SPATIAL POISSON MODELS OF STATIONARY STORM RAINFALL: PARAMETERIZATION, EVALUATION AND NUMERICAL SIMULATION

by
NEIL M. FENNESSEY
WANG QINLIANG
PETER S. EAGLESON
and
IGNACIO RODRIGUEZ-ITURBE

RALPH M. PARSONS LABORATORY
HYDROLOGY AND WATER RESOURCE SYSTEMS

Report No. 308

Prepared under the support of

The National Science Foundation
Grant No. ATM 8420781

August 1986

MIT

DEPARTMENT
OF
CIVIL
ENGINEERING

SCHOOL OF ENGINEERING
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, Massachusetts 02139

R86-20

SPATIAL-POISSON MODELS OF STATIONARY STORM RAINFALL: PARAMETERIZATION, EVALUATION AND NUMERICAL SIMULATION

by

Neil M. Fennessey

Wang Qinliang

Peter S. Eagleson

and

Ignacio Rodriguez-Iturbe

RALPH M. PARSONS LABORATORY
HYDROLOGY AND WATER RESOURCE SYSTEMS

Department of Civil Engineering
Massachusetts Institute of Technology

Report No. 308

Prepared with the support of the

National Science Foundation
Grant No. ATM 8420781

AUGUST, 1986

PREFACE

This report is one of four in a sequence describing and evaluating the modeling of the total storm rainfall due to stationary events. These reports are:

Report No. 305:	"Spatial Poisson Models of Stationary Storm Rainfall: Theoretical Development" by I. Rodriguez-Iturbe, Q. Wang, P. S. Eagleson and B. L. Jacobs.
Report No. 306:	"Spatial Analysis of Storm Depths from an Arizona Raingage Network" by N. M. Fennessey, P. S. Eagleson, Q. Wang, and I. Rodriguez-Iturbe.
Report No. 307: (Vols. 1 and 2)	"Spatial Characteristics of Observed Precipitation Fields: A Catalog of Summer Storms in Arizona" by N. M. Fennessey, P. S. Eagleson, Q. Wang and I. Rodriguez-Iturbe.
Report No. 308:	"Spatial Poisson Models of Stationary Storm Rainfall: Parameterization, Evaluation and Numerical Simulation" by N. M. Fennessey, Q. Wang, P. S. Eagleson and I. Rodriguez-Iturbe.

They are all available from

Director
Ralph M. Parsons Laboratory
Room 48-311
Massachusetts Institute of Technology
Cambridge, MA 02139

The raw data were provided by the Agricultural Research Service (U.S. Department of Agriculture), and are available from their data center in Beltsville, Maryland.

Data tables for the 428 identified storms are available on computer tape from the above MIT address, along with a computer program for retrieving the data for a particular storm.

ABSTRACT

Eight years of summer raingage observations are analyzed for a dense, 93 raingage, network operated by the U.S. Department of Agriculture, Agricultural Research Service, in the 150 km^2 Walnut Gulch catchment near Tucson, Arizona. Storms are defined by the total depths collected at each raingage during the noon-to-noon period for which there was depth recorded at any of the gages. For each of the resulting 428 storm days, the 93 gage depths are interpolated onto a dense grid and the resulting random field is analyzed to obtain storm depth isohyets, the first three moments of point storm depth, the spatial correlation function, the spatial variance function, and the spatial distribution of total rainstorm depth. The sample is split and half is used to estimate for each storm day the three parameters of each of three conceptual spatial Poisson process models proposed elsewhere [Rodriguez-Iturbe, Cox and Eagleson, 1986]. The distributions of these parameters are estimated and used to evaluate the absolute and relative worth of the three Poisson models in comparison with the second half of the sample.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. ATM-8420781.

The work was performed by Mr. Neil M. Fennessey, Research Assistant in Civil Engineering, and this document is a part of his thesis submitted to the Massachusetts Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Civil Engineering. The work was supervised by Dr. Peter S. Eagleson, Edmund K. Turner Professor of Civil Engineering. Mr. Wang Qinliang, Visiting Engineer, provided continuing assistance while on leave from the Yangtze Valley Planning Office, Ministry of Water Resources and Electric Power, People's Republic of China. Dr. Ignacio Rodriguez-Iturbe, Graduate Program in Hydrology and Water Resources, Universidad Simon Bolivar, Caracas, Venezuela, provided guidance on a consulting basis. Mr. Bruce L. Jacobs, Research Assistant in Civil Engineering, gave valuable assistance.

Special thanks are extended to Dr. David Woolhiser and Ms. Fatima Lopez of the U.S.D.A. Southwest Rangeland Watershed Research Center in Tucson, Arizona for generously providing the data used in this study.

TABLE OF CONTENTS

Page No.

PREFACE i

ABSTRACT ii

ACKNOWLEDGMENT iii

TABLE OF CONTENTS iv

LIST OF FIGURES vi

LIST OF TABLES viii

LIST OF NOTATION ix

CHAPTER 1 Introduction
- 1.1 Background 1
- 1.2 Goals and Objectives 1

CHAPTER 2 Review of the Analytical Models
- 2.1 Overview 2
- 2.2 Theoretical Background 2
- 2.3 Model I 8
- 2.4 Model II 10
- 2.5 Model III 14

CHAPTER 3 Model Parameter Estimation
- 3.1 Introduction 16
- 3.2 Estimation of Model Parameters 17
- 3.3 Bias of Estimators 20

CHAPTER 4 Distribution of Model Parameters
- 4.1 Distribution of Model Parameters 24

CHAPTER 5 Numerical Simulation of the Spatial Poisson Models
- 5.1 Simulation of Total Storm Depth 39
- 5.2 The Spatial Poisson Process 39
  - 5.2.1 Theory 39
  - 5.2.2 Simulation 40
- 5.3 Numerical Simulation of Total Rainfall Depth 42
- 5.4 Storm Day Total Rainfall Depth Simulation and Sampling from the Random Field 46
  - 5.4.1 Simulation and Sampling from the Fine Mesh Simulated Random Field 47
  - 5.4.2 Sampling from the Coarse Mesh Simulated Random Field 49
  - 5.4.3 Discussion of Simulation Results 50
- 5.5 Simulation Post-Processing 63

Page No.

CHAPTER 6 Model Evaluation
6.1 Goodness of Fit 66
6.2 Variation Among Storms 72
6.3 Model Evaluation 76

CHAPTER 7 Assessment of the Surface Fitting Interpolator
7.1 An Assessment of the Bivariate Surface Fitting Interpolator 84

CHAPTER 8 Summary and Conclusions
8.1 Summary 90
8.2 Conclusions 90

CHAPTER 9 The Computer Codes
9.1 Introduction 93
9.2 CORRFIT.FORTRAN 93
9.3 UNBIAS.FORTRAN 94
9.4 TIMEDIST.FORTRAN 94
9.5 PARAM.FORTRAN 94
9.6 RADIUS.FORTRAN and POLAR.FORTRAN 95
9.7 SIMWET.FORTRAN 95
9.8 SIMCRGAM.FORTRAN 95
9.9 TABLE5.FORTRAN 96
9.10 SIMPLOT.FORTRAN 96
9.11 SIM_STORMDAY.FORTRAN 96
9.12 WETPLOT.FORTRAN 97
9.13 CORRGAM_PLOT.FORTRAN 97
9.14 SIMINTRP.FORTRAN 97
9.15 SIMCORR.FORTRAN 98
9.16 Computer Tape 99

REFERENCES 103

APPENDIX I MODEL_PARAMETERS.DATA 104

APPENDIX II UNBIASED_MODEL_PARAMETERS.DATA 120

APPENDIX III 136
CORRFIT.FORTRAN 137
UNBIAS.FORTRAN 147
TIMEDIST.FORTRAN 152
PARAM.FORTRAN 179
RADIUS.FORTRAN 182
POLAR.FORTRAN 184
SIMWET.FORTRAN 186
SIMCRGAM.FORTRAN 210
TABLE5.FORTRAN 229
SIMPLOT.FORTRAN 235
SIM STORMDAY.FORTRAN 245
WETPLOT.FORTRAN 257
CORR_GAM_PLOT.FORTRAN 267
SIMINTRP.FORTRAN 279
SIMCORR.FORTRAN 293
OBS_EVAL.FORTRAN 309
MODEL_EVAL.FORTRAN 312

List of Figures

Figure	Title	Page
2.2.1	Conceptual Model of Total Rainstorm Depth	4
2.2.2	Conceptual Model Spread Functions	7
2.4.2	Characteristic Function	13
4.1.1	Distribution of a_1	30
4.1.2	Distribution of a_2	31
4.1.3	Distribution of a_3	32
4.1.4	Distribution of λ_1	33
4.1.5	Distribution of λ_2	34
4.1.6	Distribution of λ_3	35
4.1.7	Distribution of $E[\alpha_1]$	36
4.1.8	Distribution of $E[\alpha_2]$	37
4.1.9	Distribution of $E[\alpha_3]$	38
5.3.1	Maximum Simulation Distance	43
5.4.1	Observed 22 June 1970 Storm Day	51
5.4.2 Model 1	Simulation of 22 June 1970 Storm Day; $a_1 = 0.566$ km^{-1}, $\lambda_1 = 0.019$ km^{-2}, $E[\alpha_1] = 2.860$ mm	52
5.4.3	22 June 1970 Theoretical, Observed and Simulated Spatial Correlation	53
5.4.4	22 June 1970 Theoretical, Observed and Simulated Variance Functions	55
5.4.5	22 June 1970 Theoretical, Observed and Simulated Spatial Distribution of Total Storm Depth	56
5.4.6	Observed 24 July 1970 Storm Day	57
5.4.7 Model 1	Simulation of 24 July 1970 Storm Day; $a_1 = 0.275$ km^{-1}, $\lambda_1 = 0.063$ km^{-2}, $E[\alpha_1] = 6.128$ mm	58

List of Figures

(Continued)

Figure	Title	Page
5.4.8	24 July 1970 Theoretical, Observed and Simulated Spatial Correlation	59
5.4.9	24 July 1970 Theoretical, Observed and Simulated Variance Functions	61
5.4.10	24 July 1970 Theoretical, Observed and Simulated Spatial Distribution of Total Storm Depth	62
6.1.1	Model I Split Sample Evaluation	69
6.1.2	Model II Split Sample Evaluation	70
6.1.3	Model III Split Sample Evaluation	71
6.2.1	Effect of Interstorm Variability on the Spatial Correlation Function: Model 1	75
6.3.1	Spatial Distribution of Total Storm Depth: Evaluation of Models 1 and 2	77
6.3.2	Spatial Distribution of Total Storm Depth: Evaluation of Model 3	79
7.1.1	Simulated 24 July 1970 Storm Day	85
7.1.2	Interpolation of the Simulated 24 July 1970 Storm Day	86
7.1.3	Simulated 22 June 1970 Storm Day	87
7.1.4	Interpolation of the Simulated 22 June 1970 Storm Day	88

List of Tables

Table	Title	Page
4.1.1	Moments of Estimated Parameters	26
4.1.2	Parameter Variability and Correlation	27
4.1.3	Moments of Point Storm Depth	29
6.1.1	Split Sample Goodness of Fit Assessment	72
6.3.1	Numerical Evaluation of Models	82
6.3.2	Ensemble Average Parameters: Model 1	83
7.1.1	Comparison of "Observed" and Interpolated Rainfalls	89

Notation

A	area, km^2
A_c	catchment area, km^2
A_{cd}	dry area of catchment, km^2
A_{cw}	wetted area of catchment
A_{cd}/A_c	dry fraction of catchment area
A_{cw}/A_c	wetted fraction of catchment area
A_s	storm area, km^2
A_{sc}	storm-catchment overlap area, km^2
C.S.	skewness coefficient
$F_Y(y)$	cumulative probability function of $Y(\underline{x})$
$I_i(\ )$	ith order modified Bessel function of the first kind
$K_i(\ )$	ith order modified Bessel function of the second kind.
L	length, km
$L_i(\ )$	ith order Struve function
R_2	2-dimensional plane
R_k	spatial Poisson "nearest neighbor" simulation distance, km
R_L	simulation distance, km
R_o	maximum simulation distance, km
Y	point storm depth, mm
Y_A	average storm depth over area A, mm.
Y_k	finite element storm depth, mm

Notation

(Continued)

Y_o	station observation storm depth, mm
a	storm cell spread function parameter, km^{-1} or mm/km
erf( )	the Error function
f(a)	standard square error function of parameter a
$f(\alpha)$	probability density function of storm cell parameter α
$f_{R_k}(r)$	probability density function of the spatial Poisson "nearest neighbor distances"
$f_Y(y)$	probability density function of $Y(\underline{x})$
$g(\underline{x},\underline{z},\alpha)$	storm cell depth spread function
nR	number of simulation storm cells
q	coefficient of polynomial bivariate surface interpolator
r	radial distance, km
u	complex variable
$\underline{x}$	2-dimensional position vector
$\bar{x}$	expected value of the node depth, mm
$\bar{\bar{x}}$	expected value of the variance function element depth, mm
y	point storm depth, mm
$\underline{z}$	2-dimensional position vector
α	storm cell center depth parameter, mm
$\hat{\alpha}$	maximum simulation distance parameter
β	exponential distribution parameter = μ_α^{-1}, mm^{-1}

Notation

(Continued)

$\gamma[A]$	variance function of the average of $Y(\underline{x})$ over area A
$\delta(\ )$	Dirac delta function
δ	Gamma distribution parameter
ε	exponential distribution parameter
θ	Gamma distribution parameter
κ	Gamma distribution parameter
λ	storm cell density parameter, km^{-2}
μ_α	expected value of α, mm
ν	lag distance, km
$\rho(\nu)$	spatial correlation coefficient at lag ν
σ_A	standard deviation of $Y(\underline{x})$ averaged over area A, mm
σ_k	standard deviation of the finite element storm depth, mm
σ_Y	standard deviation of $Y(\underline{x})$
$\phi_Y(\ )$	characteristic function
χ	Gamma distribution parameter

CHAPTER 1

Introduction

1.1 Background

Current atmospheric general circulation models (GCMs) use complex numerical techniques to solve the hydrodynamic and thermodynamic equations for the atmosphere, but they generally treat the landsurface moisture boundary condition in a rather simplistic way. In order to improve upon existing GCMs, a reasonable first step is to parameterize the subgrid spatial distribution of rainfall.

Recently, Rodriguez-Iturbe et al. [1986-a] derived a conceptual spatial Poisson model of total rainstorm depth. This work constitutes the theoretical background of the present study. This study focuses on the analysis of the spatial characteristics of total storm rainfall using observed station data and on the parameterization and evaluation of the existing conceptual models. In addition, numerical simulation of these analytical models is conducted and analyzed.

1.2 Goals and Objectives

The general objective of this study is to perform a probabilistic analysis of the spatial random field of total rainfall depth resulting from air-mass thunderstorms with a view toward developing general descriptions of the spatial characteristics of such storms.

The specific objective of this work is to estimate the parameters of three existing spatial point process models of storm rainfall and to make absolute and comparative evaluations of the models' performance.

CHAPTER 2

Review of the Analytical Models

2.1 Overview

Rodriguez-Iturbe et al. [1986-a] derived a probabilistic model of total rainstorm depth. The model describes the spatial distribution of storm depth in the two-dimensional plane. It is a conceptual model which assumes that total storm depth at any point on the plane results from the superimposed contributions from a finite number of spatially stationary raincells distributed in space according to a Poisson spatial process. There are three variations on the model according to the choice of an assumed "spread function". The spread function is an axially-symmetric surface which describes the spatial distribution of total rainfall depth resulting from a single storm cell. This contribution from a single cell is assumed to have its greatest value at the center of the cell, and to decrease with radial distance away from the cell's center.

The following portions of this chapter are devoted to a brief review of the analytical models. The reader is referred to Rodriguez-Iturbe et al. [1986-a, 1986-b] for the precise details.

2.2 Theoretical background

The conceptual model [Rodriguez-Iturbe et al., 1986-b] represents the total rainstorm depth as the sum of separate precipitations contributed by a finite number of stationary storm cells as shown in Fig. 2.2.1. The precipitation from each cell has a maximum value, α, (a random variable) at the center of the cell located at $\underline{z}$. The rainfall depth surface of each cell decays with increased radial distance away from the cell's center. The rainfall depth at position $\underline{x}$ from a single cell located at $\underline{z}$ is defined by one of three spread functions $g(\underline{x},\underline{z},\alpha)$ where

$g(\underline{x},\underline{z},\alpha)$ = precipitation at $\underline{x}$ due to a single cell centered at $\underline{z}$

$f(\alpha)$ = the probability density function of the cell center depth α.

The cell centers are assumed to be distributed over the entire two-dimensional space R_2 according to a two-dimensional spatial Poisson process where the cell centers have a spatial density of $\lambda(\text{area}^{-1})$. The contributions of total rainfall depth Y at point $\underline{x}$ from all the cells in R_2 is

$$Y(\underline{x}) = \lambda \int_{R_2} g(\underline{x},\underline{z},\alpha)\, d\underline{z} \tag{2.2.1}$$

The random field $Y(\underline{x})$ may be described by

$$E(Y) = \lambda \int_{R_2} \left[\int_0^\infty g(\underline{x},\underline{z},\alpha) f(\alpha) d\alpha \right] d\underline{z}$$

$$= \lambda \int_{R_2} E(g(\underline{x},\underline{z},\alpha))\, d\underline{z} \tag{2.2.2}$$

$$VAR(Y) = \sigma_y^2$$

$$= \lambda \int_{R_2} E_\alpha[g^2(\underline{x},\underline{z},\alpha)] d\underline{z} \tag{2.2.3}$$

The coefficient of skewness, C.S. is

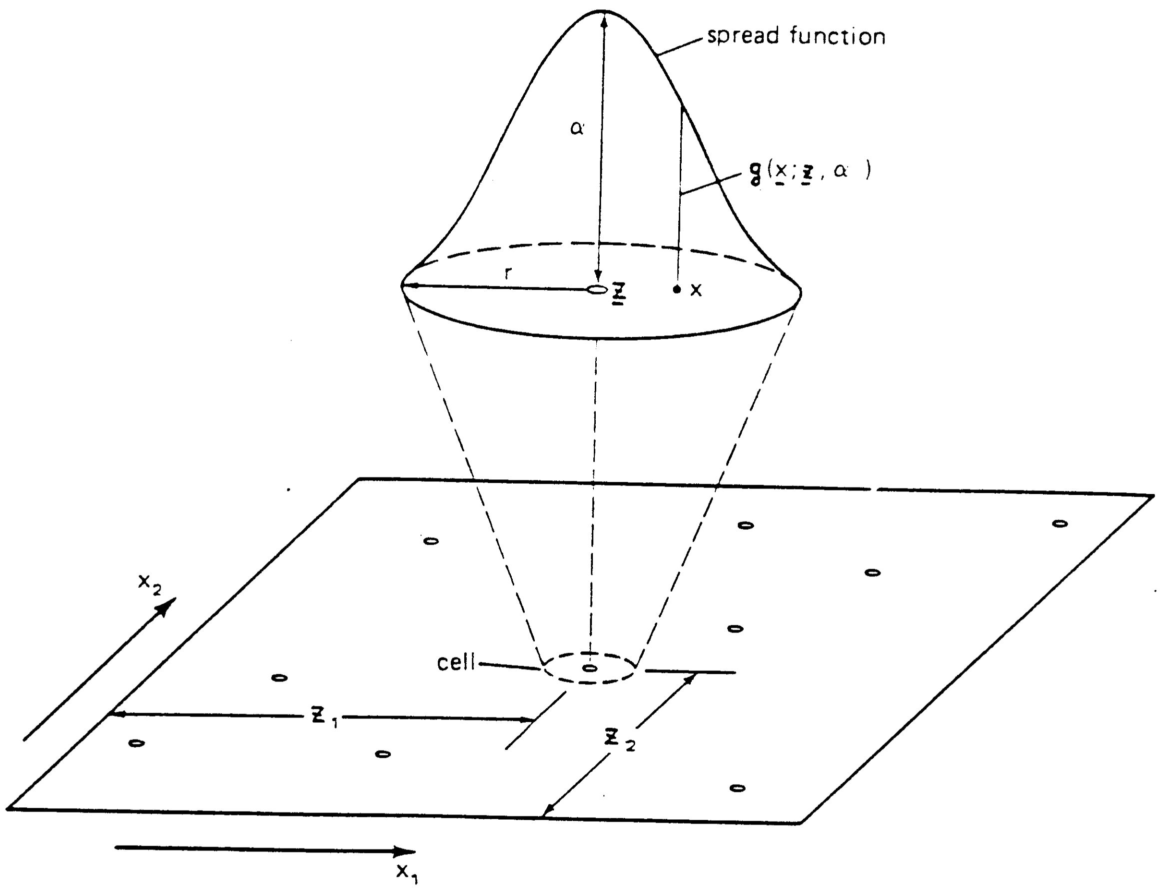


Fig. 2.2.1 Conceptual Model of Total Rainstorm Depth

$$C.S. = \frac{\int_{R_2} E_\alpha[g\ (\underline{x},\underline{z},\alpha)]\ d\underline{z}}{\lambda^{1/2}\{\int_{R_2} E_\alpha[g^2(\underline{x},\underline{z},\alpha)]\ d\underline{z}\}^{3/2}} \qquad (2.2.4)$$

The Covariance is

$$COV[Y(\underline{x}_1),Y(\underline{x}_2)] = \lambda \int_{R_2} E_\alpha[g(\underline{x}_1,\underline{z},\alpha)g(\underline{x}_2,\underline{z},\alpha)]\ d\underline{z} \qquad (2.2.5)$$

The variance function is

$$\gamma(A) = \frac{\sigma_A^2}{\sigma^2} \qquad (2.2.6)$$

where σ^2 is the variance of $Y(\underline{x})$ and σ_A^2 is the variance of the areal average of $Y(\underline{x})$ taken over area A.

The spatial distribution of $Y(\underline{x})$ is described by

$$F_Y(y) = \int_o^y f_Y(y)\ dy \qquad (2.2.7)$$

where $f_Y(y)$ is the probablilty density function of $Y(\underline{x})$, and $F_Y(y)$ is the cumulative probability function of $Y(\underline{x})$, which represents the fraction of R_2 covered by a depth of total rainfall $Y \leq y$.

The pdf $f_Y(y)$ may be obtained by inverting the characteristic function $\phi_Y(u)$ which is described by

$$\phi_Y(u) = E(e^{iuY}) \qquad (2.2.8)$$

$$E(Y) = \exp[\lambda \int_{R_2} (\int_o^\infty e^{iug(\underline{x},\underline{z},\alpha)} f(\alpha)d\alpha - 1)\ d\underline{z}] \qquad (2.2.8)$$

The probablility density function $f(\alpha)$ describes the distribution of the randomly variable cell center depth, α. This distribution is assumed to be exponential such that

$$f(\alpha) = \beta e^{-\beta\alpha}$$

where

$$\beta = E^{-1}(\alpha) \equiv \mu_\alpha^{-1} \quad (2.2.9)$$

Three different conceptual models representing the total rainfall depth from a single storm cell are developed. These models are described by the spread function $g_i(\underline{x},\underline{z},\alpha_i)$. An illustration of the total depth surface for a single raincell is shown in Fig. 2.2.2 for each of the three spread functions.

Model I is referred to as the Quadratic Exponential Spread Function where

$$g_1(\underline{x},\underline{z},\alpha_1) = \alpha_1 \exp(-2a_1^2 r^2) \quad (2.2.10)$$

Model II is referred to as the Simple Exponential Spread Function where

$$g_2(\underline{x},\underline{z},\alpha_2) = \alpha_2 \exp(-a_2 r) \quad (2.2.11)$$

Model III is referred to as the Linear Spread Function where

$$g_3(\underline{x},\underline{z},\alpha_3) = \begin{cases} \alpha_3(1 - a_3 r), & 0 < r < \alpha_3/a_3 \\ 0 \quad , & \text{otherwise.} \end{cases} \quad (2.2.12)$$

In each model $r = |\underline{x} - \underline{z}|$.

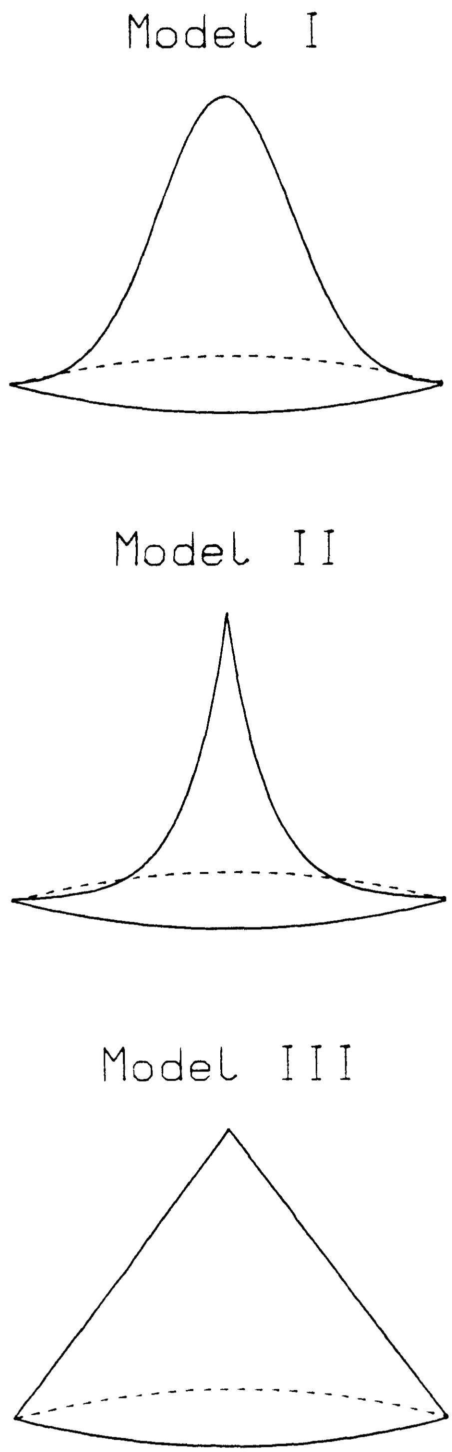


Fig. 2.2.2 Conceptual Model Spread Functions

2.3 Model I: The Quadratic Spread Function Model

For the Model I spread function, the random field ($\underline{x}$) is described by the following series of equations.

The characteristic function $\phi_Y(u)$

$$\phi_{Y_1}(u) = \left(\frac{\beta_1}{\beta_1 - iu}\right)^{\frac{\pi\lambda_1}{2\alpha_1^2}} \qquad (2.3.1)$$

where

u = a complex variable

$\beta_1 = E^{-1}(\alpha_1)$ = Model I depth parameter (length^{-1})

a_1 = Model I spread function parameter (length^{-1})

λ_1 = Model I raincell density parameter (area^{-1})

$\phi_{Y_1}(u)$ can be recognized as the characteristic function of the two parameter Gamma distribution $f_Y(y)$ [Vanmarcke, 1983, p. 47] where

$$f_Y(y) = \frac{\beta_1(\beta_1 y)^{\theta_1 - 1} e^{-\beta_1 y}}{\Gamma(\theta_1)} \qquad (2.3.2)$$

and

$$\theta_1 = \frac{\pi\lambda_1}{2a_1^2}$$

The expected value of $Y(\underline{x})$, $E(Y)$, is given by

$$E(Y) = \theta_1 \mu_{\alpha_1} \qquad (2.3.3)$$

The variance of $Y(\underline{x})$, $VAR(Y)$, is given by

$$VAR(Y) = \theta_1 \mu_{\alpha_1}^2 \tag{2.3.4}$$

The coefficient of skewness, C.S., is described by

$$C.S. = 2a_1 \left[\frac{2}{\pi\lambda_1}\right]^{1/2} \tag{2.3.5}$$

The covariance of $Y(\underline{x})$ for an isotropic field and for exponential $f(\alpha)$, is

$$COV(\nu) = \theta_1 \mu_{\alpha_1}^2 \, e^{-\alpha_1^2 \nu^2} \tag{2.3.6}$$

where

$$\nu = |\underline{x}_1 - \underline{x}_2|$$

The spatial correlation function, $\rho(\nu)$, is described by

$$\rho(\nu) = \frac{COV(\nu)}{\sigma_Y^2} = e^{-a_1^2 \nu^2} \tag{2.3.7}$$

The variance function, $\gamma(A)$, is given for a square area by

$$\gamma(A) = \left[\frac{1}{(a_1 L)^2} \left[a_1 L \pi^{1/2} \operatorname{erf}(aL) + e^{-a_1^2 L^2} - 1\right]\right]^2 \tag{2.3.8}$$

where

$$\operatorname{erf}(\ ) = \text{the Error function}$$

and

$$L = A^{1/2} \quad \text{(length)}$$

2.4 Model II: The Simple Exponential Spread Function Model

For the Model II spread function, the total depth random field Y(x) is described by the following series of equations.

The characteristic function is given by

$$\phi_{Y_2}(u) = \exp\left[\frac{2\pi\lambda_2}{a_2^2} \sum_{k=1}^{\infty} \frac{\left(\frac{iu}{\beta_2}\right)^k}{k^2}\right] \tag{2.4.1}$$

where

u = a complex variable

$\beta_2 = E^{-1}[\alpha_2]$ = Model II depth parameter (length^{-1})

a_2 = Model II spread function parameter (length^{-1})

λ_2 = Model II raincell density parameter (area^{-1})

k = a counting variable

The expected value of Y(x) is given by

$$E(Y) = 4\theta_2\mu_{\alpha_2} \tag{2.4.2a}$$

where

$$\theta_2 = \pi\lambda_2/(2\alpha_2^2) \tag{2.4.2b}$$

The variance of Y(x) is described by

$$VAR(Y) = 2\theta_2\mu_{\alpha_2}^2 \tag{2.4.3}$$

The coefficient of skewness of Y(x) is given by

$$C.S. = \frac{4a_2}{3[\pi\lambda_2]^{1/2}} \tag{2.4.4}$$

The covariance of $Y(\underline{x})$ for an isotropic field is given by

$$COV(\nu) = \pi\lambda_2 E^2[\alpha_2][\frac{\nu}{\alpha_2} K_1(a_2\nu) + \frac{\nu^2}{2} K_o(a_2\nu)] \tag{2.4.5}$$

The spatial correlation of $Y(\underline{x})$ is described by

$$\rho(\nu) = a_2\nu K_1(a_2\nu) + \frac{a_2^2\nu^2}{2} K_o(a_2\nu) \tag{2.4.6}$$

where

$$\nu = |\underline{x}_1 - \underline{x}_2|$$

$K_0(\ )$ = Zeroth order Bessel Function of the Second Kind

$K_1(\ )$ = First Order Bessel Function of the Second Kind

The variance function is given by

$$\gamma(A) = \{ \frac{3\pi}{2} [K_1(a_2A^{1/2})L_o(a_2A^{1/2}) + K_o(a_2A^{1/2})L_1(a_2A^{1/2})] + $$
$$+ \frac{8}{a_2A^{1/2}} K_1(a_2A^{1/2}) + 4K_o(a_2A^{1/2}) - \frac{8}{a_2^2A} \}^2 \tag{2.4.7}$$

where

$L_0(z)$ = Zeroth order Struve function

$$= \frac{z}{2} \sum_{k=0}^{\infty} \frac{(-1)^k(\frac{z}{2})^{2k}}{\Gamma^2(k + \frac{3}{2})} \tag{2.4.8}$$

$L_1(z)$ = First order Struve function

$$= \left(\frac{z}{2}\right)^2 \sum_{k=0}^{\infty} \frac{(-1)^k \left(\frac{z}{2}\right)^{2k}}{\Gamma\left(\frac{3}{2}+k\right)\ \Gamma\left(\frac{5}{2}+k\right)} \qquad (2.4.9)$$

The characteristic function (Eq. 2.4.1) is too complicated to invert analytically but may be approximated by

$$\phi_{Y_2}(u) \simeq \exp\left[4\theta_2 \sum_{k=1}^{\infty} \frac{\left(\frac{iu}{\delta_2}\right)^k}{k}\right] \qquad (2.4.10)$$

which is again the characteristic function of the two parameter Gamma distribution.

The real parts of the analytical (Eq. 2.4.1) and approximating (Eq. 2.4.10) characteristic functions are plotted in Fig. 2.4.2. From this approximation, the spatial distribution of total rainfall depth will be approximated by

$$f_Y(y) \simeq \frac{2\beta_2 (2\beta_2 y)^{8\theta_2 - 1}}{\Gamma(8\theta_2)} e^{-2\beta_2 y} \qquad (2.4.11)$$

where

$$\beta_2 = \mu_{\alpha_2}^{-1}$$

The difference between the exact and approximate characteristic functions corresponds [see Rodriguez-Iturbe et al., 1986-a, p.31] to an exact distribution $f_Y(y)$ that is somewhat "fatter" in the tail than the gamma approximation.

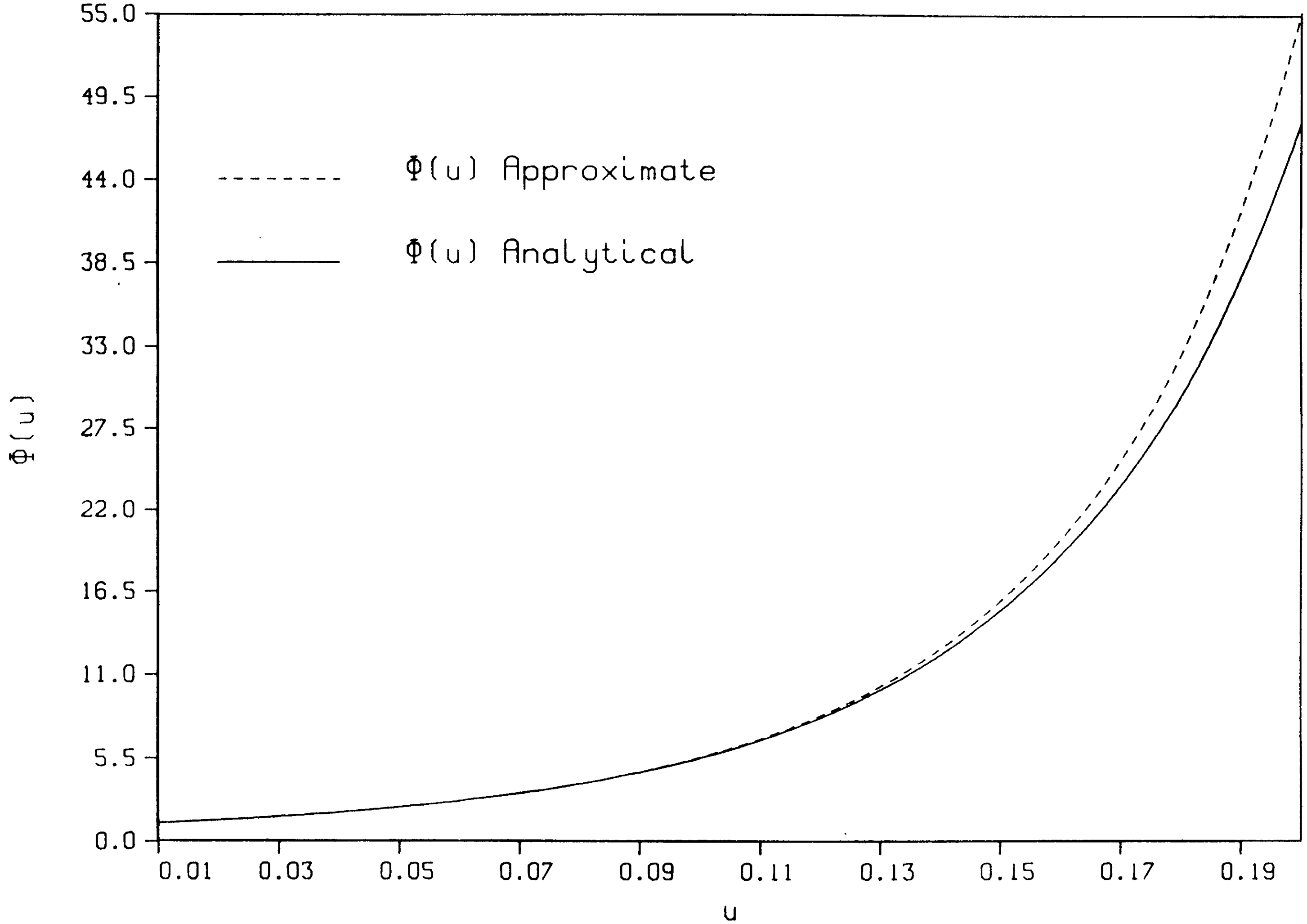


Fig. 2.4.2 Characteristic Function

2.5 Model III: The Linear Spread Function Model

For the Model III spread function, the random field $Y(\underline{x})$, is described by the following series of equations:

The characteristic function, $\phi_Y(u)$, is given by

$$\phi_{Y_3}(u) = \exp[\frac{-2\pi\lambda_3}{a_3^2\beta_3^2}(1 - \frac{\beta_3}{\beta_3 - iu})] \qquad (2.5.1)$$

where

u = a complex variable

$\beta_3 = E^{-1}[\alpha_3]$ = Model III depth parameter (length^{-1})

a_3 = Model III spread function parameter (mm/km)

λ_3 = Model III raincell concentration parameter (area^{-1})

The total storm depth probability density function, $f_Y(y)$, is given by

$$f_Y(y) = e^{-4\theta_3/\beta_3^2}\{\delta(y) + 2[\theta_3/(\beta_3 y)]^{1/2} e^{-\beta_3 y} I_1[4(\theta_3 y/\beta_3)^{1/2}]\} \qquad (2.5.2)$$

where

$$\delta(y) = \text{the Delta-Dirac Function} = \begin{cases} 1 & , y = 0 \\ 0 & , \text{otherwise} \end{cases}$$

$$\theta_3 = \pi\lambda_3/(2a_3^2)$$

$$\beta_3 = \mu_{\alpha_3}^{-1}$$

$$I_1(z) = \sum_{k=0}^{\infty} \frac{(z/2)^{2k+1}}{k!(k+1)!}$$

The expected value of $Y(\underline{x})$ is given by

$$E(Y) = 4\theta_3/\mu_{\alpha_3}^{3} \tag{2.5.3}$$

The variance of $Y(\underline{x})$ is described by

$$VAR(Y) = 8\theta_3/\mu_{\alpha_3}^{4} \tag{2.5.4}$$

The coefficient of skewness is given by

$$C.S. = \frac{3a_3}{2E[\alpha_3][\pi\lambda_3]^{1/2}} \tag{2.5.5}$$

The covariance of $Y(\underline{x})$ is approximated by

$$\begin{aligned} cov(Y) \approx \frac{4\pi\lambda_3}{a_3^2\beta_3^4} \Big\{ & K_1\Big(\frac{a_3\beta_3\nu}{2}\Big) e^{\frac{-a_3\beta_3\nu}{2}} \Big[\frac{a_3\beta_3\nu}{2} + \Big(\frac{a_3\beta_3\nu}{2}\Big)^2 + \frac{1}{4}\Big(\frac{a_3\beta_3\nu}{2}\Big)^3 \Big] + \\ & + K_0\Big(\frac{a_3\beta_3\nu}{2}\Big) e^{\frac{-a_3\beta_3\nu}{2}} \Big[\frac{1}{2}\Big(\frac{a_3\beta_3\nu}{2}\Big)^2 + \frac{1}{2}\Big(\frac{a_3\beta_3\nu}{2}\Big)^3 + \frac{3}{16}\Big(\frac{a_3\beta_3\nu}{2}\Big)^4 \Big]\Big\} \end{aligned} \tag{2.5.6}$$

The variance function is approximated by

$$\gamma[A] \approx \frac{1}{[1+(\frac{8}{3\pi}a_3\beta_3)^2 A]^{1/2}\,[1+(\frac{8}{3\pi}a_3\beta_3)^2 A\{\frac{1+(\frac{3}{25}a_3\beta_3)^2 A}{1+(\frac{8}{3\pi}a_3\beta_3)^2 A}\}]^{1/2}} \tag{2.5.7}$$

CHAPTER 3

Model Parameter Estimation

3.1 Introduction

In this chapter, we will focus on estimation of the parameters of each model from real station observation data. The storm day precipitation data came from the Walnut Gulch Experimental Watershed which is located in southeast Arizona. The basin and the dense raingage network are described in a separate volume.

From the eight summer rainy seasons, a data set of 428 separate stormdays is obtained. These fields are used to estimate the parameters of each analytical model for each of the storm days.

In Chapter 2, we saw that each of the three models require only three independent parameters:

a_i: the rain cell spread function parameter

λ_i: the rain cell (spatial Poisson) density parameter

$E[\alpha_i]$: the expected value of the rain cell center depth

We use the spatial correlation of $Y(\underline{x})$, $\rho(\nu)$ and the storm day spatial correlation estimator $\hat{\rho}(\nu)$ to estimate the model spread function parameters, a_i. E(Y) and VAR(Y) of $Y(\underline{x})$ and the storm day estimators $\hat{E}(Y)$ and $\hat{VAR}(Y)$ are then used to estimate $E(\alpha_i)$ for each model. Finally, given estimates for a_i and $E(\alpha_i)$, we use E(Y) of the random field and the $E(\hat{Y})$ estimator to arrive at an estimate of λ_i. From these relationships, we obtain a set of nine estimated parameters (3 per model) for each of the 428 storm days.

The correlation function was chosen over the variance function for the estimation of a_i for these reasons:

1. small sample size for a 154 km^2 area of irregular shape which introduces very large estimation error in using the variance function.
2. analytical simplicity of the correlation function given that an optimal fitting procedure must be used, and
3. stability of parameter estimates since numerical simulations showed $\gamma(A)$ to be more sensitive to parameter variability than $\rho(\nu)$

The coefficient of skewness was not used in the parameter estimation procedure.

3.2 Estimation of Model Parameters

In order to determine the optimum value of the spread function decay parameter a_i we use the method of least squares estimation. For a single storm day we minimize the standard square error between the analytical expression for the spatial correlation and the estimator of the spatial correlation at discrete lag distances.

The Model II and Model III spatial correlation expressions each contain zeroth and first order Bessel Functions of the second kind. The analytical expression for the Model III covariance function in particular, is rather complicated. The numerical Bessel functions are typically expressed as polynomial approximations of the analytical Bessel functions. For this reason it is unwise to perform multiple partial differentiation in order to perform a parameter estimation by least squares fitting using a Newton-Raphson or Secant Method technique. We thus used a basic number line search algorithmn to optimize the spread function parameter a_i.

A computer program called CORRFIT.FORTRAN is developed which minimizes the standard square error between the observed (i.e. interpolated) storm day spatial correlation $\rho(\nu)$ and each model's analytical expression. In addition to determining the "best" value of a_i, the program also determines the values for $E(a_i)$ and λ_i, for each model, storm day by storm day.

The minimization algorithm in CORRFIT.FORTRAN subdivides the abscissa into somewhat coarse intervals of width Δa_j. The standard square error of $f(a_i)$ is then determined for a value a_i close to the origin, and then again at $a_{i+1} = a_i + \Delta a_j$. The relative difference in magnitude between $f(a_i)$ and $f(a_{i+1})$ determines the approximate slope of the standard square error function. As long as the slope of the function is negative, the algorithm continues to calculate $a_{i+k} = a_{i+k-1} + \Delta a_j$. When $f(a_{i+k}) > f(a_{i+k-1})$, the algorithm moves back toward the origin a distance of $2\Delta a_j$, and subdivides these two intervals into twenty intervals of width Δa_n. The scanning calculations are begun again beginning at a_{i+k-2} until the slope becomes positive once more. Again the algorithm backs up a distance of $2\Delta a_n$, and subdivides these two intervals by twenty again.

This procedure is finally discontinued when the difference between the optimum value of a_i and the scanned value of a_i is less than

10^{-4}. The basic procedure is invoked in turn for each of the three analytical expressions for the spatial correlation until the set of three optimum values of $\hat{a}_1$, $\hat{a}_2$, and $\hat{a}_3$ are determined for the particular storm day. The CORRFIT.FORTRAN program next calculates the values of $E(\alpha_i)$ and λ_i for each model from the individual model expressions.

For Model I,

$$\hat{E}(\alpha_1) = \frac{\hat{Var}[Y]}{\hat{E}[Y]} \qquad (3.2.1)$$

Solving for λ_1, from eq. (2.3.3)

$$\hat{\lambda}_1 = \frac{2\hat{a}_1^2\hat{E}[Y]}{\pi\ \hat{E}[\alpha_1]} \qquad (3.2.2)$$

For Model II,

$$\hat{E}(\alpha_2) = \frac{2\ \hat{VAR}[Y]}{\hat{E}[Y]} \qquad (3.2.3)$$

Solving for λ_2, from eq. (2.4.2)

$$\hat{\lambda}_2 = \frac{a_2^2\ \hat{E}[Y]}{2\pi\hat{E}[\alpha_3]} \qquad (3.2.4)$$

Similarly for Model III,

$$\hat{E}(\alpha_3) = \frac{\hat{VAR}[Y]}{2\hat{E}[Y]} \qquad (3.2.5)$$

and from eq. (2.5.3)

$$\hat{\lambda}_3 = \frac{\hat{a}_3^2 \hat{E}[Y]}{2\pi \hat{E}^3[\alpha_3]} \qquad (3.2.6)$$

When the set of nine parameters are calculated, the program writes the results to a file, and then moves on to perform the parameter estimation for the next storm day. It is worth mentioning that the CORRFIT.FORTRAN computer code is a modification of the program called FUTURE.FORTRAN described in one of the companion volumes belonging to this study.

CORRFIT.FORTRAN accesses each of the eight summer season storm day archive files one at a time. When the execution of CORRFIT.FORTRAN is complete, the estimates of the three model parameters for each model, plus some additional information about each of the 428 storm days are written to a file. These are archived in a file entitled MODEL_PARAMETERS.DATA. A hard copy of these data may be found in Appendix I.

3.3 Bias of Estimators

If we consider each storm day set of observations to be a single 154 km^2 sample from a parent field of infinite spatial extent, then we must ask if our estimates $\hat{E}(Y)$, $\hat{VAR}(Y)$ and $\hat{\rho}(\nu)$ as determined from the sample are biased estimates of the corresponding population statistics. Vanmarcke [1983, pp. 332-333] examines this question in the standard way by seeing if the expectation of the estimator equals the desired statistic. He finds

$$E[\hat{E}(Y)] = E(Y) \qquad (3.3.1)$$

but

$$E[\hat{VAR}(Y)] = VAR(Y)[1 - \gamma(A_c)] \qquad (3.3.2)$$

and for $A_c^{1/2} >> \nu$

$$E[\hat{COV}(\nu)] = COV(\nu) - \sigma_Y^2 \gamma(A_c)$$

Using eq. 3.3.1 and 3.3.2 we can write

$$E[\hat{\rho}(\nu)] \simeq \frac{E[COV(\nu)]}{E[VAR(Y)]} = \frac{\rho(\nu) - \gamma[A_c]}{1 - \gamma[A_c]} \qquad (3.3.3)$$

Thus, to obtain an unbiased estimate $\rho(\nu)$, we must modify $\hat{\rho}(\nu)$ according to

$$\rho(\nu) = \hat{\rho}(\nu)[1 - \gamma(A_c)] + \gamma[A_c] \qquad (3.3.4)$$

For $A_c = 154$ km^2 the prior estimate is $\gamma[A_c] = 0(10^{-1})$ whereupon, for small ν at least, we may neglect the correction to $\hat{\rho}(\nu)$. To the first approximation therefore, the values of a_i estimated as described above need no bias correction and $a_i \simeq \hat{a}_i$.

Using eq. 3.3.1 and 3.3.2 we find the unbiased estimator of $E(\alpha_i)$ and λ_i for each of the three models. In applying these corrections we need an a priori estimate of $\gamma_i(A_c)$. In the manner of Vanmarcke [1983] we can approximate $\gamma_i(A_c)$ using the variance functions for each model as described in Chapter 2. We do this by using $A_c = 154.0$ km^2 and using the expected value of the population of a_i. $\gamma(A_c)$ for model III also requires the expected value of $E(\alpha_3)$. The reader is referred to Chapter 2 for the particular variance function expressions.

For Model I, we find

$$E[\alpha_1] = \frac{\hat{VAR}[Y]}{[1 - \gamma[A_c]]\hat{E}[Y]} \tag{3.3.5}$$

and

$$\lambda_1 = \frac{2a_1^2[1 - \gamma[A_c]]\hat{E}^2[Y]}{\pi \; \hat{VAR}[Y]} \tag{3.3.6}$$

For model II

$$E[\alpha_2] = \frac{2\hat{VAR}[Y]}{[1 - \gamma[A_c]] \; \hat{E}[Y]} \tag{3.3.7}$$

and

$$\lambda_2 = \frac{a_2^2[1 - \gamma[A_c]] \; \hat{E}^2[Y]}{4\pi \; \hat{VAR}[Y]} \tag{3.3.8}$$

Similarly for Model III

$$E[\alpha_3] = \frac{\hat{VAR}[Y]}{2[1 - \gamma[A_c]] \; \hat{E}[Y]} \tag{3.3.9}$$

and

$$\lambda_3 = \frac{4a_3^2[1 - \gamma[A_c]]^3 \; \hat{E}^4[Y]}{\pi \; \hat{VAR}^3[Y]} \tag{3.3.10}$$

In order to solve for the unbiased model parameters $E(\alpha_i)$ and λ_i, a computer program called UNBIAS.FORTRAN is written. In order to implement UNBIAS.FORTRAN, the population expectation of a_i and $E(\alpha_3)$ must be determined because these parameters are needed to estimate $\gamma_i(A_c)$. These parameters are determined from the 428 storm day population of model parameters archived as MODEL_PARAMETERS.DATA.

Given these parameters, UNBIAS.FORTRAN solves eqs. 3.3.5 through 3.3.10, and the results are archived as UNBIASED_MODEL_PARAMETERS.DATA. The data will be the subject of further analysis in Chapter 4. A hard copy of these data is given in Appendix II.

CHAPTER 4

4.1 Distribution of Model Parameters

In this chapter we estimate the distributions of the model parameters across the ensemble of 428 summer season stormdays. In performing this analysis we assume the storms to be independent samples from the same homogeneous population of air mass thunderstorms. The resulting distributions will provide a quantitative measure of the variability of this class of storm.

As described in Chapter 3, the nine unbiased model parameters are calculated for each storm day and are archived as the UNBIASED MODEL PARAMETERS.DATA file. Each of these 428 member parameter sets is rank-ordered and the largest and smallest values are discarded as probable outliers. The remaining 426 values are sorted into quantiles and plotted as histograms of relative frequency as is shown in Figures 4.1.1 through 4.1.9.

The distribution of a_i (see Figures 4.1.1-4.1.3) is fitted with a two parameter Gamma distribution

$$f_t(a_i) = \frac{\chi(\chi a_i)^{\kappa-1} e^{-\chi a_i}}{\Gamma(\kappa)} \qquad (4.1.1)$$

by the method of maximum likelihood. The fitted values of the "order" parameter k and the "scale" parameter χ are given in the upper right hand corner of the plots along with the values of the first two moments of the 426 member sample. These moments are subscripted with "t" to indicate moments of a sample in time.

Figures 4.1.4-4.1.6 show λ_i to be essentially constant.

The distribution of $E[\alpha_i]$ (see Figures 4.1.7-4.1.9) is fitted with an exponential distribution

$$f(E[\alpha_i]) = \varepsilon_i \exp\{-\varepsilon_i E[\alpha_i]\} \tag{4.1.2}$$

using the method of moments.

The first two moments of the distributions of the biased and unbiased parameters are compared for all three models in Table 4.1.1.

Various parameter moments, coefficients of variation [CV], and correlation coefficients are given in Table 4.1.2.

The moments of observed and interpolated point storm depth as calculated from all points in the 428 stormday ensemble are listed for comparison in Table 4.1.3. The agreement is reassuring.

TABLE 4.1.1

Moments of Estimated Parameters

(Sample Size = 426)

		Biased Estimates			Unbiased Estimates		
		Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
$E_t[a]$	(km^{-1})	0.426	1.108	1.771*	0.426	1.108	1.771*
$VAR_t[a]$	(km^{-2})	0.027	0.189	2.175*	0.027	0.189	2.175*
$E_t[\lambda]$	($cells/km^2$)	0.120	0.103	0.468	0.11	0.09	0.28
$VAR_t[\lambda]$	($cells^2/km^4$)	0.052	0.041	0.810	0.04	0.03	0.29
$E_t\{E[\alpha]\}$	(mm)	3.646	7.293	1.823	4.05	8.01	2.16
$VAR_t\{E[\alpha]\}$	(mm^2)	14.224	56.896	3.556	17.58	68.68	5.00

*mm/km

TABLE 4.1.2

Parameter Variability and Correlation

(Sample size = 426)

(Parameter)	(dimension *)	Model 1	Model 2	Model 3
		Unbiased Estimates		
$E[\alpha]$	(km^{-1})	0.43	1.11	1.77
var[a]	(km^{-2})	0.03	0.19	2.18
$E[\lambda]$	($cells/km^2$)	0.11	0.09	0.28
$var[\lambda]$	($cells^2/km^4$)	0.04	0.03	0.29
$E[\mu_\alpha]$	(mm)	4.05	8.01	2.16
$var[\mu_\alpha]$	(mm^2)	17.58	68.68	5.00
CV[a]		0.39	0.39	0.83
$CV[\lambda]$		1.92	1.92	1.92
$CV[\mu_\alpha]$		1.04	1.04	1.04
$CV[\theta]$		1.96	1.47	----
$\rho[a\lambda]$		0.06	-0.13	0.18
$\rho[\lambda\mu_\alpha]$		-0.19	0.16	0.17
$\rho[\mu_\alpha a]$		-0.44	0.43	-1.00
$\rho[\theta\mu_\alpha]$		-0.12	-0.24	----
$E[a^{-2}]$	(km^2)	8.46	1.30	3.61

(continued)

Table 4.1.2
(Continued)

(Parameter)	(dimension *)	Unbiased Estimates Model 1	Model 2	Model 3
$E[\lambda a^{-2}\mu_\alpha]$	(mm)	2.47	0.62	0.58
$\mathrm{var}[\lambda a^{-2}\mu_\alpha]$	(mm^2)	13.03	0.81	8.50
$E[\lambda a^{-2}] \equiv \frac{2}{\pi} E[\theta]$		0.80	0.10	---
$E[\lambda a^{-2}\mu_\alpha^{-2}]$	(mm^2)	15.98	7.90	0.37
$E[\lambda a^{-2}\mu_\alpha^{3}]$	(mm^3)	---	---	0.62
$E[\lambda a^{-2}\mu_\alpha^{4}]$	(mm^4)	---	---	2.13

*Note that a_3 is dimensionless. All listed dimensions are for Models 1 and 2.

