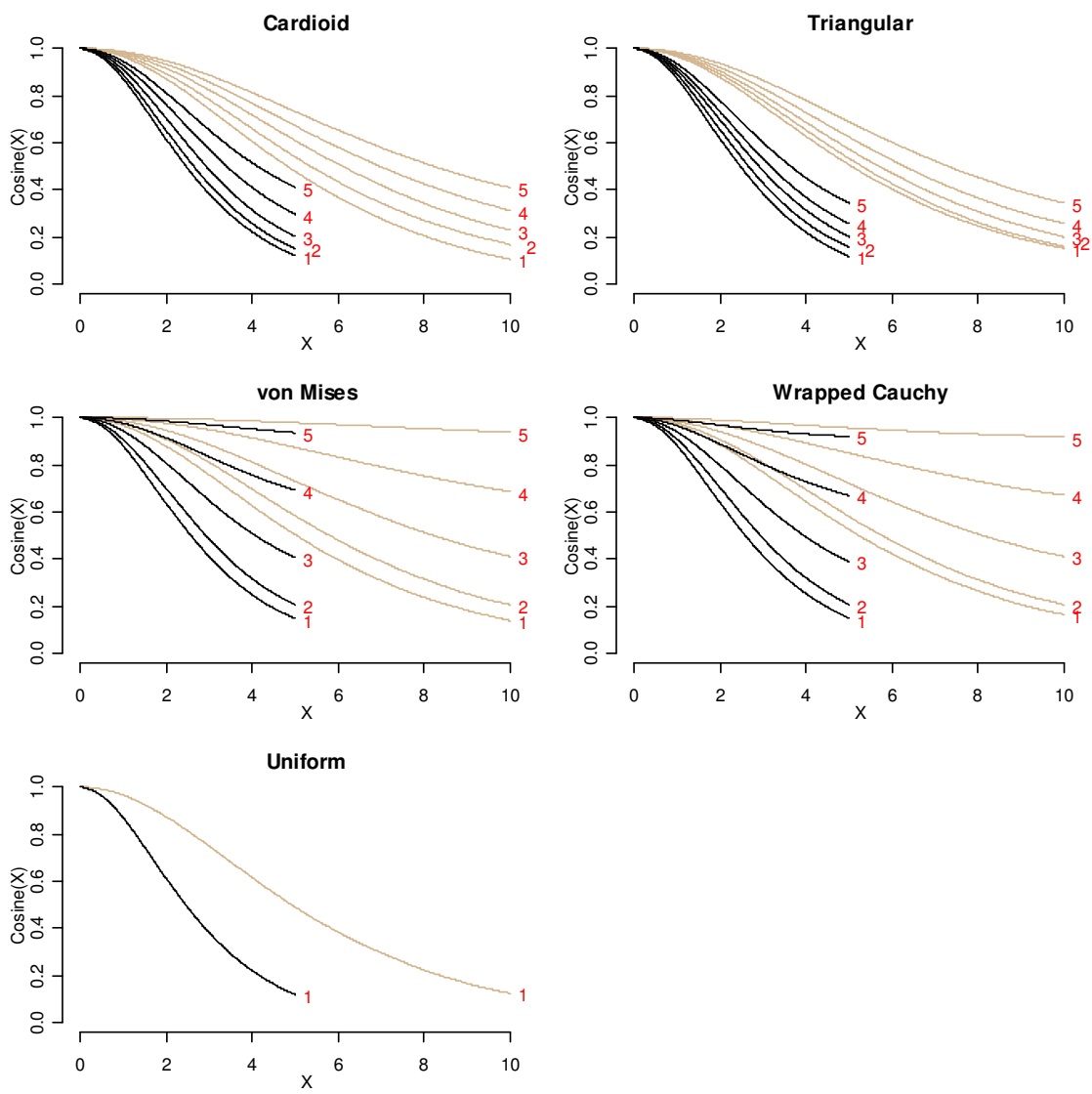


Figure M-3. Family of Cosine vs. Distance Curves from the GRF with Gaussian Covariance. The black curves have range = 5 and the tan curves have range = 10. Within the classes of circular probability distributions and range value, the parameter ρ increases from bottom to top (Cross reference red numbers to Index values in Table M-1). The Gaussian covariance produces a curve that has an “S” shape similar to the Gaussian covariance model.

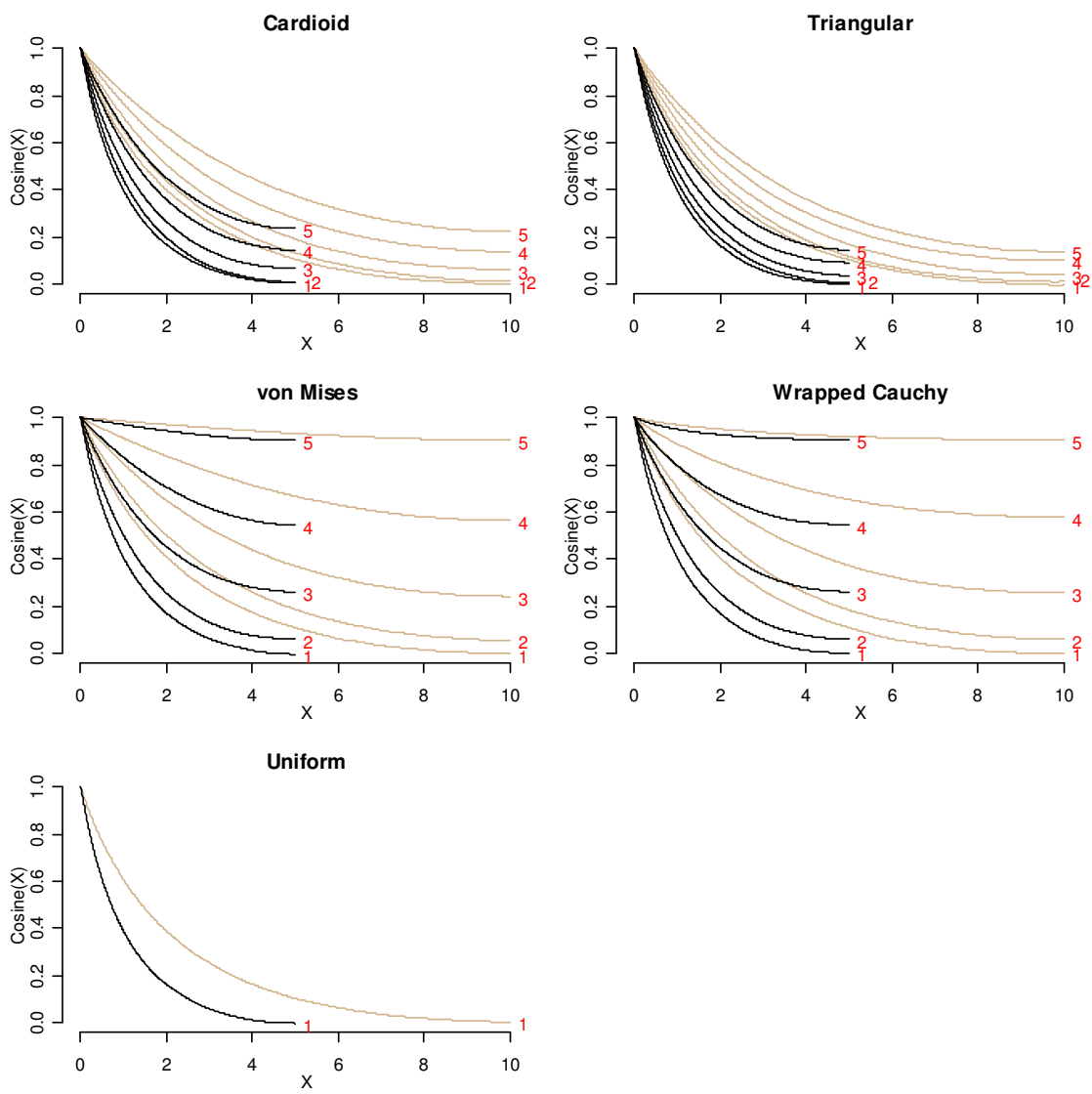


Figure M-4. Family of Cosine vs. Distance Curves from the GRF with Spherical Covariance. The black curves have range = 5 and the tan curves have range = 10. Within the classes of circular probability distributions and range value, the parameter ρ increases from bottom to top (Cross reference red numbers to Index values in Table M-1). The spherical covariance produces a curve that has a horizontal tangent at distance = range similar to the spherical covariance model.