TABLE 4.1.3

Moments of Point Storm Depth

		Station Observations (Y_o)	Interpolated (Y)
Mean	(mm)	3.86	3.89
Variance	(mm^2)	55.20	54.66

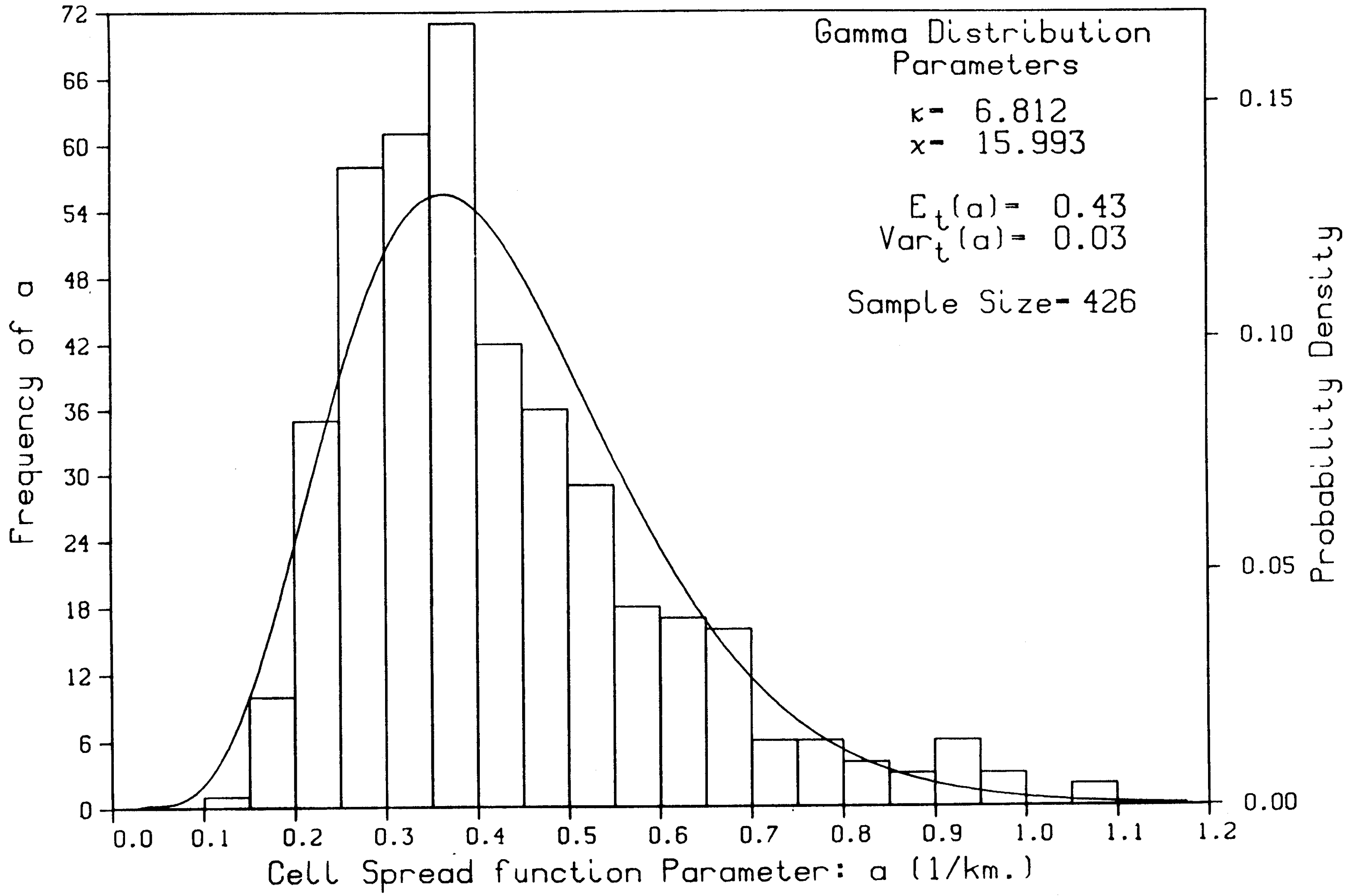


Fig. 4.1.1 Distribution of a_1

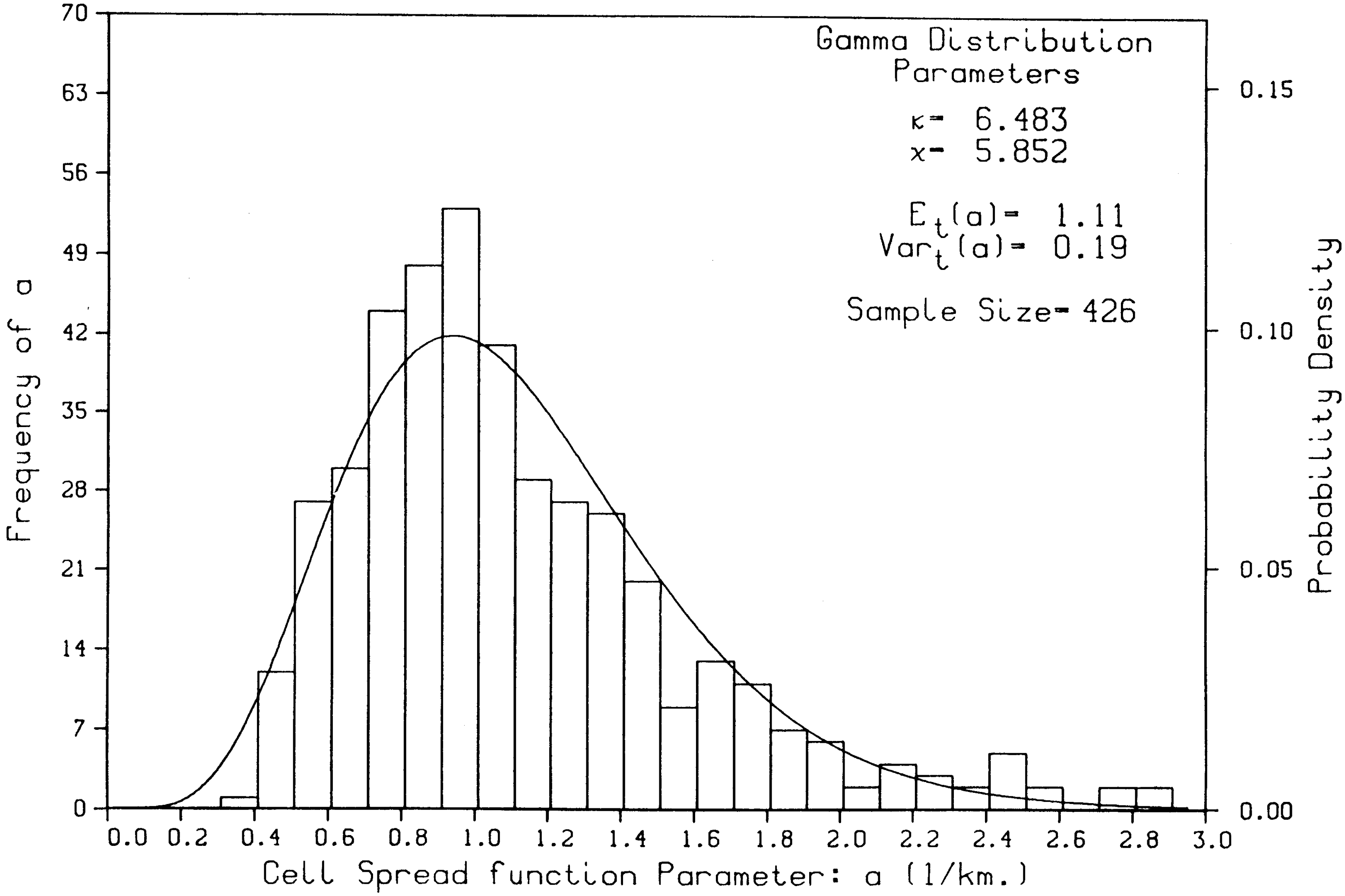


Fig. 4.1.2 Distribution of a_2

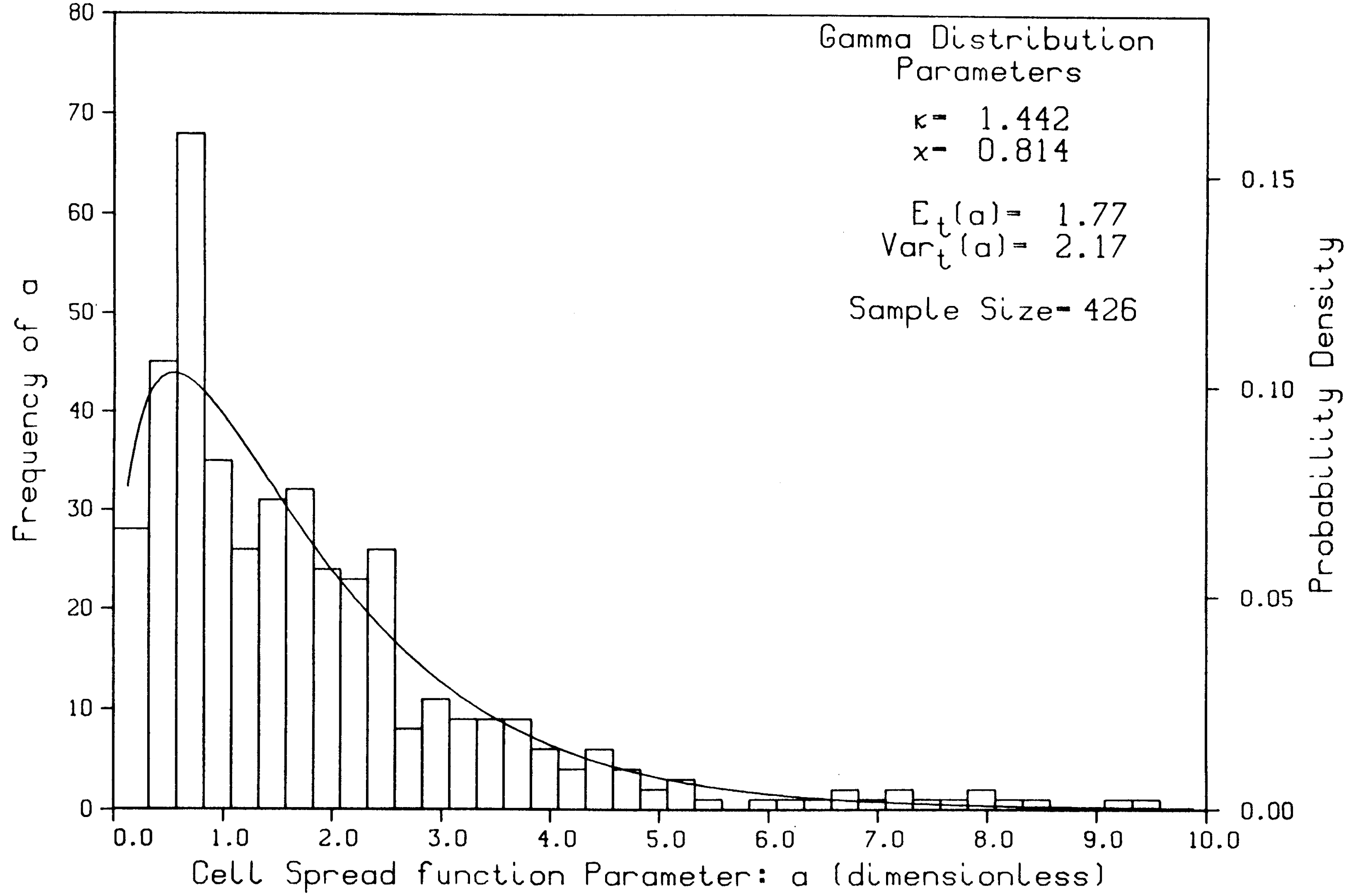


Fig. 4.1.3 Distribution of a_3

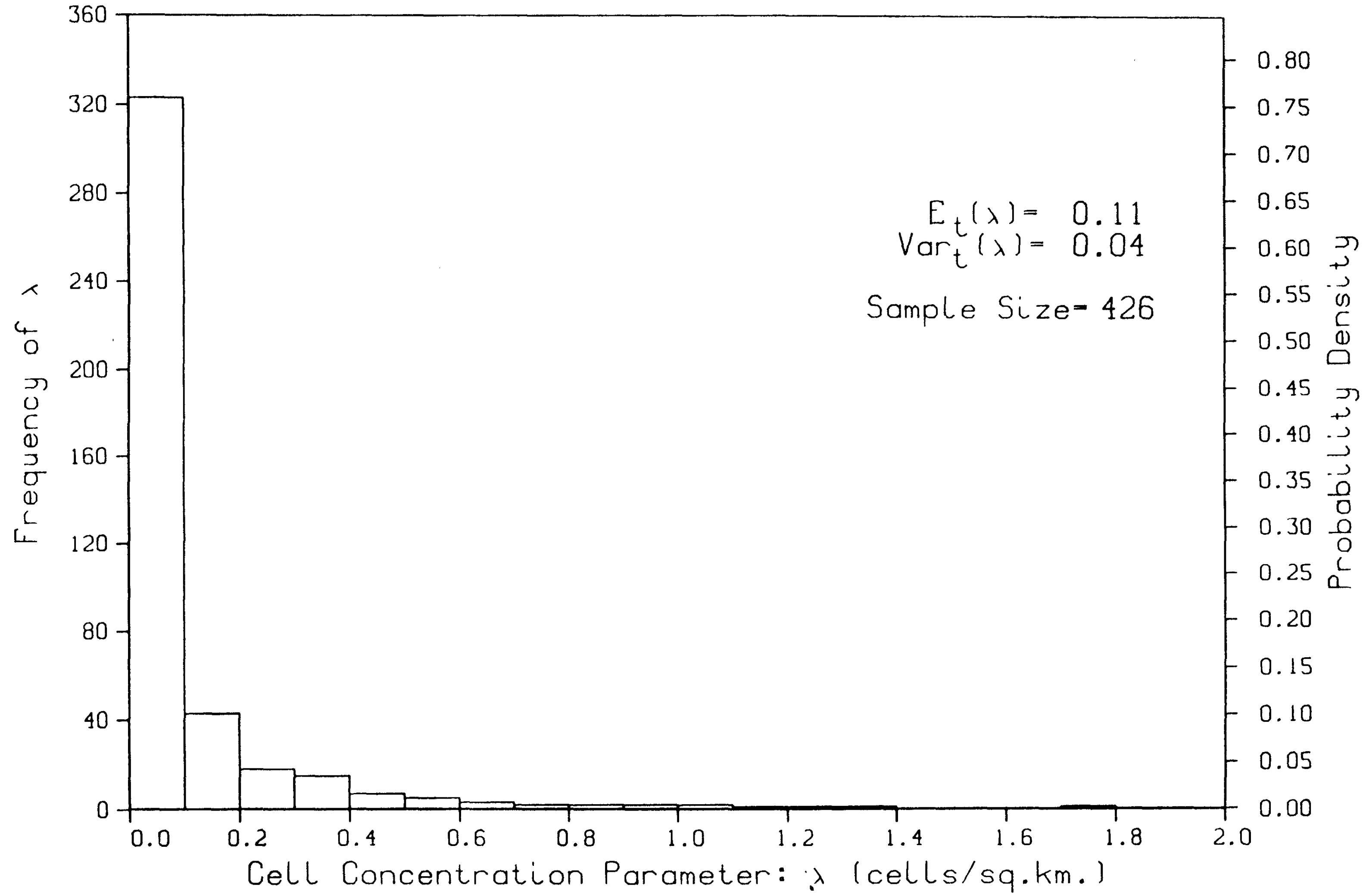


Fig. 4.1.4 Distribution of λ_1

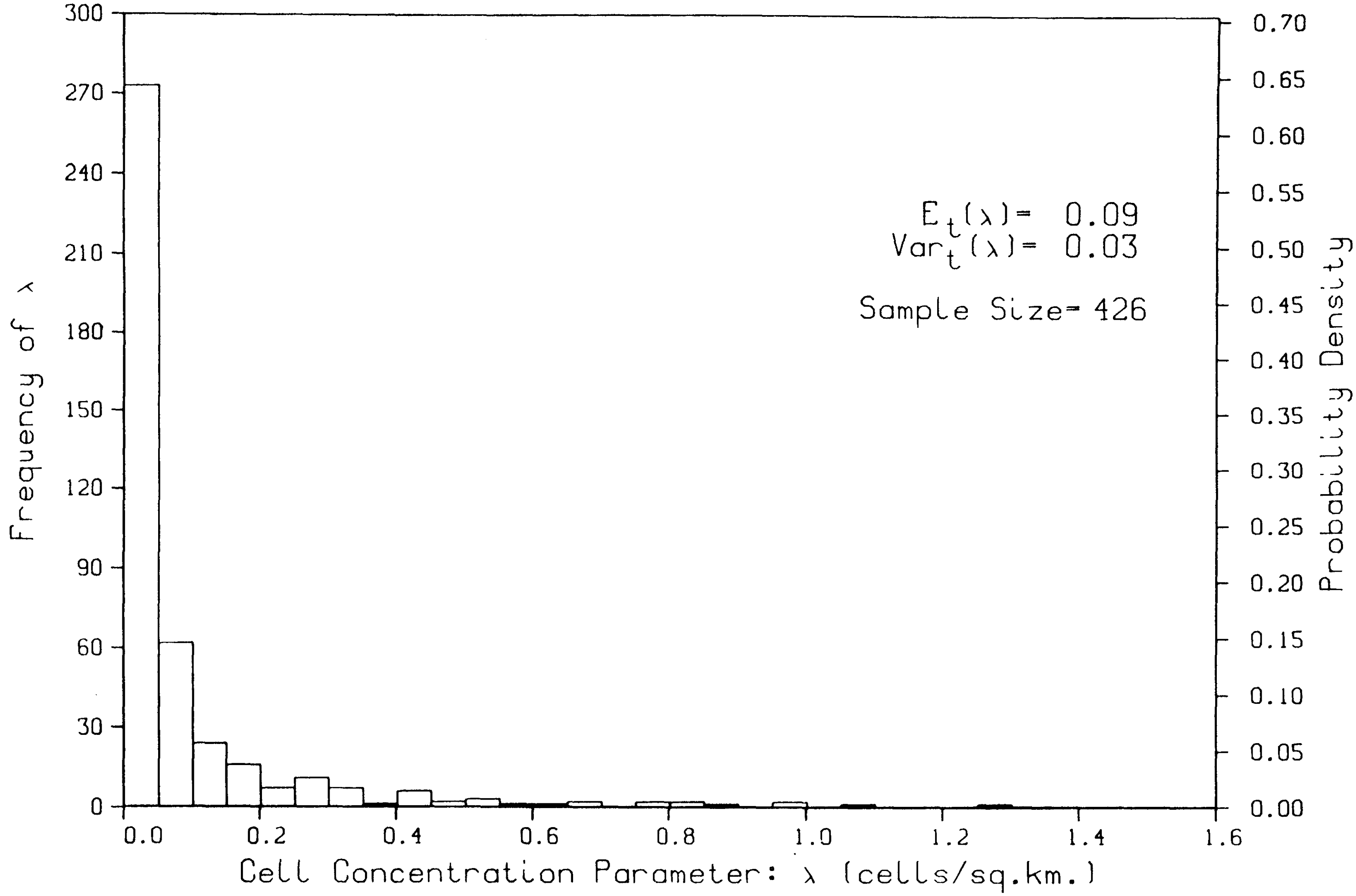


Fig. 4.1.5 Distribution of λ_2

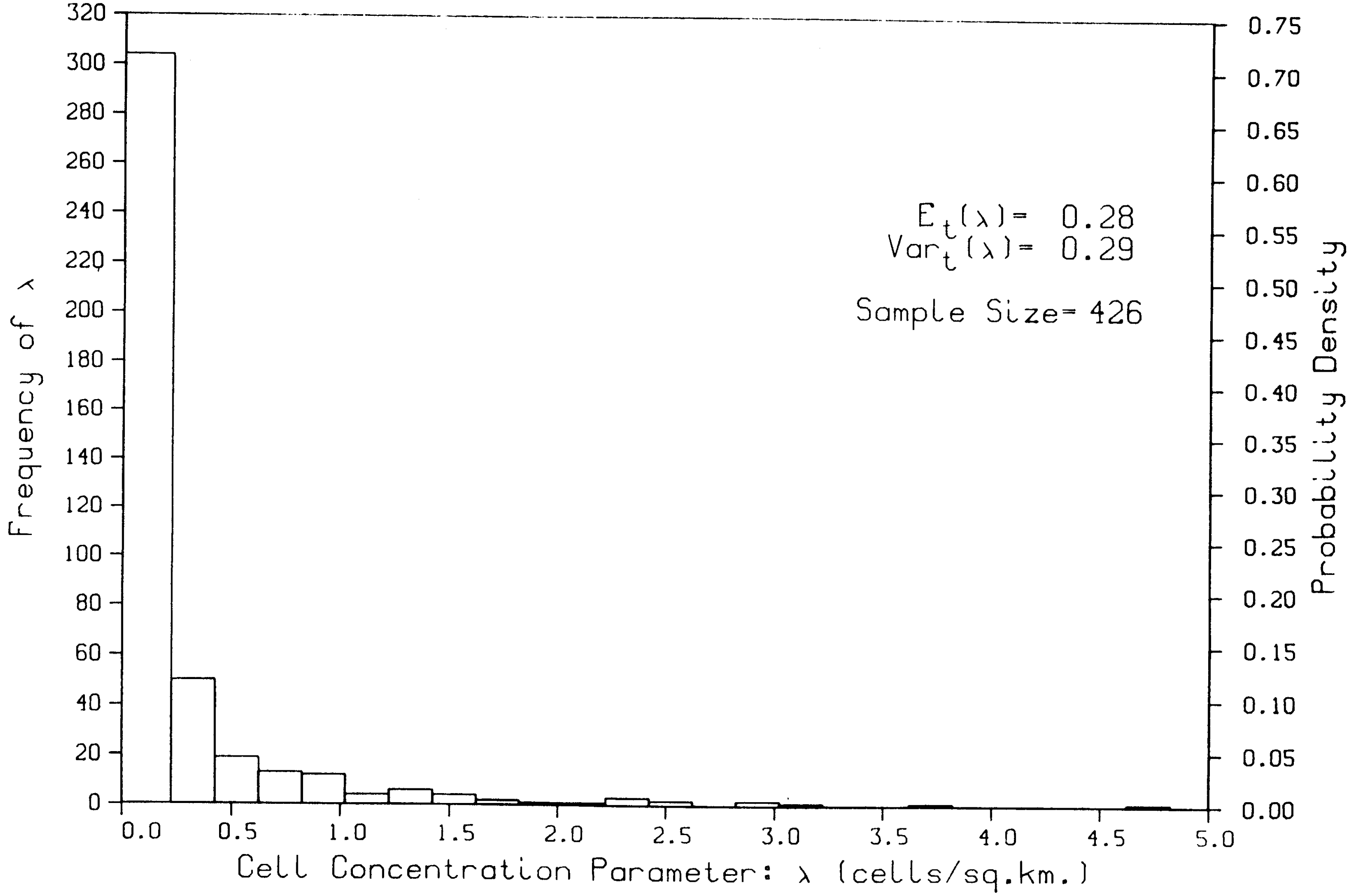


Fig. 4.1.6 Distribution of λ_3

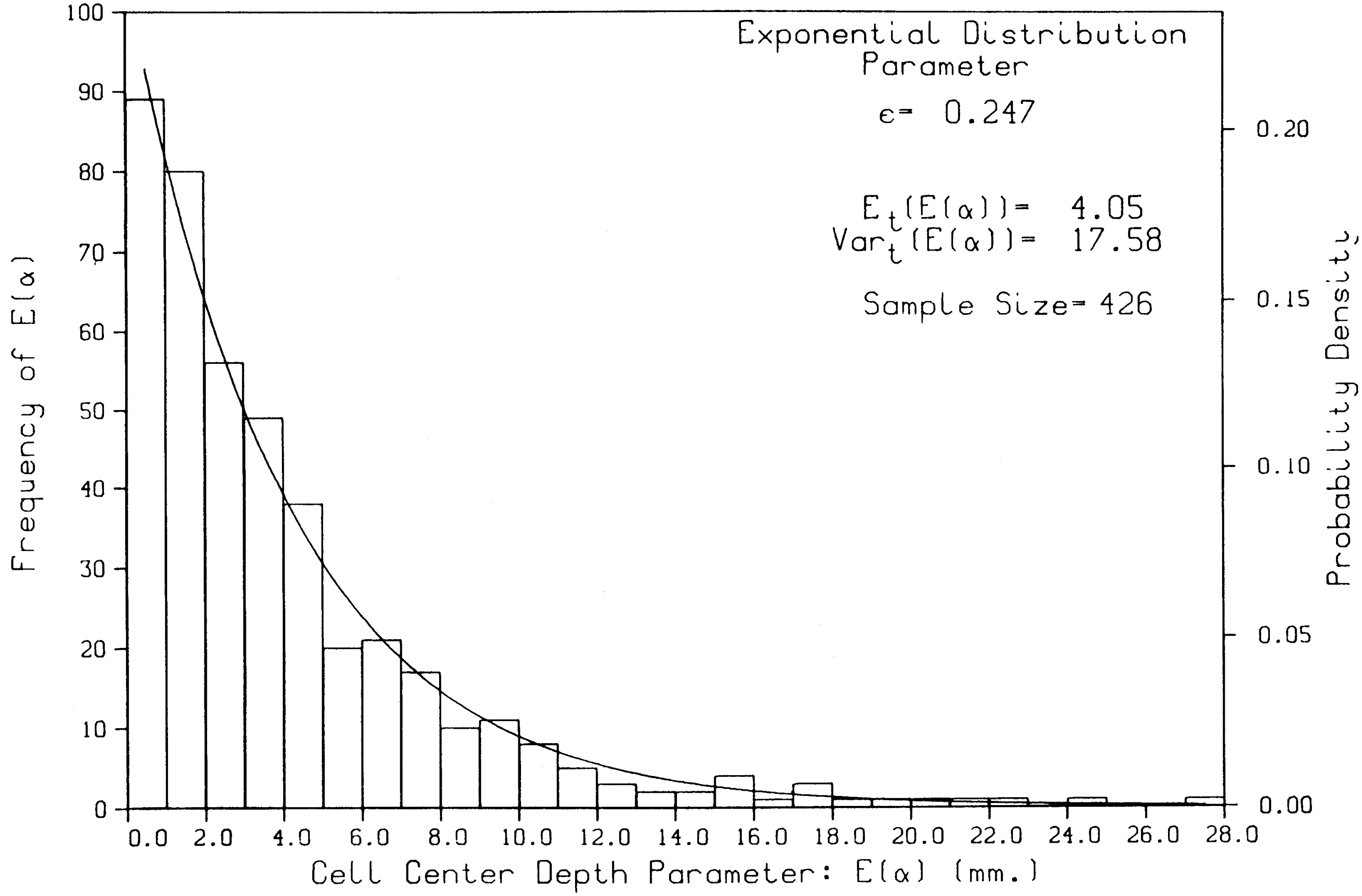


Fig. 4.1.7 Distribution of $E[\alpha_1]$

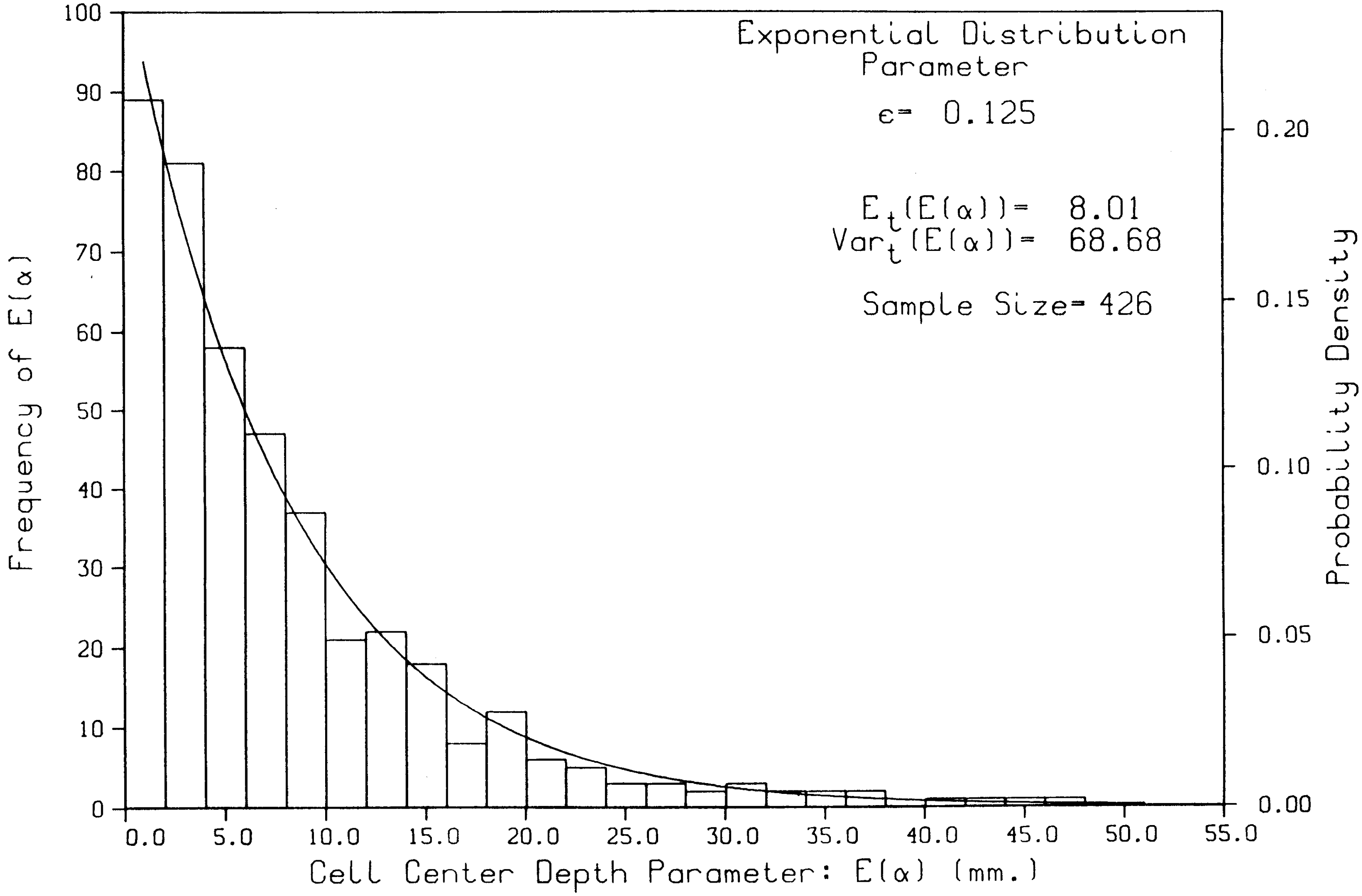


Fig. 4.1.8 Distribution of E[α_2]

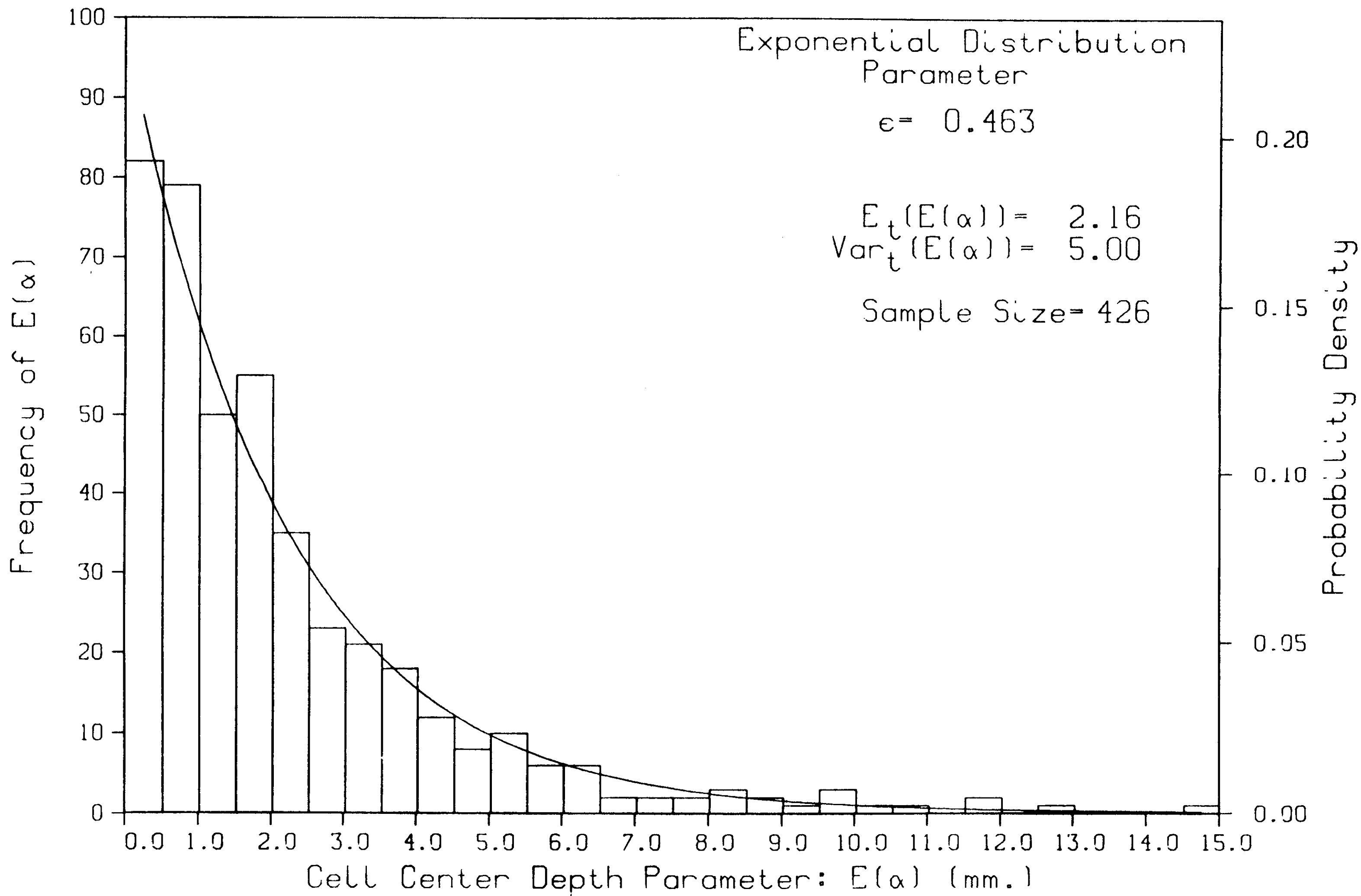


Fig. 4.1.9 Distribution of $E[\alpha_3]$

CHAPTER 5

Theory and Numerical Simulation of the Spatial Poisson Models

5.1 Simulation of Total Storm Depth

A numerical simulation of the rainfall depth models is a useful tool in their evaluation. In this chapter a review of the spatial Poisson process is presented and a numerical simulation is developed for each of the three conceptual models.

5.2 The Spatial Poisson Process: Theory and Simulation

5.2.1 Theory

The temporal Poisson process has been used as a stochastic point precipitation model in many hydrological applications [e.g. Eagleson, 1978]. It is not difficult to extend the temporal formulation to the spatial domain [see Cox and Isham, 1980].

Let the Poisson density parameter λ be a positive constant with dimension (area^{-1}). Let R_1 be the distance to a Poisson-distributed point from an arbitrary origin in space. $R_1 > r$ only if a disk of area πr^2 contains no point other than that point which defines the origin. Then the probability that $R_1 > r$ is described by

$$P[R_1 > r] = e^{-\lambda\pi r^2} \tag{5.2.1}$$

and the pdf of R_1 is given by

$$f_{R_1}(r) = 2\pi\lambda r e^{-\lambda\pi r^2} \tag{5.2.2}$$

Cox and Isham [1980, p. 147] call this the density of the "nearest neighbor distances". More explicitly, if $A_1 = \pi R_1^2$ is the area covered by a "disk" centered at the origin, the nearest neighbor "point" lies somewhere along the perimeter of A_1. If R_k describes the distance between the origin and the kth point, where the kth point lies on the perimeter of $A_k = \pi R_k^2$, then the areas A_1, $A_2 - A_1$, $A_3 - A_2$,... are independent and exponentially distributed with parameter λ. Then, A_k and R_k^2 are Gamma distributed such that

$$f_{R_k}(r) = \frac{2(\pi\lambda)^k r^{2k-1} e^{-\lambda\pi r^2}}{(k-1)!} \tag{5.2.3}$$

Each of the k points in space can be located by polar coordinates R_k and θ_k from a fixed origin. The Poisson process is obtained by generating a sequence of points in space (R_k, θ_k) such that

$$R_1^2, \ (R_k^2 - R_{k-1}^2) \qquad k=2,3,...$$

are independent and exponentially distributed with parameter λ. θ_k is a sequence of independent random variables (angles) which are uniformly distributed over $[0,2\pi]$, and are independent of R_k. 5.2.2 Simulation

5.2.2 Simulation

The actual simulation of total rainfall depth requires several steps. The first is the creation of the spatial Poisson points according to the procedure outlined in the prior section. The second step involves the creation of the storm cells, the center of each being located at one of the Poisson spatial points. The third step involves the integration of the total depth at each point in space from contributions by a finite number of storm cells.

First, the angle θ_i of the point located at (R_i, θ_i) with respect to the origin, is created from the transformation of a uniformly distributed number generated over [0,1] to the domain of $[0,2\pi]$. This is accomplished by using the IMSL (International Mathematical Statistical Libraries) subroutine called GGUBFS. Next, the difference in the areas of the overlapping disks centered at the origin, is created. This set of difference values of A_j, where $A_j = \pi[R_{j+1}^2 - R_j^2]$ is created by sampling from an exponential variate using the storm cell spatial parameter λ. This is done by using the IMSL subroutine called GGEXN. A_j is then transformed to radius R_j, and thus the polar coordinates of each cell center on the 2-D plane are known. The polar coordinates are then transformed to cartesian coordinates in order to conform to a two-dimensional grid mesh.

From the assumption that the center depths of the storm cells in one realization are exponentially distributed, we generate the set of α_j. Using $E[\alpha]$, we sample from an exponential variate, which as before is the IMSL subroutine called GGEXN. By using a different seed, this sampling is independent of that which generates A_j. At this stage of the simulation, we now have a series of points located in the two-dimensional plane, each having a maximum storm cell depth.

The simulation then focuses on the creation of the individual storm cells. Each cell has an identical spread function but a unique center depth. Because of this, the simulation first determines the domain of influence of the cell. For Models I and II, the region of influence is limited by the delineation of 0.01 mm total depth which marks the cell's edge. Next, the simulation adds this cell's contribution to the total rainfall within its sphere of influence. The simulation then turns to the next cell and the process is repeated. When the contribution of total rainfall by the last cell is made to the simulated field, the simulation is complete.

5.3 Numerical Simulation of Total Rainfall Depth

The actual simulation generates storm cells beyond the edge of the desired finite field. This is necessary because rain cells located outside the perimeter of the simulation area contribute to the node depths at the edge of the field and the peripheral nodes must be representative of the characteristics of the infinite random field. This extra distance away from the edge of the field, called the maximum simulation distance, is unique for each model and depends on the values of the raincell concentration parameter λ_i and the raincell center depth parameter $E(\alpha_i)$. The total number of cells to be generated by the simulation is a function of the maximum simulation distance.

The maximum simulation distance is a radial distance between the origin or center of the simulation field and a circle, beyond which the storm cells no longer contribute significant rainfall to the nodes located at the edge of the mesh. This distance equals $R_L + R_O$, where (for a 30 km by 30 km field) R_L equals 21.21 km, the distance from the center of the field to one of its corners. R_O is the distance from this corner to the edge of the circular simulation field. Figure 5.3.1 shows the maximum simulation distance, the rectangular simulation field of interest and the limits of the simulation represented by the circle.

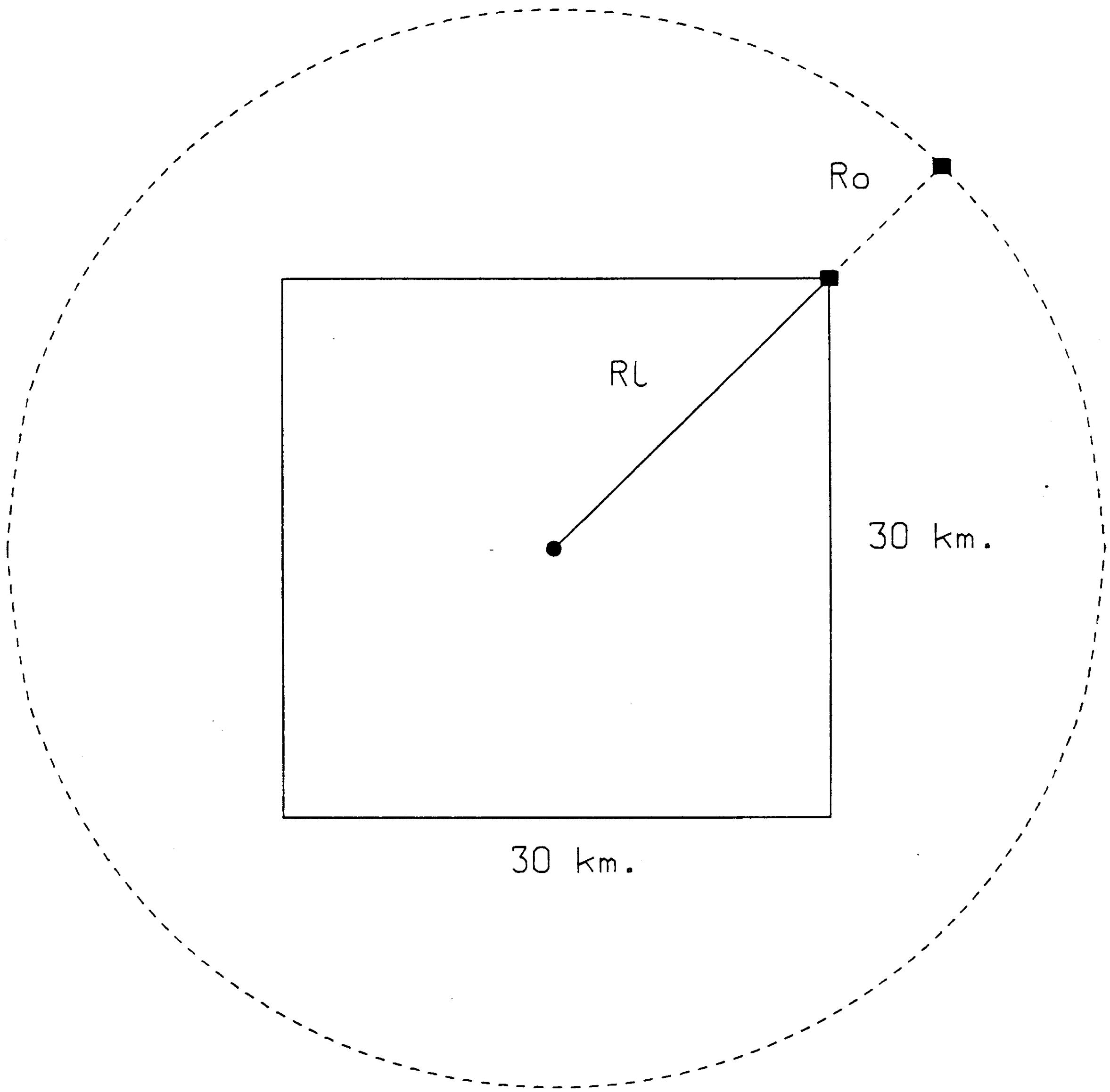


Fig. 5.3.1 Maximum Simulation Distance

Because of the infinite domain of the Model I and Model II spread functions, the cell "skirt" is terminated at the radius where $Y \leq 0.01$ mm. For larger radius the cell depth is set equal to zero. (This truncation corresponds to the minimum depth filter imposed on the interpolated observed storm day random field). R_o will now be the radius to $Y = 0.01$ mm of the cell with the largest likely center depth. We arbitrarily set this depth $\hat{\hat{\alpha}}_i$ at

$$\hat{\hat{\alpha}}_i = E(\alpha_i) + 3\sigma_{\alpha_i} \tag{5.3.1}$$

Because α is exponentially distributed over each storm, eq. 5.3.1 becomes

$$\hat{\hat{\alpha}}_i = 4E(\alpha_i) \tag{5.3.2}$$

For Model I, the quadratically exponential spread function, the surface of each cell is described by

$$g_1(\underline{x},\underline{z},\alpha_1) = \alpha_1 e^{-2a_1^2 r^2} \tag{5.3.3}$$

Then we have

$$0.01 = \hat{\hat{\alpha}}_1 \; e^{-2a_1^2 R_{o_1}^2} \tag{5.3.4}$$

or

$$R_{o_1} = \left[\frac{\ell n\, \hat{\hat{\alpha}}_1 - \ell n(0.01)}{2a_1^2}\right]^{1/2} \tag{5.3.5}$$

In a similar way, we solve for R_o when simulating the Model II random field.

$$R_{o_2} = \frac{\ell n\, \hat{\hat{\alpha}}_2 - \ell n(0.01)}{a_2} \tag{5.3.6}$$

where

$$\hat{\hat{\alpha}}_2 = 4E(\alpha_2)$$

Because the surface of the Model III cell is conical, no truncation is necessary to calculate R_o.

$$R_{o_3} = \frac{\hat{\hat{\alpha}}_3}{a_3} \tag{5.3.7}$$

where

$$\hat{\hat{\alpha}}_3 = 4E(\alpha_3)$$

The number of storm cells generated during the simulation is proportional to the storm day value of λ_i. Before the simulation is started, one must first know the number of cells to be generated, nR. An arbitrarily large value of nR will result in a very lengthy (and costly) simulation. Recalling that the "nearest neighbor" distance is exponentially distributed in the spatial Poisson process, we create these distances by sampling from an exponential variate. Given the maximum simulation distance, nR is then found.

Three computer programs must be executed before the numerical simulation is begun. The first program is called PARAM.FORTRAN. This program collects the three spatial parameters for the storm day of interest and for the specific model. It then calculates the maximum simulation distance. The program will prompt the user for the model of choice and for the date of the desired storm day.

The second pre-simulation program is called RADIUS.FORTRAN. The purpose of RADIUS.FORTRAN is to calculate nR, the number of cells the simulation must generate. This program will prompt the user for a guess estimate of nR and for the value of λ_i. The program will output the values of $R_L + R_{o_i}$ corresponding to the necessary number of cells to be generated to attain this total radius as calculated by PARAM.FORTRAN. The program outputs these results to the terminal screen in increments of 50 cells. Given the estimate of nR from RADIUS.FORTRAN, the third preparatory program should be executed.

The third program is called POLAR.FORTRAN. As in RADIUS.FORTRAN, POLAR.FORTRAN will prompt the user for the guess estimate of nR from RADIUS.FORTRAN and the storm day value of λ_i. The value of nR necessary to exceed the maximum simulation distance will be outputed to the terminal screen with the corresponding value of $R_L + R_{o_i}$.

At this stage, the simulation may begin now that the three spatial parameters plus nR are known. The three computer programs, PARAM.FORTRAN, RADIUS.FORTRAN and POLAR.FORTRAN will be discussed further in Chapter 9.

5.4 Storm Day Total Rainfall Depth Simulation and Sampling From the Random Field

Simulation of the spatial Poisson model is storm day dependent as well as dependent on the chosen spread function. The simulated random field is a square area of 900 km^2 represented by a 301 by 301 matrix of node points. The simulation results in total simulation storm day depth at each of the 90,601 nodes in the fine resolution mesh. The actual simulation is accomplished by executing a single program called SIMWET.FORTRAN.

5.4.1 Simulation and Sampling From the Fine Mesh Simulated Random Field

SIMWET.FORTRAN is a multifaceted computer code which accomplishes several tasks. The program simulates the random field for the storm day and model of choice. The grid mesh resolution of SIMWET.FORTRAN is 100 meters. Following the generation of the simulated random field described in Section 2, the program creates two additional random fields, each of which are a subset of the original random field. This is to determine the sensitivity of sampling to the area of the random field. The sensitivity analysis will be discussed in a later section.

The original random field is called the Level III random field and is 900 km^2 in area. The Level III area contains all 90601 nodes and their corresponding simulated storm depths. The second random field is called the Level II random field. It incorporates the nodes within a rectangular region located in the middle of the Level III random field. The Level II field is 29.4 km wide (west to east) by 14.0 km (from south to north). Thus, Level II occupies the middle rows and almost all of the Level III columns. The Level II random field represents an area equal to 411.6 km^2 and shares 41,595 of the Level III nodes.

The Level I random field is located within the Level II random field. The Level I field is the Walnut Gulch basin and is identical in shape and dimension to the observed (interpolated) storm day random field. A numerical watershed boundary filter is applied to the Level II random field which results in the Level I random field being 154.21 km^2 in area.

Therefore, Level I shares 15,905 nodes with Level II and Level III. The details of the numerical watershed boundary are presented in a companion volume.

SIMWET.FORTRAN next examines each node in the Level I, Level II and Level III random fields and imposes the minimum depth filter. As previously mentioned, this filter assigns a value of zero on all nodes with depths that are less than or equal to 0.01 mm.

SIMWET.FORTRAN then writes a coarse mesh version of the Level III random field to a file for further analysis. The coarse grid mesh has a resolution of 200 meters. The mesh is created by retaining the nodes in every other row and every other column from the fine mesh Level III field. The representative area is still 900 km^2 but the random field is now represented by a 151 by 151 matrix containing 22,801 nodes. Each node has the original simulated depth except those which were modified by the minimum depth filter.

The program next retains the simulated depths at 93 nodes, each of which corresponds to the location of a single raingage in the Walnut Gulch network. These nodes are taken from the Level II simulation area as opposed to Level II because each share the same coordinate system and a number of the actual raingages lie outside the perimeter of the watershed. The 93 "simulation raingage" values are written to a file to be used for later analysis. The details of this analysis will be presented in Chapter 7.

The next task accomplished by the SIMWET.FORTRAN program is to sample each of the simulated random fields for their respective spatial distribution of total storm depth. The program creates finite elements of 0.01 km^2 and assigns each element a depth which equals the average of the 4 nodes located at the corner of the element. The precise details of this technique are described in a companion volume.

The level III random field has 90,000 of these elements, the Level II random field has 40,727 elements and the Level I random field has 15,421 elements. After the program has sampled each random field, the results of the spatial distribution of simulated total rainfall depth are written to a file. Further details about the SIMWET.FORTRAN computer program are presented in Chapter 8.

5.4.2 Sampling from the Coarse Mesh Simulated Random Field

In this section, we focus on the sampling of the simulated field on the coarse mesh (200 meter resolution). A computer program called SIMCRGAM.FORTRAN is developed to accomplish this. The code samples the random field for the first two moments, the coefficient of skewness, the spatial correlation and the variance function.

Because the simulation is computationally intensive the random field is not resimulated at the coarse resolution. Rather as described in the preceeding section, the coarse mesh simulation field was created by an execution of SIMWET.FORTRAN. SIMCRGAM.FORTRAN uses the coarse mesh output file created by SIMWET.FORTRAN as its input file.

SIMCRGAM.FORTRAN first creates the three separate random fields; Level III, Level II and Level I as before. Next, each random field is sampled for the first two moments and for the coefficient of skewness of the simulated total storm depth. The program then samples each field for the spatial correlation and then the variance function using techniques described in a companion volume. The spatial correlation is carried out to a maximum lag of 6 km in each random field. Correspondingly, the maximum variance function area is 36.0 km^2.

SIMCRGAM.FORTRAN then writes the results of this sampling to an output file, and the execution of the program is complete. More details about the SIMCRGAM.FORTRAN computer program is presented in Chapter 9.

5.4.3 Discussion of Simulation Results

The results of two simulations are presented here for the sake of illustration and discussion. Each simulation uses the model I spread function. The first is a simulation of the 22 June 1970 storm day and the second is a simulation of the 24 July 1970 storm day.

Figure 5.4.1 shows the isohyet plot of the observed 22 June 1970 storm day along with the estimated spatial correlation and variance function plots. This illustration shows the sparsity of rainfall which was confined to the southern and northwestern parts of the Walnut Gulch watershed. The maximum rainfall depth was 8 mm, and about 70% of the catchment received no rainfall.

Figure 5.4.2 shows the results from the Model I simulation (i.e., a single realization) of 22 June 1970. The simulated total depth field is also quite sparse. The maximum rainfall depth is also about 10 mm. Figure 5.4.3 shows a comparison among the observed storm day spatial correlation, the analytical model spatial correlation, and the correlations from multi-level samplings of the simulated field. Because the $\rho(\nu)$ model was fitted to $\hat{\rho}(\nu)$, the theoretical and observed spatial correlations agree quite well. The correlation of the Level III simulation is

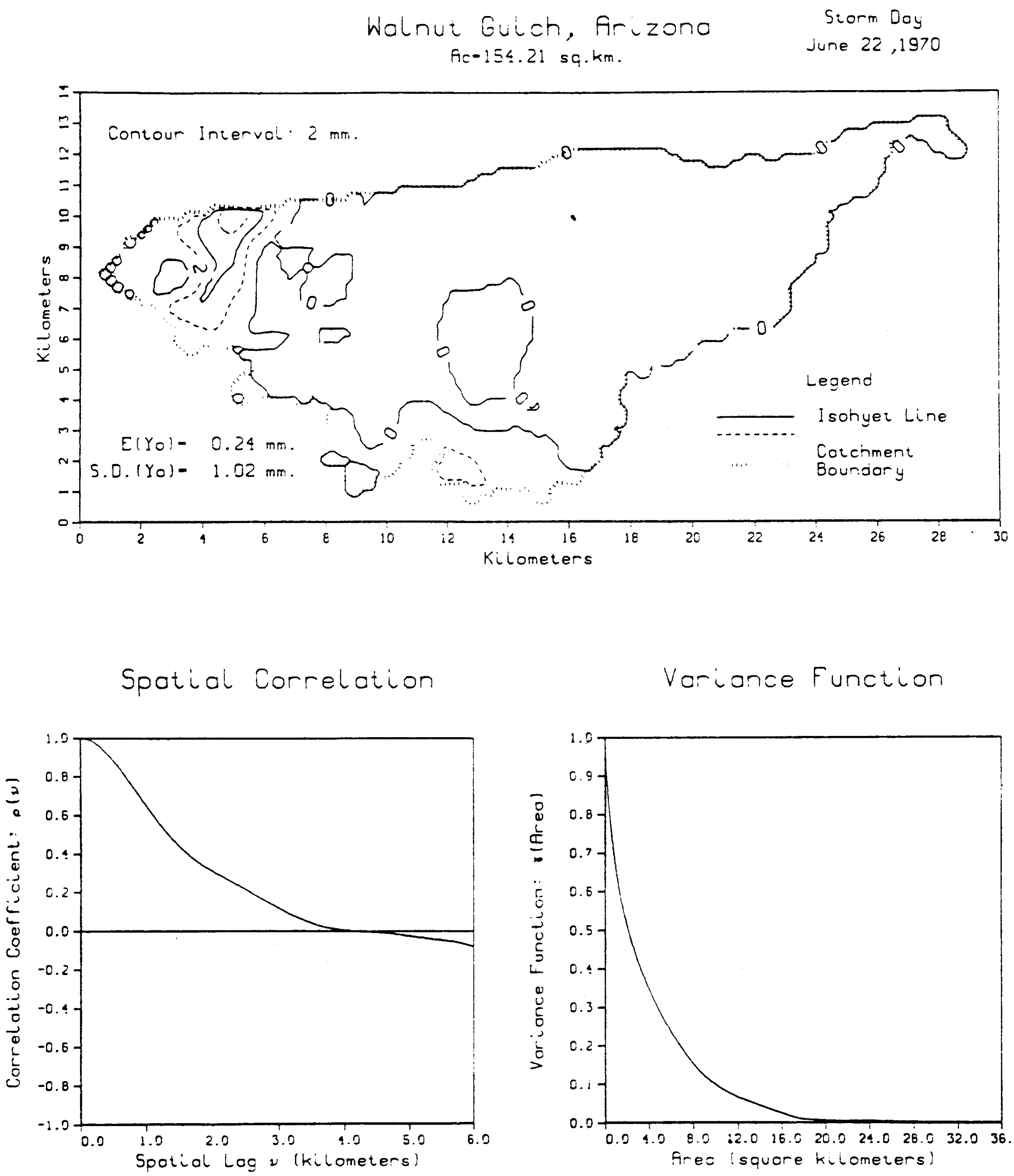
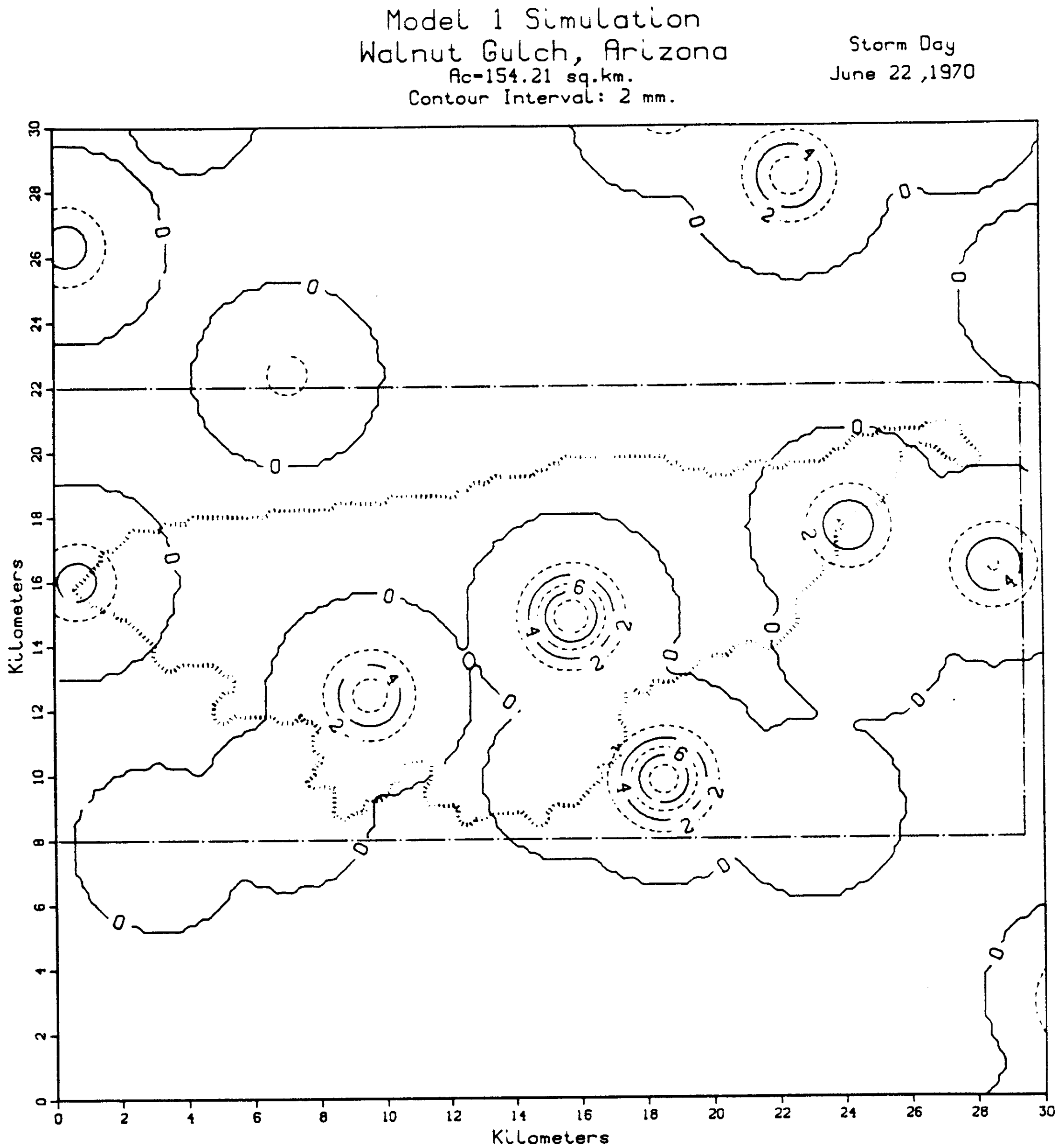


Fig. 5.4.1 Observed 22 June 1970 Storm Day



SAMPLING AREA LEGEND

Level III Area: (30.0 km. by 30.0 km.)
Level II Area: (29.4 km. by 14.0 km.)
Level I Area: (154.21 sq. km.)

Fig. 5.4.2 Model 1 Simulation of 22 June 1970 Storm Day; $a_1 = 0.566$ km^{-1}, $\lambda_1 = 0.019\ km^{-2}$, $E[\alpha_1] = 2.860$ mm

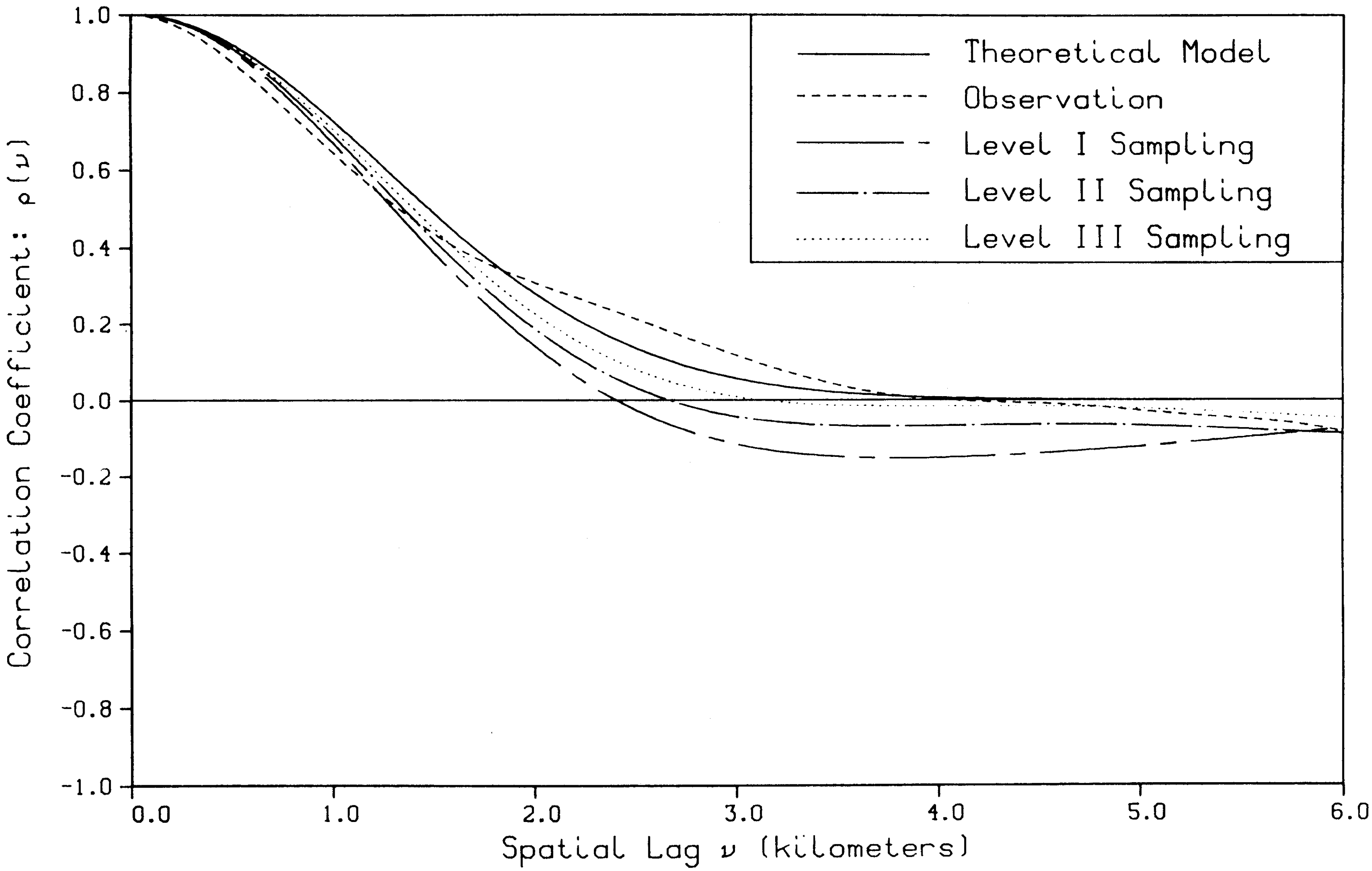


Fig. 5.4.3 22 June 1970 Theoretical, Observed and Simulated Spatial Correlation

closest to the theoretical because of its large area.

Figure 5.4.4 shows the same comparison among the five variance function curves for the 22 June 1970 storm day. This illustration shows that again, the best agreement with the infinite domain variance function is with the estimate variance function from the Level III simulated field. The apparent erratic behavior of the Level I function is due to a combination of chance and restricted area. Therefore, the Level I variance function is not truly significant.

Figure 5.4.5 compares the spatial distribution of total rainfall among the theory, observation, and simulation. We see that the theory and Level III simulations both underestimate the total dry area by about 10% and overestimate the maximum total depth by only 3 mm. Because of chance and reduced sampling area, the Level I and Level II underestimate the total dry area by about 30%. However, each overestimated the maximum total depth by only 3 mm. In general, the agreement among the observation, theory and simulation is quite good.

Figure 5.4.6 shows the 24 July 1970 storm day isohyets of observed total rainfall plus the associated spatial correlation and variance functions. The cellular structure of the isohyets reflects the stationarity of actual convective cells which developed during the course of this event. The maximum rainfall is about 33 mm in the north central part of the basin, and only 6 km southwest of the local maximum there was no rainfall.

Figure 5.4.7 shows the Model I simulation (i.e., a single realization) of the 24 July 1970 storm day. Figure 5.4.8 presents a comparison among the five spatial correlation functions. The agreement among the

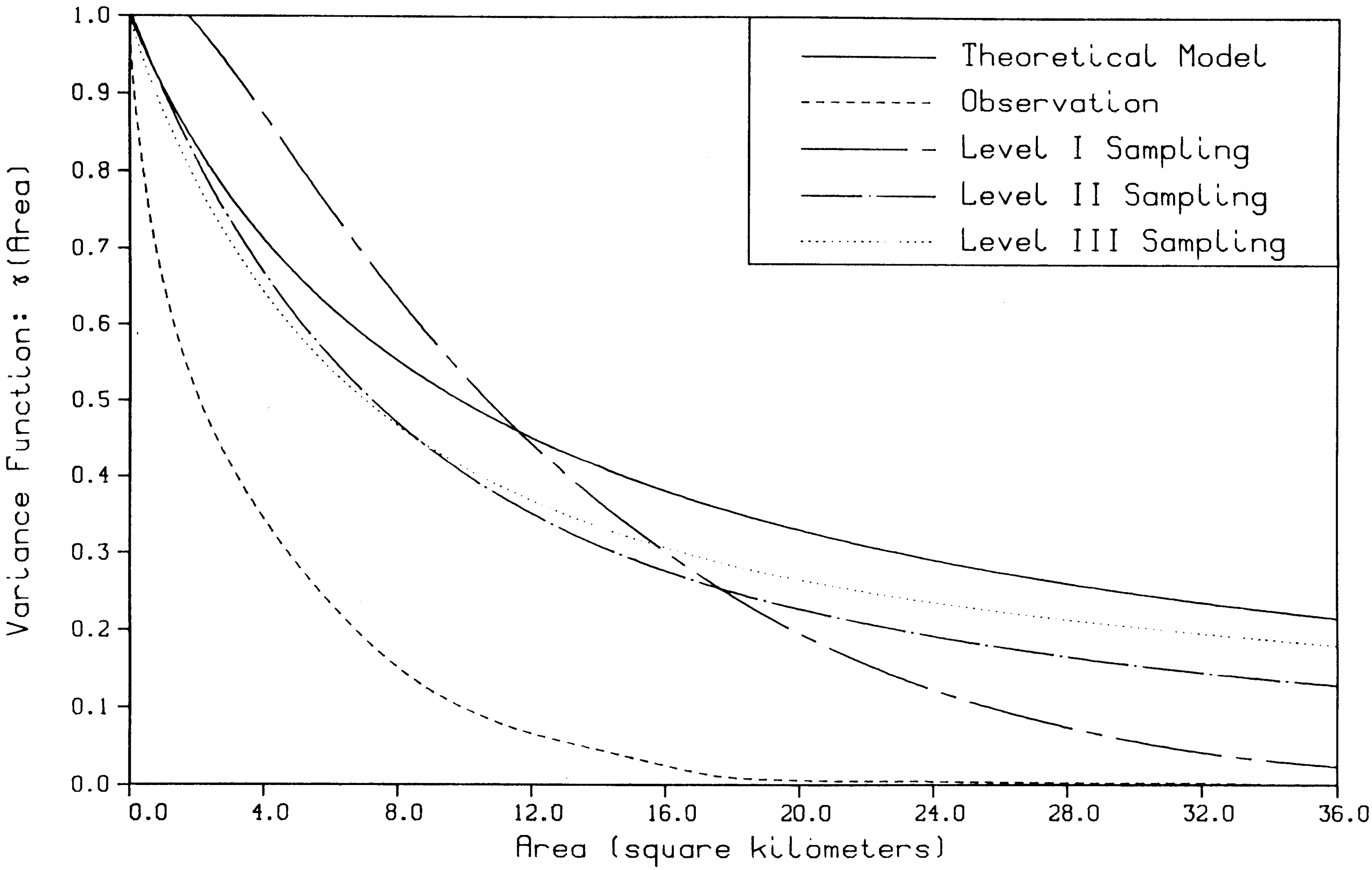


Fig. 5.4.4 22 June 1970 Theoretical, Observed and Simulated Variance Functions

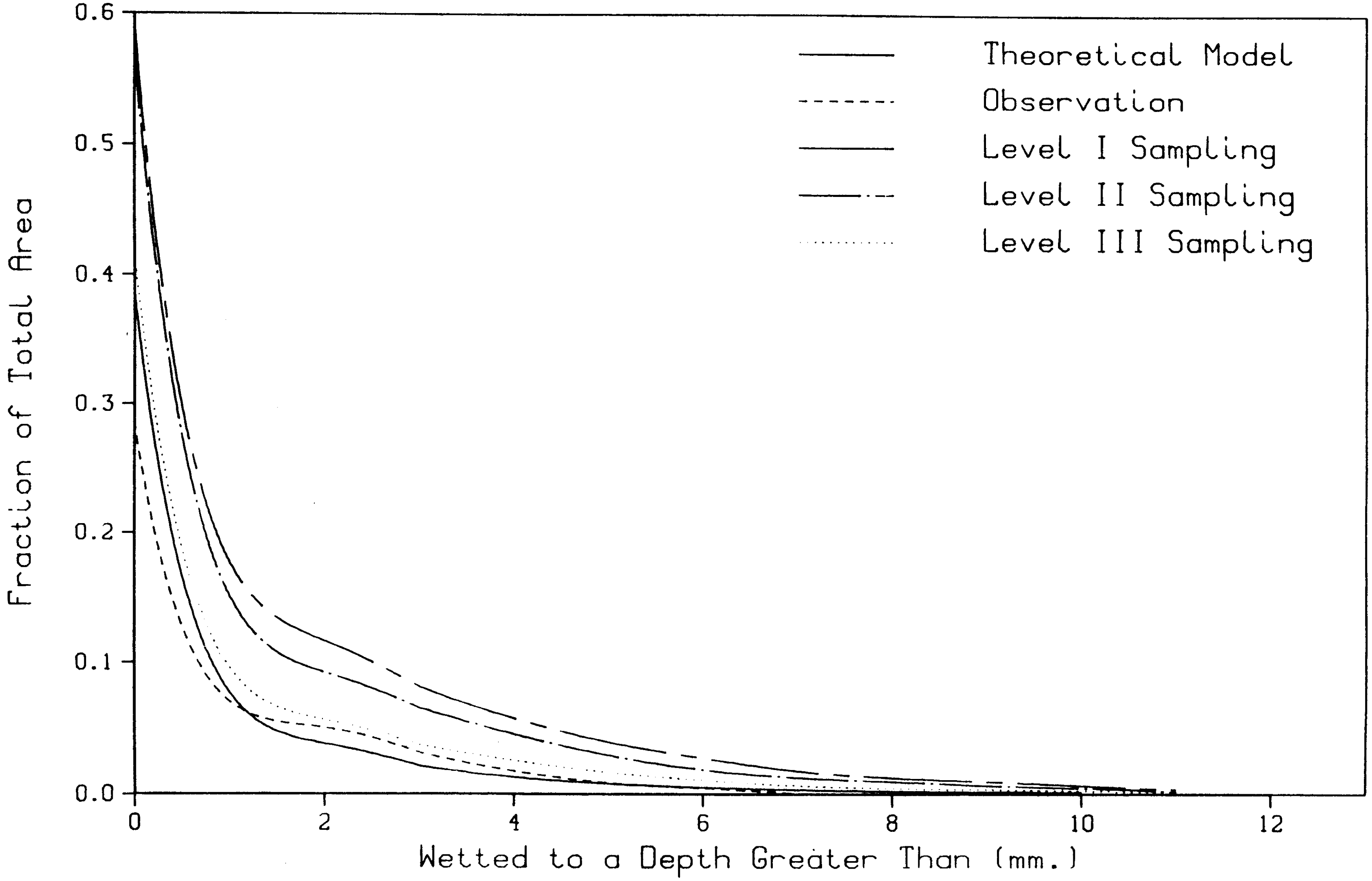
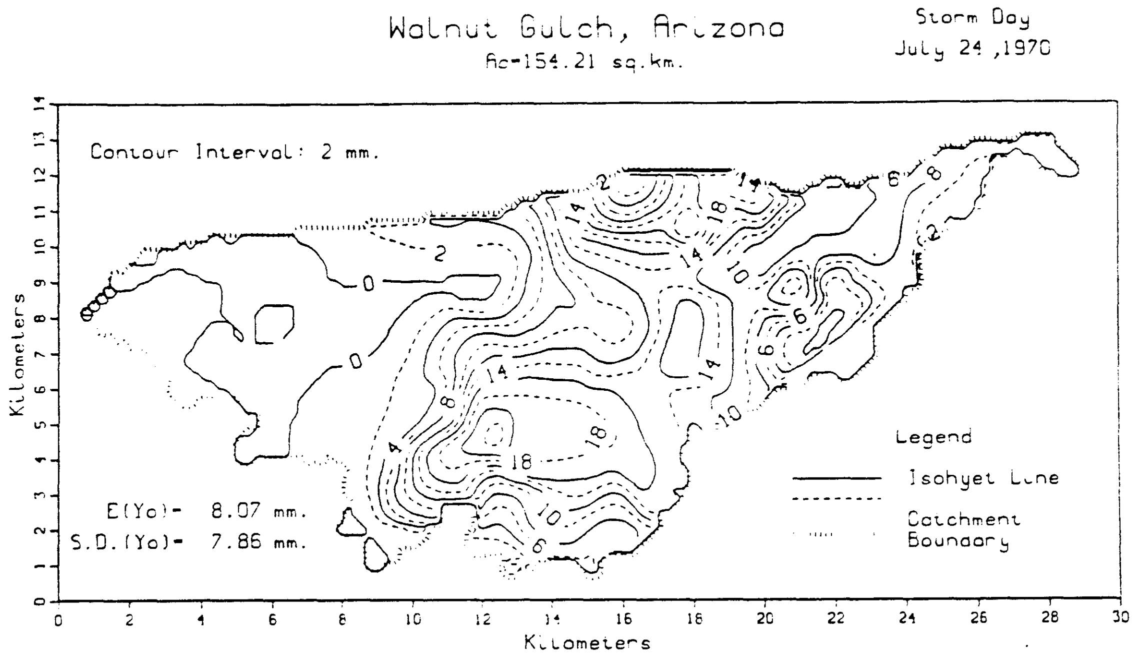
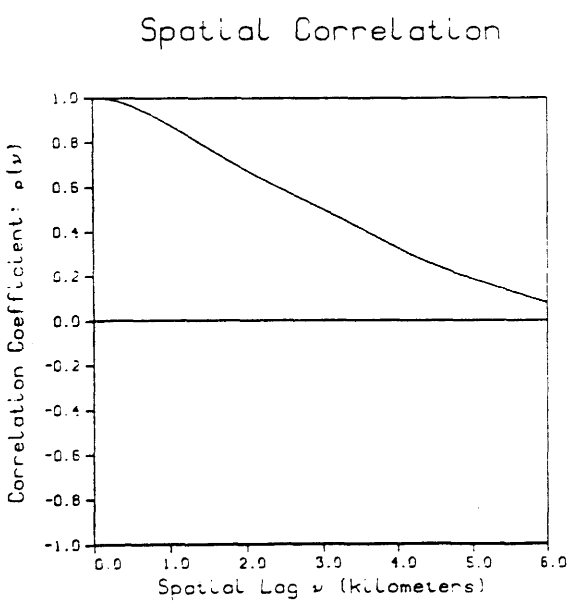


Fig. 5.4.5 22 June 1970 Theoretical, Observed and Simulated Spatial Distribution of Total Storm Depth





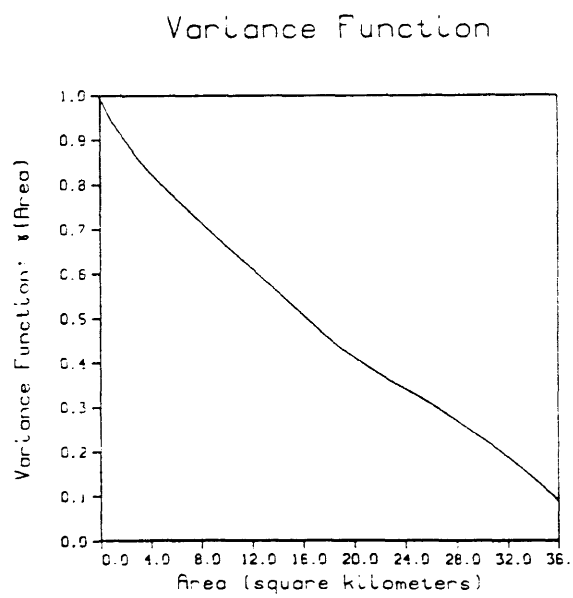
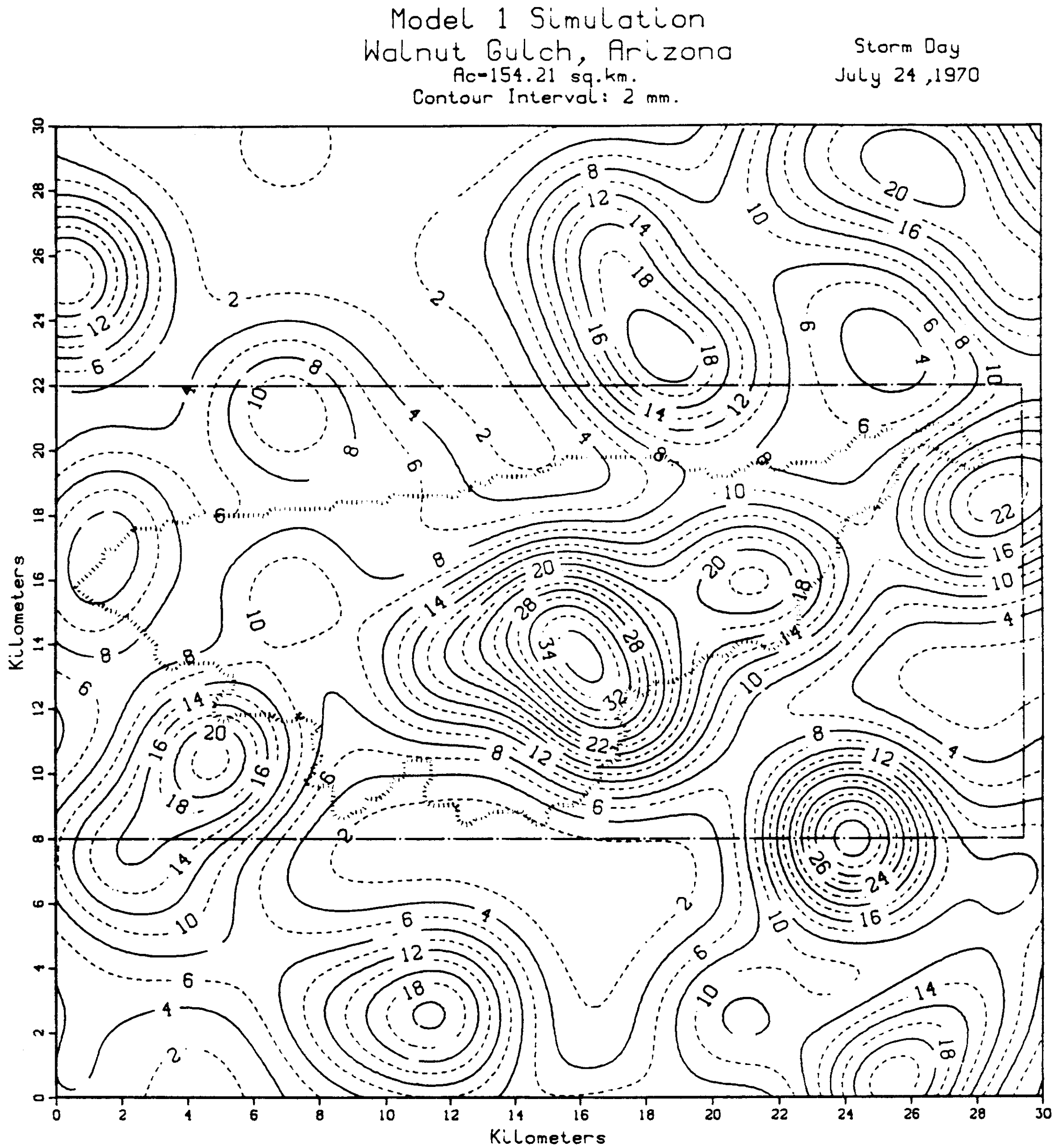


Fig. 5.4.6 Observed 24 July 1970 Storm Day



SAMPLING AREA LEGEND

Level III Area: (30.0 km. by 30.0 km.)
Level II Area: (29.4 km. by 14.0 km.)
Level I Area: (154.21 sq. km.)

Fig. 5.4.7 Model 1 Simulation of 24 July 1970 Storm Day;
a_1 = 0.275, km^{-1}, λ_1 = 0.063 km^{-2}, $E[\alpha_1]$ =

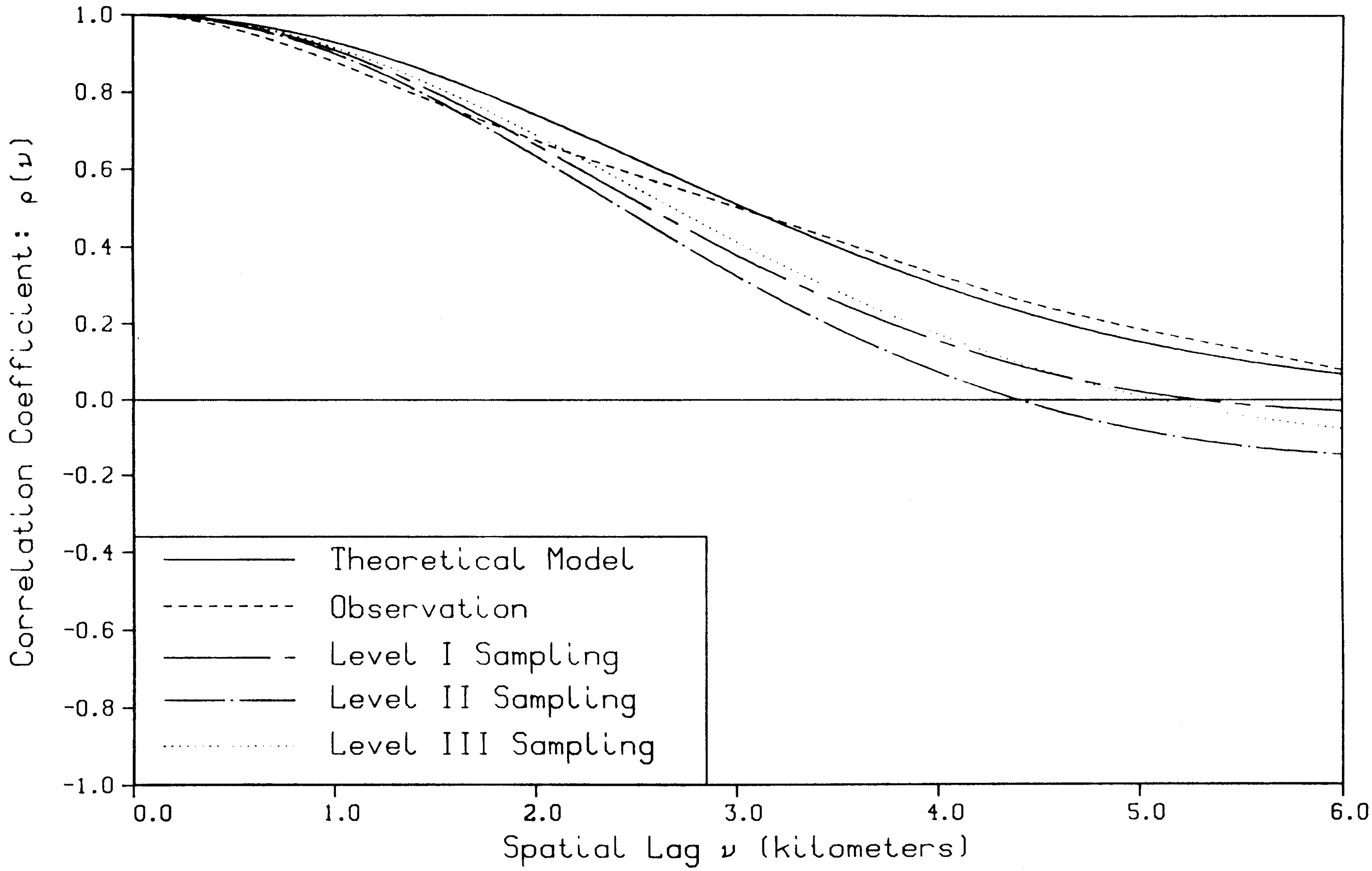


Fig. 5.4.8 24 July 1970 Theoretical, Observed and Simulated Spatial Correlation

theory, storm day observation, and each Level of simulation is very good, reflecting the stability of the correlation calculation for more continuous fields. The Level I spatial correlation does become negative at about ν = 4 km which is expected due to the small sample area.

Figure 5.4.9 shows the variance function for the observed storm day, theory and the simulated random fields. Here, as expected, the theory and the Level III simulated random field agree reasonably well. The close agreement between the Level I variance function and the observed variance function must be judged fortuitous.

Figure 5.4.10 illustrates the five separate spatial distribution curves. The theory and each simulation level underestimates the observed dry fraction of catchment area by about 10%. The theory overestimated the maximum total depth by about 10 mm. The simulation, however, overestimated the actual maximum storm day depth by 6 mm.

By chance, the most pronounced storm cell is located near the center of the Level I simulated field. Because of the reduced sampling area, we conclude that the Level I spatial distribution curve is not significant. In general, we see that the theory and the observation agree quite well.

An important issue to bear in mind is that the observations and each simulation represent only a single realization of a random process. The only definitive comparison in these plots is that between the theory and the Level III simulation which serves as a check on the accuracy of the simulation algorithm. Simulation has further utility (as shown later) in exploring the effect of interpolation and it allows visual appreciation of the realization of model storm patterns and of the effect of parameter change on these patterns.

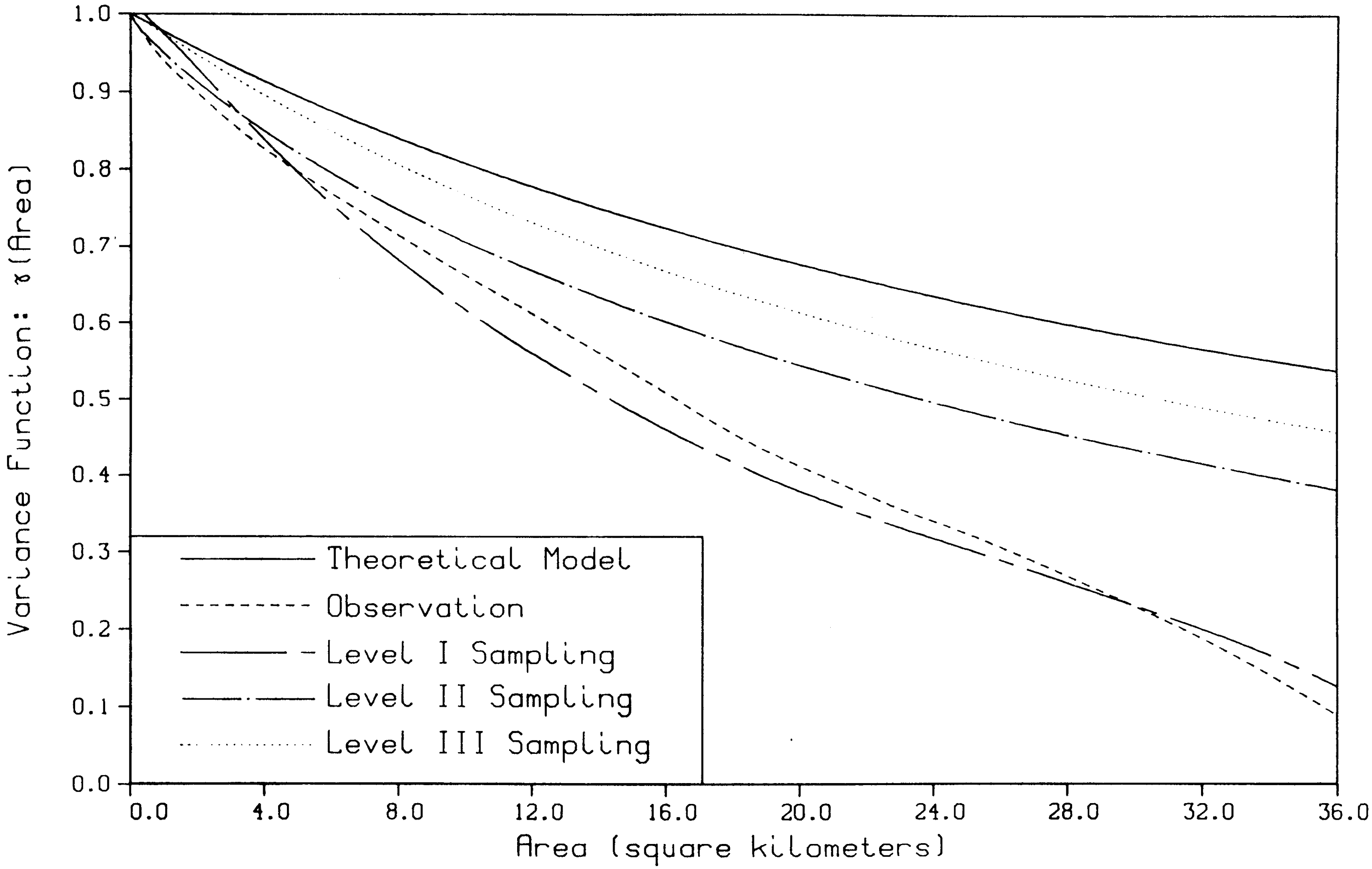


Fig. 5.4.9 24 July 1970 Theoretical, Observed and Simulated Variance Functions

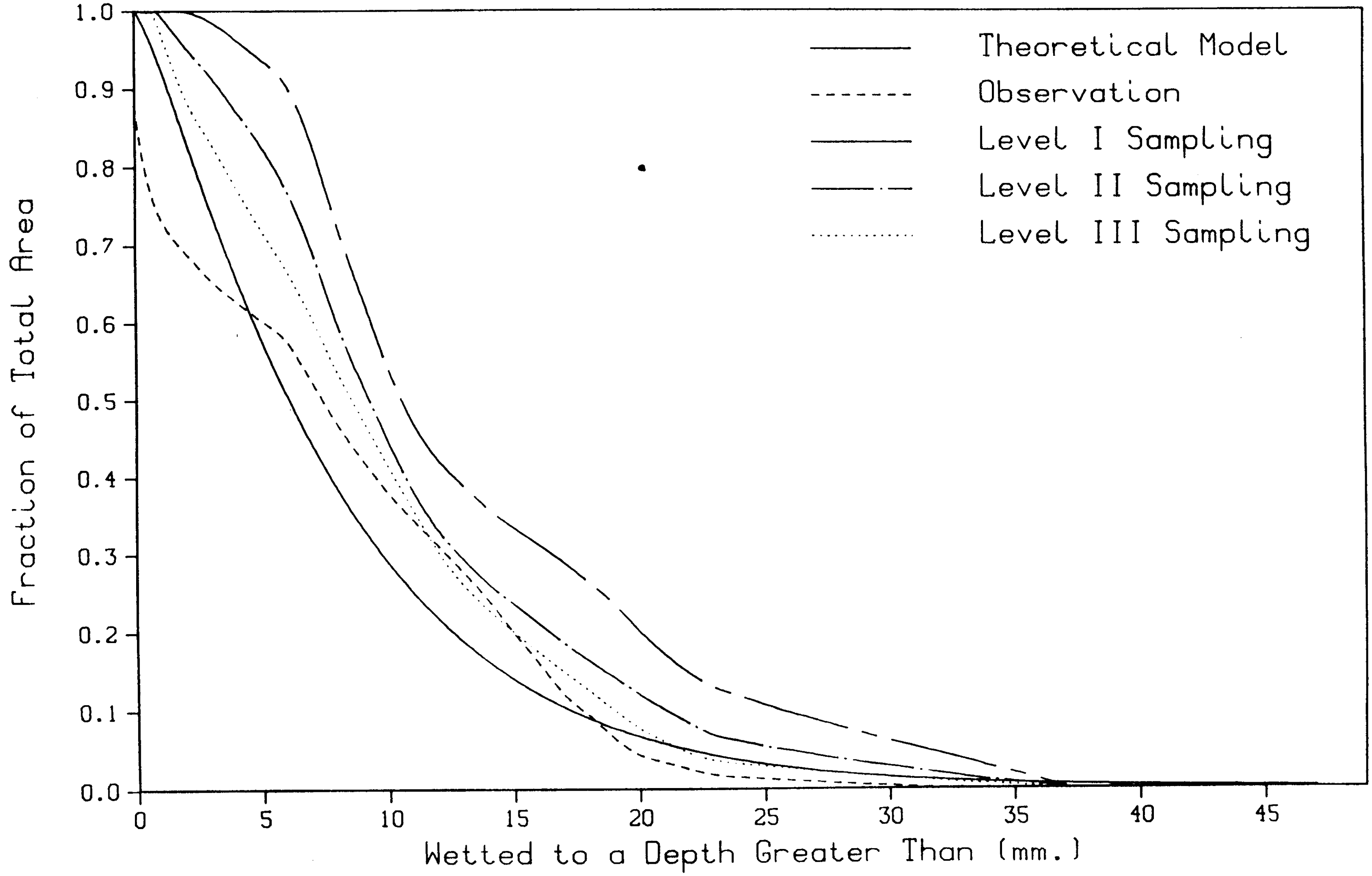


Fig. 5.4.10 24 July 1970 Theoretical, Observed and Simulated Spatial Distribution of Total Storm Depth

In a practical sense, however, simulation must be used sparingly due to the large computational effort involved. Depending on the number of storm cells to be generated, a single simulation may require several hours of computer central processing unit (cpu) time.

5.5 Simulation Post-Processing

Following the numerical simulation and the sampling process, several computer programs must be executed for purposes of further analysis, creation of data tables and graphical presentations of these analyses.

The first program is called TABLE5.FORTRAN. The purpose of TABLE5.FORTRAN is to tabulate the Level I, Level II, and Level III analyses into three separate data tables. Each of these tables is nicely formatted and is suitable for presentation.

The next post-simulation program to be run is called SIMPLOT.FORTRAN. This program plots the total depth isohyets of the Level III simulation random field. Because both Level II and Level I share Level III nodes, only their respective boundaries are plotted in order to illustrate their limits. SIMPLOT.FORTRAN uses the coarse mesh output file from the execution of SIMWET.FORTRAN program as its input file.

The next program is called SIM_STORMDAY.FORTRAN. The program plots the Level I total depth isohyets, the spatial correlation and the variance function results in the same format as STORMDAY.FORTRAN plots the observed storm day results. These two plots may be used for graphical comparison with the parent storm day. The program requires two output files from SIMWET.FORTRAN and one output file from SIMCRGAM.FORTRAN as input files.

The next program to be executed is called WETPLOT.FORTRAN. The program uses the three estimated model parameters to calculate the theoretical spatial distribution of total storm day depth for the infinite domain random field. WETPLOT.FORTRAN next plots the spatial distribution of total simulated storm depth sampled from Level I, Level II and Level III. Superimposed on the same plot is the actual storm day spatial distribution of total storm depth and the analytical model spatial distribution curves.

WETPLOT.FORTRAN requires three separate input files. One file must contain the data from the observed storm day processing results. The second two are output files created by SIMWET.FORTRAN.

Another program must now be executed. CORR_GAM_PLOT.FORTRAN determines the analytical expressions for the spatial correlation or variance function for the model of interest. The program then plots the analytical curve, the actual storm day curve, and the Levels I, II and III simulation curves. The program prompts the analyst for his choice of the variance function or the spatial correlation curve, and for the placement of the plot legend.

Execution of this program requires three separate input files. The first input file must contain the corresponding observed storm day data table. The second input file is one of the output files from SIMWET.FORTRAN and the third is an output file created by SIMCRGAM.FORTRAN.

There are two optional post simulation programs which the analyst may elect to execute. Each program is used to assess the adequacy of the bivariate surface interpolator. This analysis will be presented in Chapter 7. These programs are only optional in the sense that should the analyst choose to perform this analysis, the programs must be executed before beginning the next simulation.

The first program is called SIMINTRP.FORTRAN. This program creates an interpolation of a Level I simulation field. This program requires the output file from SIMWET.FORTRAN that contains the "simulation raingage" data. This data set consists of the 93 simulation node points which correspond to the spatial coordinates of the 93 raingages belonging to the Walnut Gulch raingage network. The interpolated field is analyzed in the same way the Level I is analyzed. The interpolated "simulation" isohyets are plotted along with the spatial correlation and the variance function sampling results. This plot may be graphically compared with that resulting from SIM_STORMDAY.FORTRAN.

The second optional post-simulation program is called SIMCORR.FORTRAN. SIMCORR.FORTRAN determines the correlation between the Level I simulated random field and the field created by interpolating the 93 "simulation raingages". More will be said about this program in Chapter 7. This program requires two input files, each of which are output files from SIMWET.FORTRAN.

Both SIM_STORMDAY.FORTRAN and SIMCORR.FORTRAN will be discussed further in Chapter 9.

CHAPTER 6

Model Evaluation

6.1 Goodness of Fit

We choose to evaluate each model by comparing the storm day fitted analytical distribution of storm day rainfall depth with the observed storm day depth using a split sample.

The fraction of the total catchment area that is wetted to a depth greater than y is given by the probability P that the point depth Y exceeds y. This is obtained by

$$P[Y > y] = 1 - \int_0^y f_y(x)\, dx \qquad (6.1.1)$$

Of particular interest is the fraction of the catchment area that remains dry during the occurrence of a local event. We have mentioned that "dry" is defined to be $Y \leq 0.01$ mm (the minimum depth filter imposed on both the simulation and the observed storm day random fields). The dry fraction is described by

$$P(dry) = \int_0^{0.01} f_Y(x)\, dx \qquad (6.1.2)$$

The probabilities defined by equation 6.6.1 and unique to each model are compared with the observed values from the actual storm days by measuring the standard square error between them. Two computer programs are developed to accomplish part of this comparison. The first is called MODEL_EVAL.FORTRAN and the second is called called OBS_EVAL.FORTRAN.

These programs perform an averaging process in the following way. The archive of observed spatial distribution of storm day rainfall is recorded by integer mm of depth ranging from Y > 0 through the maximum depth recorded for the particular storm day as well as the estimate for the fraction of dry area in the catchment. By limiting the cross evaluation to integer depths we can assess each model in turn.

OBS_EVAL.FORTRAN reads in the eight archive files created by the storm day processing described in a companion volume. Storm day by storm day, the program examines the spatial distribution curves and after the last discrete storm day spatial distribution curve is read in, the program calculates the expected value and variance of the wetted fraction of total catchment area at integer depths y_i to a maximum of 50 mm. These results are written to a file and set aside. The equations used are:

$$E\left[\frac{A_{c_w}(Y \leq y_i)}{A_c}\right] = \frac{1}{n_j} \sum_{j=1}^{n_j} \frac{A_{c_{w_j}}(Y \leq y_i)}{A_c} \qquad (6.1.3)$$

and

$$VAR\left[\frac{A_{c_w}(Y \leq y_i)}{A_c}\right] = \frac{1}{n_j - 1} \sum_{j=1}^{n_j} \frac{A^2_{c_{w_j}}(Y \leq y_i)}{A_c} - E^2\left[\frac{A_{c_w}(Y \leq y_i)}{A_c}\right] \qquad (6.1.4)$$

where n_j = number of storm days receiving a depth equal to at least y_i mm.

MODEL_EVAL.FORTRAN does something similar. This program reads the data file called UNBIASED_MODEL_PARAMETERS.DATA. Storm day by storm day, the three model parameters for each of the three models are used by three separate analytical expressions which evaluate equations 6.1.1 and 6.1.2

by numerical integration. These results are written to a file and set aside.

In this split sample assessment, the average "spatial distribution curve" is developed for the observed storm day records for the period 1970-1973. In this sample, there are 226 events. Model "spatial distribution curves" are developed in the same way using the parameters estimated from the 202 storm days in the 1974-1977 period. Figures 6.1.1, 6.1.2 and 6.1.3 show this split sample comparison between model and observation for each of the three models in turn. The Model I and Model II spatial distribution curves are identical because the Model II density function $f_Y(y)$ was approximated by the Gamma function, which is the analytical solution for the Model I pdf. In interpreting these split sample comparisons we should remember that the moments of the observations at discrete depths of y_i are compared with the moments of $P[Y > y_i]$ at discrete depths of y_i. The comparisons are thus only demonstrative of "goodness of fit" to $f_Y(y)$.