M.4 Characterization of the Cosine Curves

To characterize the cosine curves, covariance models for linear kriging were shifted and scaled to conform to the spatial correlation properties of the CRF. With $\zeta(d)$ the mean cosine of the angle between random components of direction as a function of distance d , ρ the mean resultant length of the circular probability distribution, $0 \leq \rho < 1$, n_g the nugget, $0 \leq n_g \leq 1 - \rho^2$, and $c(d)$ the covariance model with a maximum value of one, the cosine model was defined as

$$\zeta(d) = \begin{cases} 1, & d = 0 \\ \rho^2 + (1 - n_g - \rho^2)c(d), & d > 0. \end{cases} \quad (\text{M.1})$$

A cosine model with shape similar to the cosine curve was chosen. The cosine curve was fit using a plot such as Figure M-5, which was produced using the R code in Appendix K, Section K.15. The lower plot shows the mean absolute difference (MAD) between the cosine curve and cosine model as computed over distances from 0 to the range. The three parameter cosine model was fit as follows: The value of a parameter was changed to determine how to decrease the MAD. Adjustment of the parameter continued until the MAD began to increase. Then, the parameter was changed in the opposite direction at a finer resolution. When a local minimum was achieved, another parameter was adjusted to a local minimum. Next, the third parameter was adjusted to a local minimum. The entire process was repeated with the modification that when a parameter could not be adjusted, adjustment of the next parameter was attempted. When any parameter was adjusted, adjustment of the remaining parameters was attempted. When no further adjustment could be made, the parameters and the MAD were recorded.

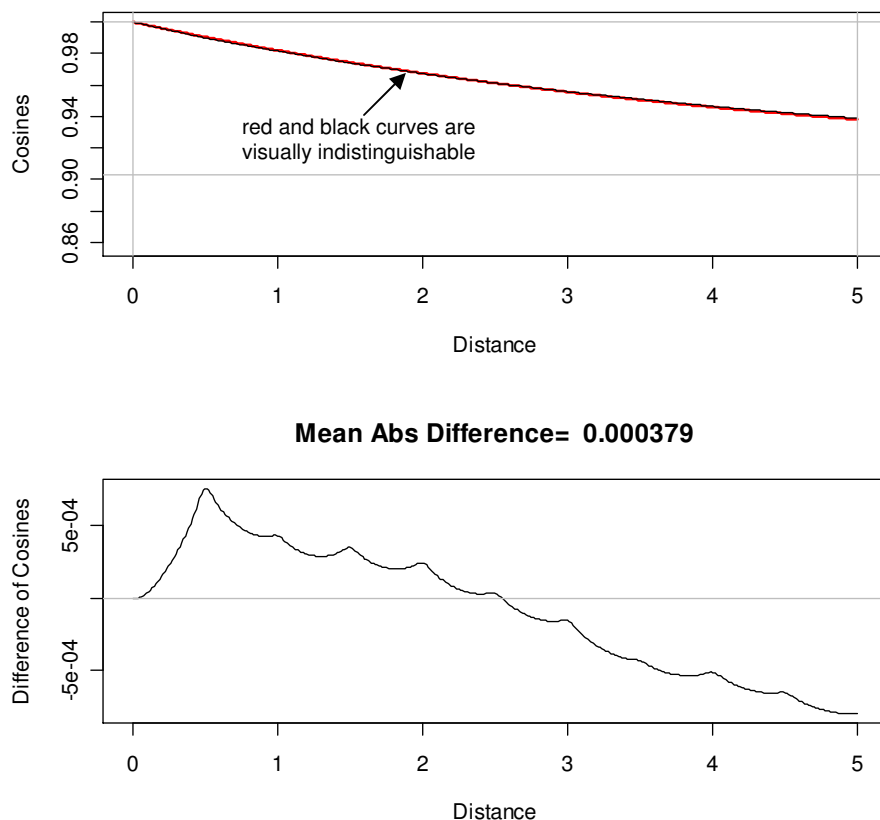


Figure M-5. Whittlematern Cosine Model ($a=.493$) Approximates the Cosine Curve of the von Mises CRF, $\rho = 0.95$, Transformed from an Exponential GRF, Range $r = 5$.

The characterizing models of reasonably close fit are tabulated in Table M-2. The input GRF covariance models are listed in the top rows of the sub tables. The dataset names in the left margin code the other simulation inputs of the distribution {Uniform (U), Cardioid (card), Triangular (tri), von Mises (vM), Wrapped Cauchy (WrC)}, the parameter ρ {.05, .95}, and the range r {5, 10}. The second row of the sub tables lists the characterizing models, which are expressed in Section M.5. In the main part of the table, the cosine model parameter values are listed in order a, b, c. Where fit was not reasonably close, additional models were evaluated. Where more than one model fits a cosine curve, the model with the minimum MAD may be chosen.

Table M-2. Cosine Models Approximating CRF Cosine Curves.

Cosine Models	Gaussian GRF							
	cauchybm a in (0,2] b > 0, c > 1		gencauchy a in (0,2] b > 0		hyperbolic a > 0, b > 0, c >= 0, or a > 0, b = 0, c > 0, or a >= 0, b < 0, c > 0		stable a in (0,2]	
	Parameters	MAD	Parameters	MAD	Parameters	MAD	Parameters	MAD
card.05.5	180, 2.90, 2.40	0.005	184, 5.39	0.004				
card.05.10	180, 2.85, 2.40	0.016	184, 5.60	0.007				
card.95.5	2.00, 1.60, 1.45	0.004	2.00, 3.93	0.004				
card.95.10	2.00, 1.58, 1.37	0.004	2.00, 3.92	0.005				
tri.05.5	184, 2.97, 2.40	0.003	185, 5.40	0.005				
tri.05.10	180, 2.68, 2.19	0.008	183, 5.02	0.002				
tri.95.5	2.00, 1.96, 1.80	0.001	2.00, 4.16	0.003				
tri.95.10	2.00, 1.94, 1.80	0.001	2.00, 4.13	0.002				
U.5	183, 2.65, 2.29	0.002	185, 5.47	0.003				
U.10	184, 2.85, 2.23	0.002	185, 5.40	0.003				
vM.05.5	180, 2.55, 2.10	0.004	186, 5.12	0.002				
vM.05.10	182, 2.71, 2.20	0.003	190, 5.50	0.002				
vM.95.5	fit not close	-	fit not close	-	7.45, 0.00, 3.30	0.002	1.95	0.001
vM.95.10	fit not close	-	fit not close	-				
WrC.05.5	186, 2.75, 2.32	0.002	190, 5.26	0.002				
WrC.05.10	186, 2.69, 2.33	0.003	190, 5.11	0.003				
WrC.95.5	190, 2.60, 2.20	0.001	191, 4.78	0.0003				
WrC.95.10	1.70, 2.60, 3.68	0.002	fit not close	-	0.50, -0.72, 0.47	0.002		

Cosine Models	Exponential GRF						Spherical GRF	
	cauchybm a in (0,2] b > 0, c >= 1		gencauchy a in (0,2] b > 0		whittlematern a > 0		cauchybm a in (0,2] b > 0, c >= 1	
	Parameters	MAD	Parameters	MAD	Parameters	MAD	Parameters	MAD
card.05.5	0.90, 1.29, 1.00	0.002	0.95, 2.74	0.002			0.96, 2.42, 1.19	0.003
card.05.10	0.92, 1.35, 1.12	0.003	0.94, 2.65	0.003			0.97, 2.45, 1.19	0.002
card.95.5	1.02, 1.01, 1.01	0.006	1.03, 2.05	0.005			1.16, 2.12, 1.07	0.005
card.95.10	1.05, 1.05, 1.01	0.003	1.09, 2.29	0.003			1.17, 2.07, 1.00	0.006
tri.05.5	0.88, 1.27, 1.00	0.008	0.92, 2.64	0.006			0.98, 2.52, 1.20	0.002
tri.05.10	0.92, 1.51, 1.30	0.002	0.92, 2.60	0.005			1.00, 2.70, 1.29	0.003
tri.95.5	1.02, 1.09, 1.02	0.003	1.04, 2.28	0.002			1.13, 2.11, 1.00	0.005
tri.95.10	1.02, 1.08, 1.02	0.005	1.04, 2.28	0.003			1.14, 2.25, 1.04	0.005
U.5	0.93, 1.57, 1.24	0.002	0.93, 2.84	0.005			0.94, 2.24, 1.07	0.003
U.10	0.91, 1.34, 1.00	0.003	0.96, 2.92	0.004			0.98, 2.67, 1.31	0.002
vM.05.5	0.95, 1.55, 1.33	0.002	0.96, 2.76	0.001			0.97, 2.30, 1.09	0.003
vM.05.10	0.95, 1.55, 1.32	0.002	0.96, 2.77	0.003			0.98, 2.45, 1.18	0.003
vM.95.5	fit not close	-	fit not close	-	0.46	0.0009	1.42, 1.99, 1.00	0.001
vM.95.10	fit not close	-	fit not close	-	0.48	0.0003	1.42, 2.00, 1.00	0.001
WrC.05.5	0.93, 1.56, 1.40	0.002	0.93, 2.63	0.002			0.99, 2.55, 1.21	0.003
WrC.05.10	0.93, 1.52, 1.26	0.003	0.93, 2.71	0.006			0.98, 2.48, 1.20	0.003
WrC.95.5	0.96, 1.30, 1.47	0.0003	0.97, 2.15	0.0006			1.05, 2.25, 1.22	0.0006
WrC.95.10	0.95, 1.29, 1.00	0.001	0.99, 2.80	0.001			1.07, 2.26, 1.19	0.0008

M.5 Expressions for the Cosine Models of Table M-2

The models in Table M-2, which approximate the cosine curves, and hence characterize them, were adapted from R package RandomFields (Schlather 2001) function CovarianceFct by scaling and shifting (Chapter 5, Subsection 5.4.2.2, (5.2)). In this section, these cosine models $\zeta(x)$ will be simplified by assuming the nugget $n_g = 0$, and expressed in terms of distance d , range r , $x = d/r$, ρ the resultant mean length parameter of the circular probability distribution, $0 \leq \rho < 1$, and covariance parameters a , b , and c . Figures M-6 to M-10 were computed using the R code in Appendix L, Section L.13.