Table 6.1.1 shows the results of the standard square error measurement between the observed and model curves and also the magnitude of the difference between the observed P(dry) and the model P(dry). The former is calculated in the usual way where

$$\text{std. sq. err.} = \sum_{i=0}^{50\text{mm}} \left[\frac{A_{c_w}}{A_c} (Y \leq y_i) - P(Y \leq y_i)\right]^2 \qquad (6.1.5)$$

and the latter

$$\text{dry error} = \frac{A_{cw}(Y=0)}{A_c} - P[\text{dry}] \qquad (6.1.6)$$

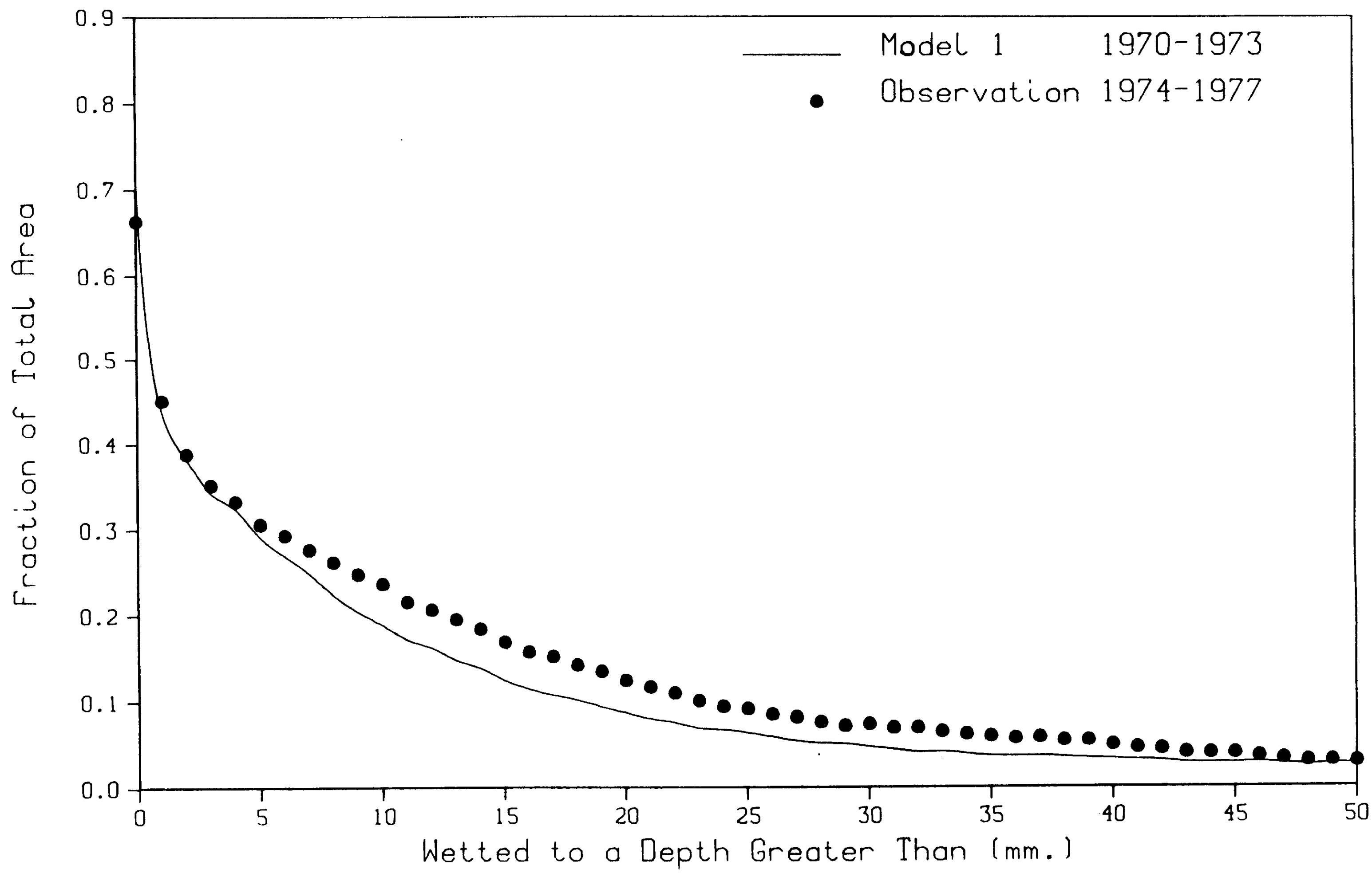


Fig. 6.1.1 Model I Split Sample Evaluation

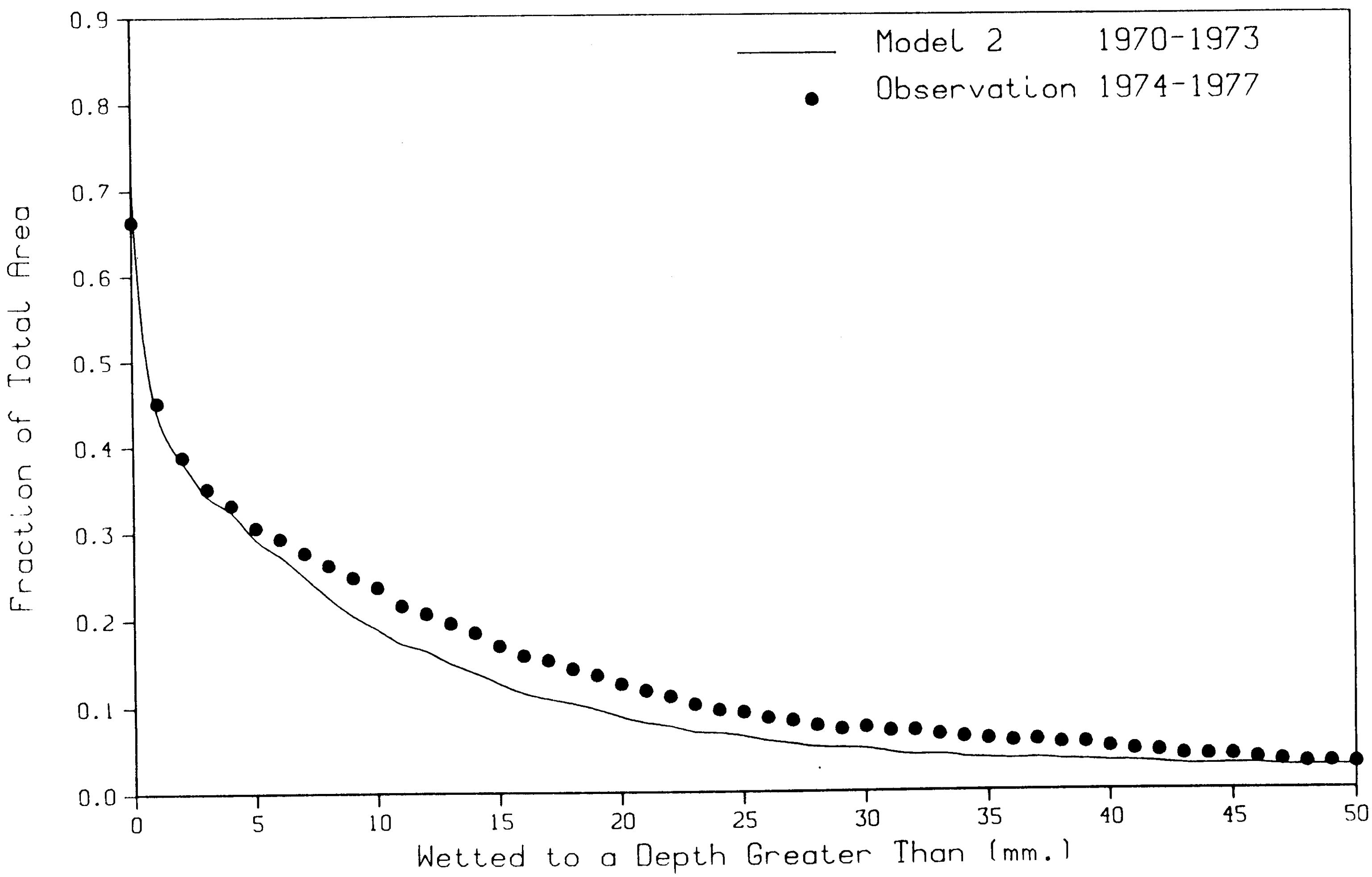


Fig. 6.1.2 Model II Split Sample Evaluation

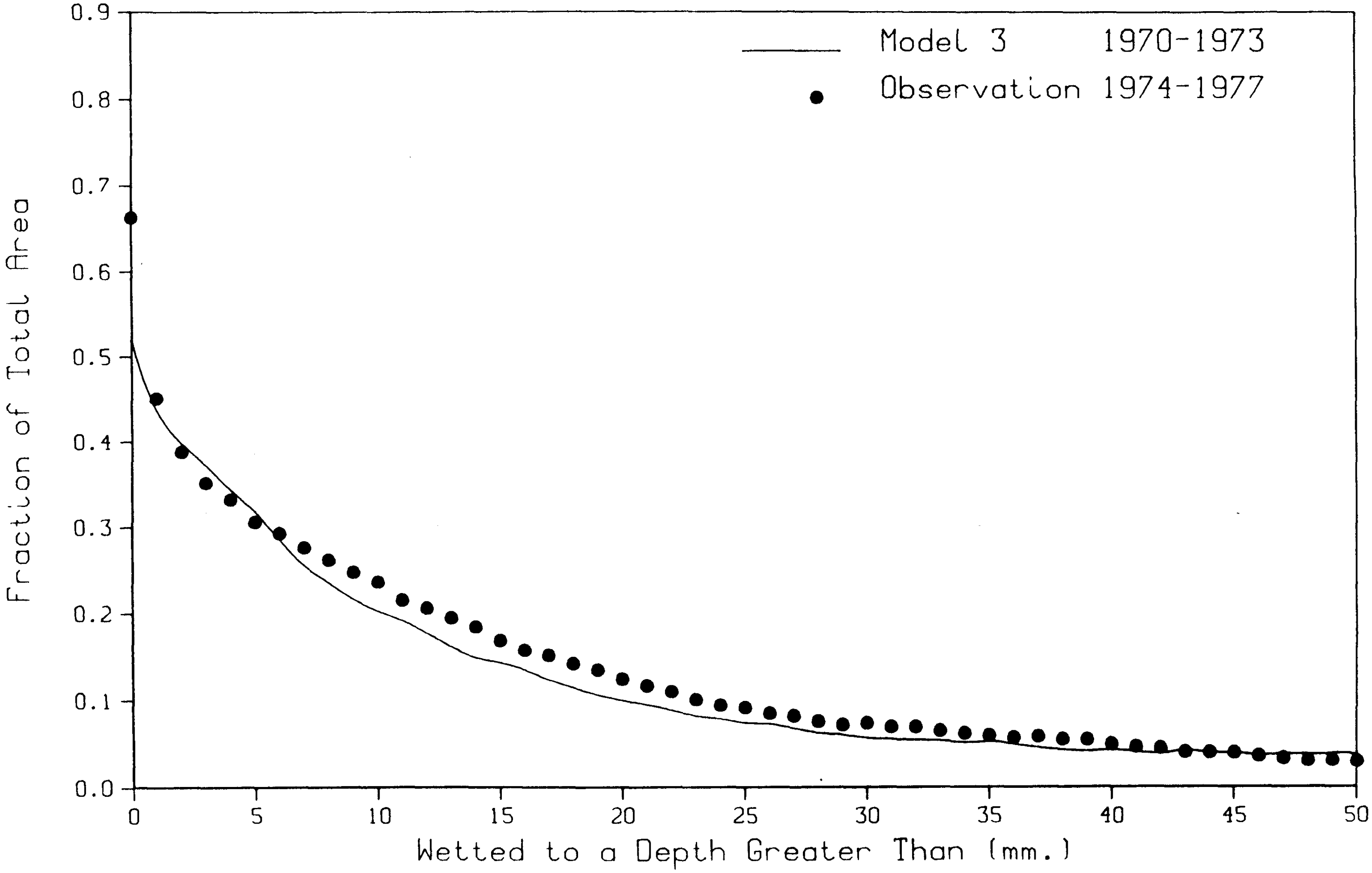


Fig. 6.1.3 Model III Split Sample Evaluation

Table 6.1.1

Split Sample Goodness of Fit Assessment

	Stand. Sq. Error
Model I	0.043
Model II	0.042
Model III	0.036

	Dry Fraction Error
Model I	0.049
Model II	0.051
Model III	-0.139

The results suggest that each model fits the observations well with the exception of the P(dry) prediction of Model III. This demonstrates convincingly that the method of moments does an excellent job of fitting this highly skewed distribution.

6.2 Variation among Storms

The previous goodness-of-fit evaluation was a conditional evaluation. In effect it compared the average of selected sets of observations (given that the member storms had depths at least equal to y) with the average of the theoretical distributions fitted to the individual members of those selected sets. To evaluate the utility of these models for practical use, however, we need to check their marginal performance. That is, how well do the models describe the observed distribution of storm depth as averaged over the entire ensemble at each depth.

The theoretical development presented earlier assumes that the parameters a_i, λ_i and μ_{α_i} are constant over an infinite field. Our observations of 426 separate finite fields, however, show an expected storm-to-storm parameter variability. The various statistical descriptors $g_Y(y)$ of local storm depth utilized thus far are therefore actually con-conditional descriptors $g_Y(y;\ a,\ \lambda,\ \mu_\alpha)$. In practice, however, removal of the conditioning generally requires assumptions of independence and/or relative invariance of one or more parameters. We will work here with Model 1 since it is the most tractable analytically.

We see from the coefficients of variation in Table 4.1.2 that the parameter a is relatively constant, but unless we can also assume λ constant there is no hope of analytically removing the conditioning of Eq. (2.3.2). Although CV[λ] is large [see Table 4.1.2] we can see from Figure 4.1.4 that 85 percent of the values of λ_1 lie between 0 and 0.2 km^{-2}. Taking this as license to assume λ constant we can write, for Model 1 [see Rodriguez-Iturbe, Cox and Eagleson, 1986, section 5]

$$E[Y] = \theta_1 E(\mu_{\alpha_1}) \tag{6.2.1}$$

$$\sigma_Y^2 = \theta_1 E^2(\mu_{\alpha_1})[2 + \theta_1] \tag{6.2.2}$$

and

$$\rho(\nu) = \frac{2\ \exp[-(a_1\nu)^2] + \theta_1}{2 + \theta_1} \tag{6.2.3}$$

The corresponding relations for Models 2 and 3 are difficult to obtain.

Eq. 6.2.3 deserves comment. It represents the spatially averaged correlation function of a nonhomogeneous random field in which the parameters a_1, λ_1 and μ_{α_1} may be regarded as constant in the near field (i.e., individual storms) but variable over the far field (i.e., ensemble of storms). The parameter θ_i is conditioned on constancy of the parameters λ_i and a_i and is thus given by writing Eq. (2.4.2b)

$$\theta = \frac{\pi \, E[\lambda_1]}{2 \, E^2[a_1]} = 0.93 \qquad (6.2.4)$$

while

$$a_1 = E[a_1] = 0.43$$

The marginal expectation of θ_1 is given in Table 4.1.2 as $(\pi/2)E[\lambda_1 a_1^{-2}] = 1.25$.

The non-zero asymptote ($\nu \to \infty$) in Eq. (6.2.3) results solely from the interstorm variability. In Figure 6.2.1 this marginal correlation function (solid line) is compared with the average observed correlation function (solid points) and with Eq. (2.3.7), the conditional correlation function (dashed line).

Having assumed a_1 and λ_1 constant, and noting from Figure 4.1.7 that μ_{α_1} is exponentially distributed, we can use the results of Bhattacharya [1967] to write the marginal density function for Model 1

$$f_Y(y) = E[\mu_{\alpha_1}] \frac{1}{E[\mu_{\alpha_1}] \, \Gamma(\theta_1)} (y/E[\mu_{\alpha_1}])^{[(\theta_1+1)/2]-1} K_{1-\theta_1}[2(y/E[\mu_{\alpha_1}])^{1/2}] \qquad (6.2.5)$$

Once again and for the same reasons, θ_1 is conditional and is calculated as shown in Eq. (6.2.4). Rounding this value off to $\theta_1 \simeq 1.0$ the marginal density function Eq. (6.2.5) reduces to

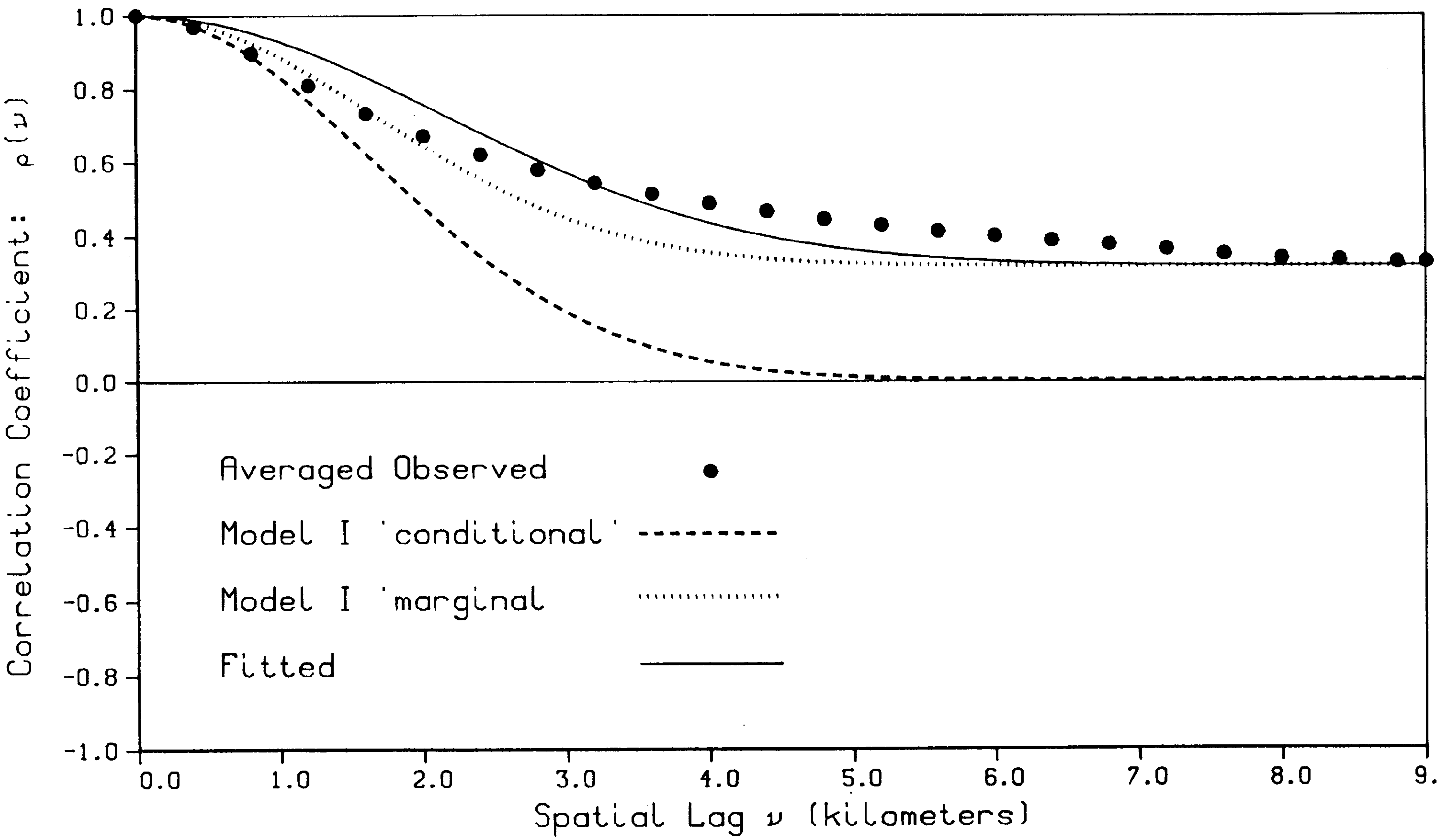


Fig. 6.2.1 Effect of Interstorm Variability on the Spatial Correlation Function: Model 1

$$f_Y(y) = 2\beta_1 K_o[2(\beta_1 y)^{1/2}] \qquad (6.2.6)$$

Using β_1 and not β_2 Eq. (6.2.6) also applies to Model 2, but the form of the conditional density function for Model 3 (Eq. 2.5.20 does not permit analytical derivation of its marginal pdf.

Statistical dependence among the parameters a, λ, μ_α and θ is examined through calculation of the correlation coefficients listed in Table 4.1.2. The principal dependence appears to be between the cell center depth and its radial decay rate.

6.3 Model Evaluation

In Figure 6.3.1 Models 1 and 2 are evaluated through a split-sample comparison of Eq. (6.1.1) with the observations. The latter are averaged at each depth across the 228 member 1974-1977 ensemble. Both the conditional (Eq. [2.3.2]) and the marginal (Eq. [6.2.6]) distributions are presented using parameters obtained from the 1970-1973 ensemble. The large difference between theory and experiment is difficult to explain conclusively but there are two possible causes:

1. The parameter θ which controls the structure of $f_Y(y)$ [see Eq. (6.2.5)] is actually highly variable [see Table 4.1.2 for CV(θ)].

2. Due to the finite storm size some, if not all, of the dry catchment area may be outside the storm coverage of the catchment [see Eagleson and Wang, 1985] and hence be unrepresentative of the storm field. Our estimated parameters are unavoidably contaminated to an unknown degree by this possibility.

The comparison for Model 2 is essentially the same as that for Model 1 because both distributions $f_Y(y)$ are gamma and the observed moments used for fitting are of course identical.

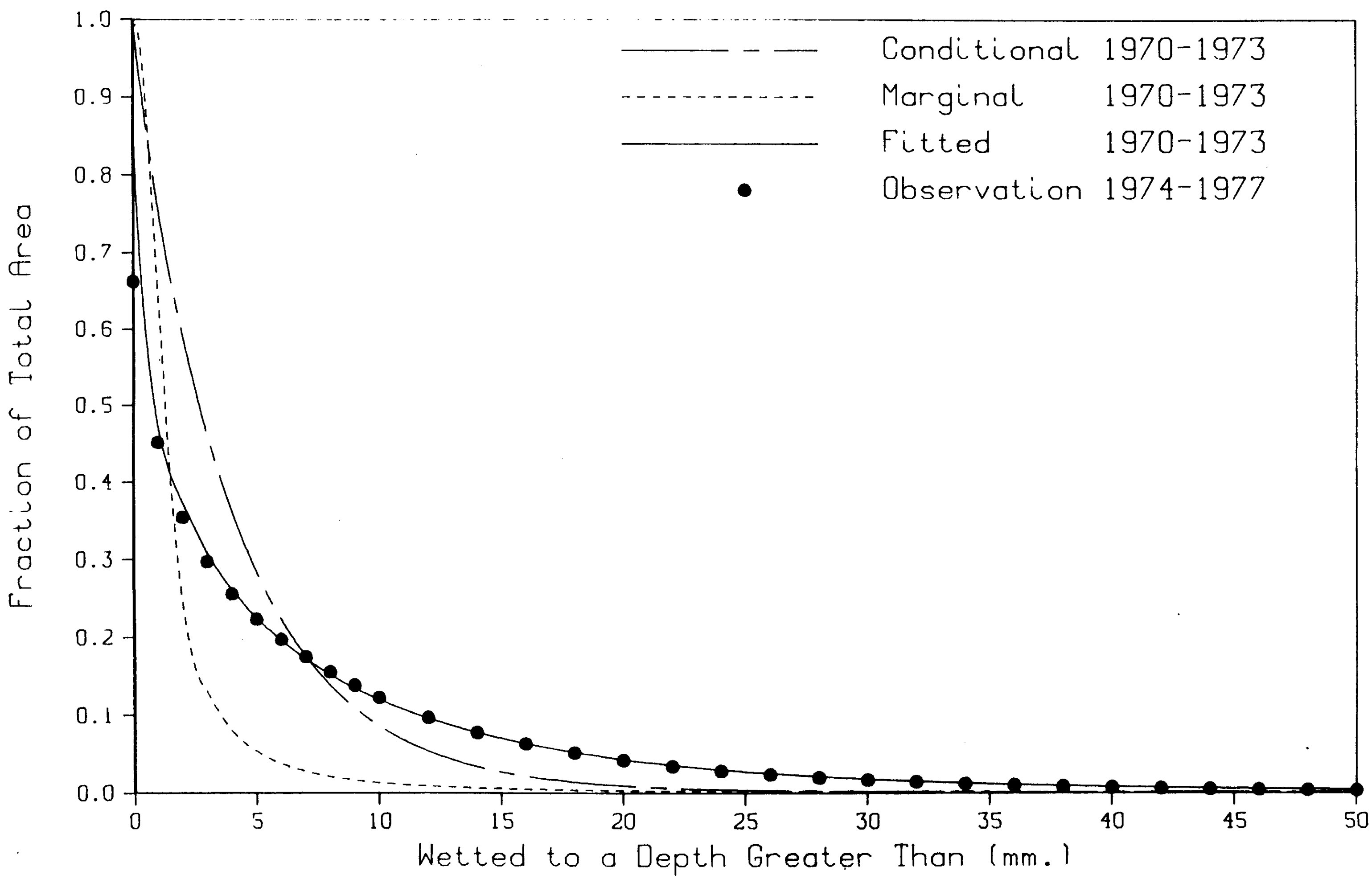


Fig. 6.3.1 Spatial Distribution of Total Storm Depth:
Evaluation of Models 1 and 2

The comparison for Model 2 is essentially the same as that for Model 1 because both distributions $f_Y(y)$ are gamma and the observed moments used for fitting are of course identical.

The evaluation of Model 3 (Figure 6.3.2) is confined to the conditional distribution because of the mathematical difficulty of unconditioning Eq. 2.5.2.

The standard squared error of these analytical distributions is summarized in the first two columns of Table 6.3.1. By this measure, Models 1 and 2 are superior to Model 3.

In the absence of an accurate analytical marginal distribution $f_Y(y)$ we, for practical purposes, fit the ensemble average distribution with the respective conditional distribution given by Eq. (2.3.2). For Model 1 the spatial distribution is then

$$P[Y > 1] = 1 - \frac{\gamma[\theta^*, \beta^* y]}{\Gamma(\theta^*)} \qquad (6.3.1)$$

where $\gamma[\cdot]$ is the incomplete gamma function. The asterisk indicates ensemble average parameters which are estimated using Eqs. (2.3.3) and (2.3.4) as

$$\beta^* = \frac{\hat{E}[Y]}{\hat{\mathrm{var}}[Y]} \qquad (6.3.2)$$

and

$$\theta^* = \frac{\hat{E}^2[Y]}{\hat{\mathrm{var}}[Y]} \qquad (6.3.3)$$

where $\hat{E}[Y]$ and $\hat{\mathrm{var}}[Y]$ are the moments of the ensemble of observations. Using the interpolated values of Table 4.1.3 we find the values of β_1^* and θ_1^* listed in Table 6.3.2. The associated ensemble parameter a_1^*, also

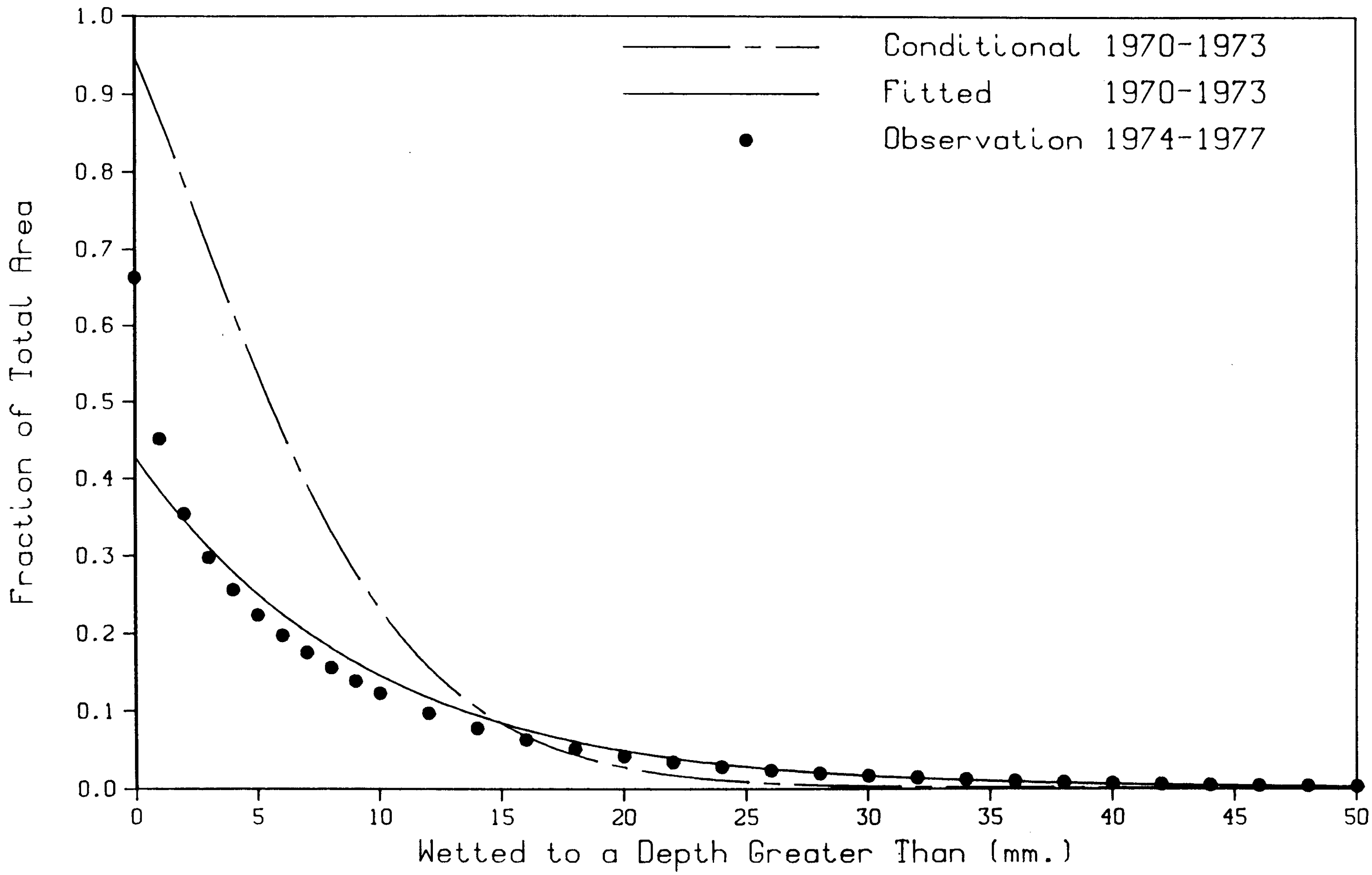


Fig. 6.3.2 Spatial Distribution of Total Storm Depth:
Evaluation of Model 3

given in Table 6.3.2, is given by fitting Eq. (6.2.3) to the averaged correlation function using the conditional value $\theta_1 = 0.93$ as shown by the solid line in Figure 6.2.1. The standard squared error of this fitted correlation function is 0.148.

Similar fitting to the observed moments gives the ensemble parameters θ^* and β^* for Models 2 and 3 listed in Table 6.3.2. Mathematical difficulty prevents determination of a^* in these cases.

Using the ensemble parameters of Table 6.3.2, Models 1 and 2 are compared with observation through Eq. 6.3.1 plotted as the solid line in Fig. 6.3.1. A similar comparison for Model 3 requires numerical integration of Eq. 2.5.2 and is shown as the solid line in Fig. 6.3.2. The standard squared error of these fittings are summarized in the last column of Table 6.3.1 and once again Models 1 and 2 are superior. Of the two we prefer Model 1 because of its analytical tractability in spite of its physically unrealistic correlation function [see Bras and Rodriguez-Iturbe, 1976].

Eq. 6.3.1 deviates from the observations only near $y = 0$. For example, at $y = 0.1$ mm, the limit of resolution of the common metric raingage, observations indicate the "dry" fraction (i.e., where $y < 0.1$ mm) to be about 40%. Eq. (6.3.1), however, gives 28%. This difference is almost certainly due to the incomplete coverage of the catchment by the storm field. The observations of $P[Y < .1 \text{ mm}]$ will reflect the exaggerated dry fraction directly while the fitted density function for the homogeneous infinite field [Eq. 2.3.2] gives a value of $P[Y < .1 \text{ mm}]$ differing from the "true" value only through the second order contamination of the moments by the partial coverage. To the first approximation, therefore, we can designate the difference, 12%, as the average dry percentage due to finite storm size. This apparent confinement of the effect of incomplete coverage to the dry fraction estimation effectively eliminates the coverage problem as one of the two possible causes of the poor agreement of the derived marginal distribution with the observations in Figure 6.3.1.

Defining the area of storm-catchment overlap to be A_{sc} and normalizing by the catchment area A_c we have in this case the marginal expectation

$$E[\frac{A_{sc}}{A_c}] = 1 - 0.12 = 0.88 \quad (6.3.4)$$

Eagleson and Wang [1985] derived the conditional expectation

$$E[\frac{A_{sc}}{A_c} ; \frac{r_c}{r_s}] = 1/[(r_c/r_s) + 1]^2 \quad (6.3.5)$$

where r_c = radius of equivalent circular catchment

r_s = radius of "characteristic" circular storm

Equating (6.3.4) and (6.3.5) gives a first order estimate of the average size of the air mass thunderstorm fields studied, r_c = 100 km.

TABLE 6.3.1

Numerical Evaluation of Models

Standard Squared Error of Spatial Distribution

	Conditional	Marginal	Fitted
Model 1	0.423	0.393	0.037
Model 2	0.447	0.393	0.037
Model 3	1.039	-----	0.067

TABLE 6.3.2

Ensemble Average Parameters

	Model 1	Model 2	Model 3
β^*	0.01 mm^{-1}	0.005 mm^{-1}	0.14 mm^{-1}
θ^*	0.28	0.035	2.81×10^{-3} mm^{-2}
a^*	0.33 km^{-1}	†	†
$\lambda^* = 2a^{*2}\theta^*/\pi$	0.02 km^{-2}	†	†

†Not calculated due to mathematical difficulty of finding equivalent of Eq. (6.2.3) for Models 2 and 3.

CHAPTER 7

Assessment of the Surface Fitting Interpolator

7.1 An Assessment of the Bivariate Surface Fitting Interpolator

This chapter focuses on an assessment of the bivariate surface fitting interpolator. The interpolator creates a smooth and continuous surface from a set of irregularly spaced data points. An extended discussion about the working of the interpolator is presented in a companion volume.

The bivariate surface interpolator is used extensively throughout the course of this study. For this reason, we feel that an evaluation of it is necessary. The filtering characteristics of the interpolator are especially important because of their effect on the estimates of the moments and the covariance structure of the random field. We have already seen in Table 4.1.3 that the ensemble moments of observed and interpolated point storm depths are essentially identical.

Two computer codes are developed to perform this analysis and were mentioned briefly in Chapter 5. They are SIMINTRP.FORTRAN and SIMCORR.FORTRAN. SIMINTRP.FORTRAN provides a graphical comparison between the Level I simulation isohyet and an interpolation of the "raingage simulations". The program also samples the interpolated simulation field for the spatial correlation and the variance function. Figure 7.1.1 shows the Model I (Level I) simulated isohyets, spatial correlation and variance function for the event of 24 July 1970. Figure 7.1.2 shows the same things as determined from the interpolated raingage simulations. By visual comparison the results are essentially identical. This test is repeated for the storm day 22 June 1970 in Figures 7.1.3 and 7.1.4 with the same result.

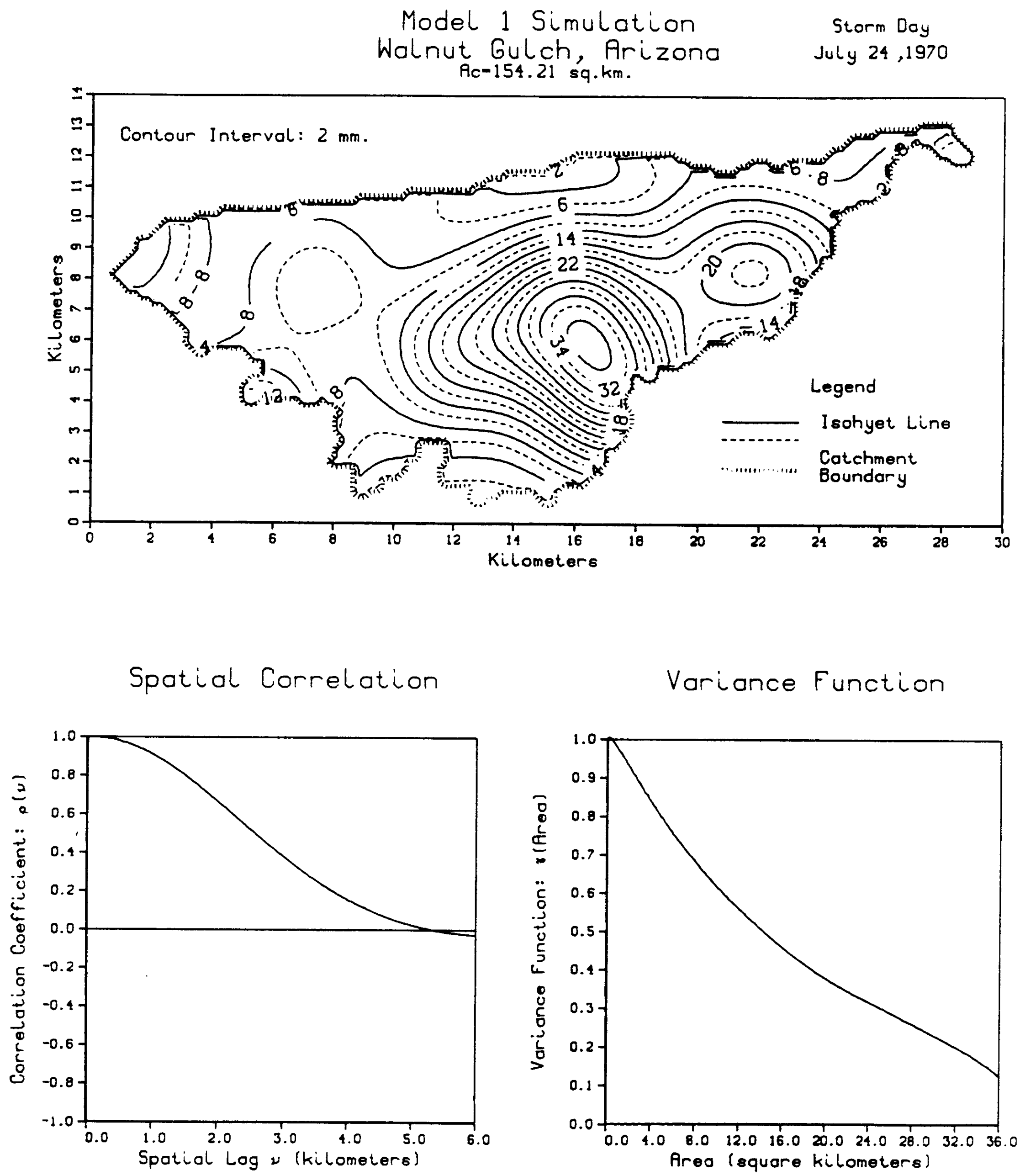


Fig. 7.1.1 Simulated 24 July 1970 Storm Day

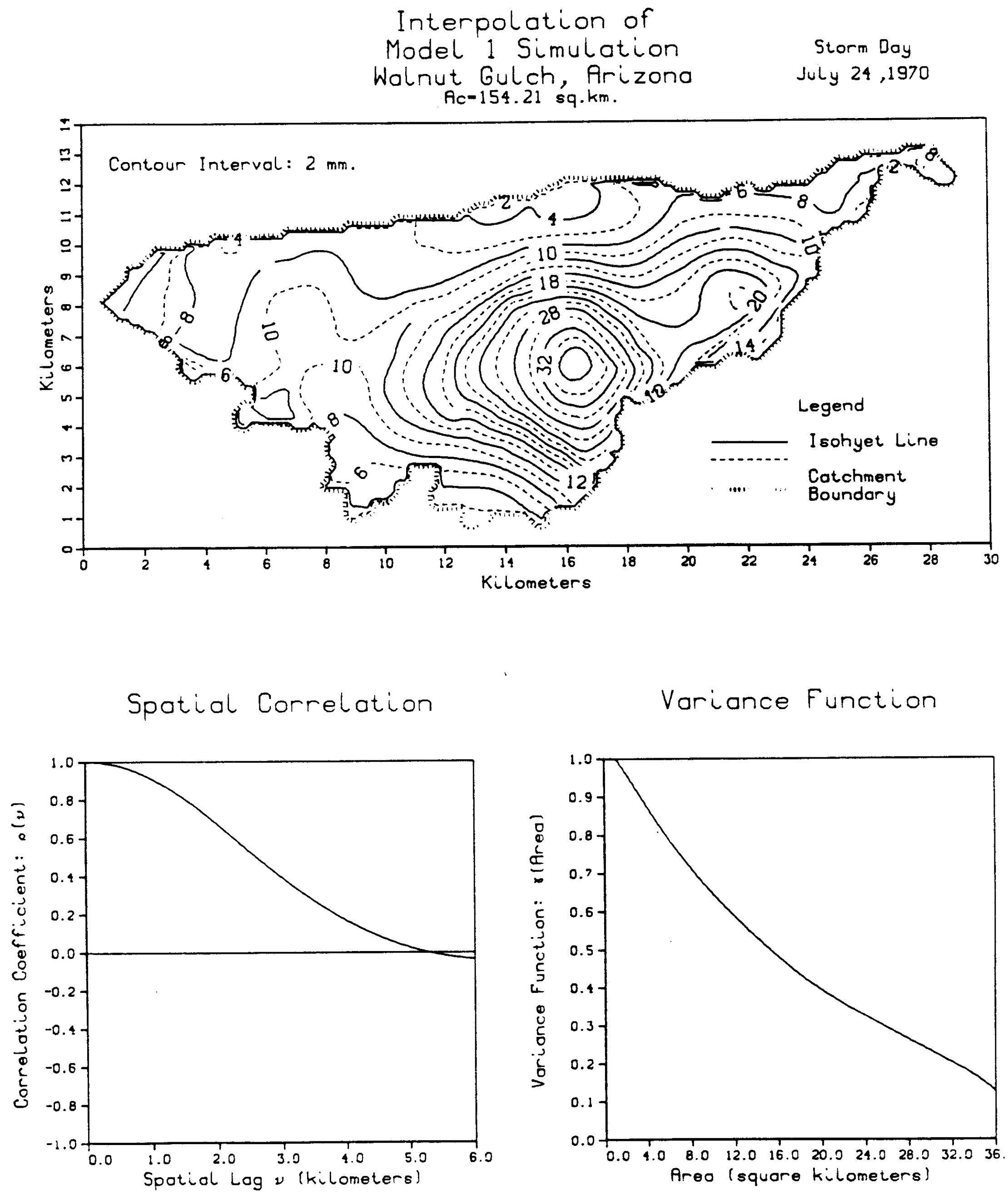


Fig. 7.1.2 Interpolation of the Simulated
24 July 1970 Storm Day

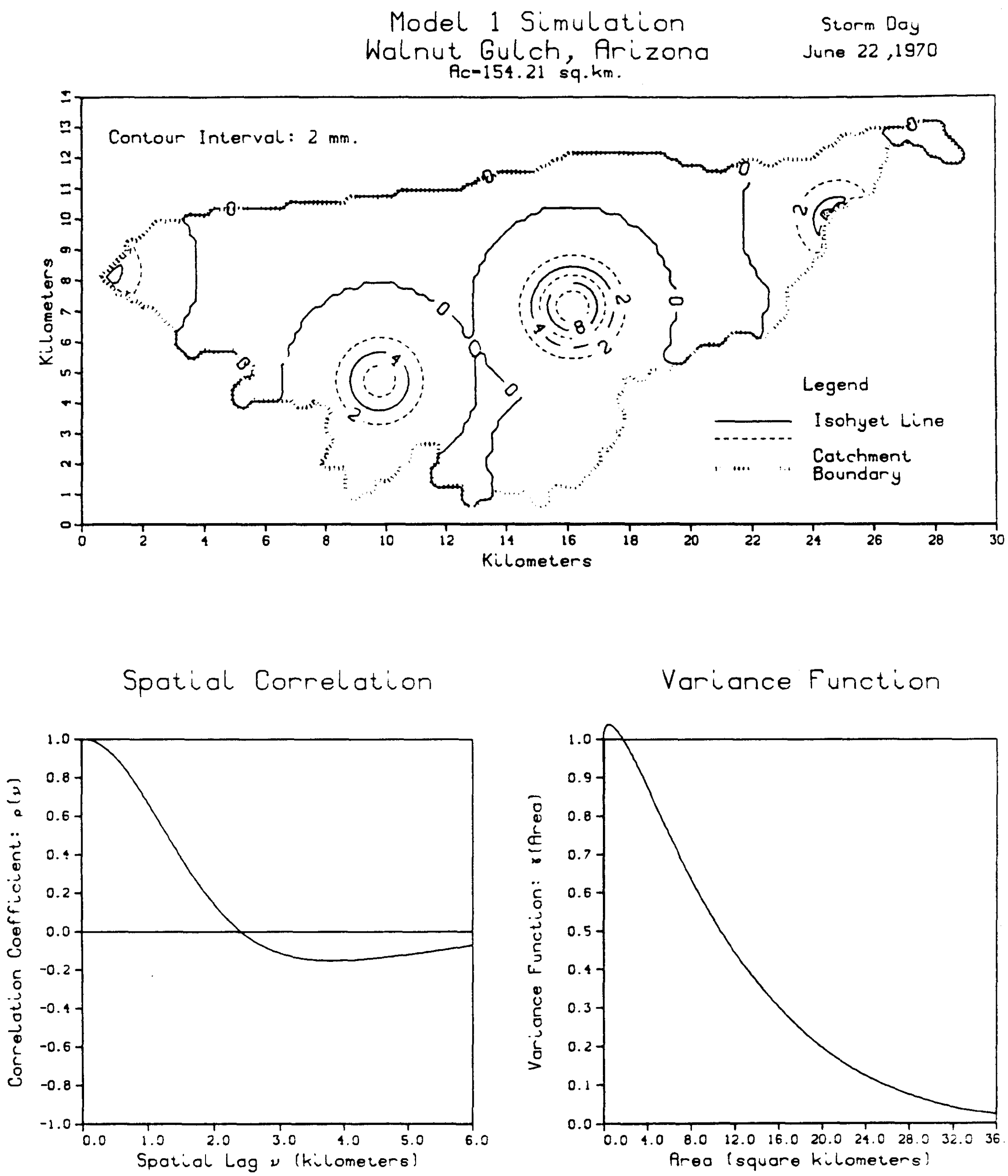


Fig. 7.1.3 Simulated 22 June 1970 Storm Day

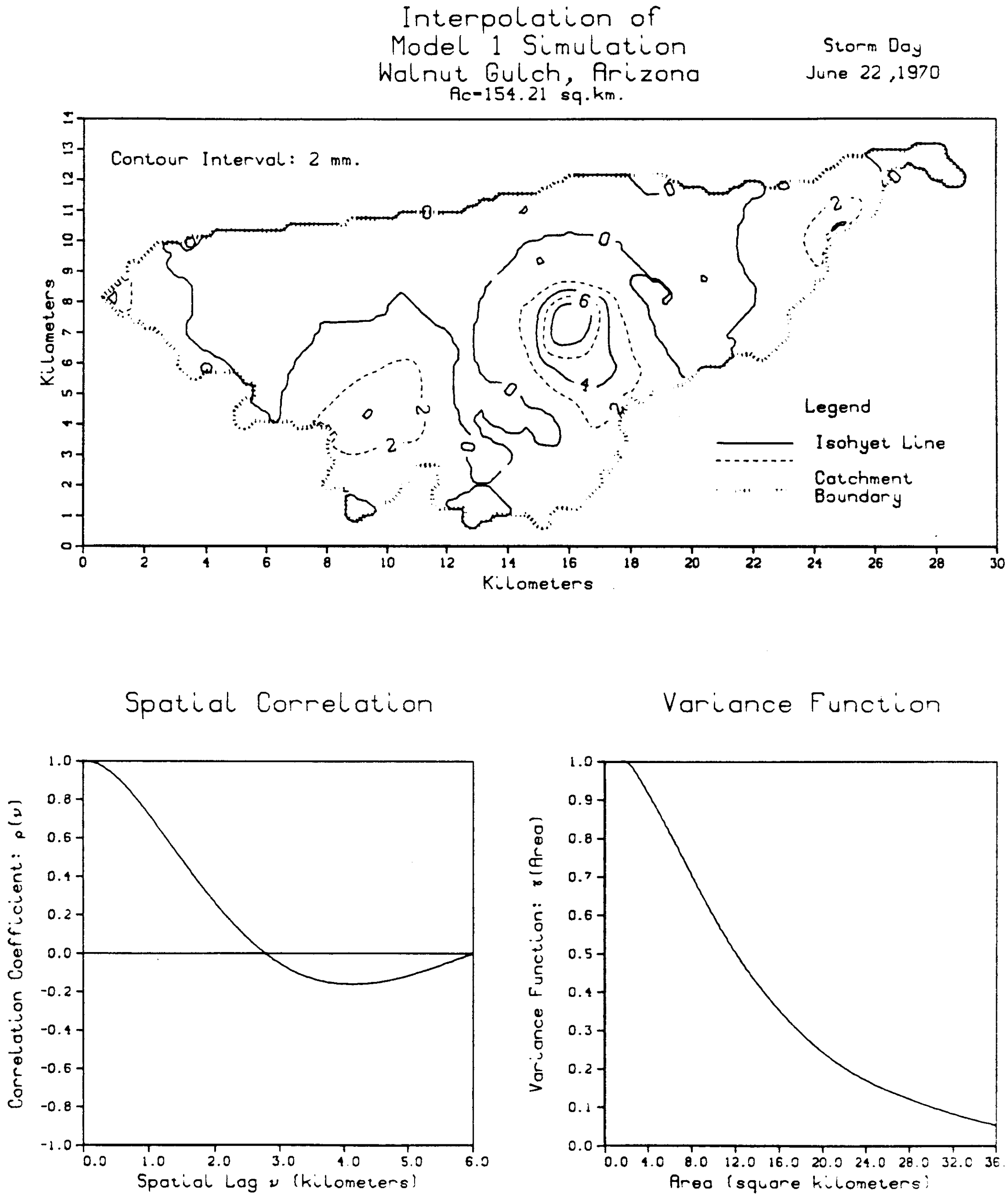


Fig. 7.1.4 Interpolation of the Simulated
22 June 1970 Storm Day

We further evaluate the surface interpolator by measuring the correlation between nodal depths in the simulated field and in the field interpolated from the simulated raingage depths.

This comparative node to node correlation was measured for five storm day simulations by the usual one dimensional estimator. The results are presented in Table 7.1.1. By comparing the first two storms, sparse and dense storms respectively, we see an increase in interpolator bias with decreasing storm coverage of the basin. This bias is not judged to be serious, however. The correlation is still quite strong and we are confident that the interpolator adequately reproduces the actual event.

Table 7.1.1

Comparison of "Observed" and Interpolated Rainfalls

Storm Day	Node to Node Correlation
22 June 1970	0.922
24 July 1970	0.988
10 August 1970	0.961
28 July 1972	0.990
26 September 1977	0.993

CHAPTER 8

Summary and Conclusions

8.1 Summary

Three existing models of the total rainfall depth from stationary thunderstorms are evaluated using observations from a dense raingage network in the Walnut Gulch experimental catchment of the A.R.S. near Tucson, Arizona.

The models each require three independent parameters:

a_i , the storm cell spread function parameter

λ_i , the storm cell spatial concentration, and

$E[\alpha_i]$, the average storm cell center depth

These parameters are estimated from each of 428 observed total storm rainfall fields and the distributions of the parameters across the ensemble of storms are estimated.

Using the estimated parameter averages the three models are compared through their ability to reproduce the observed spatial distribution of total storm depth.

Numerical simulations are also carried out.

8.2 Conclusions

These point process models appear capable of reproducing important features of the spatial distribution of total storm precipitation at least for storm types that are essentially stationary in space. The parameter sets provided should permit the spatial parameterization of storm rainfall for air mass thunderstorms in the Southwestern United States.

The conditional density functions of point storm depth derived from the three conceptual models under the assumption of constant parameters a, λ and μ_α, fit the "conditional" observations extremely well over a wide range of parameter values. This gives confidence in both the robustness of the models and the efficiency of the method of moments fitting.

Statistical dependence among the parameters, principally a and μ_α, makes it impossible to derive analytically the marginal density functions which are those of practical utility. As a substitute the desired marginal distribution is obtained by fitting the derived conditional pdfs to the ensemble average observations and the goodness of fit is again remarkable.

On the basis of these comparisons and considering analytical tractibility, Model 1 is chosen as the preferred conceptual representation of these storms.

The ensemble parameters indicate a density of convective "cells" within these airmass thunderstorms of about one cell per 50 square kilometers, having an average center depth of about 14 mm and a radius (at 1 percent of center depth) of about 6 km. Such dimensions compare favorably with those observed by Sorman [1975] for thunderstorms in the southeastern coastal plains of the United States. He found (i) an average of 5 cells in 500 km^2 (i.e., $\lambda = 10^{-2}$ km^{-2}); (ii) an average cell-center depth of 50 mm (i.e. μ_α = 50 mm) and (iii) elliptical cells with minor and major axes averaging 8 and 12 km respectively (i.e. radius of equivalent circular area, r = 5 km).

Disagreement of the fitted marginal distribution of storm depth with the observations of average dry catchment fraction indicates that while containing 28 percent dry area the Arizona air mass thunderstorms fail on the average to cover 12% of this 154 km^2 catchment. Thus, the parameter estimates are biased to an unknown degree. Using the generalized coverage statistics derived by Eagleson and Wang [1985] this 12% uncovered area translates into an estimated average thunderstorms radius of 100 km.

CHAPTER 9

The Computer Codes

9.1 Introduction

This chapter presents a brief summary of details about each of the computer codes used in this aspect of the study. The hard copies of each program are presented in Appendix III. The purpose of this chapter is not to provide complete documentation for each program but rather to comment about input and output file handling and links to the numerical algorithm and graphics libraries.

All of the computer programs developed are written in the FORTRAN V (Fortran 77) computer language. The programs were developed with other users in mind, and we feel that a small sacrifice in computational efficiency is well worth increased comprehension for the unfamiliar user.

9.2 CORRFIT.FORTRAN

The purpose of CORRFIT.FORTRAN is to estimate the model spread function parameter a_i by the method of least squares. It also calculates the other two model parameters for each of the three models. The program requires the eight summer season observed storm day data archive files entitled ARC70.DATA, ARC71.DATA, etc to be loaded into file50,file51 and so forth as input files. The program outputs to file01. The results of an execution of this program is listed in Appendix I as MODEL_PARAMETERS.DATA.

This program requires a link with the IMSL numerical library.

9.3 UNBIAS.FORTRAN

The purpose of this program is to unbias the spatial model parameters according to the method described by Vanmarcke [1983, pp.333]. This program requires the archived MODEL_PARAMETERS.DATA file as input. This data must be placed in file95 prior to execution. The results are written to file10. This data is archived as UNBIASED_MODEL_PARAMETERS.DATA and a listing is found in Appendix II.

This program does not require any links with the the numerical or graphics libraries.

9.4 TIMEDIST.FORTRAN

The purpose of this program is to plot the histograms of the spatial model parameters and fit temporal probability density functions to the appropriate ones. TIMEDIST.FORTRAN requires the archived UNBIASED_MODEL PARAMETERS.DATA as an input file. This data must be placed in file10 prior to execution. The program outputs ranked data to file01, parameter frequencies to file02 and relative frequencies to file 03. In addition, a plot of the selected parameter distribution is created.

This program requires a link to both the IMSL and DISSPLA libraries.

9.5 PARAM.FORTRAN

The purpose of PARAM.FORTRAN is to collect together the particular storm day spatial model parameters for the model of choice. The program also calculates the maximum simulation distance. PARAM.FORTRAN requires MODEL_PARAMETERS.DATA as an input file. This data must be placed in file95 prior to execution.

This program doesn't require any links with the numerical or graphics libraries.

9.6 RADIUS.FORTRAN and POLAR.FORTRAN These programs calculate the necessary number of storm cells to be generated during a storm day simulation. The programs require a guess estimate of the number of cells plus the model storm cell density parameter value. The analyst will be prompted for these. The output results will be written to the terminal screen.

These programs require access to the IMSL numerical library.

9.7 SIMWET.FORTRAN

SIMWET.FORTRAN simulates the total storm depth random field for the model and storm day of choice. It also determines the spatial distribution of total rainfall for each of the three simulation sampling random fields. The program requires the parent storm day raingage data in file84 as an input file. The program will prompt the user for the particular storm day spatial model parameters for the model of choice, and the necessary number of storm cells to be generated. SIMWET.FORTRAN writes the "simulation raingage" data to file83. The coarse grid mesh simulation field is written to file29. The simulation storm day date, and model parameters are written to file30 and the spatial distribution curves for the Level I,II and III sampling areas are written to file62.

This program requires a link with the IMSL numerical library.

9.8 SIMCRGAM.FORTRAN

The purpose of SIMCRGAM.FORTRAN is to sample the three simulated random fields for the spatial correlation and variance functions. The necessary input files are the file29, file30 and file83 output files created by SIMWET.FORTRAN. The SIMCRGAM.FORTRAN output file is file30.

This program requires a link with the IMSL numerical library.

9.9 TABLE5.FORTRAN

The purpose of TABLE5.FORTRAN is to create nicely formatted data tables, one for each of the results from sampling the Levels I, II and III random fields. The program requires the file30 output file created by SIMCRGAM.FORTRAN and file62 output file created by SIMWET.FORTRAN as its input files. The program writes the Level I results to file95, Level II to file96 and Level III to file97.

This program doesn't require access to either the numerical or graphics libraries.

9.10 SIMPLOT.FORTRAN

The purpose of SIMPLOT.FORTRAN is to plot the isohyets of the Level III simulation random field. The program requires the file29 and file30 output files from SIMWET.FORTRAN as input files.

This program requires a link with both the IMSL and DISSPLA graphics libraries during its execution.

9.11 SIM_STORMDAY.FORTRAN

SIM_STORMDAY.FORTRAN plots the Level I simulation isohyets, spatial correlation and variance function all on one page. The program requires the file29 output file from SIMWET.FORTRAN and file30 output file from SIMCRGAM.FORTRAN as its input files.

This program requires access to both the IMSL and DISSPLA libraries.

9.12 WETPLOT.FORTRAN

This program creates a single plot of the observed, Levels I,II and III simulation, and theoretical spatial distribution curves. The program creates the theoretical curves. WETPLOT.FORTRAN requires the file30, and file62 output files created by SIMWET.FORTRAN as input files. The observed storm day data set from the storm day data archive must be placed in file02 prior to execution.

This program requires a link with both the IMSL and NAG numerical libraries as well as the DISSPLA graphics library.

9.13 CORR_GAM_PLOT.FORTRAN

This program will plot either the spatial correlation or variance function for the observed storm day, the Level's I, II and III simulation random field sampling, and the theoretical curves. The program creates the theoretical curves. The program requires the file30 output file created by SIMCRGAM.FORTRAN as an input file. The observed storm day data must be taken from the storm day data archive file and placed in file02 prior to execution.

This program reuires links with both the IMSL and DISSPLA libraries.

9.14 SIMINTRP.FORTRAN

This program performs an interpolation of the 93 "simulation raingages" created by SIMWET.FORTRAN and subsequently analyses the interpolated random field. An execution results in a plot of the interpolated Level I isohyets, spatial correlation and variance functions. These three plots are placed on a single plot. This program requires the 93 "simulation raingages" written to file83 by SIMWET.FORTRAN as an input file.

This program requires links with both the IMSL and DISSPLA graphics libraries.

9.15 SIMCORR.FORTRAN

SIMCORR.FORTRAN will measure the correlation between the Level I simulated random field and the interpolation of the Level I simulation random field. The program requires the file83, and file29 output files created by SIMWET.FORTRAN as input files. The results are written to to both the terminal screen and file61.

This program requires access to the IMSL numerical library.

9.16 Computer Tape

This section describes the computer tape which may be ordered from the director of the Parsons Laboratory as mentioned in the preface of this report. This tape contains all of the FORTRAN programs and data files used and created during the course of this study and outlined in this volume and the companion volume Technical Report No. 306.

THE PHYSICAL TAPE CHARACTERISTICS ARE AS FOLLOWS:

tape density: 1600 bpi

tape mode: EBCDIC

tape type: IBM unlabeled

tape format: fixed block records

block length: 7200

Record length: 80 characters except for the last two files on the tape (at positions 46 and 47). These two files have a record length of 120 characters.

All programs are written in lower case (as opposed to the more usual upper case).

THE FILES ARE POSITIONED ON THE MAGNETIC TAPE IN THE FOLLOWING ORDER:

Program or File Name	Position on Tape
Taperead.fortran	1
summer.fortran	2
day.fortran	3
balance.fortran	4
coord.fortran	5
strmsort.fortran	6
gagecorr.fortran	7
allcorr.fortran	8
stormday.fortran	9
stormwet.fortran	10
table.fortran	11
future.fortran	12
bal70.data	13
bal71.data	14
bal72.data	15

Program or file name	Position on Tape
bal73.data	16
bal74.data	17
bal75.data	18
bal76.data	19
bal77.data	20
arc70.data	21
arc71.data	22
arc72.data	23
arc73.data	24
arc74.data	25
arc75.data	26
arc76.data	27
arc77.data	28
corrfit.fortran	29
unbias.fortran	30
timedist.fortran	31

Program or file name	Position on Tape
param.fortran	32
radius.fortran	33
polar.fortran	34
simwet.fortran	35
simcrgam.fortran	36
table5.fortran	37
simplot.fortran	38
wetplot.fortran	39
sim_stormday.fortran	40
corr_gam_plot.fortran	41
simintrp.fortran	42
sim corr.fortran	43
obs_eval.fortran	44
model_eval.fortran	45
model_parameters.data	46*
unbiased_model_parameters.data	47*

*Record length: 120 characters.

REFERENCES

Bhattacharya, S. K., A Modified Bessel Function Model in Life Testing, Metrika, 11, pp. 133-144, 1967.

Bras, R. L. and I. Rodriguez-Iturbe, Evaluation of Mean Square Error Involved in Approximating the Areal Average of a Rainfall Event by a Discrete Summation, Water Resources Research, 12(2), pp. 181-184, 1976.

Cox, D. R. and V. Isham, Point Processes, Chapman and Hall, London, England, 1980.

Eagleson, P. S., Climate, Soil and Vegetation 2. The Distribution of Annual Precipitation Derived from Observed Storm Sequences, Water Resources Research, 14(5) , pp. 713-721, 1978.

Rodriguez-Iturbe, I., D. R. Cox and P. S. Eagleson, Spatial Modelling of Total Storm Rainfall, Proc. R. Soc. Lond., A, 403, pp. 27-50, 1986-a.

Rodrigueq-Iturbe, I., Q. Wang, P. S. Eagleson and B. L. Jacobs, Spatial Models of Stationary Storm Rainfall: Theoretical Development, MIT Department of Civil Engineering, R. M. Parsons Laboratory Report No. 305, 1986-b.

Vanmarcke, E., Random Fields, Analysis and Synthesis, MIT Press, Cambridge, MA, 1983.

Sorman, A. U., Characteristics of rainfall cell patterns in the southeast coastal plain areas of the U.S.A., and a computer simulation model of thunderstorm rainfall, IAHS pub. no. 115, pp. 214-221. (Proc. Bratislava Symp. Applications of Mathematical Models in Hydrology and Water Resource Systems), 1975.

APPENDIX I

MODEL_PARAMETERS.DATA

MODEL_PARAMETERS.DATA

```
month, day, year, E(Y),VAR(Y),C.S.(Y)
   a1     E(alpha1)   lam1      a2    E(alpha2)    lam2       a3     E(alpha3)    lam3
  ......................   ........................   ........................
    428   records
June 22 1970  0.270  0.771  4.386
  0.5656   2.8556   0.0193   1.4789   5.7111   0.0165   2.2590   1.4278   0.0753
June 28 1970  0.706  0.994  1.847
  0.4573   1.4079   0.0668   1.2089   2.8159   0.0583   0.8989   0.7040   0.2603
June 29 1970  0.517  3.046  4.609
  0.3559   5.8917   0.0071   0.9189  11.7834   0.0059   2.9189   2.9458   0.0274
July  1 1970  0.382  0.380  1.693
  0.3068   0.9948   0.0230   0.8089   1.9895   0.0200   0.4289   0.4974   0.0909
July  2 1970  0.033  0.032  7.127
  0.6563   0.9697   0.0093   1.6990   1.9394   0.0078   0.8789   0.4848   0.0356
July  7 1970  2.752  9.454  1.444
  0.2926   3.4353   0.0437   0.7489   6.8706   0.0358   1.3889   1.7177   0.1667
July  8 1970  0.531  1.111  2.109
  0.3670   2.0923   0.0218   0.9689   4.1846   0.0190   1.0790   1.0461   0.0859
July 13 1970  0.238  0.452  3.682
  0.5610   1.8992   0.0251   1.4989   3.7983   0.0224   1.4989   0.9496   0.0994
July 17 1970  0.115  0.081  2.976
  0.7132   0.7043   0.0529   1.8090   1.4087   0.0425   0.6890   0.3522   0.1989
July 18 1970  0.043  0.052  7.203
  0.9337   1.2093   0.0197   2.4590   2.4186   0.0171   1.5790   0.6047   0.0772
July 19 1970  4.610 22.077  1.588
  0.2460   4.7889   0.0371   0.6189   9.5779   0.0293   1.6089   2.3945   0.1383
July 20 1970 20.640185.010  0.997
  0.1965   8.9637   0.0566   0.4789  17.9273   0.0420   2.3790   4.4818   0.2065
July 21 1970  0.257  0.117  1.182
  0.2624   0.4553   0.0247   0.6689   0.9105   0.0201   0.1690   0.2276   0.0991
July 24 1970  7.969 48.831  0.549
  0.2747   6.1276   0.0625   0.6989  12.2552   0.0506   2.3290   3.0638   0.2392
July 25 1970  2.839 20.485  2.733
  0.3403   7.2156   0.0290   0.8989  14.4311   0.0253   3.4490   3.6078   0.1145
July 27 1970  1.176  4.878  2.216
  0.3865   4.1480   0.0270   1.0190   8.2959   0.0234   2.2389   2.0740   0.1052
July 28 1970  9.744 53.787  1.191
  0.1917   5.5200   0.0413   0.4690  11.0400   0.0309   1.4290   2.7600   0.1506
July 29 1970  2.278 28.928  2.663
  0.3546  12.6989   0.0144   0.9290  25.3977   0.0123   6.2689   6.3494   0.0557
July 30 1970  5.178  6.965  0.863
  0.3781   1.3451   0.3503   0.9689   2.6902   0.2876   0.6989   0.6726   1.3232
July 31 1970 10.129 64.582  0.445
  0.2706   6.3760   0.0741   0.6889  12.7519   0.0600   2.3790   3.1880   0.2816
Aug   1 1970 12.988 41.904  0.948
  0.2112   3.2264   0.1143   0.5189   6.4527   0.0863   0.9290   1.6132   0.4250
Aug   2 1970  7.012 21.048  0.484
  0.2728   3.0017   0.1107   0.6990   6.0034   0.0908   1.1289   1.5009   0.4207
Aug   3 1970  1.067  1.681  1.123
  0.3228   1.5754   0.0449   0.8389   3.1509   0.0379   0.7090   0.7877   0.1746
Aug   4 1970  0.552  2.095  3.067
  0.3447   3.7953   0.0110   0.8989   7.5906   0.0094   1.8290   1.8976   0.0430
Aug   5 1970  0.435  0.959  2.924
  0.3183   2.2046   0.0127   0.8189   4.4092   0.0105   0.9689   1.1023   0.0485
Aug   6 1970  0.399  2.776  5.196
  0.5412   6.9574   0.0107   1.4089  13.9148   0.0091   5.2390   3.4787   0.0414
Aug   7 1970  1.393  6.158  3.315
```

	0.2604	4.4207	0.0136	0.6690	8.8413	0.0112	1.5889	2.2103	0.0518
Aug 8 1970	5.490	53.891	1.988						
	0.3212	9.8162	0.0367	0.8392	19.6324	0.0313	4.3890	4.9081	0.1424
Aug 9 1970	7.552	36.318	1.613						
	0.4278	4.8091	0.1830	1.1489	9.6181	0.1650	2.9089	2.4045	0.7316
Aug 10 1970	17.618	44.128	0.134						
	0.4541	2.5047	0.9234	1.2490	5.0094	0.8732	1.6290	1.2524	3.7882
Aug 11 1970	0.503	5.236	7.259						
	0.1819	10.4095	0.0010	0.4390	20.8191	0.0007	2.5389	5.2048	0.0037
Aug 14 1970	4.812	23.184	1.146						
	0.4803	4.8180	0.1467	1.3094	9.6359	0.1363	3.2889	2.4090	0.5926
Aug 15 1970	2.177	6.466	1.498						
	0.5439	2.9701	0.1380	1.4589	5.9403	0.1241	2.2789	1.4851	0.5494
Aug 16 1970	12.850	131.970	0.953						
	0.2817	10.2700	0.0632	0.7189	20.5401	0.0515	3.9989	5.1350	0.2415
Aug 17 1970	5.587	50.220	2.187						
	0.3445	8.9887	0.0470	0.8989	17.9774	0.0400	4.3190	4.4944	0.1827
Aug 18 1970	0.172	0.491	5.069						
	0.6172	2.8547	0.0146	1.6289	5.7093	0.0127	2.4690	1.4273	0.0574
Aug 19 1970	1.073	5.556	2.748						
	0.4423	5.1780	0.0258	1.1789	10.3560	0.0229	3.2189	2.5890	0.1020
Aug 21 1970	0.028	0.036	8.099						
	0.8267	1.2857	0.0095	2.1589	2.5714	0.0081	1.4790	0.6429	0.0367
Aug 22 1970	4.121	14.551	0.924						
	0.3186	3.5309	0.0754	0.8289	7.0619	0.0638	1.5689	1.7655	0.2934
Aug 23 1970	4.457	18.994	1.412						
	0.2978	4.2616	0.0590	0.7589	8.5232	0.0479	1.7489	2.1308	0.2243
Aug 26 1970	1.907	7.569	1.301						
	0.3945	3.9691	0.0476	1.0289	7.9381	0.0405	2.1890	1.9845	0.1861
Sept 2 1970	0.240	0.541	4.408						
	0.6602	2.2542	0.0295	1.7289	4.5083	0.0253	2.0790	1.1271	0.1153
Sept 3 1970	11.431	3.543	0.687						
	0.3853	0.3099	3.4856	0.9989	0.6199	2.9284	0.1690	0.1550	13.9607
Sept 4 1970	9.434	10.152	-0.021						
	0.4233	1.0761	1.0000	1.1389	2.1522	0.9049	0.6389	0.5381	3.9346
Sept 8 1970	6.400	105.592	1.646						
	0.3485	16.4987	0.0300	0.9189	32.9975	0.0261	8.0489	8.2494	0.1175
Sept 9 1970	8.003	10.486	0.933						
	0.3168	1.3103	0.3903	0.8189	2.6205	0.3259	0.5790	0.6551	1.5186
Sept 11 1970	0.054	0.112	7.622						
	0.7681	2.0741	0.0098	1.9989	4.1481	0.0083	2.2089	1.0370	0.0376
Sept 12 1970	0.055	0.020	2.829						
	0.3847	0.3636	0.0143	1.0191	0.7273	0.0125	0.1990	0.1818	0.0577
Sept 30 1970	1.344	1.937	1.341						
	0.5616	1.4412	0.1872	1.5590	2.8824	0.1804	1.1589	0.7206	0.7677
June 24 1971	6.621	40.557	1.471						
	0.2567	6.1255	0.0453	0.6489	12.2510	0.0362	2.1589	3.0628	0.1710
July 2 1971	3.563	9.737	0.985						
	0.6512	2.7328	0.3520	1.7189	5.4656	0.3065	2.4889	1.3664	1.3769
July 3 1971	0.377	0.993	3.691						
	0.7680	2.6340	0.0537	1.9890	5.2679	0.0451	2.8090	1.3170	0.2073
July 5 1971	1.258	5.161	2.670						
	0.2729	4.1025	0.0145	0.6989	8.2051	0.0119	1.5490	2.0513	0.0557
July 10 1971	0.561	2.365	3.282						
	0.5031	4.2157	0.0214	1.3590	8.4314	0.0196	2.9989	2.1078	0.0857
July 13 1971	0.040	0.042	6.178						
	0.4919	1.0500	0.0059	1.2889	2.1000	0.0050	0.7189	0.5250	0.0227
July 16 1971	3.065	11.816	1.985						
	0.3528	3.8551	0.0630	0.8989	7.7103	0.0511	1.8789	1.9276	0.2405
July 17 1971	1.094	11.375	3.384						

0.3079	10.3976	0.0064	0.7889	20.7952	0.0052	4.4289	5.1988	0.0243	
July 18 1971	6.209	40.297	1.646						
0.2370	6.4901	0.0342	0.5889	12.9802	0.0264	2.0989	3.2450	0.1274	
July 19 1971	0.652	1.651	2.959						
0.4945	2.5322	0.0401	1.2790	5.0644	0.0335	1.7390	1.2661	0.1546	
July 20 1971	1.905	13.831	2.764						
0.4559	7.2604	0.0347	1.2089	14.5207	0.0305	4.6489	3.6302	0.1370	
July 21 1971	1.652	22.417	3.996						
0.3528	13.5696	0.0096	0.9089	27.1392	0.0080	6.6289	6.7848	0.0370	
July 23 1971	23.769	268.325	0.355						
0.2419	11.2889	0.0784	0.6090	22.5777	0.0621	3.7290	5.6444	0.2925	
July 24 1971	12.120	25.709	2.002						
0.2979	2.1212	0.3228	0.7689	4.2424	0.2688	0.8790	1.0606	1.2492	
July 26 1971	0.871	2.054	1.989						
0.3640	2.3582	0.0312	0.9190	4.7164	0.0248	1.1790	1.1791	0.1175	
July 27 1971	3.015	5.879	1.204						
0.4861	1.9499	0.2326	1.3289	3.8998	0.2173	1.3489	0.9750	0.9421	
July 28 1971	7.076	23.848	0.646						
0.3338	3.3703	0.1489	0.8690	6.7405	0.1262	1.5589	1.6851	0.5719	
July 29 1971	2.311	8.717	1.637						
0.4307	3.7720	0.0724	1.1390	7.5439	0.0633	2.2790	1.8860	0.2848	
July 30 1971	4.077	3.159	0.775						
0.2866	0.7748	0.2751	0.7389	1.5497	0.2286	0.3094	0.3874	1.0682	
July 31 1971	5.338	13.323	1.089						
0.2917	2.4959	0.1159	0.7489	4.9918	0.0955	1.0089	1.2479	0.4450	
Aug 1 1971	0.038	0.077	9.447						
0.5609	2.0263	0.0038	1.4790	4.0526	0.0033	1.5890	1.0132	0.0147	
Aug 3 1971	5.024	4.816	0.945						
0.5568	0.9586	1.0344	1.5289	1.9172	0.9749	0.7589	0.4793	4.1823	
Aug 8 1971	2.846	17.968	1.759						
0.2362	6.3134	0.0160	0.5889	12.6268	0.0124	2.0389	3.1567	0.0599	
Aug 10 1971	27.526	174.063	0.221						
0.2727	6.3236	0.2061	0.6990	12.6472	0.1692	2.3790	3.1618	0.7844	
Aug 11 1971	1.566	9.080	2.160						
0.2965	5.7982	0.0151	0.7690	11.5964	0.0127	2.3890	2.8991	0.0584	
Aug 12 1971	20.985	53.595	0.031						
0.3540	2.5540	0.6555	0.9090	5.1079	0.5403	1.2490	1.2770	2.5021	
Aug 13 1971	0.007	0.002	7.886						
0.9316	0.2857	0.0135	2.4389	0.5714	0.0116	0.3689	0.1429	0.0520	
Aug 14 1971	4.350	12.415	1.656						
0.3633	2.8540	0.1281	0.9389	5.7080	0.1069	1.4389	1.4270	0.4933	
Aug 16 1971	1.059	4.251	3.219						
0.4219	4.0142	0.0299	1.0889	8.0283	0.0249	2.3490	2.0071	0.1150	
Aug 17 1971	2.015	12.369	1.825						
0.2163	6.1385	0.0098	0.5390	12.2769	0.0076	1.8090	3.0692	0.0363	
Aug 18 1971	26.033	56.754	-0.320						
0.4033	2.1801	1.2365	1.0589	4.3602	1.0655	1.2289	1.0900	4.8312	
Aug 19 1971	0.308	0.383	2.381						
0.4255	1.2435	0.0285	1.1389	2.4870	0.0256	0.7490	0.6218	0.1144	
Aug 20 1971	5.097	32.627	1.215						
0.4331	6.4012	0.0951	1.1689	12.8024	0.0866	3.9289	3.2006	0.3819	
Aug 21 1971	2.218	13.435	1.751						
0.3940	6.0573	0.0362	1.0489	12.1145	0.0321	3.3589	3.0286	0.1434	
Aug 23 1971	2.656	23.056	2.050						
0.3453	8.6807	0.0232	0.8889	17.3614	0.0192	4.1489	4.3404	0.0890	
Aug 24 1971	7.997	33.428	1.874						
0.3806	4.1801	0.1764	0.9889	8.3601	0.1489	2.2190	2.0900	0.6864	
Aug 27 1971	4.692	19.056	1.096						
0.3030	4.0614	0.0675	0.7890	8.1228	0.0572	1.7090	2.0307	0.2605	
Aug 28 1971	3.894	19.747	2.131						

0.3591	5.0711	0.0630	0.9290	10.1423	0.0527	2.5189	2.5356	0.2412
Aug 31 1971	0.503	2.461	3.920					
0.4027	4.8926	0.0106	1.0593	9.7853	0.0092	2.7489	2.4463	0.0413
Sept 1 1971	3.872	21.587	3.382					
0.3188	5.5752	0.0449	0.8189	11.1503	0.0371	2.4489	2.7876	0.1706
Sept 2 1971	1.063	3.560	1.711					
0.3023	3.3490	0.0185	0.7789	6.6980	0.0153	1.4090	1.6745	0.0715
Sept 3 1971	1.366	4.099	1.735					
0.5452	3.0007	0.0861	1.3789	6.0015	0.0689	2.2490	1.5004	0.3256
Sept 5 1971	0.146	0.072	2.945					
0.6936	0.4932	0.0907	1.7790	0.9863	0.0746	0.4689	0.2466	0.3408
Sept 6 1971	0.087	0.056	3.207					
0.7310	0.6437	0.0460	1.9590	1.2874	0.0413	0.6589	0.3218	0.1803
Sept 7 1971	0.051	0.135	8.958					
0.1359	2.6471	0.0002	0.3190	5.2941	0.0002	0.4790	1.3235	0.0008
Sept 8 1971	2.938	12.814	5.549					
0.3614	4.3615	0.0560	0.9390	8.7229	0.0473	2.1891	2.1807	0.2161
Sept 16 1971	0.008	0.002	5.488					
0.6403	0.2500	0.0084	1.6389	0.5000	0.0068	0.2189	0.1250	0.0312
Sept 17 1971	0.127	0.121	4.525					
0.5974	0.9528	0.0303	1.4989	1.9055	0.0238	0.7790	0.4764	0.1135
Sept 18 1971	17.799	26.270	0.568					
0.3155	1.4759	0.7642	0.8189	2.9519	0.6436	0.6490	0.7380	2.9689
Sept 28 1971	2.554	2.386	2.572					
0.4751	0.9342	0.3928	1.2489	1.8684	0.3393	0.6189	0.4671	1.5276
Sept 29 1971	0.768	3.934	3.864					
0.9237	5.1224	0.0814	2.4489	10.2448	0.0716	6.6390	2.5612	0.3207
June 4 1972	0.195	0.190	3.153					
0.3555	0.9744	0.0161	0.9389	1.9487	0.0140	0.4890	0.4872	0.0642
June 5 1972	3.963	2.439	0.467					
0.6075	0.6154	1.5129	1.6490	1.2309	1.3934	0.5289	0.3077	6.0551
June 6 1972	24.434113.449		-0.661					
0.2560	4.6431	0.2196	0.6494	9.2862	0.1766	1.6289	2.3215	0.8247
June 7 1972	0.080	0.035	2.831					
0.6511	0.4375	0.0494	1.6889	0.8750	0.0415	0.3990	0.2188	0.1936
June 8 1972	4.511	10.935	0.537					
0.3216	2.4241	0.1225	0.8290	4.8481	0.1018	1.0789	1.2120	0.4694
June 9 1972	0.047	0.018	3.481					
0.6145	0.3830	0.0295	1.3689	0.7660	0.0183	0.3090	0.1915	0.1017
June 11 1972	1.166	2.862	2.014					
0.4078	2.4545	0.0503	1.0890	4.9091	0.0448	1.4089	1.2273	0.1993
June 12 1972	4.307	14.687	1.172					
0.5110	3.4100	0.2100	1.3689	6.8201	0.1883	2.4589	1.7050	0.8362
June 14 1972	6.417	35.642	0.589					
0.2759	5.5543	0.0560	0.7090	11.1086	0.0462	2.1089	2.7772	0.2121
June 17 1972	0.017	0.008	6.629					
0.7715	0.4706	0.0137	2.0290	0.9412	0.0118	0.5090	0.2353	0.0538
June 20 1972	0.166	0.071	2.557					
0.6987	0.4277	0.1206	1.8989	0.8554	0.1114	0.4189	0.2139	0.4740
July 5 1972	1.873	10.689	3.056					
0.4030	5.7069	0.0339	1.0690	11.4138	0.0298	3.2189	2.8534	0.1329
July 6 1972	0.309	0.676	3.145					
0.4208	2.1877	0.0159	1.1090	4.3754	0.0138	1.2890	1.0939	0.0624
July 7 1972	0.868	1.606	1.554					
0.2950	1.8502	0.0260	0.7590	3.7005	0.0215	0.7489	0.9251	0.0979
July 8 1972	0.022	0.014	6.192					
0.8317	0.6364	0.0152	2.1890	1.2727	0.0132	0.7390	0.3182	0.0594
July 9 1972	0.035	0.023	4.938					
0.6526	0.6571	0.0144	1.7189	1.3143	0.0125	0.5989	0.3286	0.0563
July 10 1972	2.762	15.302	1.611					

0.2316	5.5402	0.0170	0.5789	11.0804	0.0133	1.7590	2.7701	0.0640
July 12 1972	1.691	4.393	1.795					
0.3787	2.5979	0.0594	1.0089	5.1957	0.0527	1.3890	1.2989	0.2369
July 14 1972	1.779	3.042	1.457					
0.4230	1.7099	0.1185	1.1189	3.4199	0.1036	1.0190	0.8550	0.4704
July 15 1972	15.274	32.682	0.795					
0.3185	2.1397	0.4610	0.8189	4.2794	0.3809	0.9490	1.0699	1.7878
July 16 1972	0.445	0.613	2.582					
0.4704	1.3775	0.0455	1.2189	2.7551	0.0382	0.8989	0.6888	0.1751
July 19 1972	1.330	4.441	2.146					
0.4279	3.3391	0.0464	1.1090	6.6782	0.0390	1.9789	1.6695	0.1781
July 22 1972	13.734	41.077	1.876					
0.3056	2.9909	0.2730	0.7990	5.9818	0.2333	1.2689	1.4954	1.0523
July 23 1972	0.142	0.124	3.805					
0.5520	0.8732	0.0315	1.4089	1.7465	0.0257	0.6589	0.4366	0.1179
July 24 1972	19.046	305.948	1.120					
0.2185	16.0636	0.0360	0.5389	32.1273	0.0274	4.7789	8.0318	0.1336
July 25 1972	1.224	5.817	2.969					
0.6121	4.7525	0.0614	1.5889	9.5049	0.0517	4.0390	2.3762	0.2369
July 26 1972	0.982	7.311	3.072					
0.2701	7.4450	0.0061	0.6890	14.8900	0.0050	2.7589	3.7225	0.0231
July 28 1972	7.418	25.141	0.289					
0.2324	3.3892	0.0753	0.5789	6.7784	0.0584	1.0790	1.6946	0.2825
July 29 1972	0.389	0.270	1.564					
0.5260	0.6941	0.0987	1.0990	1.3882	0.0539	0.4590	0.3470	0.3121
Aug 3 1972	1.875	8.132	2.500					
0.2613	4.3371	0.0188	0.6689	8.6741	0.0154	1.5689	2.1685	0.0720
Aug 4 1972	0.072	0.037	2.918					
0.3549	0.5139	0.0112	0.9189	1.0278	0.0094	0.2489	0.2569	0.0418
Aug 5 1972	4.687	35.809	1.306					
0.1855	7.6401	0.0134	0.4489	15.2801	0.0098	1.9089	3.8200	0.0488
Aug 6 1972	10.935	22.944	0.482					
0.3383	2.0982	0.3797	0.8689	4.1964	0.3131	0.9789	1.0491	1.4443
Aug 8 1972	9.785	29.731	0.602					
0.2289	3.0384	0.1074	0.5789	6.0769	0.0859	0.9590	1.5192	0.4085
Aug 9 1972	4.583	16.030	0.814					
0.2965	3.4977	0.0733	0.7689	6.9954	0.0616	1.4490	1.7489	0.2863
Aug 10 1972	0.205	0.258	3.289					
0.2975	1.2585	0.0092	0.7689	2.5171	0.0077	0.5189	0.6293	0.0353
Aug 12 1972	29.425	342.981	1.073					
0.2054	11.6561	0.0678	0.4989	23.3122	0.0500	3.2289	5.8281	0.2466
Aug 13 1972	0.561	2.087	3.076					
0.4421	3.7201	0.0188	1.1894	7.4403	0.0170	2.3189	1.8601	0.0746
Aug 17 1972	0.653	2.023	3.816					
0.4605	3.0980	0.0285	1.1789	6.1960	0.0233	1.9689	1.5490	0.1084
Aug 18 1972	0.779	1.716	3.278					
0.4547	2.2028	0.0465	1.1789	4.4056	0.0391	1.3889	1.1014	0.1790
Aug 19 1972	0.001	0.000	16.758					
1.1052	0.5000	0.0016	2.8390	1.0000	0.0013	0.7589	0.2500	0.0059
Aug 25 1972	7.271	3.948	0.333					
0.3212	0.5430	0.8795	0.8189	1.0860	0.7146	0.2389	0.2715	3.3006
Aug 26 1972	10.625	19.970	2.034					
0.3024	1.8795	0.3291	0.7790	3.7591	0.2730	0.7890	0.9398	1.2684
Aug 28 1972	3.061	8.325	0.949					
0.2406	2.7197	0.0415	0.6090	5.4394	0.0332	0.8990	1.3598	0.1566
Aug 29 1972	10.508	28.287	2.927					
0.4901	2.6919	0.5969	1.2690	5.3839	0.5002	1.8290	1.3460	2.2943
Aug 30 1972	0.067	0.096	8.865					
0.2863	1.4328	0.0024	0.7389	2.8657	0.0020	0.5691	0.7164	0.0094
Aug 31 1972	1.248	11.883	3.138					

0.5342 9.5216 0.0238 1.4189 19.0433 0.0210 7.1389 4.7608 0.0938
Sept 1 1972 10.329223.648 2.148
0.2801 21.6524 0.0238 0.7190 43.3049 0.0196 8.3589 10.8262 0.0905
Sept 2 1972 8.065 27.110 2.830
0.3945 3.3614 0.2377 1.0394 6.7229 0.2063 1.8489 1.6807 0.9242
Sept 5 1972 0.057 0.035 4.179
0.3846 0.6140 0.0087 0.9989 1.2281 0.0074 0.3290 0.3070 0.0339
Sept 6 1972 6.539 44.316 1.404
0.4338 6.7772 0.1156 1.1589 13.5544 0.1031 4.1389 3.3886 0.4582
Sept 7 1972 0.004 0.001 8.226
0.8666 0.2500 0.0076 2.2790 0.5000 0.0066 0.2989 0.1250 0.0291
Sept 9 1972 0.590 4.654 7.757
0.7205 7.8881 0.0247 1.8490 15.7763 0.0203 7.8290 3.9441 0.0938
Sept 10 1972 0.169 0.200 3.203
0.4229 1.1834 0.0163 1.0890 2.3669 0.0135 0.6889 0.5917 0.0616
Sept 11 1972 2.702 11.036 1.238
0.2757 4.0844 0.0320 0.7090 8.1688 0.0265 1.5590 2.0422 0.1227
Sept 13 1972 0.235 3.768 11.208
0.2785 16.0340 0.0007 0.7189 32.0681 0.0006 6.2289 8.0170 0.0028
Sept 14 1972 0.748 3.086 3.486
0.4534 4.1257 0.0237 1.1989 8.2513 0.0207 2.6189 2.0628 0.0930
Sept 15 1972 0.540 2.224 3.499
0.5026 4.1185 0.0211 1.3490 8.2370 0.0190 2.9190 2.0593 0.0839
June 1 1973 0.086 0.060 3.492
0.6759 0.6977 0.0359 1.7589 1.3953 0.0303 0.6590 0.3488 0.1400
June 11 1973 2.027 10.155 2.023
0.3420 5.0099 0.0301 0.8890 10.0197 0.0254 2.3789 2.5049 0.1162
June 12 1973 12.759 11.738 -0.829
0.4743 0.9200 1.9862 1.2789 1.8400 1.8051 0.6190 0.4600 7.9942
June 13 1973 0.035 0.010 3.955
0.7257 0.2857 0.0411 1.8990 0.5714 0.0352 0.2890 0.1429 0.1596
July 2 1973 2.723 2.531 0.408
0.5891 0.9295 0.6472 1.5689 1.8590 0.5738 0.7689 0.4647 2.5525
July 5 1973 0.850 1.320 2.159
0.5014 1.5529 0.0876 1.3390 3.1059 0.0781 1.0990 0.7765 0.3490
July 6 1973 0.958 2.109 1.673
0.2829 2.2015 0.0222 0.7290 4.4029 0.0184 0.8589 1.1007 0.0843
July 8 1973 2.752 13.070 1.283
0.3412 4.7493 0.0429 0.8790 9.4985 0.0356 2.2389 2.3746 0.1640
July 9 1973 1.107 2.055 1.233
0.2881 1.8564 0.0315 0.7389 3.7127 0.0259 0.7389 0.9282 0.1203
July 10 1973 16.499 35.523 0.597
0.2997 2.1530 0.4382 0.7789 4.3061 0.3700 0.8990 1.0765 1.7011
July 11 1973 0.454 3.185 4.755
0.2184 7.0154 0.0020 0.5392 14.0308 0.0015 2.0790 3.5077 0.0072
July 12 1973 10.291 89.424 1.222
0.4223 8.6895 0.1345 1.1490 17.3791 0.1244 5.2090 4.3448 0.5419
July 13 1973 1.563 12.760 3.126
0.3012 8.1638 0.0111 0.7789 16.3276 0.0092 3.4290 4.0819 0.0430
July 14 1973 17.386 49.469 0.958
0.4118 2.8453 0.6597 1.0790 5.6907 0.5661 1.6390 1.4227 2.5815
July 15 1973 6.229 42.863 1.546
0.3871 6.8812 0.0864 1.0189 13.7624 0.0748 3.7289 3.4406 0.3385
July 18 1973 0.711 1.489 2.233
0.2803 2.0942 0.0170 0.7189 4.1885 0.0140 0.8089 1.0471 0.0645
July 26 1973 0.288 0.341 2.552
0.3174 1.1840 0.0156 0.8189 2.3681 0.0130 0.5189 0.5920 0.0595
July 27 1973 17.630432.166 1.123
0.2286 24.5131 0.0239 0.5689 49.0262 0.0185 7.6389 12.2566 0.0889
July 28 1973 0.331 0.270 3.625

0.6312 0.8157 0.1029 1.5890 1.6314 0.0815 0.6989 0.4079 0.3793
July 29 1973 0.282 1.419 6.622
0.4406 5.0319 0.0069 1.0489 10.0638 0.0049 2.9490 2.5160 0.0245
July 30 1973 0.035 0.025 5.666
0.6207 0.7143 0.0120 1.6390 1.4286 0.0105 0.6189 0.3571 0.0468
July 31 1973 0.794 2.030 2.521
0.6611 2.5567 0.0864 1.7689 5.1134 0.0773 2.3789 1.2783 0.3423
Aug 1 1973 0.414 0.707 4.602
0.6526 1.7077 0.0657 1.7189 3.4155 0.0570 1.5590 0.8539 0.2572
Aug 2 1973 0.031 0.053 9.460
0.9666 1.7097 0.0108 2.5189 3.4194 0.0092 2.2990 0.8548 0.0417
Aug 3 1973 0.006 0.001 8.881
1.2704 0.1667 0.0370 3.2490 0.3333 0.0302 0.2889 0.0833 0.1377
Aug 5 1973 3.040 21.658 1.758
0.3588 7.1243 0.0350 0.9490 14.2487 0.0306 3.5794 3.5622 0.1371
Aug 6 1973 0.063 0.124 6.924
0.7919 1.9683 0.0128 2.0789 3.9365 0.0110 2.1790 0.9841 0.0499
Aug 9 1973 0.004 0.001 7.752
0.8108 0.2500 0.0067 2.1189 0.5000 0.0057 0.2789 0.1250 0.0254
Aug 10 1973 4.528 4.567 1.057
0.3458 1.0086 0.3418 0.9090 2.0172 0.2952 0.4890 0.5043 1.3436
Aug 19 1973 0.038 0.017 3.998
0.3837 0.4474 0.0080 0.9991 0.8947 0.0067 0.2390 0.2237 0.0309
Aug 20 1973 0.214 0.090 1.500
0.2883 0.4206 0.0269 0.7590 0.8411 0.0233 0.1689 0.2103 0.1045
Aug 21 1973 12.295102.530 1.583
0.3201 8.3392 0.0962 0.8289 16.6783 0.0806 3.7089 4.1696 0.3713
Aug 22 1973 0.449 1.984 3.945
0.3118 4.4187 0.0063 0.8090 8.8374 0.0053 1.9090 2.2094 0.0241
Aug 23 1973 3.137 37.406 2.095
0.2168 11.9241 0.0079 0.5390 23.8483 0.0061 3.5089 5.9621 0.0290
Aug 27 1973 0.370 1.535 4.255
0.3362 4.1486 0.0064 0.8693 8.2973 0.0054 1.9289 2.0743 0.0245
Aug 28 1973 0.171 0.099 3.153
0.6723 0.5789 0.0850 1.7795 1.1579 0.0744 0.5389 0.2895 0.3258
Aug 29 1973 0.838 0.800 1.307
0.4262 0.9547 0.1015 1.1390 1.9093 0.0906 0.5689 0.4773 0.3969
Aug 31 1973 0.017 0.005 4.736
0.6323 0.2941 0.0147 1.6789 0.5882 0.0130 0.2589 0.1471 0.0570
Sept 5 1973 0.030 0.031 7.237
0.8426 1.0333 0.0131 2.1990 2.0667 0.0112 1.2089 0.5167 0.0506
Sept 6 1973 0.007 0.002 7.988
0.9618 0.2857 0.0144 2.5189 0.5714 0.0124 0.3789 0.1429 0.0549
Sept 9 1973 0.144 0.412 10.366
0.4959 2.8611 0.0079 1.2690 5.7222 0.0064 1.9590 1.4306 0.0300
Sept 15 1973 0.035 0.008 2.891
0.7364 0.2286 0.0529 1.9389 0.4571 0.0458 0.2390 0.1143 0.2132
Sept 16 1973 4.790 32.955 1.151
0.3973 6.8800 0.0700 1.0489 13.7599 0.0610 3.8289 3.4400 0.2746
Sept 20 1973 2.945 12.521 2.170
0.2577 4.2516 0.0293 0.6589 8.5032 0.0239 1.5190 2.1258 0.1126
June 13 1974 0.009 0.004 10.211
1.0612 0.4444 0.0145 2.7790 0.8889 0.0124 0.6590 0.2222 0.0567
June 24 1974 0.214 0.220 2.666
0.5582 1.0280 0.0413 1.4889 2.0561 0.0367 0.8094 0.5140 0.1643
June 25 1974 0.465 0.740 2.240
0.3979 1.5914 0.0295 1.0489 3.1828 0.0256 0.8889 0.7957 0.1161
June 26 1974 3.877 4.625 0.357
0.4361 1.1929 0.3935 1.1290 2.3859 0.3297 0.7189 0.5965 1.5028
June 27 1974 0.013 0.006 7.313

0.9500	0.4615	0.0162	2.4690	0.9231	0.0137	0.6090	0.2308	0.0624
July 1 1974	0.027	0.020	6.220					
0.5347	0.7407	0.0066	1.3990	1.4815	0.0057	0.5489	0.3704	0.0255
July 2 1974	6.290	37.574	1.036					
0.3958	5.9736	0.1050	1.0589	11.9472	0.0940	3.3393	2.9868	0.4189
July 3 1974	6.672	65.137	1.499					
0.1890	9.7627	0.0155	0.4589	19.5255	0.0115	2.4989	4.8814	0.0570
July 5 1974	0.004	0.001	7.889					
0.9372	0.2500	0.0089	2.4590	0.5000	0.0077	0.3290	0.1250	0.0353
July 6 1974	0.359	0.434	2.289					
0.3556	1.2089	0.0239	0.9390	2.4178	0.0208	0.5989	0.6045	0.0928
July 7 1974	10.464	26.176	0.596					
0.2315	2.5015	0.1427	0.5789	5.0031	0.1116	0.7990	1.2508	0.5434
July 9 1974	0.001	0.000	16.758					
1.1052	0.5000	0.0016	2.8390	1.0000	0.0013	0.7589	0.2500	0.0059
July 13 1974	8.381	56.733	1.276					
0.2784	6.7692	0.0611	0.7190	13.5385	0.0509	2.5989	3.3846	0.2324
July 14 1974	8.246	27.744	0.683					
0.2090	3.3645	0.0682	0.5190	6.7291	0.0525	0.9489	1.6823	0.2482
July 15 1974	1.673	9.931	4.386					
0.4076	5.9360	0.0298	1.0590	11.8721	0.0252	3.3589	2.9680	0.1149
July 16 1974	5.585	16.026	0.509					
0.2249	2.8695	0.0627	0.5691	5.7389	0.0502	0.8890	1.4347	0.2379
July 17 1974	0.819	2.553	2.242					
0.4884	3.1172	0.0399	1.3090	6.2344	0.0358	2.1490	1.5586	0.1590
July 18 1974	10.433	14.517	0.247					
0.3010	1.3915	0.4325	0.7790	2.7829	0.3621	0.5789	0.6957	1.6524
July 19 1974	11.782	39.378	0.570					
0.3272	3.3422	0.2403	0.8490	6.6844	0.2022	1.5190	1.6711	0.9271
July 20 1974	1.439	2.839	2.242					
0.3616	1.9729	0.0607	0.9589	3.9458	0.0534	0.9989	0.9864	0.2381
July 21 1974	0.187	0.470	5.418					
0.4623	2.5134	0.0101	1.2089	5.0267	0.0087	1.6189	1.2567	0.0393
July 23 1974	0.033	0.017	4.819					
0.6468	0.5152	0.0171	1.7189	1.0303	0.0151	0.4690	0.2576	0.0676
July 26 1974	2.397	7.638	4.962					
0.5314	3.1865	0.1352	1.4389	6.3730	0.1239	2.4090	1.5932	0.5474
July 27 1974	3.507	14.548	1.338					
0.3794	4.1483	0.0775	0.9890	8.2965	0.0658	2.1890	2.0741	0.2997
July 28 1974	22.431	89.050	0.786					
0.3247	3.9700	0.3792	0.8389	7.9399	0.3164	1.7890	1.9850	1.4609
July 29 1974	2.463	6.805	1.162					
0.3416	2.7629	0.0662	0.9092	5.5258	0.0586	1.3290	1.3814	0.2626
July 31 1974	0.100	0.160	4.186					
0.3221	1.6000	0.0041	0.8390	3.2000	0.0035	0.7190	0.8000	0.0161
Aug 1 1974	14.666	49.583	0.525					
0.2871	3.3808	0.2276	0.7389	6.7616	0.1885	1.3389	1.6904	0.8663
Aug 2 1974	4.456	18.251	1.756					
0.3266	4.0958	0.0739	0.8489	8.1917	0.0624	1.8690	2.0479	0.2884
Aug 3 1974	1.243	5.806	1.930					
0.3365	4.6710	0.0192	0.8689	9.3419	0.0160	2.1789	2.3355	0.0737
Aug 4 1974	4.389	11.501	2.089					
0.3497	2.6204	0.1304	0.9189	5.2408	0.1125	1.2789	1.3102	0.5080
Aug 5 1974	10.173	268.588	1.600					
0.2540	26.4020	0.0158	0.6389	52.8041	0.0125	9.2190	13.2010	0.0598
Aug 9 1974	0.018	0.007	5.024					
0.3677	0.3889	0.0040	0.9389	0.7778	0.0032	0.1990	0.1944	0.0154
Aug 13 1974	7.859	50.299	1.189					
0.3254	6.4002	0.0828	0.8490	12.8004	0.0704	2.8990	3.2001	0.3208
Aug 14 1974	8.182	74.329	1.056					

	0.2280	9.0845	0.0298	0.5789	18.1689	0.0240	2.8689	4.5422	0.1144
Aug 15 1974	0.594	11.570	9.877						
	0.2211	19.4781	0.0009	0.5489	38.9562	0.0007	5.8689	9.7391	0.0035
Aug 18 1974	9.353	133.827	1.719						
	0.2233	14.3085	0.0207	0.5590	28.6169	0.0163	4.3590	7.1542	0.0772
Aug 19 1974	5.409	43.234	1.462						
	0.3208	7.9930	0.0443	0.8390	15.9859	0.0379	3.5689	3.9965	0.1718
Aug 22 1974	2.139	18.148	3.218						
	0.3087	8.4843	0.0153	0.7989	16.9687	0.0128	3.6489	4.2422	0.0594
Aug 23 1974	0.012	0.003	4.867						
	0.5242	0.2500	0.0084	1.3689	0.5000	0.0072	0.1789	0.1250	0.0313
Aug 24 1974	0.844	0.812	1.215						
	0.2688	0.9621	0.0404	0.6989	1.9242	0.0341	0.3589	0.4810	0.1554
Aug 25 1974	0.207	0.203	3.053						
	0.4576	0.9807	0.0281	1.1990	1.9614	0.0241	0.6290	0.4903	0.1106
Sept 1 1974	0.075	0.163	9.175						
	0.4894	2.1733	0.0053	1.2689	4.3467	0.0044	1.4789	1.0867	0.0203
Sept 2 1974	13.582	103.381	0.926						
	0.3643	7.6116	0.1508	0.9689	15.2232	0.1333	3.9090	3.8058	0.5992
Sept 3 1974	2.915	17.547	1.751						
	0.2365	6.0196	0.0172	0.5889	12.0391	0.0134	1.9389	3.0098	0.0640
Sept 4 1974	0.540	1.834	3.608						
	0.3520	3.3963	0.0125	0.9190	6.7926	0.0107	1.6589	1.6981	0.0483
Sept 5 1974	0.072	0.239	8.410						
	0.6911	3.3194	0.0066	1.8089	6.6389	0.0056	3.2090	1.6597	0.0258
Sept 6 1974	3.828	50.361	2.270						
	0.3797	13.1560	0.0267	0.9989	26.3119	0.0231	7.0089	6.5780	0.1052
Sept 9 1974	0.441	1.099	2.775						
	0.2203	2.4921	0.0055	0.5490	4.9841	0.0042	0.7490	1.2460	0.0204
Sept 12 1974	4.212	17.190	1.261						
	0.2856	4.0812	0.0536	0.7390	8.1624	0.0449	1.6190	2.0406	0.2068
Sept 14 1974	1.057	1.733	1.639						
	0.3810	1.6395	0.0596	1.0090	3.2791	0.0522	0.8790	0.8198	0.2359
Sept 15 1974	0.017	0.005	4.569						
	0.4893	0.2941	0.0088	1.2890	0.5882	0.0076	0.1989	0.1471	0.0337
Sept 16 1974	0.036	0.007	2.554						
	0.5333	0.1944	0.0335	1.3789	0.3889	0.0280	0.1389	0.0972	0.1203
Sept 18 1974	1.858	16.355	3.594						
	0.3654	8.8025	0.0179	0.9590	17.6050	0.0154	4.4890	4.4012	0.0699
Sept 19 1974	0.995	1.238	1.164						
	0.5207	1.2442	0.1380	1.3489	2.4884	0.1158	0.8992	0.6221	0.5318
Sept 20 1974	1.871	4.720	1.603						
	0.3529	2.5227	0.0588	0.9189	5.0454	0.0498	1.2389	1.2614	0.2277
Sept 21 1974	0.264	0.233	2.022						
	0.2347	0.8826	0.0105	0.5990	1.7652	0.0085	0.2890	0.4413	0.0408
Sept 22 1974	1.604	1.228	0.767						
	0.4992	0.7656	0.3324	1.3495	1.5312	0.3036	0.5389	0.3828	1.3217
Sept 24 1974	12.851	34.756	0.369						
	0.4137	2.7045	0.5177	1.0890	5.4091	0.4484	1.5589	1.3523	2.0100
Sept 25 1974	0.314	0.927	4.065						
	0.4257	2.9522	0.0123	1.1089	5.9045	0.0104	1.7489	1.4761	0.0475
Sept 26 1974	0.088	0.023	2.155						
	0.5917	0.2614	0.0750	1.4789	0.5227	0.0586	0.2089	0.1307	0.2739
Sept 27 1974	0.004	0.001	12.282						
	0.7795	0.2500	0.0062	1.9589	0.5000	0.0049	0.2690	0.1250	0.0236
June 6 1975	0.090	0.017	1.137						
	0.3716	0.1889	0.0419	0.9790	0.3778	0.0363	0.0990	0.0944	0.1666
July 1 1975	3.306	8.822	1.331						
	0.4221	2.6685	0.1405	1.1190	5.3370	0.1234	1.5790	1.3342	0.5523
July 2 1975	0.287	0.419	3.221						

0.5266 1.4599 0.0347 1.3993 2.9199 0.0306 1.0789 0.7300 0.1367
July 3 1975 0.683 0.702 1.673
0.3193 1.0278 0.0431 0.8189 2.0556 0.0355 0.4489 0.5139 0.1614
July 4 1975 0.168 0.308 4.984
0.4748 1.8333 0.0132 1.2489 3.6667 0.0114 1.2189 0.9167 0.0516
July 5 1975 11.213 35.217 0.972
0.4640 3.1407 0.4893 1.2690 6.2815 0.4575 2.0689 1.5704 1.9725
July 6 1975 0.008 0.002 5.637
0.6344 0.2500 0.0082 1.6589 0.5000 0.0070 0.2189 0.1250 0.0312
July 7 1975 9.220 11.848 0.579
0.3867 1.2850 0.6830 1.0089 2.5701 0.5812 0.6889 0.6425 2.6255
July 8 1975 2.631 14.682 1.553
0.3520 5.5804 0.0372 0.9090 11.1608 0.0310 2.7190 2.7902 0.1425
July 10 1975 0.666 3.441 4.277
0.5646 5.1667 0.0262 1.4590 10.3333 0.0218 4.0289 2.5833 0.0998
July 11 1975 1.597 2.674 2.775
0.3238 1.6744 0.0637 0.8389 3.3488 0.0534 0.7489 0.8372 0.2429
July 12 1975 13.577148.826 1.214
0.3157 10.9616 0.0786 0.8189 21.9233 0.0661 4.8090 5.4808 0.3035
July 17 1975 23.926344.302 0.562
0.3670 14.3903 0.1426 0.9689 28.7806 0.1242 7.3989 7.1951 0.5596
July 18 1975 2.205 15.988 2.356
0.3831 7.2508 0.0284 1.0190 14.5016 0.0251 3.8989 3.6254 0.1120
July 19 1975 0.706 0.899 1.750
0.4652 1.2734 0.0764 1.2090 2.5467 0.0645 0.8189 0.6367 0.2920
July 20 1975 0.219 0.154 1.975
0.4172 0.7032 0.0345 1.0790 1.4064 0.0289 0.4090 0.3516 0.1341
July 21 1975 9.960 39.055 1.742
0.2985 3.9212 0.1441 0.7690 7.8424 0.1195 1.6189 1.9606 0.5513
July 22 1975 13.452101.800 1.396
0.2988 7.5676 0.1010 0.7689 15.1353 0.0836 3.1289 3.7838 0.3869
July 23 1975 8.849 57.206 1.431
0.2458 6.4647 0.0526 0.6189 12.9294 0.0417 2.1794 3.2323 0.1981
July 24 1975 5.522 5.130 0.772
0.2652 0.9290 0.2661 0.6789 1.8580 0.2180 0.3389 0.4645 1.0071
July 25 1975 0.135 0.138 3.413
0.5015 1.0222 0.0211 1.3190 2.0444 0.0183 0.7190 0.5111 0.0832
July 26 1975 0.206 0.303 3.605
0.5087 1.4709 0.0231 1.3589 2.9417 0.0206 1.0590 0.7354 0.0924
July 27 1975 0.511 0.787 2.118
0.3799 1.5401 0.0305 0.9991 3.0802 0.0264 0.8190 0.7701 0.1195
July 28 1975 2.219 44.828 5.452
0.2802 20.2019 0.0055 0.7089 40.4038 0.0044 7.7890 10.1009 0.0208
July 29 1975 1.305 4.076 1.948
0.2717 3.1234 0.0196 0.6990 6.2467 0.0162 1.1689 1.5617 0.0745
July 30 1975 0.032 0.027 7.682
0.6494 0.8438 0.0102 1.6489 1.6875 0.0082 0.7489 0.4219 0.0380
Aug 8 1975 2.734 2.312 0.545
0.3543 0.8456 0.2584 0.9190 1.6913 0.2173 0.4190 0.4228 1.0106
Aug 9 1975 0.487 1.442 3.962
0.4458 2.9610 0.0208 1.1390 5.9220 0.0170 1.8190 1.4805 0.0790
Aug 10 1975 0.897 2.150 2.055
0.3276 2.3969 0.0256 0.8590 4.7938 0.0220 1.0990 1.1984 0.1002
Aug 11 1975 1.326 0.661 0.249
0.4948 0.4985 0.4146 1.2789 0.9970 0.3462 0.3389 0.2492 1.5654
Aug 12 1975 0.529 1.073 2.504
0.5418 2.0284 0.0487 1.4489 4.0567 0.0436 1.5489 1.0142 0.1936
Aug 13 1975 0.119 0.143 3.917
0.4103 1.2017 0.0106 1.0589 2.4034 0.0088 0.6789 0.6008 0.0402
Aug 18 1975 0.017 0.010 8.595

```
 0.6344   0.5882   0.0074   1.6689   1.1765   0.0064   0.5189   0.2941   0.0286
Aug  19 1975  0.794  0.174  0.250
 0.6083   0.2191   0.8535   1.6289   0.4383   0.7650   0.1890   0.1096   3.4314
Aug  22 1975  5.665 47.204  1.797
 0.2757   8.3326   0.0329   0.7090  16.6651   0.0272   3.1890   4.1663   0.1268
Aug  23 1975  5.399 26.373  1.147
 0.3871   4.8848   0.1054   1.0390   9.7696   0.0949   2.6690   2.4424   0.4201
Sept  1 1975  3.344 20.181  1.495
 0.3594   6.0350   0.0456   0.9390  12.0700   0.0389   3.0189   3.0175   0.1765
Sept  2 1975  0.717  1.967  2.651
 0.3632   2.7434   0.0219   0.9690   5.4868   0.0195   1.3989   1.3717   0.0865
Sept  3 1975  9.955153.880  2.464
 0.3173  15.4576   0.0413   0.8189  30.9151   0.0344   6.8090   7.7288   0.1591
Sept  4 1975  1.674  4.291  1.527
 0.3951   2.5633   0.0649   1.0489   5.1266   0.0572   1.4290   1.2817   0.2584
Sept  5 1975 14.946 85.242  1.729
 0.3959   5.7033   0.2615   1.0290  11.4067   0.2208   3.1289   2.8517   1.0042
Sept  6 1975  0.845  0.280  0.437
 0.3875   0.3314   0.2438   1.0190   0.6627   0.2107   0.1789   0.1657   0.9464
Sept  7 1975 10.665 35.272  0.518
 0.4179   3.3073   0.3585   1.1189   6.6145   0.3213   1.9489   1.6536   1.4257
Sept  8 1975  2.922 15.536  0.995
 0.1990   5.3169   0.0139   0.4890  10.6338   0.0105   1.4289   2.6585   0.0505
Sept 11 1975  0.192  0.292  3.240
 0.3164   1.5208   0.0080   0.8189   3.0417   0.0067   0.6689   0.7604   0.0311
Sept 12 1975  1.811  5.088  2.602
 0.3986   2.8095   0.0652   1.0489   5.6190   0.0564   1.5690   1.4047   0.2560
Sept 13 1975 19.634 52.091  0.464
 0.3362   2.6531   0.5325   0.8789   5.3062   0.4549   1.2389   1.3266   2.0546
Sept 30 1975  0.025  0.006  3.625
 0.6090   0.2400   0.0246   1.5992   0.4800   0.0212   0.1989   0.1200   0.0911
June 28 1976  2.413  4.232  1.532
 0.4209   1.7538   0.1552   1.0789   3.5077   0.1274   1.0189   0.8769   0.5912
June 29 1976  1.216  2.067  2.141
 0.4636   1.6998   0.0979   1.2490   3.3997   0.0888   1.1089   0.8499   0.3876
June 30 1976  1.541  5.517  1.737
 0.3597   3.5801   0.0355   0.9389   7.1603   0.0302   1.7990   1.7901   0.1384
July  1 1976  0.056  0.051  5.096
 0.6360   0.9107   0.0158   1.6689   1.8214   0.0136   0.8090   0.4554   0.0618
July  3 1976  1.236  8.155  3.108
 0.4043   6.5979   0.0195   1.0789  13.1958   0.0174   3.7689   3.2989   0.0778
July  5 1976  0.779  3.156  2.858
 0.2935   4.0513   0.0105   0.7590   8.1027   0.0088   1.6490   2.0257   0.0406
July 10 1976  2.018  3.036  0.830
 0.2809   1.5045   0.0674   0.7290   3.0089   0.0567   0.5890   0.7522   0.2618
July 11 1976 10.584 40.663  1.110
 0.4189   3.8419   0.3078   1.0889   7.6839   0.2599   2.2390   1.9210   1.1913
July 12 1976  6.179 63.370  1.853
 0.3622  10.2557   0.0503   0.9489  20.5114   0.0432   5.1990   5.1279   0.1971
July 14 1976  0.168  0.082  2.754
 0.7415   0.4881   0.1205   1.9390   0.9762   0.1030   0.4989   0.2440   0.4579
July 15 1976  6.229  7.180  0.646
 0.2398   1.1527   0.1978   0.6090   2.3053   0.1595   0.3789   0.5763   0.7435
July 16 1976  1.544  4.599  1.247
 0.3364   2.9786   0.0373   0.8689   5.9573   0.0311   1.3889   1.4893   0.1435
July 17 1976  0.655  0.302  0.813
 0.5353   0.4611   0.2591   1.4293   0.9221   0.2309   0.3490   0.2305   1.0363
July 18 1976  6.293 56.707  1.162
 0.1980   9.0111   0.0174   0.4789  18.0222   0.0127   2.4089   4.5056   0.0635
July 19 1976  5.503 15.822  1.155
```