M.5.1 Whittlematern:

$$\zeta(x) = \rho^2 + (1 - \rho^2) \frac{2^{(1-a)}}{\Gamma(a)} x^a K_a(x), \quad a > 0 \quad (\text{M.2})$$

$K_a(x)$ denotes the modified Bessel function of the third kind of order a .

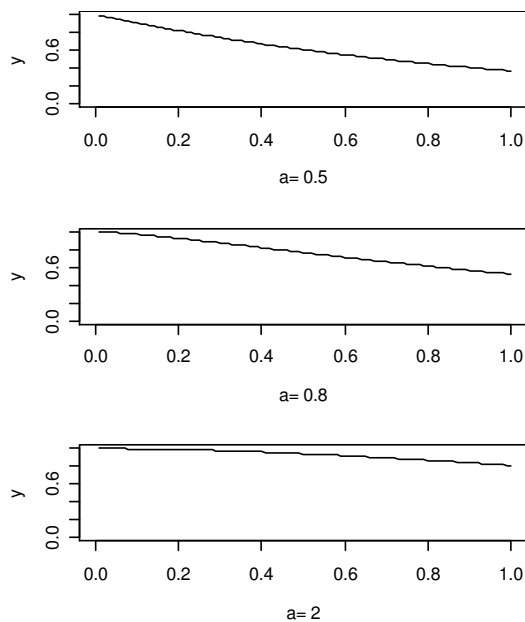


Figure M-6. Whittlematern Cosine Models for $\rho = 0$.

M.5.2 Cauchytbm:

$$\zeta(x) = \rho^2 + (1 - \rho^2) \left(1 + \left[1 - \frac{b}{c}\right] x^a\right) (1 + x^a)^{\left(\frac{b}{a} - 1\right)}, \quad a \in (0, 2], b > 0, c \geq 1 \quad (M.3)$$

Cauchytbm models are illustrated in Figure M-7 for $\rho = 0$ with parameter $c=2$ as black solid, $c=4$ as tan, and $c=6$ as thick dashed black curves.

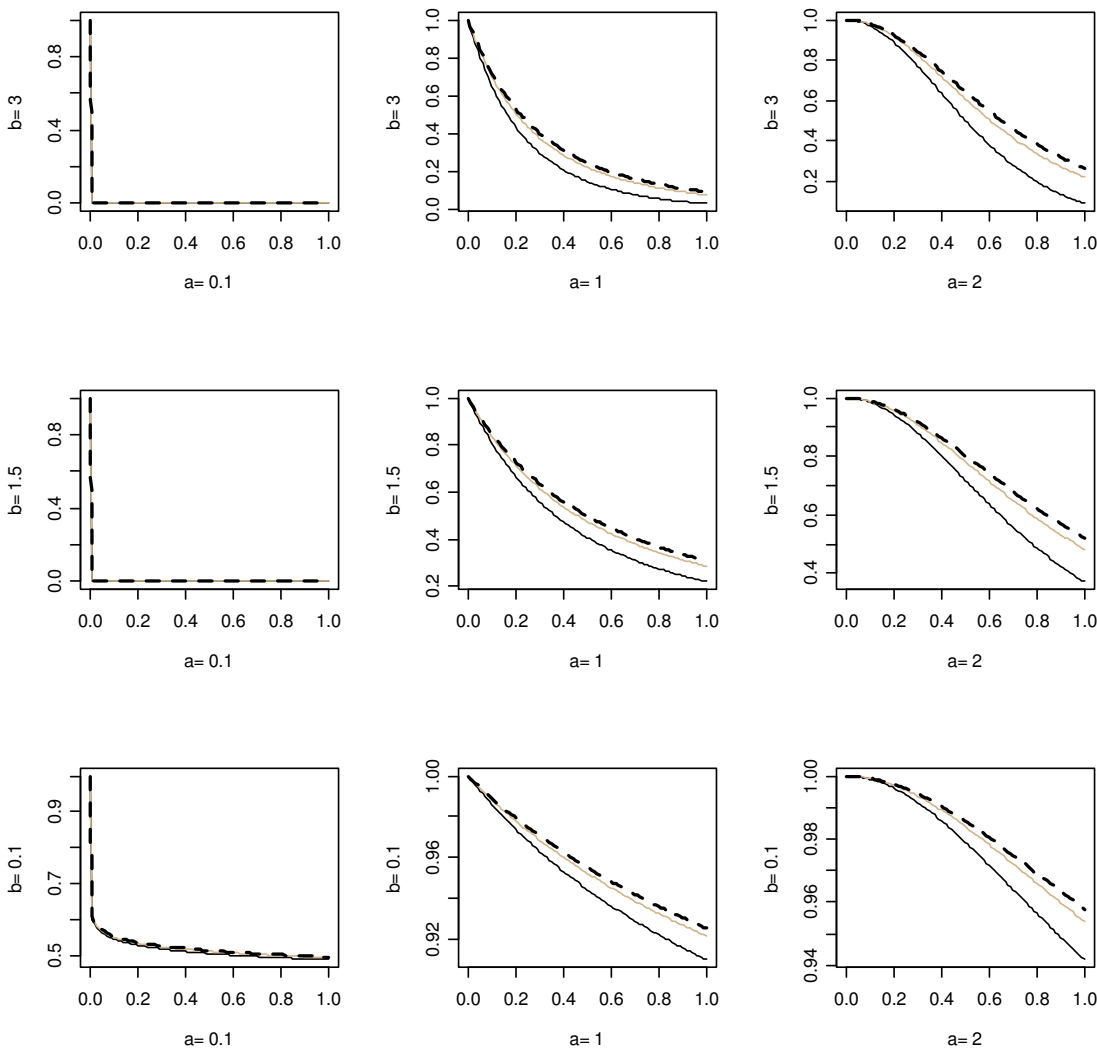


Figure M-7. Cauchytbm Cosine Models for $\rho = 0$.

M.5.3 Generalized Cauchy:

$$\zeta(x) = \rho^2 + (1 - \rho^2) (1 + x^a)^{-b/a}, \quad a \in (0, 2], b > 0 \quad (\text{M.4})$$

Generalized Cauchy cosine models are illustrated in Figure M-8.