0.2774 2.8752 0.0938 0.7089 5.7503 0.0765 1.1090 1.4376 0.3626
July 20 1976 5.340 29.719 0.887
0.2402 5.5654 0.0352 0.6090 11.1307 0.0283 1.8390 2.7827 0.1334
July 21 1976 0.963 0.889 1.093
0.3745 0.9232 0.0931 0.9989 1.8463 0.0828 0.4890 0.4616 0.3727
July 22 1976 0.574 2.497 3.622
0.3899 4.3502 0.0128 1.0290 8.7003 0.0111 2.3689 2.1751 0.0498
July 23 1976 4.129 5.717 0.041
0.2788 1.3846 0.1476 0.7189 2.7692 0.1226 0.5390 0.6923 0.5754
July 24 1976 1.172 4.425 3.145
0.3978 3.7756 0.0313 1.0490 7.5512 0.0272 2.0990 1.8878 0.1222
July 25 1976 0.386 0.216 1.099
0.4603 0.5596 0.0930 1.1689 1.1192 0.0750 0.3489 0.2798 0.3414
July 26 1976 6.422 34.026 1.496
0.2604 5.2983 0.0523 0.6690 10.5967 0.0432 1.9089 2.6492 0.2003
July 27 1976 19.132105.256 0.485
0.2914 5.5016 0.1880 0.7490 11.0031 0.1552 2.2190 2.7508 0.7203
July 28 1976 15.957293.033 4.453
0.1832 18.3639 0.0186 0.4490 36.7278 0.0139 4.5490 9.1820 0.0679
July 29 1976 0.878 15.051 6.385
0.1457 17.1424 0.0007 0.3389 34.2847 0.0005 3.3289 8.5712 0.0025
July 30 1976 0.313 0.454 3.466
0.6775 1.4505 0.0631 1.8190 2.9010 0.0568 1.3789 0.7252 0.2483
Aug 7 1976 0.060 0.051 4.478
0.5626 0.8500 0.0142 1.4890 1.7000 0.0125 0.6689 0.4250 0.0557
Aug 8 1976 1.353 0.313 0.841
0.5199 0.2313 1.0064 1.3490 0.4627 0.8470 0.1690 0.1157 3.9741
Aug 9 1976 0.043 0.042 5.479
0.5108 0.9767 0.0073 1.3389 1.9535 0.0063 0.6990 0.4884 0.0287
Aug 10 1976 9.947102.647 1.000
0.3230 10.3194 0.0640 0.8490 20.6388 0.0553 4.6489 5.1597 0.2491
Aug 12 1976 3.552 14.512 1.008
0.2520 4.0856 0.0351 0.6389 8.1712 0.0282 1.4190 2.0428 0.1335
Aug 16 1976 2.304 0.595 -0.424
0.4547 0.2582 1.1743 1.1389 0.5165 0.9209 0.1589 0.1291 4.3007
Aug 17 1976 5.007 47.026 1.659
0.2544 9.3921 0.0220 0.6490 18.7841 0.0179 3.2889 4.6960 0.0832
Aug 18 1976 0.179 0.431 6.119
0.5658 2.4078 0.0152 1.4590 4.8156 0.0126 1.8890 1.2039 0.0583
Aug 19 1976 2.776 2.013 -0.027
0.5473 0.7251 0.7300 1.4890 1.4503 0.6754 0.5589 0.3626 2.8955
Aug 21 1976 0.013 0.009 9.447
0.5618 0.6923 0.0038 1.4789 1.3846 0.0033 0.5389 0.3462 0.0145
Aug 22 1976 7.481 92.770 3.341
0.2699 12.4007 0.0280 0.6889 24.8015 0.0228 4.6089 6.2004 0.1061
Aug 23 1976 0.065 0.029 3.136
0.3350 0.4462 0.0104 0.8789 0.8923 0.0090 0.2089 0.2231 0.0407
Aug 24 1976 0.243 0.414 3.881
0.2957 1.7037 0.0079 0.7490 3.4074 0.0064 0.6889 0.8519 0.0297
Aug 25 1976 0.179 0.098 1.739
0.3692 0.5475 0.0284 0.9790 1.0950 0.0249 0.2789 0.2737 0.1080
Aug 26 1976 1.726 14.643 2.637
0.2504 8.4838 0.0081 0.6390 16.9676 0.0066 2.9189 4.2419 0.0307
Aug 27 1976 2.055 2.480 0.829
0.4218 1.2068 0.1929 1.1290 2.4136 0.1727 0.7190 0.6034 0.7696
Aug 31 1976 0.789 0.364 1.117
0.6567 0.4613 0.4695 1.7290 0.9227 0.4068 0.4189 0.2307 1.7953
Sept 1 1976 1.253 1.273 1.398
0.3781 1.0160 0.1122 0.9590 2.0319 0.0903 0.5290 0.5080 0.4257
Sept 4 1976 8.629 9.365 0.127

0.3401	1.0853	0.5855	0.8789	2.1706	0.4887	0.5089	0.5426	2.2258
Sept 5 1976	16.489	125.754	0.611					
0.2254	7.6265	0.0699	0.5589	15.2531	0.0537	2.3389	3.8133	0.2589
Sept 6 1976	15.068	64.719	0.118					
0.3757	4.2951	0.3152	0.9989	8.5903	0.2786	2.2790	2.1476	1.2576
Sept 8 1976	1.518	1.288	0.425					
0.2054	0.8485	0.0481	0.5089	1.6970	0.0369	0.2390	0.4242	0.1807
Sept 10 1976	8.179	26.529	0.029					
0.3074	3.2436	0.1517	0.7990	6.4871	0.1281	1.3789	1.6218	0.5802
Sept 23 1976	0.615	0.817	2.457					
0.4751	1.3285	0.0665	1.2589	2.6569	0.0584	0.8890	0.6642	0.2640
Sept 24 1976	6.990	5.008	0.033					
0.4712	0.7165	1.3791	1.2389	1.4329	1.1917	0.4689	0.3582	5.3209
Sept 25 1976	0.465	0.530	1.677					
0.4828	1.1398	0.0605	1.2490	2.2796	0.0506	0.7589	0.5699	0.2303
Sept 26 1976	0.046	0.049	6.002					
0.5087	1.0652	0.0071	1.2990	2.1304	0.0058	0.7389	0.5326	0.0265
June 3 1977	0.406	1.105	2.890					
0.3691	2.7217	0.0129	0.9689	5.4433	0.0111	1.4090	1.3608	0.0509
June 4 1977	1.874	2.089	0.328					
0.1714	1.1147	0.0314	0.4089	2.2295	0.0224	0.2590	0.5574	0.1156
June 7 1977	0.436	0.444	2.559					
0.3221	1.0183	0.0283	0.8389	2.0367	0.0240	0.4593	0.5092	0.1109
June 8 1977	1.809	3.912	1.294					
0.3497	2.1625	0.0651	0.9089	4.3250	0.0550	1.0489	1.0813	0.2506
June 21 1977	1.383	4.173	2.102					
0.5704	3.0174	0.0949	1.5489	6.0347	0.0875	2.4490	1.5087	0.3844
June 22 1977	5.952	31.944	0.863					
0.2578	5.3669	0.0469	0.6590	10.7339	0.0383	1.9089	2.6835	0.1786
July 1 1977	1.012	0.725	1.005					
0.5240	0.7164	0.2469	1.3790	1.4328	0.2138	0.5189	0.3582	0.9436
July 2 1977	0.510	0.210	1.100					
0.5569	0.4118	0.2445	1.4089	0.8235	0.1956	0.3190	0.2059	0.9465
July 3 1977	17.294	55.486	0.515					
0.3867	3.2084	0.5131	1.0290	6.4168	0.4542	1.7389	1.6042	2.0160
July 4 1977	0.307	0.843	3.874					
0.4662	2.7459	0.0155	1.2189	5.4919	0.0132	1.7890	1.3730	0.0604
July 5 1977	1.631	10.968	3.336					
0.3529	6.7247	0.0192	0.9189	13.4494	0.0163	3.3089	3.3624	0.0748
July 8 1977	0.006	0.001	6.922					
0.6668	0.1667	0.0102	1.7490	0.3333	0.0088	0.1489	0.0833	0.0366
July 9 1977	0.232	0.196	2.738					
0.5458	0.8448	0.0521	1.3589	1.6897	0.0404	0.6290	0.4224	0.1938
July 10 1977	0.197	0.096	1.979					
0.6038	0.4873	0.0938	1.5189	0.9746	0.0742	0.3989	0.2437	0.3449
July 11 1977	1.294	5.727	2.356					
0.3201	4.4258	0.0191	0.8390	8.8516	0.0164	1.9790	2.2129	0.0744
July 12 1977	0.290	1.508	5.854					
0.7500	5.2000	0.0200	1.8389	10.4000	0.0150	5.2589	2.6000	0.0726
July 13 1977	2.902	8.763	1.073					
0.3505	3.0196	0.0752	0.9090	6.0393	0.0632	1.4690	1.5098	0.2896
July 14 1977	1.172	0.453	0.579					
0.4231	0.3865	0.3456	1.0790	0.7730	0.2809	0.2189	0.1933	1.2383
July 16 1977	0.014	0.006	6.622					
0.8643	0.4286	0.0155	2.2590	0.8571	0.0133	0.5190	0.2143	0.0610
July 17 1977	0.027	0.034	8.886					
0.5127	1.2593	0.0036	1.3489	2.5185	0.0031	0.8989	0.6296	0.0139
July 18 1977	0.302	0.146	1.104					
0.5438	0.4834	0.1176	1.3890	0.9669	0.0959	0.3589	0.2417	0.4384
July 19 1977	1.733	7.711	2.976					

0.3591	4.4495	0.0320	0.9289	8.8990	0.0267	2.2189	2.2248	0.1233
July 20 1977	0.054	0.031	4.310					
0.6132	0.5741	0.0225	1.5889	1.1481	0.0189	0.4890	0.2870	0.0869
July 21 1977	2.245	3.028	0.730					
0.3504	1.3488	0.1301	0.8989	2.6976	0.1070	0.6489	0.6744	0.4905
July 22 1977	12.713	26.445	1.197					
0.2035	2.0802	0.1611	0.5090	4.1603	0.1260	0.5790	1.0401	0.6029
July 23 1977	5.068	4.036	0.366					
0.2103	0.7964	0.1792	0.5290	1.5927	0.1417	0.2289	0.3982	0.6694
July 24 1977	0.003	0.001	10.227					
0.9200	0.3333	0.0048	2.3989	0.6667	0.0041	0.4290	0.1667	0.0190
July 25 1977	1.176	10.192	4.290					
0.3709	8.6667	0.0119	0.9689	17.3333	0.0101	4.4989	4.3333	0.0466
July 27 1977	0.065	0.062	5.157					
0.4871	0.9538	0.0103	1.2790	1.9077	0.0089	0.6490	0.4769	0.0402
July 28 1977	0.837	1.923	2.581					
0.4347	2.2975	0.0438	1.1490	4.5950	0.0383	1.3990	1.1487	0.1720
July 30 1977	0.498	1.132	3.600					
0.2319	2.2731	0.0075	0.5889	4.5462	0.0060	0.7290	1.1365	0.0287
July 31 1977	8.881	39.312	0.456					
0.2101	4.4265	0.0564	0.5189	8.8531	0.0430	1.2690	2.2133	0.2099
Aug 1 1977	8.409	40.985	5.676					
0.3551	4.8739	0.1385	0.9489	9.7479	0.1236	2.4389	2.4370	0.5500
Aug 6 1977	2.587	6.839	2.019					
0.3325	2.6436	0.0689	0.8590	5.2872	0.0575	1.2192	1.3218	0.2650
Aug 8 1977	0.617	1.328	3.134					
0.4300	2.1524	0.0337	1.1389	4.3047	0.0296	1.2990	1.0762	0.1329
Aug 9 1977	0.389	3.825	8.884					
0.2375	9.8329	0.0014	0.5889	19.6658	0.0011	3.1889	4.9165	0.0053
Aug 10 1977	0.080	0.016	1.481					
0.5696	0.2000	0.0826	1.3490	0.4000	0.0579	0.1489	0.1000	0.2823
Aug 11 1977	5.039	11.869	2.854					
0.5363	2.3554	0.3917	1.3489	4.7109	0.3098	1.7290	1.1777	1.4677
Aug 12 1977	0.477	1.211	2.768					
0.4083	2.5388	0.0199	1.0090	5.0776	0.0152	1.3989	1.2694	0.0726
Aug 13 1977	4.176	61.200	3.111					
0.4712	14.6552	0.0403	1.2695	29.3103	0.0365	9.7690	7.3276	0.1612
Aug 14 1977	3.295	4.619	1.047					
0.3046	1.4018	0.1388	0.7890	2.8036	0.1164	0.5889	0.7009	0.5282
Aug 15 1977	29.362	167.995	0.474					
0.3395	5.7215	0.3766	0.8989	11.4430	0.3300	2.7189	2.8608	1.4755
Aug 16 1977	0.480	3.077	10.653					
1.0673	6.4104	0.0543	2.7690	12.8208	0.0457	9.4789	3.2052	0.2085
Aug 17 1977	0.010	0.004	8.661					
0.8854	0.4000	0.0125	2.2890	0.8000	0.0104	0.4889	0.2000	0.0476
Aug 18 1977	1.267	2.290	1.775					
0.4034	1.8074	0.0726	1.0590	3.6148	0.0626	1.0190	0.9037	0.2837
Aug 19 1977	0.077	0.064	3.492					
0.4564	0.8312	0.0123	1.2089	1.6623	0.0108	0.5289	0.4156	0.0478
Aug 20 1977	5.494	9.799	1.103					
0.3107	1.7836	0.1893	0.8091	3.5672	0.1605	0.7689	0.8918	0.7289
Aug 21 1977	0.006	0.001	7.454					
0.9051	0.1667	0.0188	2.3689	0.3333	0.0161	0.2089	0.0833	0.0720
Aug 22 1977	0.086	0.071	4.103					
0.5299	0.8256	0.0186	1.2690	1.6512	0.0133	0.5789	0.4128	0.0652
Aug 23 1977	1.866	7.682	3.511					
0.2687	4.1168	0.0208	0.6789	8.2337	0.0166	1.5189	2.0584	0.0786
Aug 24 1977	0.780	0.850	1.383					
0.3653	1.0897	0.0608	0.9790	2.1795	0.0546	0.5589	0.5449	0.2397
Aug 31 1977	2.130	7.910	2.169					

0.5578	3.7136	0.1136	1.4790	7.4272	0.0998	2.8993	1.8568	0.4451
Sept 1 1977	28.868104.043	-0.210						
0.4805	3.6041	1.1773	1.3089	7.2082	1.0920	2.4690	1.8020	4.7861
Sept 2 1977	2.771	26.234	2.576					
0.2763	9.4673	0.0142	0.7090	18.9347	0.0117	3.5989	4.7337	0.0539
Sept 4 1977	2.148	3.858	1.185					
0.4001	1.7961	0.1219	1.0489	3.5922	0.1047	1.0090	0.8980	0.4806
Sept 5 1977	11.748	63.639	0.226					
0.2033	5.4170	0.0571	0.4989	10.8340	0.0430	1.4889	2.7085	0.2086
Sept 10 1977	0.039	0.015	4.392					
0.6771	0.3846	0.0296	1.6689	0.7692	0.0225	0.3489	0.1923	0.1062
Sept 11 1977	10.810	18.235	1.949					
0.3664	1.6869	0.5477	0.9489	3.3737	0.4592	0.8590	0.8434	2.1158
Sept 12 1977	0.045	0.013	2.776					
0.3818	0.2889	0.0145	0.9689	0.5778	0.0116	0.1489	0.1444	0.0527
Sept 22 1977	2.303	3.799	1.484					
0.2620	1.6496	0.0610	0.6689	3.2992	0.0497	0.5990	0.8248	0.2344
Sept 25 1977	0.276	0.339	2.930					
0.4255	1.2283	0.0259	1.1189	2.4565	0.0224	0.7289	0.6141	0.1008
Sept 26 1977	31.468439.948	0.829						
0.1935	13.9808	0.0537	0.4689	27.9616	0.0394	3.6590	6.9904	0.1963
Sept 27 1977	3.947	10.067	1.795					
0.4270	2.5505	0.1796	1.1590	5.1011	0.1654	1.5490	1.2753	0.7267

APPENDIX II

UNBIASED_MODEL_PARAMETERS.DATA

UNBIASED_MODEL_PARAMETERS.DATA

```
month, day, year, E(Y),VAR(Y),C.S.(Y)
    a1     E(alpha1)  lam1      a2   E(alpha2)   lam2       a3    E(alpha3)   lam3
  ....................  ......................    ......................
    428   records
June 22 1970  0.270  0.771  4.386 June 22 1970  0.270  0.771  4.386
  0.5656   3.1747   0.0173   1.4789   6.2748   0.0150   2.2590   1.6926   0.0452
June 28 1970  0.706  0.994  1.847
  0.4573   1.5653   0.0600   1.2089   3.0938   0.0531   0.8989   0.8346   0.1562
June 29 1970  0.517  3.046  4.609
  0.3559   6.5501   0.0064   0.9189  12.9463   0.0054   2.9189   3.4923   0.0165
July  1 1970  0.382  0.380  1.693
  0.3068   1.1059   0.0207   0.8089   2.1859   0.0182   0.4289   0.5896   0.0546
July  2 1970  0.033  0.032  7.127
  0.6563   1.0781   0.0084   1.6990   2.1308   0.0071   0.8789   0.5748   0.0214
July  7 1970  2.752  9.454  1.444
  0.2926   3.8193   0.0393   0.7489   7.5487   0.0325   1.3889   2.0363   0.1001
July  8 1970  0.531  1.111  2.109
  0.3670   2.3261   0.0196   0.9689   4.5975   0.0173   1.0790   1.2402   0.0516
July 13 1970  0.238  0.452  3.682
  0.5610   2.1114   0.0226   1.4989   4.1732   0.0204   1.4989   1.1257   0.0597
July 17 1970  0.115  0.081  2.976
  0.7132   0.7831   0.0476   1.8090   1.5477   0.0387   0.6890   0.4175   0.1194
July 18 1970  0.043  0.052  7.203
  0.9337   1.3445   0.0178   2.4590   2.6573   0.0156   1.5790   0.7168   0.0463
July 19 1970  4.610 22.077  1.588
  0.2460   5.3241   0.0334   0.6189  10.5231   0.0267   1.6089   2.8386   0.0830
July 20 1970 20.640185.010  0.997
  0.1965   9.9654   0.0509   0.4789  19.6966   0.0382   2.3790   5.3132   0.1240
July 21 1970  0.257  0.117  1.182
  0.2624   0.5061   0.0223   0.6689   1.0004   0.0183   0.1690   0.2699   0.0595
July 24 1970  7.969 48.831  0.549
  0.2747   6.8124   0.0562   0.6989  13.4647   0.0460   2.3290   3.6321   0.1436
July 25 1970  2.839 20.485  2.733
  0.3403   8.0220   0.0261   0.8989  15.8554   0.0230   3.4490   4.2770   0.0687
July 27 1970  1.176  4.878  2.216
  0.3865   4.6115   0.0243   1.0190   9.1147   0.0213   2.2389   2.4587   0.0631
July 28 1970  9.744 53.787  1.191
  0.1917   6.1369   0.0371   0.4690  12.1296   0.0281   1.4290   3.2720   0.0904
July 29 1970  2.278 28.928  2.663
  0.3546  14.1181   0.0129   0.9290  27.9043   0.0112   6.2689   7.5272   0.0334
July 30 1970  5.178  6.965  0.863
  0.3781   1.4954   0.3151   0.9689   2.9557   0.2617   0.6989   0.7973   0.7942
July 31 1970 10.129 64.582  0.445
  0.2706   7.0885   0.0666   0.6889  14.0104   0.0546   2.3790   3.7793   0.1690
Aug   1 1970 12.988 41.904  0.948
  0.2112   3.5869   0.1028   0.5189   7.0896   0.0785   0.9290   1.9124   0.2551
Aug   2 1970  7.012 21.048  0.484
  0.2728   3.3372   0.0995   0.6990   6.5959   0.0827   1.1289   1.7793   0.2525
Aug   3 1970  1.067  1.681  1.123
  0.3228   1.7515   0.0404   0.8389   3.4619   0.0345   0.7090   0.9338   0.1048
Aug   4 1970  0.552  2.095  3.067
  0.3447   4.2195   0.0099   0.8989   8.3397   0.0085   1.8290   2.2497   0.0258
Aug   5 1970  0.435  0.959  2.924
  0.3183   2.4510   0.0114   0.8189   4.8444   0.0096   0.9689   1.3068   0.0291
Aug   6 1970  0.399  2.776  5.196
  0.5412   7.7349   0.0096   1.4089  15.2881   0.0082   5.2390   4.1240   0.0249
Aug   7 1970  1.393  6.158  3.315
```

0.2604	4.9147	0.0122	0.6690	9.7139	0.0102	1.5889	2.6204	0.0311
Aug 8 1970	5.490	53.891	1.988					
0.3212	10.9133	0.0330	0.8392	21.5700	0.0285	4.3890	5.8186	0.0854
Aug 9 1970	7.552	36.318	1.613					
0.4278	5.3465	0.1646	1.1489	10.5674	0.1501	2.9089	2.8506	0.4391
Aug 10 1970	17.618	44.128	0.134					
0.4541	2.7846	0.8306	1.2490	5.5038	0.7948	1.6290	1.4847	2.2737
Aug 11 1970	0.503	5.236	7.259					
0.1819	11.5729	0.0009	0.4390	22.8738	0.0007	2.5389	6.1703	0.0022
Aug 14 1970	4.812	23.184	1.146					
0.4803	5.3564	0.1319	1.3094	10.5869	0.1240	3.2889	2.8558	0.3557
Aug 15 1970	2.177	6.466	1.498					
0.5439	3.3021	0.1242	1.4589	6.5265	0.1130	2.2789	1.7606	0.3297
Aug 16 1970	12.850	131.970	0.953					
0.2817	11.4178	0.0569	0.7189	22.5672	0.0468	3.9989	6.0876	0.1450
Aug 17 1970	5.587	50.220	2.187					
0.3445	9.9933	0.0422	0.8989	19.7517	0.0364	4.3190	5.3281	0.1097
Aug 18 1970	0.172	0.491	5.069					
0.6172	3.1737	0.0131	1.6289	6.2728	0.0116	2.4690	1.6921	0.0344
Aug 19 1970	1.073	5.556	2.748					
0.4423	5.7567	0.0232	1.1789	11.3781	0.0209	3.2189	3.0693	0.0612
Aug 21 1970	0.028	0.036	8.099					
0.8267	1.4294	0.0085	2.1589	2.8252	0.0074	1.4790	0.7621	0.0220
Aug 22 1970	4.121	14.551	0.924					
0.3186	3.9256	0.0678	0.8289	7.7588	0.0581	1.5689	2.0930	0.1761
Aug 23 1970	4.457	18.994	1.412					
0.2978	4.7379	0.0531	0.7589	9.3644	0.0436	1.7489	2.5261	0.1346
Aug 26 1970	1.907	7.569	1.301					
0.3945	4.4126	0.0428	1.0289	8.7216	0.0368	2.1890	2.3527	0.1117
Sept 2 1970	0.240	0.541	4.408					
0.6602	2.5061	0.0266	1.7289	4.9533	0.0231	2.0790	1.3362	0.0692
Sept 3 1970	11.431	3.543	0.687					
0.3853	0.3446	3.1352	0.9989	0.6811	2.6654	0.1690	0.1837	8.3792
Sept 4 1970	9.434	10.152	-0.021					
0.4233	1.1964	0.8995	1.1389	2.3646	0.8236	0.6389	0.6379	2.3616
Sept 8 1970	6.400	105.592	1.646					
0.3485	18.3426	0.0270	0.9189	36.2541	0.0237	8.0489	9.7796	0.0706
Sept 9 1970	8.003	10.486	0.933					
0.3168	1.4567	0.3510	0.8189	2.8791	0.2967	0.5790	0.7767	0.9115
Sept 11 1970	0.054	0.112	7.622					
0.7681	2.3059	0.0088	1.9989	4.5575	0.0075	2.2089	1.2294	0.0226
Sept 12 1970	0.055	0.020	2.829					
0.3847	0.4043	0.0128	1.0191	0.7990	0.0114	0.1990	0.2155	0.0346
Sept 30 1970	1.344	1.937	1.341					
0.5616	1.6023	0.1684	1.5590	3.1669	0.1642	1.1589	0.8543	0.4608
June 24 1971	6.621	40.557	1.471					
0.2567	6.8101	0.0408	0.6489	13.4601	0.0330	2.1589	3.6309	0.1026
July 2 1971	3.563	9.737	0.985					
0.6512	3.0382	0.3166	1.7189	6.0050	0.2790	2.4889	1.6199	0.8264
July 3 1971	0.377	0.993	3.691					
0.7680	2.9283	0.0483	1.9890	5.7878	0.0410	2.8090	1.5613	0.1244
July 5 1971	1.258	5.161	2.670					
0.2729	4.5610	0.0131	0.6989	9.0149	0.0108	1.5490	2.4318	0.0334
July 10 1971	0.561	2.365	3.282					
0.5031	4.6868	0.0193	1.3590	9.2635	0.0178	2.9989	2.4988	0.0515
July 13 1971	0.040	0.042	6.178					
0.4919	1.1673	0.0053	1.2889	2.3073	0.0046	0.7189	0.6224	0.0136
July 16 1971	3.065	11.816	1.985					
0.3528	4.2860	0.0567	0.8989	8.4712	0.0465	1.8789	2.2851	0.1443
July 17 1971	1.094	11.375	3.384					

0.3079 11.5597 0.0057 0.7889 22.8476 0.0047 4.4289 6.1632 0.0146
July 18 1971 6.209 40.297 1.646
0.2370 7.2154 0.0308 0.5889 14.2612 0.0240 2.0989 3.8470 0.0765
July 19 1971 0.652 1.651 2.959
0.4945 2.8152 0.0361 1.2790 5.5642 0.0305 1.7390 1.5010 0.0928
July 20 1971 1.905 13.831 2.764
0.4559 8.0718 0.0312 1.2089 15.9538 0.0278 4.6489 4.3036 0.0822
July 21 1971 1.652 22.417 3.996
0.3528 15.0861 0.0087 0.9089 29.8177 0.0073 6.6289 8.0434 0.0222
July 23 1971 23.769268.325 0.355
0.2419 12.5505 0.0706 0.6090 24.8060 0.0566 3.7290 6.6915 0.1756
July 24 1971 12.120 25.709 2.002
0.2979 2.3583 0.2904 0.7689 4.6611 0.2447 0.8790 1.2573 0.7498
July 26 1971 0.871 2.054 1.989
0.3640 2.6218 0.0280 0.9190 5.1819 0.0226 1.1790 1.3978 0.0706
July 27 1971 3.015 5.879 1.204
0.4861 2.1678 0.2092 1.3289 4.2847 0.1978 1.3489 1.1558 0.5655
July 28 1971 7.076 23.848 0.646
0.3338 3.7469 0.1340 0.8690 7.4058 0.1148 1.5589 1.9977 0.3433
July 29 1971 2.311 8.717 1.637
0.4307 4.1935 0.0651 1.1390 8.2885 0.0576 2.2790 2.2358 0.1709
July 30 1971 4.077 3.159 0.775
0.2866 0.8614 0.2475 0.7389 1.7026 0.2081 0.3094 0.4593 0.6412
July 31 1971 5.338 13.323 1.089
0.2917 2.7748 0.1042 0.7489 5.4844 0.0869 1.0089 1.4794 0.2671
Aug 1 1971 0.038 0.077 9.447
0.5609 2.2528 0.0034 1.4790 4.4526 0.0030 1.5890 1.2011 0.0088
Aug 3 1971 5.024 4.816 0.945
0.5568 1.0657 0.9304 1.5289 2.1064 0.8873 0.7589 0.5682 2.5102
Aug 8 1971 2.846 17.968 1.759
0.2362 7.0190 0.0144 0.5889 13.8730 0.0113 2.0389 3.7423 0.0359
Aug 10 1971 27.526174.063 0.221
0.2727 7.0303 0.1854 0.6990 13.8954 0.1540 2.3790 3.7483 0.4708
Aug 11 1971 1.566 9.080 2.160
0.2965 6.4462 0.0136 0.7690 12.7409 0.0116 2.3890 3.4369 0.0350
Aug 12 1971 20.985 53.595 0.031
0.3540 2.8394 0.5896 0.9090 5.6121 0.4917 1.2490 1.5139 1.5017
Aug 13 1971 0.007 0.002 7.886
0.9316 0.3176 0.0122 2.4389 0.6278 0.0106 0.3689 0.1694 0.0312
Aug 14 1971 4.350 12.415 1.656
0.3633 3.1730 0.1152 0.9389 6.2714 0.0973 1.4389 1.6917 0.2961
Aug 16 1971 1.059 4.251 3.219
0.4219 4.4628 0.0269 1.0889 8.8207 0.0227 2.3490 2.3794 0.0690
Aug 17 1971 2.015 12.369 1.825
0.2163 6.8245 0.0088 0.5390 13.4886 0.0069 1.8090 3.6386 0.0218
Aug 18 1971 26.033 56.754 -0.320
0.4033 2.4237 1.1122 1.0589 4.7905 0.9698 1.2289 1.2922 2.8997
Aug 19 1971 0.308 0.383 2.381
0.4255 1.3825 0.0257 1.1389 2.7325 0.0233 0.7490 0.7371 0.0687
Aug 20 1971 5.097 32.627 1.215
0.4331 7.1166 0.0855 1.1689 14.0659 0.0788 3.9289 3.7943 0.2292
Aug 21 1971 2.218 13.435 1.751
0.3940 6.7342 0.0325 1.0489 13.3101 0.0292 3.3589 3.5904 0.0860
Aug 23 1971 2.656 23.056 2.050
0.3453 9.6509 0.0209 0.8889 19.0749 0.0175 4.1489 5.1455 0.0534
Aug 24 1971 7.997 33.428 1.874
0.3806 4.6472 0.1587 0.9889 9.1852 0.1355 2.2190 2.4777 0.4120
Aug 27 1971 4.692 19.056 1.096
0.3030 4.5153 0.0607 0.7890 8.9244 0.0521 1.7090 2.4074 0.1563
Aug 28 1971 3.894 19.747 2.131

0.3591	5.6379	0.0567	0.9290	11.1432	0.0480	2.5189	3.0059	0.1448
Aug 31 1971	0.503	2.461	3.920					
0.4027	5.4394	0.0095	1.0593	10.7510	0.0084	2.7489	2.9001	0.0248
Sept 1 1971	3.872	21.587	3.382					
0.3188	6.1982	0.0404	0.8189	12.2508	0.0337	2.4489	3.3047	0.1024
Sept 2 1971	1.063	3.560	1.711					
0.3023	3.7233	0.0166	0.7789	7.3591	0.0139	1.4090	1.9851	0.0429
Sept 3 1971	1.366	4.099	1.735					
0.5452	3.3361	0.0775	1.3789	6.5938	0.0627	2.2490	1.7787	0.1954
Sept 5 1971	0.146	0.072	2.945					
0.6936	0.5483	0.0816	1.7790	1.0836	0.0679	0.4689	0.2923	0.2045
Sept 6 1971	0.087	0.056	3.207					
0.7310	0.7156	0.0414	1.9590	1.4144	0.0376	0.6589	0.3815	0.1082
Sept 7 1971	0.051	0.135	8.958					
0.1359	2.9429	0.0002	0.3190	5.8166	0.0001	0.4790	1.5690	0.0005
Sept 8 1971	2.938	12.814	5.549					
0.3614	4.8489	0.0504	0.9390	9.5838	0.0430	2.1891	2.5853	0.1297
Sept 16 1971	0.008	0.002	5.488					
0.6403	0.2779	0.0075	1.6389	0.5493	0.0062	0.2189	0.1482	0.0187
Sept 17 1971	0.127	0.121	4.525					
0.5974	1.0592	0.0272	1.4989	2.0936	0.0217	0.7790	0.5647	0.0681
Sept 18 1971	17.799	26.270	0.568					
0.3155	1.6409	0.6874	0.8189	3.2432	0.5857	0.6490	0.8749	1.7820
Sept 28 1971	2.554	2.386	2.572					
0.4751	1.0386	0.3534	1.2489	2.0528	0.3088	0.6189	0.5538	0.9169
Sept 29 1971	0.768	3.934	3.864					
0.9237	5.6949	0.0733	2.4489	11.2559	0.0651	6.6390	3.0363	0.1925
June 4 1972	0.195	0.190	3.153					
0.3555	1.0833	0.0145	0.9389	2.1410	0.0128	0.4890	0.5776	0.0385
June 5 1972	3.963	2.439	0.467					
0.6075	0.6842	1.3608	1.6490	1.3524	1.2682	0.5289	0.3648	3.6342
June 6 1972	24.434113.449		-0.661					
0.2560	5.1620	0.1975	0.6494	10.2026	0.1607	1.6289	2.7522	0.4950
June 7 1972	0.080	0.035	2.831					
0.6511	0.4864	0.0444	1.6889	0.9614	0.0378	0.3990	0.2593	0.1162
June 8 1972	4.511	10.935	0.537					
0.3216	2.6950	0.1102	0.8290	5.3266	0.0926	1.0789	1.4369	0.2817
June 9 1972	0.047	0.018	3.481					
0.6145	0.4258	0.0265	1.3689	0.8416	0.0167	0.3090	0.2270	0.0611
June 11 1972	1.166	2.862	2.014					
0.4078	2.7289	0.0452	1.0890	5.3936	0.0408	1.4089	1.4549	0.1196
June 12 1972	4.307	14.687	1.172					
0.5110	3.7911	0.1889	1.3689	7.4932	0.1714	2.4589	2.0213	0.5019
June 14 1972	6.417	35.642	0.589					
0.2759	6.1751	0.0504	0.7090	12.2050	0.0421	2.1089	3.2923	0.1273
June 17 1972	0.017	0.008	6.629					
0.7715	0.5232	0.0123	2.0290	1.0341	0.0108	0.5090	0.2789	0.0323
June 20 1972	0.166	0.071	2.557					
0.6987	0.4755	0.1085	1.8989	0.9398	0.1014	0.4189	0.2535	0.2845
July 5 1972	1.873	10.689	3.056					
0.4030	6.3447	0.0305	1.0690	12.5402	0.0272	3.2189	3.3828	0.0798
July 6 1972	0.309	0.676	3.145					
0.4208	2.4322	0.0143	1.1090	4.8072	0.0126	1.2890	1.2968	0.0375
July 7 1972	0.868	1.606	1.554					
0.2950	2.0570	0.0234	0.7590	4.0657	0.0196	0.7489	1.0967	0.0587
July 8 1972	0.022	0.014	6.192					
0.8317	0.7075	0.0137	2.1890	1.3983	0.0120	0.7390	0.3772	0.0356
July 9 1972	0.035	0.023	4.938					
0.6526	0.7306	0.0130	1.7189	1.4440	0.0114	0.5989	0.3895	0.0338
July 10 1972	2.762	15.302	1.611					

0.2316	6.1594	0.0153	0.5789	12.1739	0.0121	1.7590	3.2839	0.0384
July 12 1972	1.691	4.393	1.795					
0.3787	2.8882	0.0535	1.0089	5.7085	0.0480	1.3890	1.5399	0.1422
July 14 1972	1.779	3.042	1.457					
0.4230	1.9011	0.1066	1.1189	3.7574	0.0943	1.0190	1.0136	0.2823
July 15 1972	15.274	32.682	0.795					
0.3185	2.3788	0.4147	0.8189	4.7018	0.3467	0.9490	1.2683	1.0731
July 16 1972	0.445	0.613	2.582					
0.4704	1.5315	0.0409	1.2189	3.0270	0.0348	0.8989	0.8165	0.1051
July 19 1972	1.330	4.441	2.146					
0.4279	3.7123	0.0418	1.1090	7.3373	0.0355	1.9789	1.9793	0.1069
July 22 1972	13.734	41.077	1.876					
0.3056	3.3252	0.2456	0.7990	6.5722	0.2123	1.2689	1.7729	0.6316
July 23 1972	0.142	0.124	3.805					
0.5520	0.9708	0.0284	1.4089	1.9188	0.0234	0.6589	0.5176	0.0708
July 24 1972	19.046	305.948	1.120					
0.2185	17.8589	0.0324	0.5389	35.2980	0.0249	4.7789	9.5217	0.0802
July 25 1972	1.224	5.817	2.969					
0.6121	5.2836	0.0553	1.5889	10.4430	0.0471	4.0390	2.8170	0.1422
July 26 1972	0.982	7.311	3.072					
0.2701	8.2771	0.0055	0.6890	16.3596	0.0045	2.7589	4.4130	0.0138
July 28 1972	7.418	25.141	0.289					
0.2324	3.7680	0.0677	0.5789	7.4474	0.0531	1.0790	2.0089	0.1695
July 29 1972	0.389	0.270	1.564					
0.5260	0.7717	0.0888	1.0990	1.5252	0.0490	0.4590	0.4114	0.1873
Aug 3 1972	1.875	8.132	2.500					
0.2613	4.8218	0.0169	0.6689	9.5302	0.0140	1.5689	2.5708	0.0432
Aug 4 1972	0.072	0.037	2.918					
0.3549	0.5713	0.0101	0.9189	1.1292	0.0086	0.2489	0.3046	0.0251
Aug 5 1972	4.687	35.809	1.306					
0.1855	8.4939	0.0121	0.4489	16.7882	0.0090	1.9089	4.5287	0.0293
Aug 6 1972	10.935	22.944	0.482					
0.3383	2.3327	0.3415	0.8689	4.6106	0.2850	0.9789	1.2437	0.8669
Aug 8 1972	9.785	29.731	0.602					
0.2289	3.3780	0.0966	0.5789	6.6766	0.0782	0.9590	1.8010	0.2452
Aug 9 1972	4.583	16.030	0.814					
0.2965	3.8886	0.0660	0.7689	7.6858	0.0561	1.4490	2.0733	0.1718
Aug 10 1972	0.205	0.258	3.289					
0.2975	1.3992	0.0083	0.7689	2.7655	0.0070	0.5189	0.7460	0.0212
Aug 12 1972	29.425	342.981	1.073					
0.2054	12.9588	0.0610	0.4989	25.6130	0.0455	3.2289	6.9092	0.1480
Aug 13 1972	0.561	2.087	3.076					
0.4421	4.1359	0.0169	1.1894	8.1746	0.0155	2.3189	2.2051	0.0448
Aug 17 1972	0.653	2.023	3.816					
0.4605	3.4442	0.0256	1.1789	6.8075	0.0212	1.9689	1.8363	0.0651
Aug 18 1972	0.779	1.716	3.278					
0.4547	2.4490	0.0419	1.1789	4.8405	0.0356	1.3889	1.3057	0.1074
Aug 19 1972	0.001	0.001	16.758					
1.1052	0.5559	0.0014	2.8390	1.0987	0.0012	0.7589	0.2964	0.0035
Aug 25 1972	7.271	3.948	0.333					
0.3212	0.6037	0.7911	0.8189	1.1931	0.6504	0.2389	0.3219	1.9810
Aug 26 1972	10.625	19.970	2.034					
0.3024	2.0896	0.2960	0.7790	4.1301	0.2485	0.7890	1.1141	0.7613
Aug 28 1972	3.061	8.325	0.949					
0.2406	3.0237	0.0373	0.6090	5.9762	0.0302	0.8990	1.6121	0.0940
Aug 29 1972	10.508	28.287	2.927					
0.4901	2.9928	0.5369	1.2690	5.9152	0.4553	1.8290	1.5957	1.3771
Aug 30 1972	0.067	0.096	8.865					
0.2863	1.5930	0.0022	0.7389	3.1485	0.0018	0.5691	0.8493	0.0056
Aug 31 1972	1.248	11.883	3.138					

0.5342	10.5858	0.0214	1.4189	20.9227	0.0191	7.1389	5.6439	0.0563
Sept 1 1972	10.329223.648		2.148					
0.2801	24.0723	0.0214	0.7190	47.5787	0.0179	8.3589	12.8345	0.0543
Sept 2 1972	8.065	27.110	2.830					
0.3945	3.7371	0.2138	1.0394	7.3864	0.1877	1.8489	1.9925	0.5547
Sept 5 1972	0.057	0.035	4.179					
0.3846	0.6827	0.0079	0.9989	1.3493	0.0067	0.3290	0.3640	0.0204
Sept 6 1972	6.539	44.316	1.404					
0.4338	7.5346	0.1040	1.1589	14.8921	0.0939	4.1389	4.0172	0.2750
Sept 7 1972	0.004	0.001	8.226					
0.8666	0.2779	0.0069	2.2790	0.5493	0.0060	0.2989	0.1482	0.0175
Sept 9 1972	0.590	4.654	7.757					
0.7205	8.7697	0.0222	1.8490	17.3333	0.0185	7.8290	4.6757	0.0563
Sept 10 1972	0.169	0.200	3.203					
0.4229	1.3157	0.0146	1.0890	2.6005	0.0123	0.6889	0.7015	0.0370
Sept 11 1972	2.702	11.036	1.238					
0.2757	4.5409	0.0288	0.7090	8.9750	0.0241	1.5590	2.4210	0.0737
Sept 13 1972	0.235	3.768	11.208					
0.2785	17.8260	0.0007	0.7189	35.2330	0.0005	6.2289	9.5042	0.0017
Sept 14 1972	0.748	3.086	3.486					
0.4534	4.5868	0.0213	1.1989	9.0657	0.0189	2.6189	2.4455	0.0558
Sept 15 1972	0.540	2.224	3.499					
0.5026	4.5788	0.0190	1.3490	9.0500	0.0173	2.9190	2.4413	0.0503
June 1 1973	0.086	0.060	3.492					
0.6759	0.7756	0.0322	1.7589	1.5331	0.0276	0.6590	0.4135	0.0840
June 11 1973	2.027	10.155	2.023					
0.3420	5.5698	0.0271	0.8890	11.0086	0.0232	2.3789	2.9696	0.0697
June 12 1973	12.759	11.738	-0.829					
0.4743	1.0228	1.7865	1.2789	2.0215	1.6430	0.6190	0.5453	4.7981
June 13 1973	0.035	0.010	3.955					
0.7257	0.3176	0.0369	1.8990	0.6278	0.0320	0.2890	0.1694	0.0958
July 2 1973	2.723	2.531	0.408					
0.5891	1.0334	0.5822	1.5689	2.0424	0.5223	0.7689	0.5510	1.5320
July 5 1973	0.850	1.320	2.159					
0.5014	1.7265	0.0788	1.3390	3.4124	0.0711	1.0990	0.9205	0.2095
July 6 1973	0.958	2.109	1.673					
0.2829	2.4475	0.0199	0.7290	4.8375	0.0168	0.8589	1.3049	0.0506
July 8 1973	2.752	13.070	1.283					
0.3412	5.2801	0.0386	0.8790	10.4360	0.0324	2.2389	2.8151	0.0984
July 9 1973	1.107	2.055	1.233					
0.2881	2.0638	0.0283	0.7389	4.0792	0.0236	0.7389	1.1004	0.0722
July 10 1973	16.499	35.523	0.597					
0.2997	2.3937	0.3941	0.7789	4.7311	0.3367	0.8990	1.2762	1.0210
July 11 1973	0.454	3.185	4.755					
0.2184	7.7995	0.0018	0.5392	15.4156	0.0014	2.0790	4.1584	0.0043
July 12 1973	10.291	89.424	1.222					
0.4223	9.6607	0.1209	1.1490	19.0943	0.1132	5.2090	5.1507	0.3252
July 13 1973	1.563	12.760	3.126					
0.3012	9.0762	0.0099	0.7789	17.9390	0.0084	3.4290	4.8391	0.0258
July 14 1973	17.386	49.469	0.958					
0.4118	3.1633	0.5933	1.0790	6.2523	0.5153	1.6390	1.6866	1.5494
July 15 1973	6.229	42.863	1.546					
0.3871	7.6502	0.0777	1.0189	15.1207	0.0681	3.7289	4.0788	0.2031
July 18 1973	0.711	1.489	2.233					
0.2803	2.3283	0.0153	0.7189	4.6018	0.0127	0.8089	1.2414	0.0387
July 26 1973	0.288	0.341	2.552					
0.3174	1.3164	0.0140	0.8189	2.6018	0.0118	0.5189	0.7018	0.0357
July 27 1973	17.630432.166		1.123					
0.2286	27.2527	0.0215	0.5689	53.8647	0.0169	7.6389	14.5301	0.0534
July 28 1973	0.331	0.270	3.625					

0.6312	0.9069	0.0926	1.5890	1.7924	0.0742	0.6989	0.4835	0.2276
July 29 1973	0.282	1.419	6.622					
0.4406	5.5943	0.0062	1.0489	11.0571	0.0045	2.9490	2.9827	0.0147
July 30 1973	0.035	0.025	5.666					
0.6207	0.7941	0.0108	1.6390	1.5696	0.0095	0.6189	0.4234	0.0281
July 31 1973	0.794	2.030	2.521					
0.6611	2.8424	0.0777	1.7689	5.6180	0.0704	2.3789	1.5155	0.2055
Aug 1 1973	0.414	0.707	4.602					
0.6526	1.8986	0.0591	1.7189	3.7525	0.0519	1.5590	1.0123	0.1544
Aug 2 1973	0.031	0.053	9.460					
0.9666	1.9008	0.0097	2.5189	3.7568	0.0083	2.2990	1.0134	0.0251
Aug 3 1973	0.006	0.001	8.881					
1.2704	0.1853	0.0333	3.2490	0.3662	0.0275	0.2889	0.0988	0.0827
Aug 5 1973	3.040	21.658	1.758					
0.3588	7.9206	0.0315	0.9490	15.6549	0.0278	3.5794	4.2230	0.0823
Aug 6 1973	0.063	0.124	6.924					
0.7919	2.1882	0.0115	2.0789	4.3250	0.0100	2.1790	1.1667	0.0300
Aug 9 1973	0.004	0.001	7.752					
0.8108	0.2779	0.0060	2.1189	0.5493	0.0052	0.2789	0.1482	0.0152
Aug 10 1973	4.528	4.567	1.057					
0.3458	1.1213	0.3074	0.9090	2.2163	0.2687	0.4890	0.5979	0.8064
Aug 19 1973	0.038	0.017	3.998					
0.3837	0.4974	0.0072	0.9991	0.9830	0.0061	0.2390	0.2652	0.0185
Aug 20 1973	0.214	0.090	1.500					
0.2883	0.4676	0.0242	0.7590	0.9241	0.0212	0.1689	0.2493	0.0627
Aug 21 1973	12.295	102.530	1.583					
0.3201	9.2711	0.0865	0.8289	18.3244	0.0734	3.7089	4.9430	0.2229
Aug 22 1973	0.449	1.984	3.945					
0.3118	4.9125	0.0057	0.8090	9.7096	0.0048	1.9090	2.6192	0.0145
Aug 23 1973	3.137	37.406	2.095					
0.2168	13.2568	0.0071	0.5390	26.2019	0.0055	3.5089	7.0680	0.0174
Aug 27 1973	0.370	1.535	4.255					
0.3362	4.6123	0.0058	0.8693	9.1162	0.0049	1.9289	2.4591	0.0147
Aug 28 1973	0.171	0.099	3.153					
0.6723	0.6437	0.0764	1.7795	1.2722	0.0677	0.5389	0.3432	0.1956
Aug 29 1973	0.838	0.800	1.307					
0.4262	1.0613	0.0913	1.1390	2.0977	0.0825	0.5689	0.5659	0.2382
Aug 31 1973	0.017	0.005	4.736					
0.6323	0.3270	0.0132	1.6789	0.6463	0.0118	0.2589	0.1743	0.0342
Sept 5 1973	0.030	0.031	7.237					
0.8426	1.1488	0.0118	2.1990	2.2706	0.0102	1.2089	0.6125	0.0304
Sept 6 1973	0.007	0.002	7.988					
0.9618	0.3176	0.0130	2.5189	0.6278	0.0113	0.3789	0.1694	0.0329
Sept 9 1973	0.144	0.412	10.366					
0.4959	3.1809	0.0071	1.2690	6.2870	0.0059	1.9590	1.6959	0.0180
Sept 15 1973	0.035	0.008	2.891					
0.7364	0.2541	0.0475	1.9389	0.5023	0.0417	0.2390	0.1355	0.1279
Sept 16 1973	4.790	32.955	1.151					
0.3973	7.6489	0.0629	1.0489	15.1179	0.0555	3.8289	4.0781	0.1648
Sept 20 1973	2.945	12.521	2.170					
0.2577	4.7268	0.0263	0.6589	9.3424	0.0218	1.5190	2.5201	0.0676
June 13 1974	0.009	0.004	10.211					
1.0612	0.4941	0.0131	2.7790	0.9766	0.0113	0.6590	0.2634	0.0340
June 24 1974	0.214	0.220	2.666					
0.5582	1.1429	0.0371	1.4889	2.2590	0.0334	0.8094	0.6094	0.0986
June 25 1974	0.465	0.740	2.240					
0.3979	1.7693	0.0265	1.0489	3.4969	0.0233	0.8889	0.9433	0.0697
June 26 1974	3.877	4.625	0.357					
0.4361	1.3263	0.3539	1.1290	2.6213	0.3000	0.7189	0.7071	0.9020
June 27 1974	0.013	0.006	7.313					

0.9500	0.5131	0.0146	2.4690	1.0142	0.0124	0.6090	0.2736	0.0375
July 1 1974	0.027	0.020	6.220					
0.5347	0.8235	0.0060	1.3990	1.6277	0.0052	0.5489	0.4391	0.0153
July 2 1974	6.290	37.574	1.036					
0.3958	6.6412	0.0945	1.0589	13.1263	0.0855	3.3393	3.5409	0.2515
July 3 1974	6.672	65.137	1.499					
0.1890	10.8538	0.0140	0.4589	21.4525	0.0104	2.4989	5.7869	0.0342
July 5 1974	0.004	0.001	7.889					
0.9372	0.2779	0.0080	2.4590	0.5493	0.0070	0.3290	0.1482	0.0212
July 6 1974	0.359	0.434	2.289					
0.3556	1.3440	0.0215	0.9390	2.6564	0.0190	0.5989	0.7166	0.0557
July 7 1974	10.464	26.176	0.596					
0.2315	2.7811	0.1284	0.5789	5.4968	0.1015	0.7990	1.4828	0.3261
July 9 1974	0.001	0.001	16.758					
1.1052	0.5559	0.0014	2.8390	1.0987	0.0012	0.7589	0.2964	0.0035
July 13 1974	8.381	56.733	1.276					
0.2784	7.5258	0.0549	0.7190	14.8746	0.0464	2.5989	4.0125	0.1395
July 14 1974	8.246	27.744	0.683					
0.2090	3.7406	0.0613	0.5190	7.3932	0.0478	0.9489	1.9943	0.1490
July 15 1974	1.673	9.931	4.386					
0.4076	6.5995	0.0268	1.0590	13.0438	0.0229	3.3589	3.5186	0.0690
July 16 1974	5.585	16.026	0.509					
0.2249	3.1902	0.0564	0.5691	6.3053	0.0457	0.8890	1.7009	0.1428
July 17 1974	0.819	2.553	2.242					
0.4884	3.4656	0.0359	1.3090	6.8497	0.0326	2.1490	1.8477	0.0954
July 18 1974	10.433	14.517	0.247					
0.3010	1.5470	0.3890	0.7790	3.0576	0.3296	0.5789	0.8248	0.9918
July 19 1974	11.782	39.378	0.570					
0.3272	3.7157	0.2161	0.8490	7.3441	0.1840	1.5190	1.9811	0.5565
July 20 1974	1.439	2.839	2.242					
0.3616	2.1934	0.0546	0.9589	4.3352	0.0486	0.9989	1.1694	0.1429
July 21 1974	0.187	0.470	5.418					
0.4623	2.7943	0.0091	1.2089	5.5228	0.0079	1.6189	1.4898	0.0236
July 23 1974	0.033	0.017	4.819					
0.6468	0.5727	0.0153	1.7189	1.1320	0.0137	0.4690	0.3054	0.0406
July 26 1974	2.397	7.638	4.962					
0.5314	3.5426	0.1216	1.4389	7.0019	0.1128	2.4090	1.8888	0.3286
July 27 1974	3.507	14.548	1.338					
0.3794	4.6119	0.0697	0.9890	9.1154	0.0599	2.1890	2.4589	0.1799
July 28 1974	22.431	89.050	0.786					
0.3247	4.4136	0.3411	0.8389	8.7235	0.2880	1.7890	2.3532	0.8768
July 29 1974	2.463	6.805	1.162					
0.3416	3.0717	0.0596	0.9092	6.0711	0.0534	1.3290	1.6377	0.1576
July 31 1974	0.100	0.160	4.186					
0.3221	1.7788	0.0037	0.8390	3.5158	0.0032	0.7190	0.9484	0.0096
Aug 1 1974	14.666	49.583	0.525					
0.2871	3.7587	0.2048	0.7389	7.4289	0.1715	1.3389	2.0040	0.5199
Aug 2 1974	4.456	18.251	1.756					
0.3266	4.5536	0.0665	0.8489	9.0001	0.0568	1.8690	2.4278	0.1731
Aug 3 1974	1.243	5.806	1.930					
0.3365	5.1930	0.0173	0.8689	10.2639	0.0146	2.1789	2.7687	0.0443
Aug 4 1974	4.389	11.501	2.089					
0.3497	2.9133	0.1173	0.9189	5.7581	0.1024	1.2789	1.5533	0.3049
Aug 5 1974	10.173	268.588	1.600					
0.2540	29.3527	0.0142	0.6389	58.0155	0.0114	9.2190	15.6498	0.0359
Aug 9 1974	0.018	0.007	5.024					
0.3677	0.4324	0.0036	0.9389	0.8545	0.0030	0.1990	0.2305	0.0093
Aug 13 1974	7.859	50.299	1.189					
0.3254	7.1155	0.0745	0.8490	14.0637	0.0641	2.8990	3.7937	0.1925
Aug 14 1974	8.182	74.329	1.056					