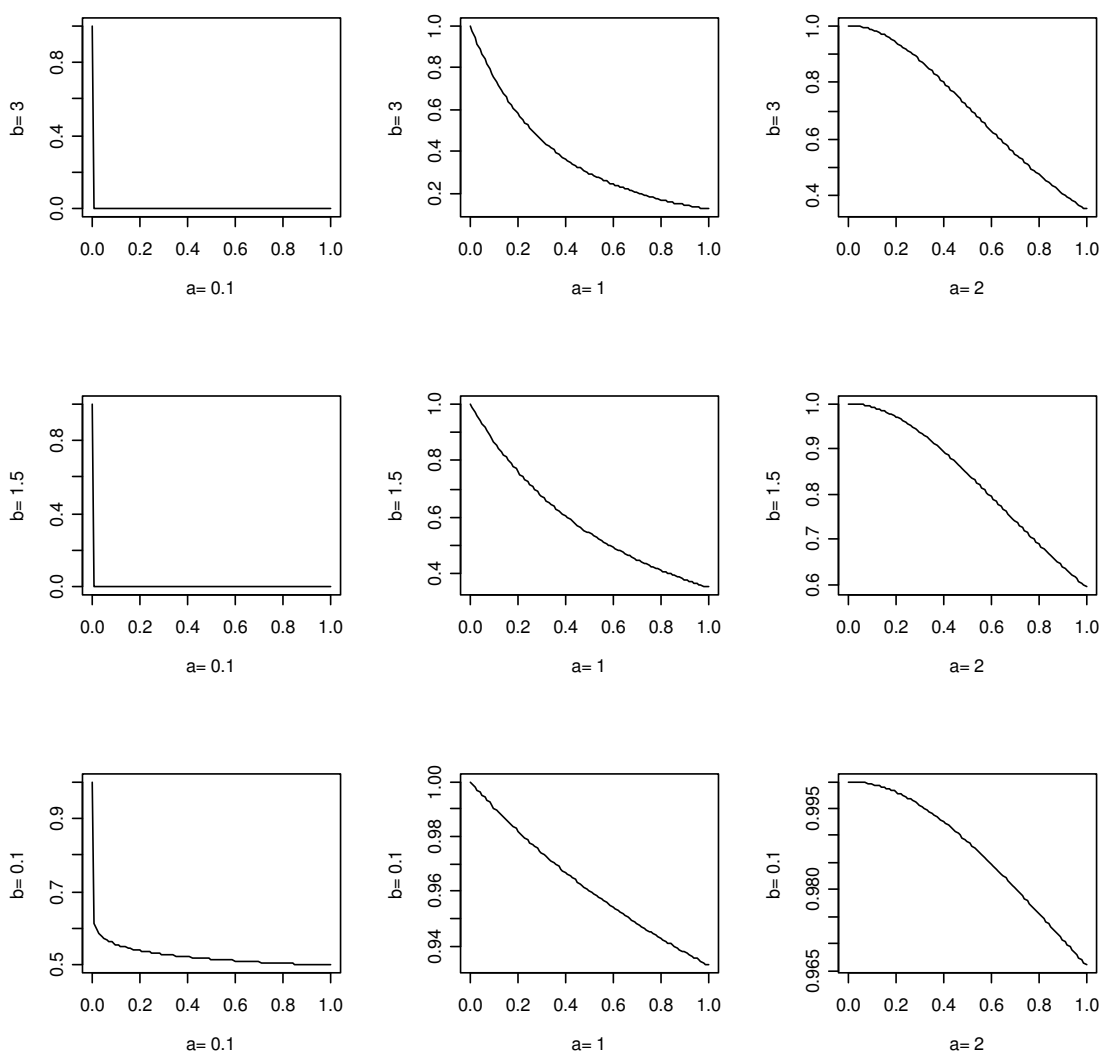


Figure M-8. Generalized Cauchy Cosine Models for $\rho = 0$.

M.5.4 Hyperbolic:

$$\zeta(x) = \rho^2 + (1 - \rho^2) * c^{-b} \frac{(c^2 + x^2)^{0.5b}}{K_b(a * c)} * K_b(a\sqrt{c^2 + x^2}) \tag{M.5}$$

The parameter constraints are {c>=0, a>0 and b>0}, or {c>0, a>0 and b=0}, or {c>0, a>=0, and b<0}. $K_b(x)$ denotes the modified Bessel function of the third kind of order b .

Hyperbolic models are illustrated in Figure M-9 for $\rho = 0$ with parameter $c=1$ as black solid, $c=2$ as tan, and $c=3$ as thick dashed black curves.

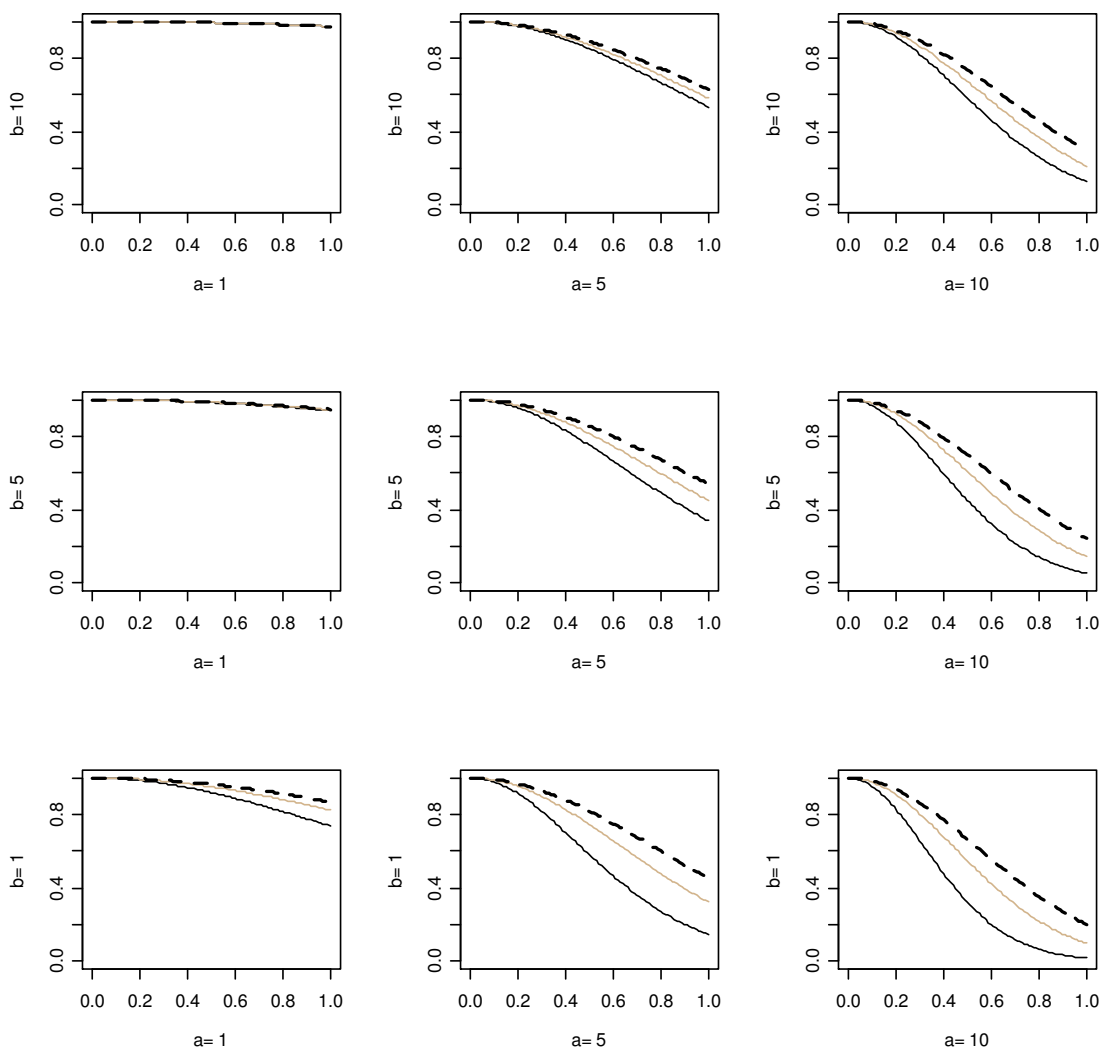


Figure M-9. Hyperbolic Cosine Models for $\rho = 0$.

M.5.5 Stable:

$$\zeta(x) = \rho^2 + (1 - \rho^2) \exp(-x^a), \quad a \in (0, 2] \quad (\text{M.6})$$

Stable models are illustrated in Figure M-10 for $\rho = 0$ with parameter $c=1$ as black solid, $c=2$ as tan, and $c=3$ as thick dashed black curves.

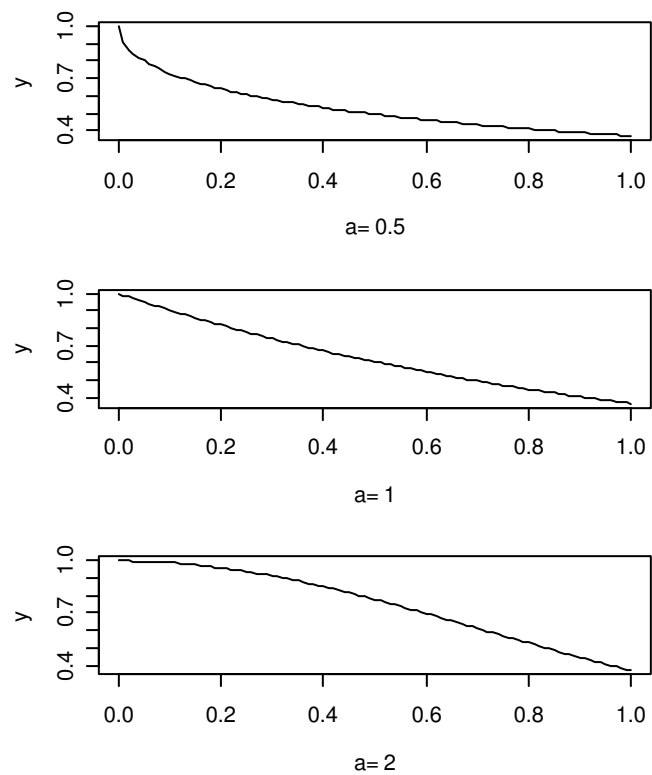


Figure M-10. Stable Cosine Models for $\rho = 0$.

M.6 Generalization of the Generation and Characterization of the Cosine Curves

The reader may choose from more than 30 GRF covariance models as provided by the R package Random Fields (Schlather 2001) function `CovarianceFct`. This function is embedded in the R code in Appendix K, Section K.14, `MakeCosineData`.

`MakeCosineData` provides 5 circular distributions (cardioid, triangular, uniform, von Mises, wrapped Cauchy) with any valid value of parameter ρ or κ , depending on the distribution, and any range r . Additional circular distributions can be implemented either as a CDF table as described in Chapter 5, Section 5.3, 3), point 3, a) “Compute a table of the desired circular CDF ...”, or as a closed form inverse CDF such as in 3), point 2 “For the triangular CRV ...”, as appropriate.

The reader may use the code in Appendix L, Section L.10 as a template for generating a large series of cosine datasets. The dataset name references the circular distribution, the GRF covariance model, and the range r . For a more complicated example, `U.GenCauchy_a_1.8_b_5.15_r_5` indicates that a uniform CRF was generated from a generalized Cauchy distribution with parameters $a = 1.8$, $b = 5.15$, and range $r = 5$. Note that these computations may take significant time, depending on the number of datasets, simulations, and observations per simulation.

The reader may use the code in Appendix L, Section L.12 as a template for plotting families of cosine curves produced.

Any of the covariance models of `CovarianceFct` may be used to fit a cosine curve because the code in Appendix K, Section K.15, `FitCosineData`, also uses the `CovarianceFct` function. The algorithm for achieving a sequence of decreasing local minima, as described in Section M.4, is tedious. The `FitCosineData` may be revised to implement other methods of fitting. To facilitate fitting, the inputs to `MakeCosineData` should be saved in the output object to be read by `FitCosineData`.

Appendix N

Additional Graphics for Circular Data

N.1 Summary Plots for Circular Data

Continuing from Chapter 1, Section 1.2, Figure N-1 (a) is a vector plot of the wind data. It shows magnitude (m/s) and direction. Quartiles of vector magnitude were colored, in order, blue, green, red, and violet, and the arrow heads of vectors were replaced with dots to eliminate coverage by the arrow heads. Figure N-1 (b) summarizes the circular data of Figure N-1 (a). The circular data plot is constructed similar to a histogram. The arc bin origin is 0° , bin width is 30° , and number of bins is 12. One dot for each observation is stacked on the outside of a circle at the center of its arc bin. The circular plot shows that the modal wind direction (arc bin with greatest number of observations) is toward about 105° (bin mid point) counterclockwise (CCW) from 0° E, or winds are blowing mostly from south-southeast to north-northwest.

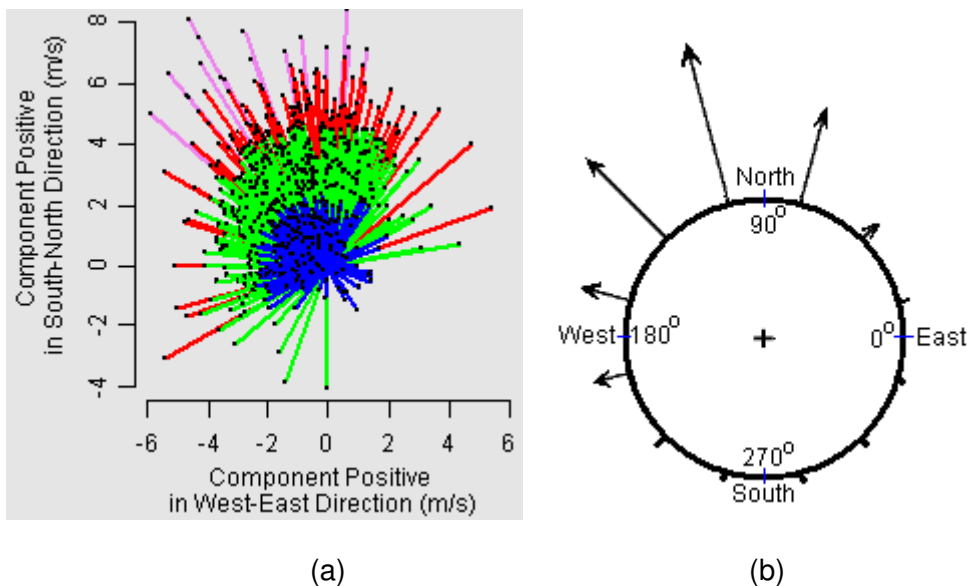


Figure N-1. Summary Plots of the Ocean Wind Data. (a) Magnitude (length) and direction wind blows toward, (b) frequency (length) and direction.

N.2 Histograms for Circular Data

The histogram is a common method of summarizing numerical data. The data are grouped into intervals and the number of observations in each interval is counted. A rectangular bar of area proportional to the count in an interval is centered above the midpoint of the interval. The vertical axis of the histogram provides a scale for bar height. With the histogram, we can see the frequency with which data occur relative to the horizontal scale value, whether frequency is uniform over the range, or has some structure such as being concentrated at some value on the horizontal scale. The histogram for circular data is constructed similarly. An arc bin origin (e.g., 0°) and arc bin width (e.g., 5° , 10° , 20° , or 30°) are selected, and the data are binned and counted. The next step is to wrap the horizontal axis of the histogram around a unit circle. The bars are aligned with the circle center and arc interval midpoint, plotted on the outside of the circle, and have length or area proportional to count or relative frequency. Vector magnitude is ignored.

The circular histograms in Figure N-2 summarize the same data as shown in Figure N-1 with the same bin origin of 0° and bin width of 30° . Figure N-2 (a) shows a rose plot (Florence Nightingale) of wind direction. The angle of the wedge is the bin width and the area of a wedge is proportional to the bin count. Figure N-2 (b) shows a circular histogram with a bar area corresponding to bin count. Figure N-2 (c) unwraps the circular histogram of Figure N-2 (b) onto a linear scale and repeats one period. The additional period facilitates visual extraction of period, counting of modes, and prevents breakup of features occurring near the cross over point of 360° (0°). Additionally, Figures N-2 (a) to (c) show the estimated mean (vector resultant direction) of 106.3° and the 95% confidence interval ($104.8^\circ, 107.8^\circ$) for the mean. This confidence interval is based on the von Mises distribution (Fisher 1993, pp. 88-89). In Figures N-2 (a) and (b),

the mean direction is indicated by a thick black radial line from the circle center with the confidence interval indicated by a black arc slightly outside the large circle. In Figure N-2 (c), the mean is indicated by a black solid vertical line enclosed in confidence limits displayed as vertical dashed lines.

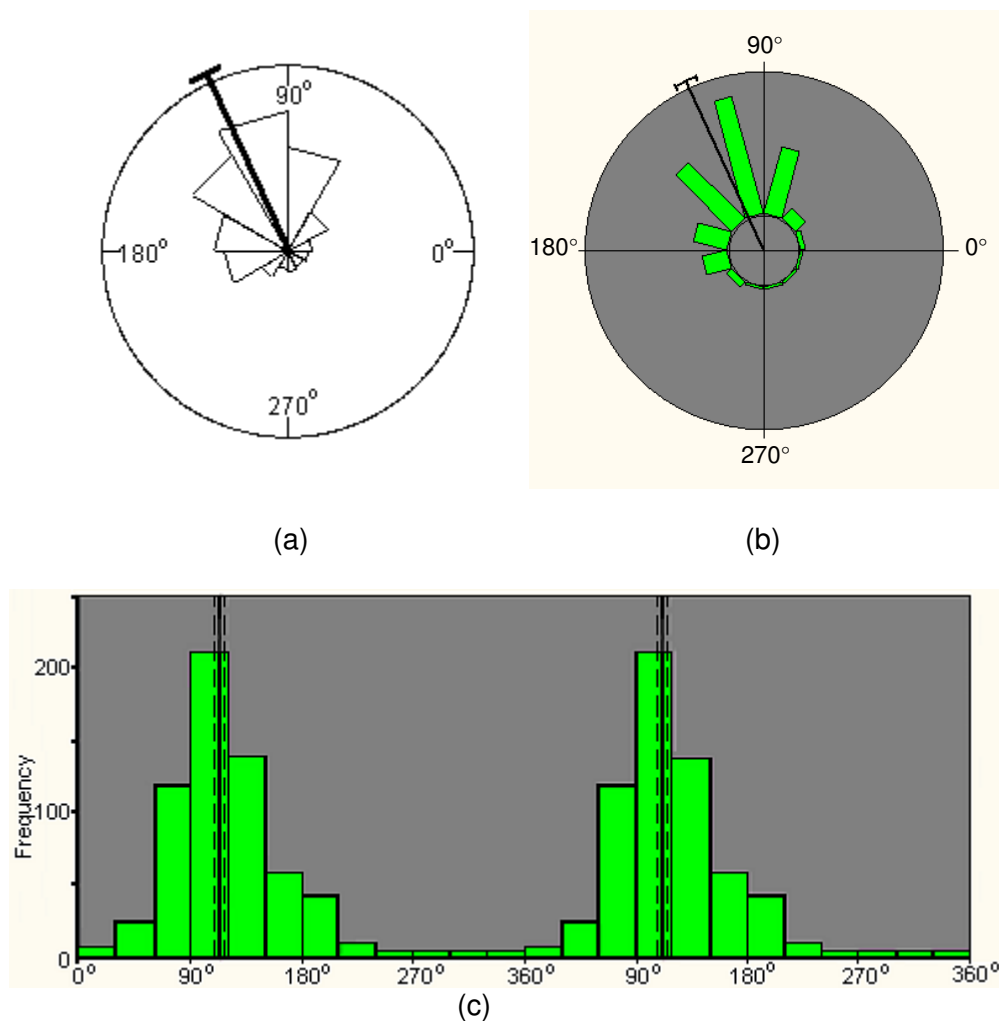


Figure N-2. Circular Histograms of the Ocean Wind Data. (a) Rose plot, (b) circular histogram, and (c) linearized circular histogram summarize circular data. (c) repeats the histogram to facilitate visual extraction of period, counting of modes, and prevent breakup of features occurring near the cross over point of 360°.