0.2280 10.0997 0.0268 0.5789 19.9620 0.0219 2.8689 5.3848 0.0686
Aug 15 1974 0.594 11.570 9.877
0.2211 21.6550 0.0009 0.5489 42.8009 0.0007 5.8689 11.5457 0.0021
Aug 18 1974 9.353133.827 1.719
0.2233 15.9076 0.0187 0.5590 31.4412 0.0148 4.3590 8.4813 0.0464
Aug 19 1974 5.409 43.234 1.462
0.3208 8.8863 0.0399 0.8390 17.5636 0.0345 3.5689 4.7378 0.1031
Aug 22 1974 2.139 18.148 3.218
0.3087 9.4325 0.0138 0.7989 18.6434 0.0117 3.6489 5.0291 0.0356
Aug 23 1974 0.012 0.003 4.867
0.5242 0.2779 0.0076 1.3689 0.5493 0.0065 0.1789 0.1482 0.0188
Aug 24 1974 0.844 0.812 1.215
0.2688 1.0696 0.0363 0.6989 2.1141 0.0310 0.3589 0.5703 0.0933
Aug 25 1974 0.207 0.203 3.053
0.4576 1.0903 0.0253 1.1990 2.1549 0.0220 0.6290 0.5813 0.0664
Sept 1 1974 0.075 0.163 9.175
0.4894 2.4162 0.0047 1.2689 4.7757 0.0040 1.4789 1.2882 0.0122
Sept 2 1974 13.582103.381 0.926
0.3643 8.4623 0.1356 0.9689 16.7257 0.1213 3.9090 4.5118 0.3596
Sept 3 1974 2.915 17.547 1.751
0.2365 6.6923 0.0155 0.5889 13.2273 0.0122 1.9389 3.5681 0.0384
Sept 4 1974 0.540 1.834 3.608
0.3520 3.7759 0.0113 0.9190 7.4630 0.0097 1.6589 2.0132 0.0290
Sept 5 1974 0.072 0.239 8.410
0.6911 3.6904 0.0059 1.8089 7.2941 0.0051 3.2090 1.9676 0.0155
Sept 6 1974 3.828 50.361 2.270
0.3797 14.6263 0.0240 0.9989 28.9087 0.0210 7.0089 7.7982 0.0631
Sept 9 1974 0.441 1.099 2.775
0.2203 2.7706 0.0049 0.5490 5.4760 0.0039 0.7490 1.4772 0.0122
Sept 12 1974 4.212 17.190 1.261
0.2856 4.5373 0.0482 0.7390 8.9680 0.0408 1.6190 2.4191 0.1241
Sept 14 1974 1.057 1.733 1.639
0.3810 1.8228 0.0536 1.0090 3.6027 0.0475 0.8790 0.9718 0.1416
Sept 15 1974 0.017 0.005 4.569
0.4893 0.3270 0.0079 1.2890 0.6463 0.0070 0.1989 0.1743 0.0202
Sept 16 1974 0.036 0.007 2.554
0.5333 0.2162 0.0302 1.3789 0.4273 0.0255 0.1389 0.1153 0.0722
Sept 18 1974 1.858 16.355 3.594
0.3654 9.7862 0.0161 0.9590 19.3424 0.0141 4.4890 5.2177 0.0420
Sept 19 1974 0.995 1.238 1.164
0.5207 1.3833 0.1242 1.3489 2.7340 0.1054 0.8992 0.7375 0.3192
Sept 20 1974 1.871 4.720 1.603
0.3529 2.8047 0.0529 0.9189 5.5434 0.0454 1.2389 1.4953 0.1367
Sept 21 1974 0.264 0.233 2.022
0.2347 0.9812 0.0094 0.5990 1.9394 0.0078 0.2890 0.5231 0.0245
Sept 22 1974 1.604 1.228 0.767
0.4992 0.8511 0.2990 1.3495 1.6823 0.2764 0.5389 0.4538 0.7933
Sept 24 1974 12.851 34.756 0.369
0.4137 3.0068 0.4657 1.0890 5.9429 0.4081 1.5589 1.6031 1.2064
Sept 25 1974 0.314 0.927 4.065
0.4257 3.2822 0.0110 1.1089 6.4872 0.0095 1.7489 1.7499 0.0285
Sept 26 1974 0.088 0.023 2.155
0.5917 0.2906 0.0675 1.4789 0.5743 0.0533 0.2089 0.1549 0.1644
Sept 27 1974 0.004 0.001 12.282
0.7795 0.2779 0.0056 1.9589 0.5493 0.0044 0.2690 0.1482 0.0142
June 6 1975 0.090 0.017 1.137
0.3716 0.2100 0.0377 0.9790 0.4151 0.0331 0.0990 0.1120 0.1000
July 1 1975 3.306 8.822 1.331
0.4221 2.9667 0.1264 1.1190 5.8637 0.1124 1.5790 1.5817 0.3315
July 2 1975 0.287 0.419 3.221

0.5266	1.6231	0.0312	1.3993	3.2080	0.0279	1.0789	0.8654	0.0820
July 3 1975	0.683	0.702	1.673					
0.3193	1.1427	0.0388	0.8189	2.2585	0.0323	0.4489	0.6092	0.0969
July 4 1975	0.168	0.308	4.984					
0.4748	2.0382	0.0118	1.2489	4.0285	0.0104	1.2189	1.0867	0.0310
July 5 1975	11.213	35.217	0.972					
0.4640	3.4917	0.4401	1.2690	6.9014	0.4164	2.0689	1.8617	1.1839
July 6 1975	0.008	0.002	5.637					
0.6344	0.2779	0.0074	1.6589	0.5493	0.0064	0.2189	0.1482	0.0187
July 7 1975	9.220	11.848	0.579					
0.3867	1.4286	0.6144	1.0089	2.8237	0.5290	0.6889	0.7617	1.5758
July 8 1975	2.631	14.682	1.553					
0.3520	6.2041	0.0335	0.9090	12.2623	0.0282	2.7190	3.3078	0.0855
July 10 1975	0.666	3.441	4.277					
0.5646	5.7441	0.0235	1.4590	11.3532	0.0199	4.0289	3.0625	0.0599
July 11 1975	1.597	2.674	2.775					
0.3238	1.8615	0.0573	0.8389	3.6793	0.0486	0.7489	0.9925	0.1458
July 12 1975	13.577	148.826	1.214					
0.3157	12.1867	0.0707	0.8189	24.0869	0.0602	4.8090	6.4975	0.1822
July 17 1975	23.926	344.302	0.562					
0.3670	15.9985	0.1282	0.9689	31.6210	0.1131	7.3989	8.5298	0.3359
July 18 1975	2.205	15.988	2.356					
0.3831	8.0611	0.0256	1.0190	15.9328	0.0229	3.8989	4.2979	0.0672
July 19 1975	0.706	0.899	1.750					
0.4652	1.4157	0.0687	1.2090	2.7981	0.0587	0.8189	0.7548	0.1752
July 20 1975	0.219	0.154	1.975					
0.4172	0.7818	0.0310	1.0790	1.5452	0.0263	0.4090	0.4168	0.0805
July 21 1975	9.960	39.055	1.742					
0.2985	4.3594	0.1296	0.7690	8.6164	0.1088	1.6189	2.3243	0.3309
July 22 1975	13.452	101.800	1.396					
0.2988	8.4134	0.0909	0.7689	16.6290	0.0761	3.1289	4.4857	0.2322
July 23 1975	8.849	57.206	1.431					
0.2458	7.1872	0.0474	0.6189	14.2054	0.0380	2.1794	3.8319	0.1189
July 24 1975	5.522	5.130	0.772					
0.2652	1.0328	0.2394	0.6789	2.0414	0.1984	0.3389	0.5507	0.6045
July 25 1975	0.135	0.138	3.413					
0.5015	1.1365	0.0190	1.3190	2.2462	0.0166	0.7190	0.6059	0.0499
July 26 1975	0.206	0.303	3.605					
0.5087	1.6353	0.0208	1.3589	3.2321	0.0187	1.0590	0.8719	0.0555
July 27 1975	0.511	0.787	2.118					
0.3799	1.7122	0.0274	0.9991	3.3842	0.0240	0.8190	0.9129	0.0717
July 28 1975	2.219	44.828	5.452					
0.2802	22.4597	0.0049	0.7089	44.3913	0.0040	7.7890	11.9747	0.0125
July 29 1975	1.305	4.076	1.948					
0.2717	3.4724	0.0177	0.6990	6.8633	0.0148	1.1689	1.8514	0.0447
July 30 1975	0.032	0.027	7.682					
0.6494	0.9380	0.0092	1.6489	1.8540	0.0075	0.7489	0.5001	0.0228
Aug 8 1975	2.734	2.312	0.545					
0.3543	0.9402	0.2324	0.9190	1.8582	0.1978	0.4190	0.5013	0.6065
Aug 9 1975	0.487	1.442	3.962					
0.4458	3.2919	0.0187	1.1390	6.5064	0.0155	1.8190	1.7551	0.0474
Aug 10 1975	0.897	2.150	2.055					
0.3276	2.6648	0.0230	0.8590	5.2669	0.0200	1.0990	1.4207	0.0601
Aug 11 1975	1.326	0.661	0.249					
0.4948	0.5542	0.3729	1.2789	1.0954	0.3151	0.3389	0.2955	0.9395
Aug 12 1975	0.529	1.073	2.504					
0.5418	2.2550	0.0438	1.4489	4.4571	0.0397	1.5489	1.2023	0.1162
Aug 13 1975	0.119	0.143	3.917					
0.4103	1.3360	0.0095	1.0589	2.6406	0.0080	0.6789	0.7123	0.0242
Aug 18 1975	0.017	0.010	8.595					

0.6344	0.6540	0.0067	1.6689	1.2926	0.0058	0.5189	0.3487	0.0172
Aug 19 1975	0.794	0.174	0.250					
0.6083	0.2436	0.7677	1.6289	0.4815	0.6963	0.1890	0.1299	2.0595
Aug 22 1975	5.665	47.204	1.797					
0.2757	9.2638	0.0296	0.7090	18.3099	0.0248	3.1890	4.9391	0.0761
Aug 23 1975	5.399	26.373	1.147					
0.3871	5.4307	0.0948	1.0390	10.7338	0.0864	2.6690	2.8955	0.2522
Sept 1 1975	3.344	20.181	1.495					
0.3594	6.7095	0.0410	0.9390	13.2612	0.0354	3.0189	3.5772	0.1060
Sept 2 1975	0.717	1.967	2.651					
0.3632	3.0500	0.0197	0.9690	6.0283	0.0178	1.3989	1.6261	0.0519
Sept 3 1975	9.955	153.880	2.464					
0.3173	17.1851	0.0371	0.8189	33.9662	0.0313	6.8090	9.1625	0.0955
Sept 4 1975	1.674	4.291	1.527					
0.3951	2.8498	0.0584	1.0489	5.6326	0.0520	1.4290	1.5194	0.1551
Sept 5 1975	14.946	85.242	1.729					
0.3959	6.3407	0.2352	1.0290	12.5324	0.2010	3.1289	3.3807	0.6027
Sept 6 1975	0.845	0.280	0.437					
0.3875	0.3684	0.2193	1.0190	0.7281	0.1918	0.1789	0.1964	0.5680
Sept 7 1975	10.665	35.272	0.518					
0.4179	3.6769	0.3225	1.1189	7.2673	0.2924	1.9489	1.9604	0.8557
Sept 8 1975	2.922	15.536	0.995					
0.1990	5.9111	0.0125	0.4890	11.6833	0.0095	1.4289	3.1516	0.0303
Sept 11 1975	0.192	0.292	3.240					
0.3164	1.6908	0.0072	0.8189	3.3419	0.0061	0.6689	0.9015	0.0187
Sept 12 1975	1.811	5.088	2.602					
0.3986	3.1235	0.0586	1.0489	6.1735	0.0514	1.5690	1.6653	0.1536
Sept 13 1975	19.634	52.091	0.464					
0.3362	2.9496	0.4790	0.8789	5.8299	0.4140	1.2389	1.5726	1.2332
Sept 30 1975	0.025	0.006	3.625					
0.6090	0.2668	0.0221	1.5992	0.5274	0.0193	0.1989	0.1423	0.0547
June 28 1976	2.413	4.232	1.532					
0.4209	1.9498	0.1396	1.0789	3.8538	0.1160	1.0189	1.0396	0.3549
June 29 1976	1.216	2.067	2.141					
0.4636	1.8898	0.0880	1.2490	3.7352	0.0808	1.1089	1.0076	0.2326
June 30 1976	1.541	5.517	1.737					
0.3597	3.9803	0.0319	0.9389	7.8670	0.0275	1.7990	2.1221	0.0831
July 1 1976	0.056	0.051	5.096					
0.6360	1.0125	0.0142	1.6689	2.0012	0.0124	0.8090	0.5398	0.0371
July 3 1976	1.236	8.155	3.108					
0.4043	7.3353	0.0175	1.0789	14.4981	0.0158	3.7689	3.9109	0.0467
July 5 1976	0.779	3.156	2.858					
0.2935	4.5041	0.0095	0.7590	8.9024	0.0080	1.6490	2.4014	0.0243
July 10 1976	2.018	3.036	0.830					
0.2809	1.6726	0.0606	0.7290	3.3059	0.0516	0.5890	0.8918	0.1571
July 11 1976	10.584	40.663	1.110					
0.4189	4.2713	0.2768	1.0889	8.4422	0.2366	2.2390	2.2773	0.7150
July 12 1976	6.179	63.370	1.853					
0.3622	11.4019	0.0453	0.9489	22.5357	0.0393	5.1990	6.0791	0.1183
July 14 1976	0.168	0.082	2.754					
0.7415	0.5426	0.1084	1.9390	1.0725	0.0937	0.4989	0.2893	0.2748
July 15 1976	6.229	7.180	0.646					
0.2398	1.2815	0.1779	0.6090	2.5329	0.1452	0.3789	0.6832	0.4462
July 16 1976	1.544	4.599	1.247					
0.3364	3.3115	0.0336	0.8689	6.5452	0.0283	1.3889	1.7656	0.0861
July 17 1976	0.655	0.302	0.813					
0.5353	0.5126	0.2331	1.4293	1.0131	0.2102	0.3490	0.2733	0.6220
July 18 1976	6.293	56.707	1.162					
0.1980	10.0182	0.0157	0.4789	19.8009	0.0116	2.4089	5.3413	0.0381
July 19 1976	5.503	15.822	1.155					

	0.2774	3.1965	0.0843	0.7089	6.3178	0.0697	1.1090	1.7043	0.2176
July 20 1976	5.340	29.719	0.887						
	0.2402	6.1873	0.0317	0.6090	12.2292	0.0258	1.8390	3.2989	0.0801
July 21 1976	0.963	0.889	1.093						
	0.3745	1.0263	0.0838	0.9989	2.0285	0.0754	0.4890	0.5472	0.2237
July 22 1976	0.574	2.497	3.622						
	0.3899	4.8363	0.0115	1.0290	9.5590	0.0101	2.3689	2.5786	0.0299
July 23 1976	4.129	5.717	0.041						
	0.2788	1.5393	0.1327	0.7189	3.0425	0.1116	0.5390	0.8207	0.3453
July 24 1976	1.172	4.425	3.145						
	0.3978	4.1976	0.0281	1.0490	8.2964	0.0247	2.0990	2.2380	0.0733
July 25 1976	0.386	0.216	1.099						
	0.4603	0.6221	0.0837	1.1689	1.2296	0.0683	0.3489	0.3317	0.2049
July 26 1976	6.422	34.026	1.496						
	0.2604	5.8905	0.0471	0.6690	11.6425	0.0393	1.9089	3.1406	0.1202
July 27 1976	19.132	105.256	0.485						
	0.2914	6.1164	0.1691	0.7490	12.0891	0.1413	2.2190	3.2611	0.4323
July 28 1976	15.957	293.033	4.453						
	0.1832	20.4163	0.0167	0.4490	40.3526	0.0127	4.5490	10.8852	0.0407
July 29 1976	0.878	15.051	6.385						
	0.1457	19.0582	0.0006	0.3389	37.6684	0.0004	3.3289	10.1611	0.0015
July 30 1976	0.313	0.454	3.466						
	0.6775	1.6126	0.0567	1.8190	3.1873	0.0517	1.3789	0.8598	0.1490
Aug 7 1976	0.060	0.051	4.478						
	0.5626	0.9450	0.0128	1.4890	1.8678	0.0113	0.6689	0.5038	0.0334
Aug 8 1976	1.353	0.313	0.841						
	0.5199	0.2572	0.9052	1.3490	0.5083	0.7709	0.1690	0.1371	2.3853
Aug 9 1976	0.043	0.042	5.479						
	0.5108	1.0859	0.0066	1.3389	2.1463	0.0057	0.6990	0.5790	0.0172
Aug 10 1976	9.947	102.647	1.000						
	0.3230	11.4727	0.0576	0.8490	22.6757	0.0503	4.6489	6.1168	0.1495
Aug 12 1976	3.552	14.512	1.008						
	0.2520	4.5422	0.0316	0.6389	8.9776	0.0257	1.4190	2.4217	0.0801
Aug 16 1976	2.304	0.595	-0.424						
	0.4547	0.2871	1.0563	1.1389	0.5675	0.8382	0.1589	0.1531	2.5813
Aug 17 1976	5.007	47.026	1.659						
	0.2544	10.4417	0.0198	0.6490	20.6380	0.0163	3.2889	5.5671	0.0500
Aug 18 1976	0.179	0.431	6.119						
	0.5658	2.6769	0.0136	1.4590	5.2909	0.0115	1.8890	1.4272	0.0350
Aug 19 1976	2.776	2.013	-0.027						
	0.5473	0.8062	0.6566	1.4890	1.5934	0.6147	0.5589	0.4298	1.7379
Aug 21 1976	0.013	0.009	9.447						
	0.5618	0.7697	0.0034	1.4789	1.5213	0.0030	0.5389	0.4104	0.0087
Aug 22 1976	7.481	92.770	3.341						
	0.2699	13.7867	0.0252	0.6889	27.2492	0.0207	4.6089	7.3505	0.0637
Aug 23 1976	0.065	0.029	3.136						
	0.3350	0.4960	0.0094	0.8789	0.9804	0.0082	0.2089	0.2645	0.0244
Aug 24 1976	0.243	0.414	3.881						
	0.2957	1.8941	0.0071	0.7490	3.7437	0.0058	0.6889	1.0099	0.0178
Aug 25 1976	0.179	0.098	1.739						
	0.3692	0.6087	0.0255	0.9790	1.2030	0.0227	0.2789	0.3245	0.0648
Aug 26 1976	1.726	14.643	2.637						
	0.2504	9.4319	0.0073	0.6390	18.6421	0.0060	2.9189	5.0288	0.0184
Aug 27 1976	2.055	2.480	0.829						
	0.4218	1.3417	0.1735	1.1290	2.6518	0.1572	0.7190	0.7153	0.4619
Aug 31 1976	0.789	0.364	1.117						
	0.6567	0.5129	0.4223	1.7290	1.0137	0.3703	0.4189	0.2735	1.0775
Sept 1 1976	1.253	1.273	1.398						
	0.3781	1.1295	0.1010	0.9590	2.2325	0.0822	0.5290	0.6022	0.2555
Sept 4 1976	8.629	9.365	0.127						

0.3401 1.2066 0.5266 0.8789 2.3848 0.4448 0.5089 0.6433 1.3359

Sept 5 1976 16.489125.754 0.611

0.2254 8.4789 0.0629 0.5589 16.7584 0.0489 2.3389 4.5206 0.1554

Sept 6 1976 15.068 64.719 0.118

0.3757 4.7752 0.2836 0.9989 9.4381 0.2535 2.2790 2.5459 0.7548

Sept 8 1976 1.518 1.288 0.425

0.2054 0.9433 0.0432 0.5089 1.8644 0.0336 0.2390 0.5029 0.1085

Sept 10 1976 8.179 26.529 0.029

0.3074 3.6060 0.1364 0.7990 7.1273 0.1166 1.3789 1.9226 0.3483

Sept 23 1976 0.615 0.817 2.457

0.4751 1.4769 0.0598 1.2589 2.9191 0.0531 0.8890 0.7874 0.1584

Sept 24 1976 6.990 5.008 0.033

0.4712 0.7965 1.2404 1.2389 1.5743 1.0846 0.4689 0.4247 3.1936

Sept 25 1976 0.465 0.530 1.677

0.4828 1.2672 0.0545 1.2490 2.5045 0.0461 0.7589 0.6756 0.1382

Sept 26 1976 0.046 0.049 6.002

0.5087 1.1843 0.0064 1.2990 2.3407 0.0053 0.7389 0.6314 0.0159

June 3 1977 0.406 1.105 2.890

0.3691 3.0258 0.0116 0.9689 5.9806 0.0101 1.4090 1.6133 0.0306

June 4 1977 1.874 2.089 0.328

0.1714 1.2393 0.0283 0.4089 2.4495 0.0204 0.2590 0.6608 0.0694

June 7 1977 0.436 0.444 2.559

0.3221 1.1322 0.0254 0.8389 2.2377 0.0218 0.4593 0.6036 0.0666

June 8 1977 1.809 3.912 1.294

0.3497 2.4042 0.0586 0.9089 4.7519 0.0501 1.0489 1.2818 0.1504

June 21 1977 1.383 4.173 2.102

0.5704 3.3546 0.0854 1.5489 6.6303 0.0796 2.4490 1.7885 0.2307

June 22 1977 5.952 31.944 0.863

0.2578 5.9667 0.0422 0.6590 11.7932 0.0349 1.9089 3.1813 0.1072

July 1 1977 1.012 0.725 1.005

0.5240 0.7965 0.2221 1.3790 1.5742 0.1946 0.5189 0.4246 0.5663

July 2 1977 0.510 0.210 1.100

0.5569 0.4578 0.2200 1.4089 0.9048 0.1781 0.3190 0.2441 0.5681

July 3 1977 17.294 55.486 0.515

0.3867 3.5670 0.4616 1.0290 7.0501 0.4134 1.7389 1.9018 1.2100

July 4 1977 0.307 0.843 3.874

0.4662 3.0528 0.0139 1.2189 6.0339 0.0120 1.7890 1.6276 0.0363

July 5 1977 1.631 10.968 3.336

0.3529 7.4763 0.0173 0.9189 14.7768 0.0148 3.3089 3.9861 0.0449

July 8 1977 0.006 0.001 6.922

0.6668 0.1853 0.0092 1.7490 0.3662 0.0080 0.1489 0.0988 0.0220

July 9 1977 0.232 0.196 2.738

0.5458 0.9392 0.0468 1.3589 1.8564 0.0367 0.6290 0.5008 0.1163

July 10 1977 0.197 0.096 1.979

0.6038 0.5418 0.0844 1.5189 1.0708 0.0676 0.3989 0.2889 0.2070

July 11 1977 1.294 5.727 2.356

0.3201 4.9204 0.0172 0.8390 9.7252 0.0149 1.9790 2.6234 0.0447

July 12 1977 0.290 1.508 5.854

0.7500 5.7812 0.0180 1.8389 11.4264 0.0137 5.2589 3.0823 0.0436

July 13 1977 2.902 8.763 1.073

0.3505 3.3571 0.0676 0.9090 6.6353 0.0575 1.4690 1.7899 0.1738

July 14 1977 1.172 0.453 0.579

0.4231 0.4297 0.3108 1.0790 0.8493 0.2557 0.2189 0.2291 0.7432

July 16 1977 0.014 0.006 6.622

0.8643 0.4765 0.0140 2.2590 0.9417 0.0121 0.5190 0.2540 0.0366

July 17 1977 0.027 0.034 8.886

0.5127 1.4000 0.0032 1.3489 2.7671 0.0028 0.8989 0.7464 0.0083

July 18 1977 0.302 0.146 1.104

0.5438 0.5375 0.1058 1.3890 1.0623 0.0873 0.3589 0.2866 0.2631

July 19 1977 1.733 7.711 2.976

0.3591	4.9468	0.0288	0.9289	9.7773	0.0243	2.2189	2.6374	0.0740
July 20 1977	0.054	0.031	4.310					
0.6132	0.6382	0.0203	1.5889	1.2615	0.0172	0.4890	0.3403	0.0522
July 21 1977	2.245	3.028	0.730					
0.3504	1.4995	0.1170	0.8989	2.9638	0.0974	0.6489	0.7995	0.2944
July 22 1977	12.713	26.445	1.197					
0.2035	2.3126	0.1449	0.5090	4.5709	0.1147	0.5790	1.2330	0.3618
July 23 1977	5.068	4.036	0.366					
0.2103	0.8854	0.1612	0.5290	1.7499	0.1290	0.2289	0.4720	0.4018
July 24 1977	0.003	0.001	10.227					
0.9200	0.3706	0.0044	2.3989	0.7325	0.0038	0.4290	0.1976	0.0114
July 25 1977	1.176	10.192	4.290					
0.3709	9.6353	0.0107	0.9689	19.0440	0.0092	4.4989	5.1372	0.0279
July 27 1977	0.065	0.062	5.157					
0.4871	1.0604	0.0093	1.2790	2.0960	0.0081	0.6490	0.5654	0.0241
July 28 1977	0.837	1.923	2.581					
0.4347	2.5543	0.0394	1.1490	5.0485	0.0348	1.3990	1.3618	0.1032
July 30 1977	0.498	1.132	3.600					
0.2319	2.5271	0.0067	0.5889	4.9949	0.0055	0.7290	1.3474	0.0172
July 31 1977	8.881	39.312	0.456					
0.2101	4.9212	0.0507	0.5189	9.7268	0.0391	1.2690	2.6238	0.1260
Aug 1 1977	8.409	40.985	5.676					
0.3551	5.4187	0.1246	0.9489	10.7099	0.1125	2.4389	2.8890	0.3301
Aug 6 1977	2.587	6.839	2.019					
0.3325	2.9391	0.0620	0.8590	5.8090	0.0523	1.2192	1.5670	0.1591
Aug 8 1977	0.617	1.328	3.134					
0.4300	2.3929	0.0304	1.1389	4.7295	0.0269	1.2990	1.2758	0.0798
Aug 9 1977	0.389	3.825	8.884					
0.2375	10.9318	0.0013	0.5889	21.6067	0.0010	3.1889	5.8285	0.0032
Aug 10 1977	0.080	0.016	1.481					
0.5696	0.2224	0.0743	1.3490	0.4395	0.0527	0.1489	0.1186	0.1694
Aug 11 1977	5.039	11.869	2.854					
0.5363	2.6187	0.3523	1.3489	5.1758	0.2819	1.7290	1.3962	0.8809
Aug 12 1977	0.477	1.211	2.768					
0.4083	2.8225	0.0179	1.0090	5.5787	0.0139	1.3989	1.5049	0.0436
Aug 13 1977	4.176	61.200	3.111					
0.4712	16.2930	0.0362	1.2695	32.2031	0.0333	9.7690	8.6869	0.0968
Aug 14 1977	3.295	4.619	1.047					
0.3046	1.5585	0.1249	0.7890	3.0803	0.1060	0.5889	0.8309	0.3170
Aug 15 1977	29.362	167.995	0.474					
0.3395	6.3609	0.3387	0.8989	12.5724	0.3003	2.7189	3.3914	0.8856
Aug 16 1977	0.480	3.077	10.653					
1.0673	7.1268	0.0488	2.7690	14.0862	0.0416	9.4789	3.7998	0.1251
Aug 17 1977	0.010	0.004	8.661					
0.8854	0.4447	0.0112	2.2890	0.8790	0.0095	0.4889	0.2371	0.0285
Aug 18 1977	1.267	2.290	1.775					
0.4034	2.0094	0.0653	1.0590	3.9716	0.0569	1.0190	1.0713	0.1703
Aug 19 1977	0.077	0.064	3.492					
0.4564	0.9241	0.0110	1.2089	1.8264	0.0098	0.5289	0.4927	0.0287
Aug 20 1977	5.494	9.799	1.103					
0.3107	1.9829	0.1703	0.8091	3.9192	0.1461	0.7689	1.0572	0.4375
Aug 21 1977	0.006	0.001	7.454					
0.9051	0.1853	0.0169	2.3689	0.3662	0.0146	0.2089	0.0988	0.0432
Aug 22 1977	0.086	0.071	4.103					
0.5299	0.9178	0.0167	1.2690	1.8141	0.0121	0.5789	0.4894	0.0391
Aug 23 1977	1.866	7.682	3.511					
0.2687	4.5769	0.0187	0.6789	9.0463	0.0151	1.5189	2.4402	0.0472
Aug 24 1977	0.780	0.850	1.383					
0.3653	1.2115	0.0547	0.9790	2.3946	0.0497	0.5589	0.6459	0.1439
Aug 31 1977	2.130	7.910	2.169					

0.5578	4.1286	0.1022	1.4790	8.1602	0.0909	2.8993	2.2012	0.2672
Sept 1 1977	28.868	104.043	-0.210					
0.4805	4.0069	1.0590	1.3089	7.9196	0.9939	2.4690	2.1363	2.8726
Sept 2 1977	2.771	26.234	2.576					
0.2763	10.5254	0.0128	0.7090	20.8034	0.0107	3.5989	5.6118	0.0323
Sept 4 1977	2.148	3.858	1.185					
0.4001	1.9968	0.1096	1.0489	3.9467	0.0953	1.0090	1.0646	0.2884
Sept 5 1977	11.748	63.639	0.226					
0.2033	6.0224	0.0513	0.4989	11.9033	0.0391	1.4889	3.2109	0.1252
Sept 10 1977	0.039	0.015	4.392					
0.6771	0.4276	0.0266	1.6689	0.8451	0.0205	0.3489	0.2280	0.0638
Sept 11 1977	10.810	18.235	1.949					
0.3664	1.8754	0.4926	0.9489	3.7067	0.4179	0.8590	0.9999	1.2699
Sept 12 1977	0.045	0.013	2.776					
0.3818	0.3212	0.0130	0.9689	0.6348	0.0106	0.1489	0.1712	0.0316
Sept 22 1977	2.303	3.799	1.484					
0.2620	1.8339	0.0549	0.6689	3.6248	0.0452	0.5990	0.9778	0.1407
Sept 25 1977	0.276	0.339	2.930					
0.4255	1.3655	0.0233	1.1189	2.6990	0.0204	0.7289	0.7281	0.0605
Sept 26 1977	31.468	439.948	0.829					
0.1935	15.5433	0.0483	0.4689	30.7212	0.0358	3.6590	8.2871	0.1178
Sept 27 1977	3.947	10.067	1.795					
0.4270	2.8356	0.1616	1.1590	5.6045	0.1506	1.5490	1.5118	0.4362