N.3 Nonparametric Density Plots for Circular Data

The kernel density estimates in Figure N-3 are based on the same data as was used in Figures N-1 and N-2. Circular histograms of Figure N-2, like histograms for linear variables, can distort the structural information in the sample about the number, locations, and sizes of modes through an arbitrary choice of bin origin and bin width. The nonparametric smoothed density estimate replaces the bin edge and origin decisions with an easier smoothing band width decision. In nonparametric smoothing, a symmetric unimodal function is centered on each observation. It effectively spreads out the mass of an observation with maximum value at the location of the observation, mimicking the stochastic process where the observation is an instance of a random direction and could have occurred at other locations in the neighborhood of the observed direction. Increasing the bandwidth increases smoothness and decreases noise. Decreasing bandwidth exposes more structural detail and noise. The smoothed histograms of Figure N-3 implement the kernel density method for a CRV given in Fisher (1993, pp. 26-27).

Figure N-3 (a) plots the estimated data density on the outside of a unit circle. The density at a given angle (direction) is the radial distance between the red and the grey curves at the given angle. Figure N-3 (b) shows the density of Figure N-3 (a) unwrapped onto a linear scale with an extra period. Like the linearized circular histogram in Figure N-2 (c), the extra period eliminates the breakup of features at the cross over point of 360° and makes it easier to assess structure. This density estimation method indicates that there is one mode at about 105° CCW from 0° E.

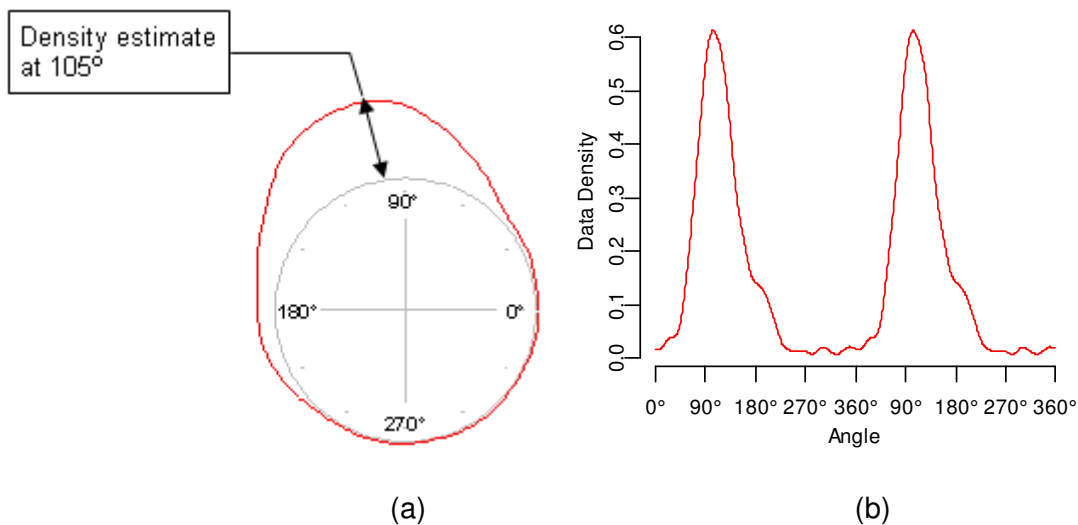


Figure N-3. Kernel Density Plots of the Ocean Wind Data. (a) Circular, (b) linear with extra period. There is one mode at about 105° CCW from 0° E.

N.4 New Cylindrical Plot of the Circular Probability Density

Figure N-3 (a) is an example of the traditional method of plotting the probability density function (PDF) of a CRV with radius equal to 1.0 + density with density depending on direction. Figure N-4 shows a new method for displaying the circular PDF. The PDF is drawn as a cylinder of unit radius with height of the cylinder at an angle equal to the probability density at the angle. Note that, as stated in Chapter 1, Section 1.1, the unit of angular measurement is radians because PDFs generally contain trigonometric functions that require angles in radian units. This method maintains the requirement that the area between the PDF and circular scale must equal 1.0 over the support. In general, the traditional circular display of the PDF does not integrate to one as does the linear display. For a circular uniform distribution with density $1/(2\pi)$ on $[0, 2\pi)$ or $[-\pi, +\pi)$ radians, we would draw circles of unit radius and radius $1 + 1/(2\pi)$. The area between the outer circle and the unit circle representing the total probability

should equal 1.0, but it is actually $1+1/(4\pi)$. However, the traditional display is easy to draw, intuitive, and is widely used.

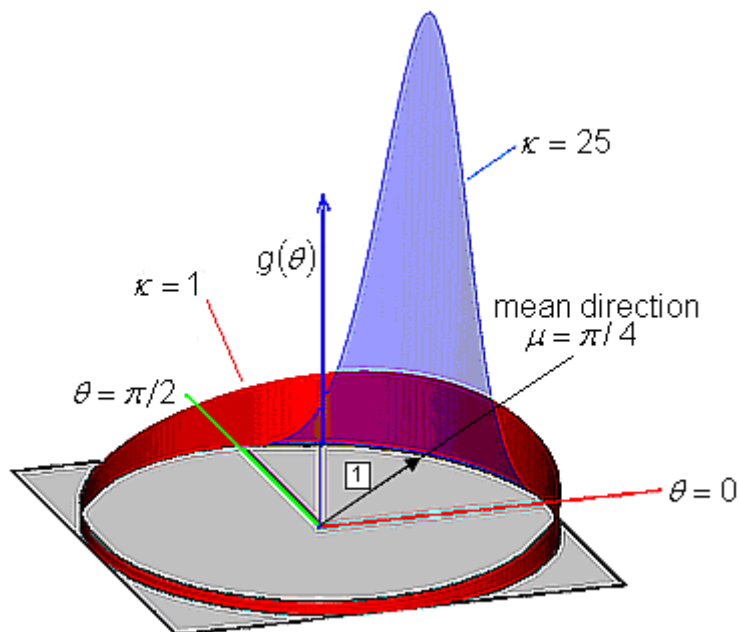


Figure N-4. New Cylindrical Plot of PDFs of von Mises Probability Densities. Direction is expressed in radian units. The distribution with $\kappa = 25$ is more concentrated and less variable than the distribution with $\kappa = 1$.

Figure N-4 shows two von Mises probability densities with concentration parameters $\kappa = 1, 25$ (see Chapter 3, Table 3-1). The concentration parameter κ is a measure of variability for the von Mises distribution equal to $\frac{1}{2}$ the log of the ratio of maximum density to the density at the opposite direction. Hence, large values correspond to distributions of low variability and concentrated about the mean.

Appendix O
Permissions

This section contains the documents for Technical Information Clearance (permissions) applicable to Figures 2-17 (b), and 2-18 (b) and (c). The first screenshot captures the request, and internal (ATK) and external (NASA) approvals for Technical Information Clearance.



ePIC FYI@Approved | PUBLICATION: PUB000073,-,6

SECTIONS OF CHAPTER OF THE PHD DISSERTATION SIMULATION, KRIGING, AND VISUALIZATI...

Step	Assignment	State	Status	Organization	First Name	MI	Last Name
10	Assign Signatures	Working	Complete	Submitter	William	J	Morphet
20	Preparer Work Assignment	InWork	Complete	Preparer	William	J	Morphet
30	Editor Review	InWork	Approved	Editor	Kathy	Noreen	Shafer
40	Intellectual Property Review	InReview	Approved	Intellectual Property	Jeremy		Sanders
40	Management Review	InReview	Approved	Management	David	E	Richardson
40	PM Review	InReview	Approved	PM	Norman	C	Allen
40	Technical Review	InReview	Approved	ResponsibleEng	Andrew	M	Eaton
40	Business Development Review	InReview	Approved	Business Development	Andrew	C	Haaland
40	Chief Engineer Review	InReview	Approved	ChiefEngineer	Frederick	M	Perkins
40	Export/Import Contol Office Review	InReview	Approved	Export/Import Control Office	Jennie	L	Campbell
50	S&E Vice President Review	InReview	Approved	S&E Vice President	David	H	Riemer
54	NASA Publication Send to Customer	InReview	Complete	NASA Publication Grp	Sheila	L	Burnette
58	NASA Publication Customer Review	Customer	Approved	NASA Publication Grp	Sheila	L	Burnette

This part of Permissions contains the documents pertaining to NASA approval of Technical Information Clearance.

Launch Systems Group
P.O. Box 707
Bingham City, UT 84302
www.atk.com

June 23, 2008
E670-CY08-031

Kim Narmore
National Aeronautics and Space Administration
Bldg. 4200, AD33
MSFC, AL 35812

This memo submits the attached; ATK-Thiokol Propulsion published Space Shuttle Reusable Solid Rocket Motor (RSRM) technical document(s) into the NASA-MSFC Scientific and Technical Document Availability Authorization (DAA) System for review and approval in accordance with NPG 2200.2.

<u>TITLE</u>	<u>AUTHOR(S)</u>	<u>CONTRACT NO.</u>	<u>CONF/FORUM</u>
Simulation, Kriging And Visualization of Circular-Spatial Data	W. Morphet	NAS8-97238	Dissertation Paper

Final disposition of this document is requested on an expedited basis, on or before 7/18/2008.

Please contact the undersigned at 544-6395 or by E-mail at Richard.Roth@atk.com with questions/comments.


Sincerely,



Richard B. Roth

Atch: a/s

JUN 23 2008

 National Aeronautics and Space Administration		NASA Scientific and Technical Information (STI) Document Availability Authorization (DAA) <i>(Instructions for completing this form are on page 4)</i>		<input checked="" type="checkbox"/> NEW DAA <input checked="" type="checkbox"/> REVISION TO EXISTING DAA <i>(Attach Copy)</i>	
SECTION 1					
1a. DOCUMENT IDENTIFICATION					
TITLE Sections of Chapter 2 of PhD dissertation "SIMULATION, KRIGING, AND VISUALIZATION OF CIRCULAR-SPATIAL DATA" * Only the data for Figures 2-16 and 2-17			AUTHOR(S) William J. Morphet		
POINT OF CONTACT (if different from author)		E-MAIL	ADDRESS		
COPIES OR CENTER-DESIGNATED OFFICIAL		E-MAIL	ADDRESS		
ORIGINATING NASA CENTER AND ORGANIZATION MFSC			PERFORMING NASA CENTER AND ORGANIZATION (if different) ATK		
CONTRACT NUMBER NAS3-97238 *		GRANT NUMBER	INTERAGENCY NUMBER	OTHER DOCUMENT NUMBER(S)	
1b. TYPE OF STI					
<input type="checkbox"/> ABSTRACT <input type="checkbox"/> BOOK NAME OF PUBLISHER: _____ <input type="checkbox"/> CONFERENCE PAPER <input type="checkbox"/> CONFERENCE PRESENTATION CONFERENCE NAME/SPONSOR: _____ CONFERENCE LOCATION: _____ DATES: _____ TO: _____ <input type="checkbox"/> ORAL/VISUAL PRESENTATION ADDITIONAL INFORMATION: _____ <input type="checkbox"/> NASA STI SERIES REPORT (See WPR 2200.2) <input type="checkbox"/> TECHNICAL PUBLICATION (TP) <input type="checkbox"/> TECHNICAL MEMORANDUM (TM) <input type="checkbox"/> CONTRACTOR REPORT (CR) <input type="checkbox"/> CONFERENCE PUBLICATION (CP) <input type="checkbox"/> SPECIAL PUBLICATION (SP) <input type="checkbox"/> TECHNICAL TRANSLATION (TT)			<input type="checkbox"/> DOCUMENT FOR PUBLIC WEB SITE (Must meet requirements in WPR 2010.1, chapter 11, for publishing on the Internet) <input type="checkbox"/> DOCUMENT FOR INTERNAL NASA WEB SITE URL: _____ PERSISTENT URL (if applicable): _____ <input checked="" type="checkbox"/> OTHER section of chapter of dissertation <i>(Periodical, format, etc.)</i> <input type="checkbox"/> JOURNAL ARTICLE JOURNAL NAME/PUBLISHER: _____ <input type="checkbox"/> IF A PREPRINT (draft or revision being sent to journal), DO YOU WANT IT DISSEMINATED TO: <input type="checkbox"/> ONLY THE PUBLISHER <input type="checkbox"/> NASA AND ITS CONTRACTORS <input type="checkbox"/> US GOVERNMENT AGENCIES AND THEIR CONTRACTORS <input type="checkbox"/> PUBLIC <input type="checkbox"/> IF A REPRINT (article published by journal) BEING SENT TO NASA CENTER FOR AEROSPACE INFORMATION TO ADD TO STI DATABASE, DID YOU GET PERMISSION FROM THE PUBLISHER TO DISTRIBUTE TO PUBLIC? <input type="checkbox"/> YES (Attach copy of permission) <input type="checkbox"/> NO		

2. NATIONAL SECURITY CLASSIFICATION (Check one)	
<input checked="" type="checkbox"/> NONE (Unclassified)	<input type="checkbox"/> CONFIDENTIAL <input type="checkbox"/> SECRET <input type="checkbox"/> TOP SECRET
IF CLASSIFIED, CENTER SECURITY OFFICER (CSO) MUST APPROVE	
CSO APPROVAL (Name)	
3. AVAILABILITY CATEGORY	
3a. PUBLICLY AVAILABLE STI (See instructions)	
<input checked="" type="checkbox"/> YES	<input type="checkbox"/> NO (See 3 (b, c, and/or d) and 4)
3b. EXPORT-CONTROLLED INFORMATION (Contact your Export Control Administrator for help)	
<input type="checkbox"/> INTERNATIONAL TRAFFIC IN ARMS REGULATIONS (ITAR) GIVE THE US MUNITIONS LIST (USML) CATEGORY: _____	<input type="checkbox"/> EXPORT ADMINISTRATION REGULATIONS (EAR) GIVE THE EXPORT CONTROL CLASSIFICATION NUMBER (ECCN): _____ OF THE COMMERCE CONTROL LIST (CCL)
3c. PROPRIETARY/SENSITIVE INFORMATION (See NPR 2200.2, Chapter 4)	
<input type="checkbox"/> LIMITED RIGHTS DATA	<input type="checkbox"/> PRODUCED UNDER SPACE ACT AGREEMENT & SUBJECT TO § 303(b) OF THE SPACE ACT
<input type="checkbox"/> SMALL BUSINESS INNOVATION RESEARCH (SBR/STTR)	<input type="checkbox"/> LIMITED UNTIL DATE (mm/dd/yyyy): _____
<input type="checkbox"/> LIMITED UNTIL DATE (mm/dd/yyyy): _____	<input type="checkbox"/> IMMEDIATE RELEASE APPROVED TO CATEGORIES INDICATED IN 3d
<input type="checkbox"/> IMMEDIATE RELEASE APPROVED TO CATEGORIES INDICATED IN 3d	<input type="checkbox"/> COPYRIGHTED (If copyrighted, check with Center Patent or Intellectual Property Counsel)
<input type="checkbox"/> TRADE SECRET/COMMERCIAL CONFIDENTIAL OTHER THAN LIMITED RIGHTS OR SBR DATA	<input type="checkbox"/> CAN BE RELEASED TO PUBLIC (Attach approval)
<input type="checkbox"/> LIMITED UNTIL DATE (mm/dd/yyyy): _____	<input type="checkbox"/> MUST BE RESTRICTED TO CATEGORIES INDICATED IN 3d
<input type="checkbox"/> IMMEDIATE RELEASE APPROVED TO CATEGORIES INDICATED IN 3d	<input type="checkbox"/> CONTAINS PUBLIC-WEBS-SENSITIVE INFORMATION PER NPR 2810.1
NAME OF APPROVING OFFICIAL	TITLE OF APPROVING OFFICIAL
_____	_____
3d. DISTRIBUTION LIMITATIONS (See instructions)	
<input type="checkbox"/> US GOVERNMENT AGENCIES AND US GOVERNMENT AGENCY CONTRACTORS ONLY	<input type="checkbox"/> DISTRIBUTION LIMITED TO US PERSONS
<input type="checkbox"/> US GOVERNMENT AGENCIES ONLY	<input type="checkbox"/> AVAILABLE ONLY WITH APPROVAL OF THE FOLLOWING ISSUING OFFICE:
<input type="checkbox"/> NASA PERSONNEL AND NASA CONTRACTORS ONLY	_____
<input type="checkbox"/> NASA CONTRACTORS AND US GOVERNMENT ONLY	<input type="checkbox"/> LIMITED UNTIL DATE (mm/dd/yyyy): _____
<input type="checkbox"/> NASA PERSONNEL ONLY	
4. DOCUMENT DISCLOSING AN INVENTION	
<input type="checkbox"/> CHECK IF THIS DOCUMENT/PRESENTATION DISCLOSES AN INVENTION AND ROUTE TO HQ OR CENTER PATENT OR INTELLECTUAL PROPERTY COUNSEL (See instructions)	
I CERTIFY THAT THIS DOCUMENT MAY BE RELEASED ON (mm/dd/yyyy): _____	
PATENT COUNSEL NAME	DATE
_____	_____

5. BLANKET AVAILABILITY AUTHORIZATION (Optional) (See instructions)			
<input type="checkbox"/> I APPROVE ALL DOCUMENTS ISSUED UNDER THE FOLLOWING CONTRACT, GRANT, OR PROJECT NUMBER TO BE PROCESSED AS CHECKED IN SECTIONS 2 AND 3(e-d) I GRANT THIS AVAILABILITY AUTHORIZATION ON (mm/dd/yyyy): _____			
<input type="checkbox"/> CONTRACT NUMBER	<input type="checkbox"/> GRANT NUMBER	<input type="checkbox"/> PROJECT NUMBER	<input type="checkbox"/> TASK NUMBER
APPROVING OFFICIAL	ORG CODE	E-MAIL	DATE
I HEREBY CHANGE THE BLANKET AVAILABILITY AUTHORIZATION DATED (mm/dd/yyyy): _____ <input type="checkbox"/> RESCINDED - FUTURE DOCUMENTS MUST HAVE INDIVIDUAL DOCUMENT AVAILABILITY AUTHORIZATIONS <input type="checkbox"/> MODIFIED - LIMITATIONS ON ALL DOCUMENTS SHOULD BE CHANGED TO CONFORM TO BLOCKS AS CHECKED IN SECTIONS 2 AND 3			
NAME	ORG CODE	E-MAIL	DATE
6. AUTHOR/ORIGINATOR VERIFICATION			
I BELIEVE THAT THIS PUBLICATION: <input type="checkbox"/> DOES CONTAIN ITAR/EAR/EXPORT-CONTROLLED, PROPRIETARY/SENSITIVE INFORMATION, AND/OR DISCLOSES AN INVENTION AND THE APPROPRIATE LIMITATION IS CHECKED IN SECTIONS 3 AND/OR 4 <input checked="" type="checkbox"/> DOES NOT CONTAIN ITAR/EAR/EXPORT-CONTROLLED, PROPRIETARY/SENSITIVE INFORMATION, NOR DOES IT DISCLOSE AN INVENTION AND MAY BE RELEASED AS INDICATED ABOVE			
NAME	ORG CODE	E-MAIL	DATE
William J. Morphet	1.53434	Bill.Morphet2@ATK.com	06/20/2008
7. NASA PROJECT OFFICIAL (DDTR/designated official, or Technical Monitor) / DIVISION CHIEF REVIEW OF 1 THROUGH 6			
<input checked="" type="checkbox"/> APPROVED FOR DISTRIBUTION AS MARKED <input type="checkbox"/> NOT APPROVED (Give reason)		REASON:	
NAME	TITLE	E-MAIL	DATE
Robert D. Woods	Supervisory Aerospace Engineer	Robert.D.Woods@nasa.gov	7/25/08
8. EXPORT CONTROL ADMINISTRATOR CONFIRMATION			
<input checked="" type="checkbox"/> PUBLIC RELEASE IS APPROVED (Information is not export controlled.) <input type="checkbox"/> EXPORT-CONTROLLED LIMITATIONS IN BLOCKS 3b, 3d, AND 6 ARE APPROVED AND ASSIGNED TO THIS DOCUMENT		<input type="checkbox"/> EXPORT-CONTROLLED LIMITATION(S) INDICATED IN SECTIONS 3b AND 3d HAVE BEEN CHANGED TO: ITAR USML CATEGORY: _____ NUMBER: _____ EAR CCL ECCN NUMBER: _____	
CENTER EXPORT CONTROL ADMINISTRATOR NAME	E-MAIL	DATE	
John M. Pea	john.m.pea@nasa.gov	7/25/08	
9. FINAL APPROVAL BY PROGRAM MANAGER			
<input checked="" type="checkbox"/> APPROVED FOR DISTRIBUTION AS MARKED <input type="checkbox"/> NOT APPROVED (Give reason)		REASON:	
DATE DOCUMENT PUBLISHED/RELEASED: _____			
NAME	E-MAIL	DATE	
Steven F. Cole	STEVEN.F.COLE@NASA.GOV	7/25/08	
10. DISPOSITION			
ADDITIONAL APPROVALS BY CENTER IF APPLICABLE (OPTIONAL):			
NAME/TITLE	E-MAIL	DATE	
Angela D. Storey	Angela.D.Storey@NASA.gov	30 July 08	
NAME/TITLE	E-MAIL	DATE	
SEND TO CENTER DAA REPRESENTATIVE (SEE INSTRUCTIONS)			
<input type="checkbox"/> SEND TO CENTER TECHNICAL PUBLICATIONS OFFICE		<input type="checkbox"/> SEND TO NASA CENTER FOR AEROSPACE INFORMATION (e-mail: help@af.nasa.gov)	
NAME OF DAA REPRESENTATIVE	DATE		
[Signature]	7/29/08		

National Aeronautics and
Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812



Reply to Attn of: IS20

July 31, 2008

ATK
Launch Systems Group
P.O. Box 707
Brigham City, UT 84302

Attn: Richard Roth

Dear Mr. Roth:

Re: E670-CY08-031

This is in response to your request of June 23, 2008 for review and approval of sections in chapter 2 of a dissertation entitled "Simulation, Kriging and Visualization of Circular-Spatial Data.

The technical presentation is publicly available as indicated on the attached NASA Form 1676.

Sincerely,

A handwritten signature in black ink, appearing to read "Kim Narmore".

Kim Narmore
MSFC Document Availability (DAA) Representative

Enclosure

The following is email confirming approval of Technical Information Clearance.

From: MSFC-STI
Sent: Friday, August 01, 2008 12:40 PM
To: 'Bill.Morphet@ATK.com'
Cc: 'Burnette, Sheila'; Pea, John M. (MSFC-MP02)
Subject: Approved NF 1676

Mr. Morphet,

The dissertation entitled "Simulation, Kriging, and Visualization of Circular-Spatial Data" has been approved for release and distribution as indicated by the attached NF 1676. The original form will be returned to you for record retention.

As required by NASA policy, your document will be forwarded to the Center for Aerospace Information (CASI) for inclusion in the NASA Aeronautics and Space Database (<http://www.sti.nasa.gov>).

Thank you,

Kim Narmore
NASA/Marshall Space Flight Center
Document Availability Authorization (DAA) Representative
IS20, Bldg. 4200, Rm. 500
MSFC, AL 35812
(256) 544-4512

By 'The dissertation entitled "Simulation, Kriging, and Visualization of Circular-Spatial Data" ' Mr. Narmore is referring to Subsections 2.8.2 and 2.8.3 of Chapter 2, which was submitted for Technical Information Clearance.

The following authorization applies to Figure 2-19.



ATK THIOKOL

Memorandum

Date	May 18, 2005	From	Olga B. Siggins <i>Olga B. Siggins</i>
Subject	Permission to Use Copyrighted Artwork	Organization	Office of General Counsel
To	Bill Morphet	MS	UT03-EL2
		Telephone	801-251 2205

This memo confirms that you have permission to use ATK Thiokol Inc.'s copyright protected artwork that shows the roll maneuver of the Space Shuttle system in connection with the paper you have authored for The Journal of Computational and Graphical Statistics.

CURRICULUM VITAE

William James Morphet
(June 2009)

CAREER OBJECTIVE:

To develop technical expertise, involving creativity, leadership and proactive behavior. Special areas of interest: statistics - bringing out the message in the data in graphical display.

EDUCATION:

PhD in Statistics, Utah State University, Logan, Utah. (2009) GPA: 4.0 (4.0=A).

Master of Science in Statistics, Utah State University. (12/14/2001) GPA: 4.0.

Bachelor of Science in Manufacturing Engineering, Brigham Young University, Provo, Utah. (4/21/78) GPA: 3.78 Magna Cum Laude.

EXPERIENCE:

12/07+	ATK Launch Systems, Nozzle Structural Analysis (material properties and general statistical support)
5/97 – 12/07	ATK Launch Systems, Post Launch/Test Evaluation Engineering (anomaly review process)
12/92 - 5/97	ATK Launch Systems, Drafting (drafting, geometric dimensioning & tolerancing, and simulation of assembly tolerance stack up)
3/92 - 12/92	ATK Launch Systems, Computer Aided Engineering & Trending (math, statistics, and nondestructive engineering support)
5/81 - 3/92	ATK Launch Systems, Materials & Processes (numerical control programming, manufacturing drawings)
5/78 - 5/81	Lockheed Missiles & Space Co. (computer aided manufacturing applications)
66 - 72	United States Navy (nuclear reactor operator, electronics, health physics)

PUBLISHED PAPERS/AWARDS:

- | | |
|------------|--|
| 2/08 | Pinnacle National Honor Society for Adult and other nontraditional students |
| 11/07 | Gold Medal International Honour Society, nominated to Phi Kappa Phi |
| 4/04 | “Space Shuttle SRM <Solid Rocket Motor> Nozzle Erosion & Char Statistical Assessment,” Joint Army Navy NASA Air Force (JANNAF) 23 rd Rocket Nozzle Technology Subcommittee, 2004, New Orleans, Louisiana. |
| 12/01 | “Relationship Between Nozzle Erosion and Vectoring for the Space Shuttle Reusable Solid Rocket Motor,” 33 rd International Society for the Advancement of Material and Process Engineering (SAMPE) Technical Conference |
| 5/01 | Outstanding Student Graduate |
| 4/01 | Phi Kappa Phi |
| 8/00 | “Vectoring Effects on Nozzle Erosion,” TWR-66980, ATK Space Launch Systems |
| 8/93 | Monetary award from NASA Inventions and Contributions Board for New Technology Evaluation-MFS 28886, “Flat Pattern Development for Aiming X-Ray Machine Which Is Equipped with a Laser Spotting Beam That Is Aligned with the X-Ray Beam Centerline” |
| 7/92 | “Effect of Surface Curvature on OSEE [Optically Stimulated Electron Emission],” TER-92033, ATK Space Launch Systems |
| 81-Present | 30+ employer recognitions including monetary awards |
| 78 | Tau Beta Pi |
| 72 | United States Navy, Honorable Discharge |