APPENDIX III

Computer Program Listings

```
c                         CORRFIT.FORTRAN
c
c      This program will determine the spread function parameters
c      a1,a2 and a3 by a least square error best fit algorithmn
c      to the observed spatial correlation curves from the archive
c      files: file50, file51,file52, etc.
c
c
c     This program was developed by Neil M. Fennessey at M.I.T. during
c     the course of his Master's Degree research about the Areal Distribution
c     of Rainfall
c
c      ...................................................................
c
c                        Definition of Variable
c
c                           Storm Day Date
c
c     iyear            year
c     amnth            month
c     iiday            day
c     idate            day
c     ave              E(Y)
c     avedepth         E(Y)
c     vardp            VAR(Y)
c     vardpth          VAR(Y)
c     skwdpth          C.S.(Y)
c     skwdp            C.S.(Y)
c
c
c                         Program Variables
c
c     corrobs        Observed spatial correlation
c     corr1          Observed spatial correlation
c     coothry        Theoretical spatial correlation
c     lagmod         Model spatial lag distance (km.)
c     lagobs         Observed spatial lag distance (km.)
c     lag1           Observed spatial lag distance (km.)
c     wetfrct        fraction of wetted area
c     xarea          variance function averaged area
c     ygamma         variance function
c
c
c                           Model parameters
c
c     parama1          Model 1 spread function parameters: a1
c     parama2          Model 2 spread function parameters: a2
c     parama3          Model 3 spread function parameters: a3
c
c     ealpha1           Model 1 cell center depth parameter: E(alpha 1)
c     ealpha2           Model 2 cell center depth parameter: E(alpha 2)
c     ealpha3           Model 3 cell center depth parameter: E(alpha 3)
c
c     ramda1            Model 1 cell density parameter: lambda 1
c     ramda2            Model 2 cell density parameter: lambda 2
c     ramda3            Model 3 cell density parameter: lambda 3
c
c                              Subroutines
c
c     MMBSK0           I.M.S.L. Zeroth Order Modified Bessel Funct. 2nd. kind
```

```
c      MMBSK1            I.M.S.L. First Order Modified Bessel Funct. 2nd. kind
c      error1            calculates the std. sk. error for model 2
c      error3            calculates the std. sk. error for model 1
c      error2            calculates the std. sk. error for model 3
c      corr3             calculates the model 3 spatial correlation
c
c      ......................................................................
c
       dimension   iyear(1),iday(1),imonth(1) ,corr(31),lag(31)
       dimension   amonth(6),wetfrct(-2:200),xzero(31),yzero(31)
       dimension   xarea(31),ygamma(31),corr1(31),lag1(31),alpha(100)
       dimension   lagmod(31),lagobs(31),corrmod1(31),chk(2)
       dimension   xthry(200),ythry(200),xobs(200),yobs(200)
c
       common /pitch/corrobs(31),corrthry(31)
c
       real lag1,lag,l1,lagmod
c
       double precision mmbsk0,mmbsk1,arg,c1,c2,c3,c4
c
       data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
       data pi/3.14159265389793/
       api=pi
c
c      ************************************************************************
c
c      loop through files 50,51,...,57
c
c      ************************************************************************
c
       do 770 jk=50,57
c
c      ******************************************************************************
c
c      read storm day results from file jk for the observed correlation
c      and variance functions
c
c      ******************************************************************************
c
       read(jk,1370)n,iiyear
c
c       begin the storm day loop here
c
       do 550 jj8=1,n
c
       read(jk,1310)amnth,iiday,iiyear
       write(06,1310)amnth,iiday,iiyear
       read(jk,1320)avedepth,vardepth,skwdpth
           ave=avedepth
           vardp=vardepth
           var=vardp
           iiyr=iiyear
           skwdp=skwdpth
       read(jk,1330)wetfrct(-1)
       read(jk,1340)wetfrct(0)
       read(jk,1300)kk,k1
       if(kk.gt.31)  then
           k1=kk
           b=1
       else
```

```
      k1=31
      b=2
    end if
    read(jk,1350)
    read(jk,1360)
    do 10 i=1,k1
c
c     cumulative wetted fraction is greater than 31 mm.
c
    if ((b.eq.1).and.(i.le.31)) then
    read(jk,1230) i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
    else if((b.eq.1).and.(i.gt.31)) then
    read(jk,1240) i,wetfrct(i)
c
c     cumulative wetted fraction is less than 31 mm.
c
    else if ((b.eq.2).and.(i.le.(kk))) then
    read(jk,1230) i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
    else
    read(jk,1250) lag1(i),corr1(i),xarea(i),ygamma(i)
    end if
10  continue
c
c     filejk format statements
c
1370 format(' there are ',i5,' storm days in this file for this year:'i4)
1360 format(7x,'Y (mm.) ',2x,'Acw/Ac (Y>y) ',10x,'v (km.) ',1x,'rho(v) ',8x,'A
    & (km.sq.) ',1x,'Gamma(A) ')
1350 format(3x,'Cumulative Wetted Fraction',5x,'Spatial Correlation',5x,
    &'Variance Function')
1340 format(' Wetted Fraction of Total Basin Area: (Acw/Ac)=',f5.3)
1330 format(' Dry Fraction of Total Basin Area: (Acd/Ac)=',f5.3)
1320 format(' point depth E(Y)=',f7.3,' Var(Y)=',f7.3,' S.C.(Y)=',f7.3 )
1310 format(' Storm Day ',a4,i3,i5)
1300 format(' there are ',i3,' wetted area curve pts, there are ',i3,
    &' total data points this day')
1250 format(37x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
1240 format(7x,i3,7x,f5.3)
1230 format(7x,i3,7x,f5.3,15x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
c
c     ************************************************************************
c
c     Create the lag variables for models I,II and III
c
c     ************************************************************************
c
    do 50 i=0,30
      ai=i
      j=ai
      lagmod(j+1)=ai/5.0
      lagobs(j+1)=lag1(i+1)
      corrobs(j+1)=corr1(i+1)
      xzero(j+1)=lagmod(j+1)
      yzero(j+1)=0.0
50  continue
c
c     ************************************************************************
c
c     This part of the program will minimize the std. sq. error between
c     the observed spatial correlation curves (at discrete points)
```

```
c     and the theoretical spatial correlation curves in order to perform
c     a least squares best fit test for the spread function parameter
c
c     ******************************************************************************
c
c
      do 540 jj7=1,3
          model=jj7
c
      if (model.eq.1) then
          go to 51
      else if (model.eq.2) then
          go to 70
      else if (model.eq.3) then
          go to 80
      end if
c
c     ***************************************************************************
c
c     Fit Model I  spread function parameter a1
c
c     ***************************************************************************
c
51    continue
c
          chk(2)=100
c
c
      do 55 i=1,1000
          bbb=i
          x1=bbb/100
      call error3(x1,y1)
          chk(1)=y1
      if(chk(2).gt.chk(1)) then
          chk(2)=y1
          go to 55
      else
          chk(2)=100
 1802 format('down to level II')
          j1=(i-2)*10
          j2=i*10
      do 56 ii=j1,j2
          bbb=ii
          x1=bbb/1000
      call error3(x1,y1)
          chk(1)=y1
      if(chk(2).gt.chk(1)) then
          chk(2)=y1
          go to 56
      else
          chk(2)=100
 1803 format('down to level III')
          jj1=(ii-2)*10
          jj2=ii*10
      do 57 iii=jj1,jj2
          bbb=iii
          x1=bbb/10000
      call error3(x1,y1)
          chk(1)=y1
      if(chk(2).gt.chk(1)) then
```

```
            chk(2)=y1
            go to 57
        else
            parama1=x1-0.0001
            err1=chk(2)
            go to 530
        end if
57      continue
        end if
56      continue
        end if
55      continue
c
c       ************************************************************************
c
c       Fit Model II  spread function parameter a2
c
c       ************************************************************************
c
70      continue
c
            chk(2)=100
c
        do 75 i=1,1000
            bbb=i
            x1=bbb/100
        call error1(x1,y1)
            chk(1)=y1
        if(chk(2).gt.chk(1)) then
            chk(2)=y1
            go to 75
        else
            j1=(i-1)*10.0
            j2=i*10
        do 76 ii=j1,j2
            bbb=ii
            x1=bbb/1000
        call error1(x1,y1)
            chk(1)=y1
        if(chk(2).gt.chk(1)) then
            chk(2)=y1
            go to 76
        else
            jj1=(ii-1)*10.0
            jj2=ii*10
        do 77 iii=jj1,jj2
            bbb=iii
            x1=bbb/10000
        call error1(x1,y1)
            chk(1)=y1
        if(chk(2).gt.chk(1)) then
            chk(2)=y1
            go to 77
       else
            parama2=x1-0.0001
            err2=chk(2)
            go to 530
        end if
77      continue
        end if
```

```
76    continue
      end if
75    continue
c
c     *************************************************************************
c
c     Fit Model III  spread function parameter a3
c
c     *************************************************************************
c
80    continue
c
      if(vardepth.eq.0.0) then
          vardepth=0.0005
      end if
c
          ealph=vardepth/(2*avedepth)
c
          chk(2)=100
c
      do 85 i=1,1000
          bbb=i
          x1=bbb/100
      call error2(x1,ealph,y1)
          chk(1)=y1
      if(chk(2).gt.chk(1)) then
          chk(2)=y1
          go to 85
      else
          j1=(i-1)*10.0
          j2=i*10
      do 86 ii=j1,j2
          bbb=ii
          x1=bbb/1000
      call error2(x1,ealph,y1)
          chk(1)=y1
      if(chk(2).gt.chk(1)) then
          chk(2)=y1
          go to 86
      else
          jj1=(ii-1)*10.0
          jj2=ii*10
      do 87 iii=jj1,jj2
          bbb=iii
          x1=bbb/10000
      call error2(x1,ealph,y1)
          chk(1)=y1
      if(chk(2).gt.chk(1)) then
          chk(2)=y1
          go to 87
     else
          parama3=x1-0.0001
          err3=chk(2)
          go to 530
      end if
87    continue
      end if
86    continue
      end if
85    continue
```

```
c
1801  format(' x1=',f7.4,' y1=',f9.6)
1800  format(' x1=',f9.5,' y1=',f9.2,' x2=',f9.5,' y2=',f9.2,' x3=',f9.5,
     &' y3=',f9.2)
          go to 530
c
530  continue
c
c
      if(model.eq.2) then
      write(06,1570) parama2,err2
      else if (model.eq.3) then
      write(06,1575) parama3,err3
      else if (model.eq.1) then
      write(06,1577) parama1,err1
      end if
1570  format(' parameter a2=',f7.4,' std. sq. err.=',f9.6)
1575  format(' parameter a3=',f7.4,' std. sq. err.=',f9.6)
1577  format(' parameter a1=',f7.4,' std. sq. err.=',f9.6)
c
540   continue
c
c     ********************************************************************
c
c     This portion determines E(alpha) and Lambda for the three models
c
c     ********************************************************************
c
      if(vardp.eq.0.0) then
          vardp=0.0005
      end if
c
c     Model 1
c
          ealpha1=vardp/ave
          ramda1=2*(parama1**2.0)*(ave**2.)/(api*vardp)
c
c     Model 2
c
          ealpha2=2*vardp/ave
          ramda2=(parama2**2.)*(ave**2.)/(4*api*vardp)
c
c     Model 3
c
          ealpha3=vardp/(2*ave)
          ramda3=(parama3**2.0)*ave/(2*api*(ealpha3**3.))
c
c      write the results to file01
c
      write(01,1880)amnth,iiday,iiyr,ave,var,skwdp
      write(01,1890)parama1,ealpha1,ramda1,parama2,ealpha2,ramda2,
     &parama3,ealpha3,ramda3
1880  format(1x,a4,i3,i5,3(f7.3))
1890  format(9(1x,f9.4))
550    continue
c
770   continue
      end
c
c     ********************************************************
```

```
c
c     Subroutine Error1
c
c     calculate the std. sq error for the model 2 correlation
c     function Fitting
c
c     ************************************************************
c
      subroutine error1(xx,yy)
c
      common /pitch/corrobs(31),corrthry(31)
c
      real lag
c
      double precision mmbsk0,mmbsk1,arg,c1,c2,c3,c4
c
          yy1=0.0
      do 10 i=0,30
          aaa=i
          lag=aaa/5.0
          arg=xx*lag
      if(i.eq.0) then
          corrthry(i+1)=1.
      else
          iopt=1
          c1=mmbsk0(iopt,arg,ier)
          c2=mmbsk1(iopt,arg,ier)
          c3=0.5*(arg**2.)
          c4=arg*c2
          corrthry(i+1)=(c3*c1)+c4
      end if
          yy1=((corrthry(i+1)-corrobs(i+1))**2.0)+yy1
10    continue
          yy=yy1
          yy1=0.0
      return
      end
c     ***********************************************************
c
c     Subroutine Error3
c
c     calculate the std. sq error for the model 1 correlation
c     function Fitting
c
c     ************************************************************
c
      subroutine error3(xx,yy)
c
      common /pitch/corrobs(31),corrthry(31)
c
      real lag
c
          yy1=0.0
c
      do 10 i=0,30
          aaa=i
          lag=aaa/5.0
          corrthry(i+1)=exp(-(xx*lag)**2.0)
          yy1=((corrthry(i+1)-corrobs(i+1))**2.0)+yy1
10    continue
```

```
         yy=yy1
         yy1=0.0
      return
      end
c
c     ************************************************************************
c
c     Subroutine Error2
c
c     Calculate the standard  square error between the obsverved
c     and the calculated spatial correlation  Model 3
c
c     ************************************************************************
c
      subroutine error2(xx,ealph,yy)
c
      dimension lag(31),corr(31)
c
      common /pitch/corrobs(31),corrthry(31)
c
      real lag
c
          yy1=0.0
          param=xx
c
      do 10 j=0,30
          call corr3(j,param,ealph,lag,corr)
          corrthry(j+1)=corr(j+1)
          yy1=((corrthry(j+1)-corrobs(j+1))**2.0)+yy1
10    continue
          yy=yy1
          yy1=0.0
      return
      end
c      ***************************************************************
c
c
c      subroutine corr3(i,parama,ealpha,lagmod,corrmod1)
c
c
c      ***************************************************************
      subroutine corr3(i,parama,ealpha,lagmod,corrmod1)
c
      dimension   corrmod1(31),lagmod(31)
c
      double precision mmbsk0,mmbsk1,arg,a2,a3,a4,c1,c2
c
      real lagmod,lambda
c
      data pi/3.14159265389793/
c
          ai=i
          lagmod(i+1)=ai/5
          beta=1/ealpha
          arg=parama*lagmod(i+1)*beta/2.0
       if(i.eq.0) then
          corrmod1(i+1)=1.
      else
          iopt=1
          a2=arg+(sqrt(pi)*(arg**2.0))+(((pi/2.0)-0.25)*(arg**3.0))
```

```
          a3=((arg**2.0)/2.0)+(sqrt(pi)*(arg**3.0)/2.0)+(((pi/4.0)-(1/16.0))*
     &(arg**4.0))
          a4=exp(-sqrt(pi)*arg)
          c1=mmbsk0(iopt,arg,ier)
          c2=mmbsk1(iopt,arg,ier)
          corrmod1(i+1)=(a2*c2*a4)+(a3*c1*a4)
      end if
c
      return
      end
```

```
c                         UNBIAS.FORTRAN
c
c
c        this will "unbias" the storm day model parameters based on the
c        the correction by the VAR climate=VAR(Y)/(1-gamma(A))
c        where A=154 sq. km. (see Vanmarcke pp. 333)
c
c       This program was developed by Neil M. Fennessey at M.I.T. during
c       the course of his Master's Degree research about the Areal Distribution
c       of Rainfall
c
c        ................................................................
c
c                              Definition of Variable
c
c                                 Storm Day Date
c
c       iiyr              year
c       amont             month
c       iiday             day
c
c                                     Moments
c
c       ave               expected value of storm day depth
c       var               variance of storm day depth
c       skwdp             skewness coefficient of storm day depth
c
c       aveal             428 storm day expected value of parameter a1
c       avea2             428 storm day expected value of parameter a2
c       avea3             428 storm day expected value of parameter a3
c       aveealp3          428 storm day expected value of parameter ealpha3
c
c                 Model parameters (used for both input and output)
c
c       parama1           Model I spread function parameters: a1
c       parama2           Model II spread function parameters: a2
c       parama3           Model III spread function parameters: a3
c
c       ealpha1           Model I cell center depth parameter: E(alpha 1)
c       ealpha2           Model II cell center depth parameter: E(alpha 2)
c       ealpha3           Model III cell center depth parameter: E(alpha 3)
c
c       ramda1            Model I cell density parameter: lambda 1
c       ramda2            Model II cell density parameter: lambda 2
c       ramda3            Model III cell density parameter: lambda 3
c
c                              Output Variables
c
c       varclim            unbiased storm day VAR(Y)
c
c                             Subroutines
c
c       varfunc1           Model I variance function
c       varfunc2           Model II variance function
c       varfunc3           Model III variance function
c       mmbsk0             I.M.S.L zeroth order Bessel Function (second kind)
c       mmbsk1             I.M.S.L first order Bessel Function (second kind)
c       ..................................................................
c
        data pi/3.14159265389793/
```

```
      api=pi
c
          avea1=0.42724
          avea2=1.11099
          avea3=1.78566
          aveealp3=1.84572
c
      read(95,*)k
      do 10 i=1,k
c
      read(95,1880)amnth,iiday,iiyr,ave,var,skwdp
      read(95,1890)parama1,ealpha1,ramda1,parama2,ealpha2,ramda2,
     &parama3,ealpha3,ramda3
1880  format(1x,a4,i3,i5,3(f7.3))
1890  format(9(f9.4))
c
c     This portion determines  the unbiased parameter values of
c     E(alpha) and Lambda for the three models
c
      if(var.eq.0.0) then
          var=0.0005
      end if
c
c     Model 1
c
      call varfunc1(avea1,gamod1)
          varclim=var/(1-gamod1)
          ealpha1=varclim/ave
          ramda1=2*(parama1**2.0)*(ave**2.)/(api*varclim)
c
c     Model 2
c
      call varfunc2(avea2,gamod2)
          varclim=var/(1-gamod2)
          ealpha2=2*varclim/ave
          ramda2=(parama2**2.)*(ave**2.)/(4*api*varclim)
c
c     Model 3
c
     call varfunc3(avea3,aveealp3,gamod3)
          varclim=var/(1-gamod3)
          ealpha3=varclim/(2*ave)
          ramda3=(parama3**2.0)*ave/(2*api*(ealpha3**3.))
c
c      write the results to file10
c
      write(10,1880)amnth,iiday,iiyr,ave,var,skwdp
      write(10,1890)parama1,ealpha1,ramda1,parama2,ealpha2,ramda2,
     &parama3,ealpha3,ramda3
c
10    continue
c
      end
c
c**********************************************************************
c
c      subroutine varfunc1(param1,gammod1)
c     Model 1 Theoretical Variance Function (approximation)
c
c**********************************************************************
```

```
c
       subroutine varfunc1(param1,gammod1)
c
      data pi/3.14159265389793/
c
          gammod1=0.0
c
          api=pi
          area=154.0
          gammod1=api/(api+((param1**2.0)*area))
c
      return
      end
c
c***********************************************************************
c
c      subroutine varfunc3(param3,ealpha3,gammod3)
c      Model 3 Theoretical Variance Function (approximation)
c
c***********************************************************************
c
c      subroutine varfunc3(param3,ealpha3,gammod3)
c      Model 3 Theoretical Variance Function (approximation)
c
c***********************************************************************
c
      subroutine varfunc3(param3,ealpha3,gammod3)
c
          gammod3=0.0
c
          area=154.0
          beta=1/ealpha3
          eta=((param3**2.0)*(beta**2.0)*area)
          a1=1+(0.03392*eta)
          a2=1+(0.04266*eta)
          a3=0.03392*eta
          bb=1+(a3*a2/a1)
          gammod3=1/(sqrt(a1)*sqrt(bb))
c
      return
      end
c
c***********************************************************************
c
c      subroutine varfunc2(parama,gammod2)
c
c     Model 2 Theoretical Variance Function
c
c***********************************************************************
c
      subroutine varfunc2(parama,gammod2)
c
      data pi/3.14159265389793/
c
          api=pi
c
      double precision mmbsk0,mmbsk1,arg
c
          gammod2=0.0
c
```

```
c
c      create the zeroth and first order modified Bessel Functions
c      of the Second kind
c
          area=154.0
          a4=parama*sqrt(area)
          arg=a4
          iopt=1
          ck0=mmbsk0(iopt,arg,ier)
          ck1=mmbsk1(iopt,arg,ier)
c
c        first order Modified Struve Function
c
      do 310 i=0,100
          aa=(arg/2.0)**((2*i)+2)
          bb=i+1.5
          cc=i+2.5
          gam1=gamma(bb)
          gam2=gamma(cc)
          ee=alog(aa)-alog(gam1)-alog(gam2)
          gam3=exp(ee)
          soln=soln1+(gam3)
      if(soln.gt.0.0) then
          diff=abs(gam3)
      end if
          soln1=soln
      if((diff.lt.10E-8).or.(aa.gt.10E30)) then
          go to 320
      end if
310   continue
c
320   continue
c
          stru1=soln1
          soln1=0.0
          soln=0.0
c
c       Zero order Modified Struve Function
c
      do 330 i=0,100
          aa=(arg/2.0)**((2*i)+1)
          bb=i+1.5
          gam1=gamma(bb)
          ee=alog(aa)-2*alog(gam1)
          gam1=exp(ee)
          soln=soln0+gam1
          diff=abs(gam1)
          soln0=soln
      if((diff.lt.10E-8).or.(aa.gt.10E30)) then
          go to 350
      end if
330   continue
c
350   continue
c
          stru0=soln0
          soln0=0.0
c
c     Create the variance Function here
c
```

```
          part1=3*pi*((ck1*stru0)+(ck0*stru1))/2.0
          part2=(8*ck1/arg)+(4*ck0)
          part3=(8/(arg**2.0))
          gammod2=(part1+part2-part3)**2.0
c
      return
      end
```

```
c                              TIMEDIST.FORTRAN
c
c      The purpose of this program is to split up the unbiased model
c      spatial parameter distributions into histograms, fit density functions
c      and plot the results
c
c
c      This program was developed by Neil M. Fennessey at M.I.T. during
c      the course of his Master's Degree research about the Areal Distribution
c      of Rainfall
c
c       ..............................................................
c
c                              Definition of Variable
c
c                                 Storm Day Date
c
c      iyr                 year
c      amnth               month
c      iiday               day
c      ave                 E(Y)
c      var                 VAR(Y)
c      skwdp               C.S.(Y)
c      modelnum            model number
c      pamam               parameter of choice
c
c
c                              Program Variables
c
c    lama1              Gamma distribution parameter
c    lama2              Gamma distribution parameter
c    lama3              Gamma distribution parameter
c    lamlam1            Gamma distribution parameter
c    lamlam2            Gamma distribution parameter
c    lamlam3            Gamma distribution parameter
c    lam                Gamma distribution parameter
c    kapa1              Gamma distribution parameter
c    kapa2              Gamma distribution parameter
c    kapa3              Gamma distribution parameter
c    kaplam1            Gamma distribution parameter
c    kaplam2            Gamma distribution parameter
c    kaplam3            Gamma distribution parameter
c    kap                Gamma distribution parameter
c
c                                Model parameters
c
c      parama1           Model 1 spread function parameters: a1
c      parama2           Model 2 spread function parameters: a2
c      parama3           Model 3 spread function parameters: a3
c
c      ealpha1            Model 1 cell center depth parameter: E(alpha 1)
c      ealpha2            Model 2 cell center depth parameter: E(alpha 2)
c      ealpha3            Model 3 cell center depth parameter: E(alpha 3)
c
c      inealpha1          Inverse Model 1 cell center depth parameter: Beta 1
c      inealpha2          Inverse Model 2 cell center depth parameter: Beta 2
c      inealpha3          Inverse Model 3 cell center depth parameter: Beta 3
c
c      ramda1             Model 1 cell density parameter: lambda 1
c      ramda2             Model 2 cell density parameter: lambda 2
```

```
c     ramda3            Model 3 cell density parameter: lambda 3
c
c                                   Subroutines
c
c     GAMMA             I.M.S.L. two parameter Gamma Distribution density funct.
c
c     ............................................................................
c
c
      dimension   parama2(450),parama3(450),x(450),y(450)
      dimension   parama1(450),a(450),b(450),c(450),count1(450)
      dimension   count2(450),sort1(0:450),sort2(0:450),d(450),e(450),f(450)
      dimension   ramda1(450),ramda2(450),ramda3(450),ealpha1(450)
      dimension   ealpha2(450),ealpha3(450),sort3(0:450),sort4(0:450)
      dimension   sort5(0:450),sort6(0:450),g(450),p(450),q(450)
      dimension   sort7(0:450),sort8(0:450),sort9(0:450),ygamma(450)
      dimension   inealph1(450),inealph2(450),inealph3(450)
c
      real  lam,kap,kapealp1,lamealp1,kapealp2,lamealp2,kapealp3,lamealp3
      real  lam1sq,lam2sq,lam3sq,kaplam1,kaplam2,kaplam3
      real  lamlam1,lamlam2,lamlam3,kapa1,kapa2,kapa3,lama1,lama2,lama3
      real  inealph1,inealph2,inealph3
c
      data pi/3.14159265389793/
c
          api=pi
c
          humbug=2.0
c
c
c     read parameters from file10  (which is model_parameters.data)
c
      read(10,*)k
      do 10 i=1,k
      read(10,1880)amnth,iiday,iiyr,ave,var,skwdp
      read(10,1890)parama1(i),ealpha1(i),ramda1(i),parama2(i),ealpha2(i),
     &ramda2(i),parama3(i),ealpha3(i),ramda3(i)
1880  format(1x,a4,i3,i5,3(f7.3))
1890  format(9(f9.4))
c
c
c     ****************************************************************************
c
c     this portion of the program will determine the values of other
c     model parameters: the model 2 and model 3 spread function parameters
c
c     ****************************************************************************
c
c     Model 1
c
           inealph1(i)=1.0/ealpha1(i)
c
c     Model 2
c
           inealph2(i)=1.0/ealpha2(i)
c
c     Model 3
c
           inealph3(i)=1.0/ealpha3(i)
c
```

```
c
10    continue
c
c
1200  format(3(i2),4(1x,f8.4),6(1x,f8.5))
c
c
     print,'Please select the model parameter distribution to be plotted'
     print,' enter the Model number of interest: 1,2 or 3'
     input,modelnum
      if(modelnum.eq.1) then
          go to 640
      else if (modelnum.eq.2) then
          go to 650
      else if (modelnum.eq.3) then
          go to 660
      end if
640   continue
      print,' Model 1 Parameters'
      print,' if you select E(alpha), enter 1'
      print,'  if you select Lambda 1, enter 2'
      print,' if you select A1, enter 3'
      print,' if you select Beta 1, enter 4'
      input,param
      if(param.eq.1) then
          go to 670
      else if (param.eq.2) then
          go to 680
      elseif (param.eq.3) then
          go to 690
      elseif (param.eq.4) then
          go to 770
      end if
650   continue
      print,' Model 2 Parameters'
      print,' if you select E(alpha), enter 1'
      print,'  if you select Lambda 2, enter 2'
      print,' if you select A2, enter 3'
      print,' if you select Beta 2, enter 4'
      input,param
      if(param.eq.1) then
          go to 710
      else if (param.eq.2) then
          go to 720
      else if (param.eq.3) then
          go to 730
      elseif (param.eq.4) then
          go to 780
      end if
660   continue
      print,' Model 3 Parameters'
      print,' if you select E(alpha), enter 1'
      print,'  if you select Lambda 3, enter 2'
      print,' if you select A3, enter 3'
      print,' if you select Beta 3, enter 4'
      input,param
      if(param.eq.1) then
          go to 740
      else if (param.eq.2) then
          go to 750
```

```
      else if (param.eq.3) then
          go to 760
      elseif (param.eq.4) then
          go to 790
      end if
c     ************************************************************************
c
c     sort the model one parameters into histogram
c     cells from the smallest to the largest.
c
c     ************************************************************************
c
c     sort the model 1 cell center depth parameter into histogram cells
c
c     ************************************************************************
c
670   continue
      do 30 j=1,k
      do 40 i=1,k-1
          aa=ealpha1(i)
          bb=ealpha1(i+1)
      if(aa.lt.bb) then
          a(i)=bb
          a(i+1)=aa
          ealpha1(i)=bb
          ealpha1(i+1)=aa
          go to 40
      else
          a(i)=aa
          a(i+1)=bb
      end if
40    continue
30    continue
      write (01,1000)k
      write (01,1010)a(k-1)
      write (01,1020)a(2)
      do 50 i=2,k-1
      write (01,1030)i-1,a(i)
50    continue
1000  format(10x,'The number of storms comprising this histogram=',i4)
1010  format(10x,'The min. value of Model 1 E(alpha)=',f9.4)
1020  format(10x,'The max. value of Model 1 E(alpha)=',f9.4)
1030  format(10x,i5,2x,f9.5)
c
c
c     Sort model1 parameter e(aplha)
c
          aincr=1.0
      do 240 i=k-1,2,-1
          totealp1=totealp1+a(i)
          totvral1=totvral1+(a(i)**2.0)
      do 250 s1=0,a(2),aincr
      if((a(i).lt.(s1+aincr)).and.(a(i).ge.s1)) then
          aj=s1/aincr
          j=aj
          sort1(j)=sort1(j)+1
      end if
250   continue
240   continue
c
```

```
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
          aveealp1=totealp1/(k-2)
          av=aveealp1
          ealph1sq=aveealp1**2.0
          varealp1=(totvrall/(k-3))-ealph1sq
          var=varealp1
          kapealp1=ealph1sq/varealp1
          lamealp1=aveealp1/varealp1
          gamealp1=gamma(kapealp1)
          lam=lamealp1
          kap=kapealp1
          gam=gamealp1
      write(03,1995)av,var,lam,kap,gam
1995  format('ave=',f9.5,' var=',f9.5,' lam=',f9.5,' kap=',f9.5,' gam=',f9.5)
c
c     sorting results
c
      do 280 i=0,27
          ai=i
          aii=ai*aincr
      write(02,1172)aii,aii+aincr,sort1(i)
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort1(i)
          x1=x(i+1)
c         ygamma(i+1)=aincr*lam*((lam*x1)**(kap-1))*exp(-lam*x1)/gam
c      exponential distribution
          eps=1.0/av
          arg=eps*x1
      if(arg.gt.80) then
          ygamma(i+1)=0
      else
          ygamma(i+1)=aincr*eps*exp(-eps*x1)
      end if
      write(03,1173)x1,y(i+1)/(k-2),ygamma(i+1)
1173  format('x1=',f7.4,' y(i+1)=',f9.4,' gamaa(i+1)=',f9.3)
280   continue
      write(01,1212)totealp1/(k-2)
1172  format('freq. between ',f7.3,' and ',f7.3,' is ',f7.3)
1212  format('expected value of model 1 E(alpha)=',f9.5)
c
          grafmax=28
          grafstp=2
          vertmax=100
          vertstp=10.
          numpts=28
          model=1
          go to 10000
c
c
c     sort the inverse of the model 1 cell center depth parameter into
c     histogram cells :1/E(alpha)
c
770   continue
      do 35 j=1,k
      do 45 i=1,k-1
          aa=inealph1(i)
          bb=inealph1(i+1)
      if(aa.lt.bb) then
```

```
            a(i)=bb
            a(i+1)=aa
            inealph1(i)=bb
            inealph1(i+1)=aa
            go to 45
        else
            a(i)=aa
            a(i+1)=bb
        end if
45      continue
35      continue
        write (01,1005)k-2
        write (01,1015)a(k-1)
        write (01,1025)a(2)
        do 55 i=2,k-1
        write (01,1035)i-1,a(i)
55      continue
1005    format(10x,'The number of storms comprising this histogram=',i4)
1015    format(10x,'The min. value of inverse of Model 1 E(alpha)=',f9.5)
1025    format(10x,'The max. value of inverse of Model 1 E(alpha)=',f9.5)
1035    format(10x,i5,2x,f9.5)
c
c
c       Sort inverse of model1 parameter e(aplha)
c
            aaa=anint(a(2)*10)
            aaa=aaa/10
            aincr=0.2
        do 245 i=k-1,2,-1
            totealp1=totealp1+a(i)
            totvral1=totvral1+(a(i)**2.0)
        do 255 s1=0,aaa,aincr
        if((a(i).lt.(s1+aincr)).and.(a(i).ge.s1)) then
            aj=s1/aincr
            j=anint(aj)
            sort1(j)=sort1(j)+1
        end if
255     continue
245     continue
c
c       Determine the first two moments and compute the gamma density function
c       parameters Kappa and Lambda
c
            aveealp1=totealp1/(k-2)
            av=aveealp1
            ealph1sq=aveealp1**2.0
            varealp1=(totvral1/(k-3))-ealph1sq
            var=varealp1
            kapealp1=ealph1sq/varealp1
            lamealp1=aveealp1/varealp1
            gamealp1=gamma(kapealp1)
            lam=lamealp1
            kap=kapealp1
            gam=gamealp1
        write(03,1995)av,var,lam,kap,gam
c
c       sorting results
c
        do 285 i=0,29
            ai=i
```

```
          aii=ai*aincr
      write(02,1179)aii,aii+aincr,sort1(i)
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort1(i)
          x1=x(i+1)
c         ygamma(i+1)=aincr*lam*((lam*x1)**(kap-1))*exp(-lam*x1)/gam
c      exponential distribution
          eps=1.0/av
          arg=eps*x1
      if(arg.gt.80) then
          ygamma(i+1)=0.0
      else
          ygamma(i+1)=aincr*eps*exp(-eps*x1)
      end if
      write(03,1178)x1,y(i+1)/k,ygamma(i+1)
1178  format('x1=',f7.4,' y(i+1)=',f9.4,' gamaa(i+1)=',f9.3)
285   continue
      write(01,1214)totealp1/k
1179  format('freq. between ',f7.3,' and ',f7.3,' is ',f7.3)
1214  format('expected value of inverse of model 1 E(alpha)=',f9.5)
c
          grafmax=6.0
          grafstp=0.5
          vertmax=120.
          vertstp=10.
          numpts=30
          model=1
          go to 10000
c
680   continue
c
c     sort the model 1 cell conecentration parameter into histogram
c     cells from the smallest to the largest.: Lambda
c
      do 60 j=1,k
      do 70 i=1,k-1
          aa=ramda1(i)
          bb=ramda1(i+1)
      if(aa.lt.bb) then
          b(i)=bb
          b(i+1)=aa
          ramda1(i)=bb
          ramda1(i+1)=aa
          go to 70
      else
          b(i)=aa
          b(i+1)=bb
      end if
70    continue
60    continue
      write (01,1050)b(k-1)
      write (01,1060)b(2)
      do 80 i=2,k-1
      write (01,1070)i-1,b(i)
80    continue
1050  format(/10x,'The min.value of model 1 cell concentration Lambda=',f9.4)
1060  format(10x,'The max. value of model 1 cell concentration Lambda=',f9.4)
1070  format(10x,i5,2x,f9.5)
c
c     Sort model 1 cell concentration parameter lambda in to histogram cells
```

```
c
          aaa=anint(b(2)*10.0)
          aaa=aaa/10
          aincr=0.1
      do 260 i=k-1,2,-1
          totlam1=totlam1+b(i)
          tovrlam1=tovrlam1+(b(i)**2.0)
      do 270 s2=0,aaa,aincr
      if((b(i).lt.(s2+aincr)).and.(b(i).ge.s2)) then
          aj=s2/aincr
          j=anint(aj)
          sort2(j)=sort2(j)+1
      end if
270   continue
260   continue
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
          avelam1=totlam1/(k-2)
          av=avelam1
          lam1sq=avelam1**2.0
          varlam1=(tovrlam1/(k-3))-lam1sq
          var=varlam1
c
c     sorting results
c
      do 290 i=0,19
          ai=i
          aii=ai*aincr
      write(02,1185)aii,aii+aincr,sort2(i)
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort2(i)
          x1=x(i+1)
290   continue
      write(01,1225)totlam1/k
1185  format('freq. between ',f6.3,' and ',f6.3,' is ',f7.3)
1225  format(/'expected value of Model 1 lambda=',f9.5/)
c
          grafmax=2.
          grafstp=0.2
          vertmax=360
          vertstp=40.
          numpts=20
          model=1
          go to 10000
c
690   continue
c
c     sort the model one spread function parameter into histogram
c     cells from the smallest to the largest.:A1
c
      do 440 j=1,k
      do 450 i=1,k-1
          aa=paramal(i)
          bb=paramal(i+1)
      if(aa.lt.bb) then
          g(i)=bb
          g(i+1)=aa
          paramal(i)=bb
```

```
            paramal(i+1)=aa
            go to 450
        else
            g(i)=aa
            g(i+1)=bb
        erd if
450      continue
440     continue
        write (01,1375)k-2
        write (01,1380)g(k-1)
        write (01,1390)g(2)
        do 460 i=2,k-1
        write (01,1400)i-1,g(i)
460     continue
1375    format(10x,'The number of storms comprising this histogram=',i4)
1380    format(10x,'The min. value of Model 1 Sprd. Fnct. Param. a1=',f9.4)
1390    format(10x,'The max. value of Model 1 Sprd. Fnct. Param. a1=',f9.4)
1400    format(10x,i5,2x,f9.5)
c
c
c       Sort model 1 spread function parameter a1 in to histogram cells
c
            aaa=anint(g(2)*10)
            aaa=aaa/10
            aincr=0.05
        do 540 i=k-1,2,-1
            totparal=totparal+g(i)
            c1=c1+alog(g(i))
        do 550 s2=0,aaa,aincr
        if((g(i).lt.(s2+aincr)).and.(g(i).ge.s2)) then
            aj=s2/aincr
            j=anint(aj)
            sort7(j)=sort7(j)+1
        end if
550     continue
540     continue
c
c        do a maximum likelyhood best fit for the gamma distribution
c
            c1=c1/(k-2)
            c2=alog(totparal/(k-2))
            c3=1/2.
            c4=1/12.
            c5=1/120.
            c6=1/252.
c
c       solve for kappa via newton-raphson method
c
            tol=10E-5
            aa1=5.0
        do 551 i=1,20
            ff1=(-c3/aa1)-(c4/(aa1**2.0))+(c5/(aa1**4.0))-(c6/(aa1**6.0))+c2-c1
            fprm=(c3/(aa1**2.0))+((2*c4)/(aa1**3.0))-((4*c5)/(aa1**5.0))+
       &((6*c6)/(aa1**7.0))
            kapa1=aa1-(ff1/fprm)
            chk=abs((kapa1-aa1)/aa1)
        if(chk.le.tol) then
            go to 552
        else
            aa1=kapa1
```

```
      end if
551   continue
      if(i.gt.19) then
      print,'failed to converge'
      end if
552   continue
c
          lamal=kapal/(totparal/(k-2))
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
          aveal=kapal/lamal
          av=aveal
          varal=kapal/(lamal**2.)
          var=varal
          gamal=gamma(kapal)
          lam=lamal
          kap=kapal
          gam=gamal
      write(03,1995)av,var,lam,kap,gam
c
c     sorting results
c
       do 560 i=0,23
          ai=i
          aii=ai*aincr
      write(02,1490)aii,aii+aincr,sort7(i)
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort7(i)
          x1=x(i+1)
          ygamma(i+1)=aincr*lam*((lam*x1)**(kap-1))*exp(-lam*x1)/gam
      write(03,1173)x1,y(i+1)/k,ygamma(i+1)
560   continue
      write(01,1500)aveal
1490  format('freq. between ',f6.3,' and ',f6.3,' is ',f7.3)
1500  format(/'expected value of Model 1 a1=',f9.5/)
c
          grafmax=1.2
          grafstp=0.1
          vertmax=80.
          vertstp=6.
          numpts=24
          model=1
          go to 10000
c
810   continue
1376  format(10x,'The number of storms comprising this histogram=',i4)
1381  format(10x,'The min. value of Model 1 scale of fluctuation=',f9.4)
1391  format(10x,'The max. value of Model 1 scale of fluctuation=',f9.4)
c
710   continue
c
c     ************************************************************************
c
c     sort the model two parameters into histogram
c     cells from the smallest to the largest.
c
c     ************************************************************************
c
```

```
c     model 2 E(alpha)
c
      do 90 j=1,k
      do 110 i=1,k-1
          aa=ealpha2(i)
          bb=ealpha2(i+1)
      if(aa.lt.bb) then
          c(i)=bb
          c(i+1)=aa
          ealpha2(i)=bb
          ealpha2(i+1)=aa
          go to 110
      else
          c(i)=aa
          c(i+1)=bb
      end if
110   continue
90    continue
      write (01,1080)k-2
      write (01,1090)c(k-1)
      write (01,1100)c(2)
      do 120 i=2,k-1
      write (01,1110)i-1,c(i)
120   continue
1080  format(10x,'The number of storms comprising this histogram=',i4)
1090  format(10x,'The min. value of Model 2 E(alpha)=',f9.4)
1100  format(10x,'The max. value of Model 2 E(alpha)=',f9.4)
1110  format(10x,i5,2x,f9.5)
c
          aaa=anint(c(2)*10)
          aaa=aaa/10
          aincr=2.0
      do 310 i=k-1,2,-1
          totealp2=totealp2+c(i)
          totvral2=totvral2+(c(i)**2.0)
      do 320 s1=0,aaa,aincr
      if((c(i).lt.(s1+aincr)).and.(c(i).ge.s1)) then
          aj=s1/aincr
          j=anint(aj)
          sort3(j)=sort3(j)+1
      end if
320   continue
310   continue
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
          aveealp2=totealp2/(k-2)
          av=aveealp2
          ealph2sq=aveealp2**2.0
          varealp2=(totvral2/(k-3))-ealph2sq
          var=varealp2
          kapealp2=ealph2sq/varealp2
          lamealp2=aveealp2/varealp2
          gamealp2=gamma(kapealp2)
          lam=lamealp2
          kap=kapealp2
          gam=gamealp2
      write(03,1995)av,var,lam,kap,gam
c
```

```
c      sorting results
c
       do 350 i=0,25
           ai=i
           aii=ai*aincr
       write(02,1305)aii,aii+aincr,sort3(i)
           x(i+1)=aii+(aincr/2.0)
           y(i+1)=sort3(i)
           x1=x(i+1)
c          ygamma(i+1)=aincr*lam*((lam*x1)**(kap-1))*exp(-lam*x1)/gam
c       exponential distribution
           eps=1.0/av
           arg=eps*x1
       if(arg.gt.80) then
           ygamma(i+1)=0.0
       else
           ygamma(i+1)=aincr*eps*exp(-eps*x1)
       end if
       write(03,1173)x1,y(i+1)/k,ygamma(i+1)
350    continue
       write(01,1315)totealp2/k
1300   format('freq. between ',f7.3,' and ',f7.3,' is ',f7.3)
1310   format(/'expected value of model 2 E(alpha)=',f9.5/)
c
           grafmax=55.
           grafstp=5
           vertmax=100.
           vertstp=10.
           numpts=26
           model=2
           go to 10000
c
780    continue-
c
c      model 2 inverse of E(alpha)
c
       do 95 j=1,k
       do 115 i=1,k-1
           aa=inealph2(i)
           bb=inealph2(i+1)
       if(aa.lt.bb) then
           c(i)=bb
           c(i+1)=aa
           inealph2(i)=bb
           inealph2(i+1)=aa
           go to 115
       else
           c(i)=aa
           c(i+1)=bb
       end if
115    continue
95     continue
       write (01,1085)k-2
       write (01,1095)c(k-1)
       write (01,1105)c(2)
       do 125 i=2,k-1
       write (01,1115)i-1,c(i)
125    continue
1085   format(10x,'The number of storms comprising this histogram=',i4)
1095   format(10x,'The min. value of inverse of Model 2 E(alpha)=',f9.4)
```

```
1105  format(10x,'The max. value of inverse of Model 2 E(alpha)=',f9.4)
1115  format(10x,i5,2x,f9.5)
c
c
c     Sort inverse of model 2 parameter e(aplha)
c
         aaa=anint(c(2)*10)
         aaa=aaa/10
         aincr=0.1
      do 315 i=k-1,2,-1
         totealp2=totealp2+c(i)
         totvral2=totvral2+(c(i)**2.0)
      do 325 s1=0,aaa,aincr
      if((c(i).lt.(s1+aincr)).and.(c(i).ge.s1)) then
         aj=s1/aincr
         j=anint(aj)
         sort3(j)=sort3(j)+1
      end if
325   continue
315   continue
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
         aveealp2=totealp2/(k-2)
         av=aveealp2
         ealph2sq=aveealp2**2.0
         varealp2=(totvral2/(k-3))-ealph2sq
         var=varealp2
         kapealp2=ealph2sq/varealp2
         lamealp2=aveealp2/varealp2
         gamealp2=gamma(kapealp2)
         lam=lamealp2
         kap=kapealp2
         gam=gamealp2
      write(03,1995)av,var,lam,kap,gam
c
c     sorting results
c
      do 355 i=0,29
         ai=i
         aii=ai*aincr
      write(02,1305)aii,aii+aincr,sort3(i)
         x(i+1)=aii+(aincr/2.0)
         y(i+1)=sort3(i)
         x1=x(i+1)
c        ygamma(i+1)=aincr*lam*((lam*x1)**(kap-1))*exp(-lam*x1)/gam
c      exponential distribution
         eps=1.0/av
         arg=eps*x1
      if(arg.gt.80) then
         ygamma(i+1)=0.0
      else
         ygamma(i+1)=aincr*eps*exp(-eps*x1)
      end if
      write(03,1173)x1,y(i+1)/k,ygamma(i+1)
355   continue
      write(01,1315)totealp2/k
1305  format('freq. between ',f7.3,' and ',f7.3,' is ',f7.3)
1315  format(/'expected value of model 2 E(alpha)=',f9.5/)
```

```
c
          grafmax=3.0
          grafstp=0.2
          vertmax=120.
          vertstp=10.
          numpts=30
          model=2
          go to 10000
c
720   continue
c
c     sort the model 2 cell concentration parameter into histogram
c     cells from the smallest to the largest.
c
      do 130 j=1,k
      do 140 i=1,k-1
          aa=ramda2(i)
          bb=ramda2(i+1)
      if(aa.lt.bb) then
          d(i)=bb
          d(i+1)=aa
          ramda2(i)=bb
          ramda2(i+1)=aa
          go to 140
      else
          d(i)=aa
          d(i+1)=bb
      end if
140   continue
130   continue
c
      write (01,1120)d(k-1)
      write (01,1130)d(2)
      do 150 i=2,k-1
      write (01,1140)i-1,d(i)
150   continue
c
1120  format(/10x,'The min. value of model 2 cell concentration Lambda=',f9.4)
1130  format(10x,'The max. value of model 2 cell concentration Lambda=',f9.4)
1140  format(10x,i5,2x,f9.5)
c
c     Sort model 2 cell concentration parameter lambda in to histogram cells
c
          aaa=anint(d(2)*10)
          aaa=aaa/10
          aincr=0.05
      do 330 i=k-1,2,-1
          totlam2=totlam2+d(i)
          tovrlam2=tovrlam2+(d(i)**2.0)
      do 340 s2=0,aaa,aincr
      if((d(i).lt.(s2+aincr)).and.(d(i).ge.s2)) then
          aj=s2/aincr
          j=anint(aj)
          sort4(j)=sort4(j)+1
      end if
340   continue
330   continue
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
```

```
c
          avelam2=totlam2/(k-2)
          av=avelam2
          lam2sq=avelam2**2.0
          varlam2=(tovrlam2/(k-3))-lam2sq
          var=varlam2
c
c     sorting results
c
      do 360 i=0,28
          ai=i
          aii=ai*aincr
      write(02,1320)aii,aii+aincr,sort4(i)
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort4(i)
          x1=x(i+1)
360   continue
      write(01,1330)totlam2/(k-2)
1320  format('freq. between ',f6.3,' and ',f6.3,' is ',f7.3)
1330  format(/'expected value of Model 2 lambda=',f9.5/)
c
          grafmax=1.6
          grafstp=0.2
          vertmax=300.
          vertstp=30.
          numpts=29
          model=2
          go to 10000
c
c
730   continue
c
c     sort the model two spread function parameter a2 into histogram
c     cells from the smallest to the largest.
c
      do 470 j=1,k
      do 480 i=1,k-1
          aa=parama2(i)
          bb=parama2(i+1)
      if(aa.lt.bb) then
          p(i)=bb
          p(i+1)=aa
          parama2(i)=bb
          parama2(i+1)=aa
          go to 480
      else
          p(i)=aa
          p(i+1)=bb
      end if
480    continue
470   continue
      write (01,1410)k-2
      write (01,1420)p(k-1)
      write (01,1430)p(2)
      do 490 i=2,k-1
      write (01,1440)i-1,p(i)
490   continue
1410  format(10x,'The number of storms comprising this histogram=',i4)
1420  format(10x,'The min. value of Model 2 Sprd. Fnct. Param. a2=',f9.4)
1430  format(10x,'The max. value of Model 2 Sprd. Fnct. Param. a2=',f9.4)
```

```
1440  format(10x,i5,2x,f9.5)
c
c     Sort model 2 spread function parameter a2 in to histogram cells
c
          aaa=anint(p(2)*10)
          aaa=aaa/10
          aincr=0.1
      do 570 i=k-1,2,-1
          totpara2=totpara2+p(i)
          c1=c1+alog(p(i))
      do 580 s2=0,aaa,aincr
      if((p(i).lt.(s2+aincr)).and.(p(i).ge.s2)) then
          aj=s2/aincr
          j=anint(aj)
          sort8(j)=sort8(j)+1
      end if
580   continue
570   continue
c
          c1=c1/(k-2)
          c2=alog(totpara2/(k-2))
          c3=1/2.
          c4=1/12.
          c5=1/120.
          c6=1/252.
c
c     solve for kappa via newton-raphson method
c
          tol=10E-5
          aa1=5.0
      do 562 i=1,20
          ff1=(-c3/aa1)-(c4/(aa1**2.0))+(c5/(aa1**4.0))-(c6/(aa1**6.0))+c2-c1
          fprm=(c3/(aa1**2.0))+((2*c4)/(aa1**3.0))-((4*c5)/(aa1**5.0))+
     &((6*c6)/(aa1**7.0))
          kapa2=aa1-(ff1/fprm)
          chk=abs((kapa2-aa1)/aa1)
      if(chk.le.tol) then
          go to 564
      else
          aa1=kapa2
      end if
562   continue
      if(i.gt.19) then
      print,'failed to converge'
      end if
564   continue
c
          lama2=kapa2/(totpara2/(k-2))
c
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
          avea2=kapa2/lama2
          av=avea2
          vara2=kapa2/(lama2**2.)
          var=vara2
          gama2=gamma(kapa2)
          lam=lama2
          kap=kapa2
```

```
          gam=gama2
      write(03,1995)av,var,lam,kap,gam
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
c     sorting results
c
       do 590 i=0,29
          ai=i
          aii=ai*aincr
      write(02,1510)aii,aii+aincr,sort8(i)
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort8(i)
          x1=x(i+1)
          ygamma(i+1)=aincr*lam*((lam*x1)**(kap-1))*exp(-lam*x1)/gam
      write(03,1173)x1,y(i+1)/(k-2),ygamma(i+1)
590   continue
      write(01,1520)totpara2/(k-2)
1510  format('freq. between ',f6.3,' and ',f6.3,' is ',f7.3)
1520  format(/'expected value of Model 2 a2=',f9.5/)
c
          grafmax=3.0
          grafstp=0.2
          vertmax=70.
          vertstp=7.
          numpts=30
          model=2
          go to 10000
c
820   continue
c
740   continue
c     ************************************************************************
c
c     sort the model three parameters into histogram
c     cells from the smallest to the largest.
c
c     ************************************************************************
c
c     sort model 3 E(alpha)
c
      do 160 j=1,k
      do 170 i=1,k-1
          aa=ealpha3(i)
          bb=ealpha3(i+1)
      if(aa.lt.bb) then
          e(i)=bb
          e(i+1)=aa
          ealpha3(i)=bb
          ealpha3(i+1)=aa
          go to 170
      else
          e(i)=aa
          e(i+1)=bb
      end if
170   continue
160   continue
      write (01,1150)k
      write (01,1160)e(k-1)
      write (01,1170)e(2)
```

```
c
      do 180 i=2,k-1
      write (01,1180) i-1,e(i)
180   continue
c
1150  format(10x,'The number of storms comprising this histogram=',i4)
1160  format(10x,'The min. value of Model 3 E(alpha)=',f9.4)
1170  format(10x,'The max. value of Model 3 E(alpha)=',f9.4)
1180  format(10x,i5,2x,f9.5)
c
c
c     Sort model 3 parameter e(aplha)
c
          aaa=anint(e(2)*10)
          aaa=aaa/10
          aincr=0.5
      do 370 i=k-1,2,-1
          totealp3=totealp3+e(i)
          totvral3=totvral3+(e(i)**2.0)
      do 380 s1=0,aaa,aincr
      if((e(i).lt.(s1+aincr)).and.(e(i).ge.s1)) then
          aj=s1/aincr
          j=anint(aj)
          sort5(j)=sort5(j)+1
      end if
380   continue
370   continue
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
          aveealp3=totealp3/(k-2)
          av=aveealp3
          ealph3sq=aveealp3**2.0
          varealp3=(totvral3/(k-3))-ealph3sq
          var=varealp3
          kapealp3=ealph3sq/varealp3
          lamealp3=aveealp3/varealp3
          gamealp3=gamma(kapealp3)
          lam=lamealp3
          kap=kapealp3
          gam=gamealp3
      write(03,1995)av,var,lam,kap,gam
c
c     sorting results
c
      do 420 i=0,29
          ai=i
          aii=ai*aincr
      write(02,1340)aii,aii+aincr,sort5(i)
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort5(i)
          x1=x(i+1)
c         ygamma(i+1)=aincr*lam*((lam*x1)**(kap-1))*exp(-lam*x1)/gam
c      exponential distribution
          eps=1.0/av
          arg=eps*x1
      if(arg.gt.80) then
          ygamma(i+1)=0.0
      else
```

```
          ygamma(i+1)=aincr*eps*exp(-eps*x1)
      end if
      write(03,1173)x1,y(i+1)/(k-2),ygamma(i+1)
420   continue
      write(01,1350)totealp3/(k-2)
1340  format('freq. between ',f7.3,' and ',f7.3,' is ',f7.3)
1350  format(/'expected value of model 3 E(alpha)=',f9.5/)
c
          grafmax=15.0
          grafstp=1.0
          vertmax=100.
          vertstp=10.
          numpts=30
          model=3
          go to 10000
c
c
790   continue
c
c     sort the inverse of model 3 E(alpha)
c
      do 165 j=1,k
      do 175 i=1,k-1
          aa=inealph3(i)
          bb=inealph3(i+1)
      if(aa.lt.bb) then
          e(i)=bb
          e(i+1)=aa
          inealph3(i)=bb
          inealph3(i+1)=aa
          go to 175
      else
          e(i)=aa
          e(i+1)=bb
      end if
175   continue
165   continue
      write (01,1155)k-2
      write (01,1165)e(k-1)
      write (01,1175)e(2)
c
      do 185 i=2,k-1
      write (01,1187)i-1,e(i)
185   continue
c
1155  format(10x,'The number of storms comprising this histogram=',i4)
1165  format(10x,'The min. value of inverse of Model 3 E(alpha)=',f9.4)
1175  format(10x,'The max. value of inverse of Model 3 E(alpha)=',f9.4)
1187  format(10x,i5,2x,f9.5)
c
c
c     Sort inverse of model 3 parameter e(aplha)
c
          aaa=anint(e(2)*10)
          aaa=aaa/10
          aincr=0.5
      do 375 i=k-1,2,-1
          totealp3=totealp3+e(i)
          totvral3=totvral3+(e(i)**2.0)
      do 385 s1=0,aaa,aincr
```

```
      if((e(i).lt.(s1+aincr)).and.(e(i).ge.s1))  then
          aj=s1/aincr
          j=anint(aj)
          sort5(j)=sort5(j)+1
      end if
385   continue
375   continue
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
          aveealp3=totealp3/k
          av=aveealp3
          ealph3sq=aveealp3**2.0
          varealp3=(totvral3/(k-1))-ealph3sq
          var=varealp3
          kapealp3=ealph3sq/varealp3
          lamealp3=aveealp3/varealp3
          gamealp3=gamma(kapealp3)
          lam=lamealp3
          kap=kapealp3
          gam=gamealp3
      write(03,1995)av,var,lam,kap,gam
c
c     sorting results
c
      do 425 i=0,22
          ai=i
          aii=ai*aincr
      write(02,1345)aii,aii+aincr,sort5(i)
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort5(i)
          x1=x(i+1)
c         ygamma(i+1)=aincr*lam*((lam*x1)**(kap-1))*exp(-lam*x1)/gam
c      exponential distribution
          eps=1.0/av
          arg=eps*x1
      if(arg.gt.80)  then
          ygamma(i+1)=0.0
      else
          ygamma(i+1)=aincr*eps*exp(-eps*x1)
      end if
      write(03,1173)x1,y(i+1)/(k-2),ygamma(i+1)
425   continue
      write(01,1355)totealp3/(k-2)
1345  format('freq. between ',f7.3,' and ',f7.3,' is ',f7.3)
1355  format(/'expected value of model 3 E(alpha)=',f9.5/)
c
          grafmax=11.0
          grafstp=1.0
          vertmax=180.
          vertstp=20.
          numpts=23
          model=3
          go to 10000
c
c
750   continue
c
c     sort the model 3 cell conecentration parameter into histogram
```

```
c     cells from the smallest to the largest.
c
      do 190 j=1,k
      do 210 i=1,k-1
          aa=ramda3(i)
          bb=ramda3(i+1)
      if(aa.lt.bb) then
          f(i)=bb
          f(i+1)=aa
          ramda3(i)=bb
          ramda3(i+1)=aa
          go to 210
      else
          f(i)=aa
          f(i+1)=bb
      end if
210   continue
190   continue
c
      write (01,1205)f(k-1)
      write (01,1210)f(2)
      do 220 i=2,k-1
      write (01,1220)i-1,f(i)
220   continue
c
1205  format(/10x,'The min. value of model 3 cell concentration Lambda=',f9.4)
1210  format(10x,'The max. value of model 3 cell concentration Lambda=',f9.4)
1220  format(10x,i5,2x,f9.5)
c
c     Sort model 3 cell concentration parameter lambda in to histogram cells
c
          aaa=anint(f(2)*10)
          aaa=aaa/10
          aincr=0.2
      do 390 i=k-1,2,-1
          totlam3=totlam3+f(i)
          tovrlam3=tovrlam3+(f(i)**2.0)
      do 410 s2=0,aaa,aincr
      if((f(i).lt.(s2+aincr)).and.(f(i).ge.s2)) then
          aj=s2/aincr
          j=anint(aj)
          sort6(j)=sort6(j)+1
      end if
410   continue
390   continue
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
          avelam3=totlam3/(k-2)
          av=avelam3
          lam3sq=avelam3**2.0
          varlam3=(tovrlam3/(k-3))-lam3sq
          var=varlam3
c
c     sorting results
c
       do 430 i=0,24
          ai=i
          aii=ai*aincr
```

```
      write(02,1360)aii,aii+aincr,sort6(i)
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort6(i)
          x1=x(i+1)
430   continue
      write(01,1370)totlam3/(k-2)
1360  format('freq. between ',f6.3,' and ',f6.3,' is ',f7.3)
1370  format(/'expected value of Model 3 lambda=',f9.5/)
c
          grafmax=5.
          grafstp=0.5
          vertmax=320
          vertstp=20.
          numpts=25
          model=3
          go to 10000
c
c
760   continue
c
c
c     sort the model three spread function parameter a3 into histogram
c     cells from the smallest to the largest.
c
      do 510 j=1,k
      do 520 i=1,k-1
          aa=parama3(i)
          bb=parama3(i+1)
      if(aa.lt.bb) then
          q(i)=bb
          q(i+1)=aa
          parama3(i)=bb
          parama3(i+1)=aa
          go to 520
      else
          q(i)=aa
          q(i+1)=bb
      end if
520    continue
510   continue
      write (01,1450)k-2
      write (01,1460)q(k-1)
      write (01,1470)q(2)
      do 530 i=2,k-1
      write (01,1480)i-1,q(i)
c
530   continue
1450  format(10x,'The number of storms comprising this histogram=',i4)
1460  format(10x,'The min. value of Model 3 Sprd. Fnct. Param. a3=',f9.4)
1470  format(10x,'The max. value of Model 3 Sprd. Fnct. Param. a3=',f9.4)
1480  format(10x,i5,2x,f9.5)
c
c     Sort model 3 spread function parameter a3 in to histogram cells
c
          aincr=0.25
          aaa=anint(q(2)*10)
          aaa=aaa/10
      do 610 i=k-1,2,-1
          totpara3=totpara3+q(i)
          c1=c1+alog(q(i))
```

```
      do 620 s2=0,aaa,aincr
      if((q(i).lt.(s2+aincr)).and.(q(i).ge.s2)) then
         aj=s2/aincr
         j=anint(aj)
         sort9(j)=sort9(j)+1
      end if
620   continue
610   continue
c
c      do a maximum likelyhood best fit for the gamma distribution
c
          c1=c1/(k-2)
          c2=alog(totpara3/(k-2))
          c3=1/2.
          c4=1/12.
          c5=1/120.
          c6=1/252.
c
c     solve for kappa via newton-raphson method
c
          tol=10E-4
          aa1=1.0
      do 571 i=1,50
          ff1=(-c3/aa1)-(c4/(aa1**2.0))+(c5/(aa1**4.0))-(c6/(aa1**6.0))+c2-c1
          fprm=(c3/(aa1**2.0))+((2*c4)/(aa1**3.0))-((4*c5)/(aa1**5.0))+
     &((6*c6)/(aa1**7.0))
          kapa3=aa1-(ff1/fprm)
          chk=abs((kapa3-aa1)/aa1)
      if(chk.le.tol) then
          go to 572
      else
          aa1=kapa3
      end if
571   continue
      if(i.gt.19) then
      print,'failed to converge'
      end if
572   continue
c
          lama3=kapa3/(totpara3/(k-2))
c
c     Determine the first two moments and compute the gamma density function
c     parameters Kappa and Lambda
c
          avea3=kapa3/lama3
          av=avea3
          vara3=kapa3/(lama3**2.)
          var=vara3
          gama3=gamma(kapa3)
          lam=lama3
          kap=kapa3
          gam=gama3
      write(03,1995)av,var,lam,kap,gam
c
c     sorting results
c
       do 630 i=0,39
          ai=i
          aii=ai*aincr
      write(02,1530)aii,aii+aincr,sort9(i)
```

```
          x(i+1)=aii+(aincr/2.0)
          y(i+1)=sort9(i)
          x1=x(i+1)
          ygamma(i+1)=aincr*lam*((lam*x1)**(kap-1))*exp(-lam*x1)/gam
      write(03,1173)x1,y(i+1)/(k-2),ygamma(i+1)
630   continue
      write(01,1540)totpara2/(k-2)
1530  format('freq. between ',f6.3,' and ',f6.3,' is ',f7.3)
1540  format(/'expected value of Model 3 a3=',f9.5/)
c
          grafmax=10.0
          grafstp=1.0
          vertmax=80.
          vertstp=10.
          numpts=40
          model=3
          go to 10000
c
10000 continue
c
c
      print,'entering the plotting routine '
      call comprs
      call page(11.,8.5)
      call physor(2.0,1.25)
      call basalf('stand')
      call mx2alf('l/cgr',1h*)
c
      if(param.eq.1) then
      call xname('Cell Center Depth Parameter: E(*a)) (mm.)$',100)
      call yname('Frequency of E(*a))$',100)
      else if(param.eq.2) then
      call xname('Cell Concentration Parameter: *l) (cells/sq.km.)$',100)
      call yname('Frequency of *l)$',100)
      else if(param.eq.3) then
      if(model.ne.3) then
      call xname('Cell Spread function Parameter: a (1/km.) $',100)
      else if (model.eq.3) then
      call xname('Cell Spread function Parameter: a (dimensionless) $',100)
      end if
      call yname('Frequency of a$',100)
      else if(param.eq.4) then
      call xname('Inverse of Cell Center Depth Parameter: *b) (1/mm.)
     &$',100)
      call yname('Frequency of *b)$',100)
      end if
c
      call yaxang(0.0)
        xax=7.0
        yax=5.0
      call grace(0.)
      call area2d(xax,yax)
c
      if((model.eq.1).and.(param.eq.1)) then
      call headin('Distribution of E(*a)1)$',100,1.5,2)
      call headin('Model 1$',100,1.2,2)
      else if((model.eq.2).and.(param.eq.1)) then
      call headin('Distribution of E(*a)2)$',100,1.5,2)
      call headin('Model 2$',100,1.2,2)
      else if((model.eq.3).and.(param.eq.1)) then
```

```
      call headin('Distribution of E(*a)3)$',100,1.5,2)
      call headin('Model 3$',100,1.2,2)
      else if((model.eq.1).and.(param.eq.2)) then
      call headin('Distribution of *l)1$',100,1.5,2)
      call headin('Model 1$',100,1.2,2)
      else if((model.eq.2).and.(param.eq.2)) then
      call headin('Distribution of *l)2$',100,1.5,2)
      call headin('Model 2$',100,1.2,2)
      else if((model.eq.3).and.(param.eq.2)) then
      call headin('Distribution of *l)3$',100,1.5,2)
      call headin('Model 3$',100,1.2,2)
      else if((model.eq.1).and.(param.eq.3)) then
      call headin('Distribution of a1$',100,1.5,2)
      call headin('Model 1$',100,1.2,2)
      else if((model.eq.2).and.(param.eq.3)) then
      call headin('Distribution of a2$',100,1.5,2)
      call headin('Model 2$',100,1.2,2)
      else if((model.eq.3).and.(param.eq.3)) then
      call headin('Distribution of a3$',100,1.5,2)
      call headin('Model 3$',100,1.2,2)
      else if((model.eq.1).and.(param.eq.4)) then
      call headin('Distribution of *b)1$',100,1.5,2)
      call headin('Model 1$',100,1.2,2)
      else if((model.eq.2).and.(param.eq.4)) then
      call headin('Distribution of *b)2$',100,1.5,2)
      call headin('Model 2$',100,1.2,2)
      else if((model.eq.3).and.(param.eq.4)) then
      call headin('Distribution of *b)3$',100,1.5,2)
      call headin('Model 3$',100,1.2,2)
      end if
c
      call thkfrm(0.01)
      call frame
      call yintax
      call graf(0.,grafstp,grafmax,0.,vertstp,vertmax)
c
      if(param.eq.3) then
      call messag('Gamma Distribution',18,4.5,4.8)
      call messag('Parameters',10,5.0,4.6)
      call messag('*k)=',4,5.1,4.3)
      call realno(kap,3,5.4,4.3)
      call messag('*h)=',4,5.1,4.1)
      call realno(lam,3,5.4,4.1)
      else if((param.eq.1).or.(param.eq.4)) then
      call messag('Exponential Distribution',24,3.9,4.8)
      call messag('Parameter',9,4.7,4.6)
      call messag('*e)=',4,4.8,4.3)
      call realno(eps,3,5.1,4.3)
      end if
c
      call messag('Sample Size=',12,4.7,3.1)
          k=k-2
      call intno(k,6.2,3.1)
c
      call realno(av,2,5.9,3.7)
      call realno(var,2,5.9,3.5)
c
      if(param.eq.1) then
      call messag('E (E(*a)))=',11,4.7,3.7)
      call messag('t',1,4.85,3.62)
```

```
      call messag('Var  (E(*a)))=',13,4.5,3.5)
      call messag('t',1,4.85,3.42)
c
      else if(param.eq.2) then
      call messag('E  (*1))=',8,5.1,3.7)
      call messag('t',1,5.25,3.62)
      call messag('Var  (*1))=',10,4.9,3.5)
      call messag('t',1,5.25,3.42)
c
      else if(param.eq.3) then
      call messag('E  (a)=',6,5.1,3.7)
      call messag('t',1,5.25,3.62)
      call messag('Var  (a)=',8,4.9,3.5)
      call messag('t',1,5.25,3.42)
c
      else if(param.eq.4) then
      call messag('E  (*b))=',8,5.1,3.7)
      call messag('t',1,5.25,3.62)
      call messag('Var  (*b))=',10,4.9,3.5)
      call messag('t',1,5.25,3.42)
c
      end if
c
      if((model.eq.1).and.(param.eq.1)) then
      call bars(0.248)
      else if((model.eq.1).and.(param.eq.2)) then
      call bars(0.35)
      else if((model.eq.1).and.(param.eq.3)) then
      call bars(0.290)
      else if((model.eq.1).and.(param.eq.4)) then
      call bars(0.231)
      else if((model.eq.2).and.(param.eq.1)) then
      call bars(0.262)
      else if((model.eq.2).and.(param.eq.2)) then
      call bars(0.234)
      else if((model.eq.2).and.(param.eq.3)) then
      call bars(0.278)
      else if((model.eq.2).and.(param.eq.4)) then
      call bars(0.231)
      else if((model.eq.3).and.(param.eq.1)) then
      call bars(0.258)
      else if((model.eq.3).and.(param.eq.2)) then
      call bars(0.351)
      else if((model.eq.3).and.(param.eq.3)) then
      call bars(0.281)
      else if((model.eq.3).and.(param.eq.4)) then
      call bars(0.319)
      end if
c
c    plot the histogram cells
c
      call curve(x,y,numpts,0)
c
c     plot the gamma or exponential density function
c
      call reset('bars')
      call reset('yintax')
          vertstp=0.05
          vertmax=vertmax/k
c
```

```
c      relabel the righthand side vertical axis
c
      call ygraxs(0.,vertstp,vertmax,yax,'Probability Density$',-100,xax,0.)
      if((param.eq.1).or.(param.eq.3).or.(param.eq.4)) then
      call poly5
      call curve(x,ygamma,numpts,0)
      end if
      call endpl(0)
      call donepl
c
      end
```

```
c                               PARAM.FORTRAN
c
c      The program will collect the three spatial model parameters
c      for the storm day of interest, and the model of interest
c      and calculate the maximum radius necessary to calculate the
c      number of storm cells to be generated prior to simulation
c
c      This program was developed by Neil M. Fennessey at M.I.T. during
c      the course of his Master's Degree research about the Areal Distribution
c      of Rainfall
c       ...............................................................
c
c                               Definition of Variable
c
c                                  Storm Day Date
c
c      iyear              year
c      iiiyear            year
c      iiyr               year
c      imonth             month
c      amonth             month
c      amont              month
c      iiday              day
c      idate              day
c
c                                      Moments
c
c      ave                expected value of storm day depth
c      var                variance of storm day depth
c      skwdp              skewness coefficient of storm day depth
c
c                                  Model parameters
c
c      parama1            Model I spread function parameters: a1
c      parama2            Model II spread function parameters: a2
c      parama3            Model III spread function parameters: a3
c
c      ealpha1            Model I cell center depth parameter: E(alpha 1)
c      ealpha2            Model II cell center depth parameter: E(alpha 2)
c      ealpha3            Model III cell center depth parameter: E(alpha 3)
c
c      ramda1             Model I cell density parameter: lambda 1
c      ramda2             Model II cell density parameter: lambda 2
c      ramda3             Model III cell density parameter: lambda 3
c
c                               Output Variables
c
c      alpha1ht           Model I E(alpha 1) + 3 std. dev. of alpha
c      alpha2ht           Model II E(alpha 2) + 3 std. dev. of alpha
c      alpha3ht           Model III E(alpha 3) + 3 std. dev. of alpha
c
c      radius             Maximum Simulation distance (km.)
c
c      .................................................................
c
      dimension  amonth(6)
c
      data  (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
c
      print,'enter the last two digits of the year of the storm date of
```

```
     &interest'
      input,iyear
      print,'enter the digits of the month of the storm date of interest'
      input,imonth
      print,'enter the calander date of the storm of interest'
      input,idate
c
          iiiday=idate
          iiiyear=1900+iyear
          iii=imonth-5
          amont=amonth(iii)
      print,'...........................  '
      print1005,amont,iiiday,iiiyear
      print,'...........................  '
1005  format('storm day ',a4,i3,i5)
      print,' '
      print,' enter the Model number of interest: 1,2 or 3'
      input,modelnum
c
c      **************************************************************************
c
c     read the model_parameters.data which is placed in file95
c
c      **************************************************************************
c
c
       read(95,*)n
c
       do 10 i=1,n
      read(95,1880)amnth,iiday,iiyr,ave,var,skwdp
      read(95,1890)parama1,ealpha1,ramda1,parama2,ealpha2,ramda2,
     &parama3,ealpha3,ramda3
      if((iiyr.eq.iiiyear).and.(amont.eq.amnth).and.(iiday.eq.idate)) then
          go to 20
      end if
10    continue
1880  format(1x,a4,i3,i5,3(f7.3))
1890  format(9(f9.4))
c
20    continue
c
c     **************************************************************************
c
c      calculate alphahat for use in determining the number of cells
c     (nR) necessary to generate for a simulation
c
c     **************************************************************************
c
          alph1ht=4*ealpha1
          alph2ht=4*ealpha2
          alph3ht=4*ealpha3
c
c
c     Model 1

      if(modelnum.eq.1) then
c
      radius=21.214+sqrt((alog(alph1ht)-alog(0.01))/
     &(2*(parama1**2.0)))
c
```

```
c     Model 2
c
      else if(modelnum.eq.2) then
c
      radius=(21.214+((alog(alph2ht)-alog(0.01))/parama2))
c
c     Model 3
c
      else
c
      radius=(21.214+(alph3ht/parama3))
c
      end if
c
c
      if(modelnum.eq.1) then
      print1010
1010  format(' Model 1 parameters')
      print1020,ramda1
1020  format('Cell concentration parameter: lambda1=',f7.4)
      print1030,ealpha1
1030  format('Cell center depth parameter: E(alpha1)=',f7.4)
      print1040,parama1
1040  format('Cell Spread Function Parameter: a1=',f7.4)
      print1045,radius
1045  format('Maximum radius necessary for nR calculation=',f6.2)
c
      else if(modelnum.eq.2) then
      print1050
1050  format(' Model 2 parameters')
      print1060,ramda2
1060  format('Cell concentration parameter: lambda2=',f7.4)
      print1070,ealpha2
1070  format('Cell center depth parameter: E(alpha2)=',f7.4)
      print1080,parama2
1080  format('Cell Spread Function Parameter: a2=',f7.4)
      print1085,radius
1085  format('Maximum radius necessary for nR calculation=',f6.2)
c
      else if(modelnum.eq.3) then
      print1110
1110  format(' Model 3 parameters')
      print1120,ramda3
1120  format('Cell concentration parameter: lambda3=',f7.4)
      print1130,ealpha3
1130  format('Cell center depth parameter: E(alpha3)=',f7.4)
      print1140,parama3
1140  format('Cell Spread Function Parameter: a3=',f7.4)
      print1145,radius
1145  format('Maximum radius necessary for nR calculation=',f6.2)
      end if
c
          nr=radius
      end
```

```
c                             RADIUS.FORTRAN
c
c     This program will calculate the number of storm cells to be generated
c     during a storm day simulation.
c
c     This program was developed by Wang Qinliang during the course of his
c     research at M.I.T. as a Visiting Scholar.
c
c     ..................................................................
c
c                           Definition of Variables
c
c         nr        number of storm cells to be generated
c         ramda     Model I storm cell concentration parameter: lambda
c         r         Maximum simulation distance (km.)
c
c                               Subroutines
c
c         GGEXN     I.M.S.L. exponential random deviate
c
c     ......................................................................
c
c
      integer nr
      print,"nr=?"
      read,nr
      call pc(nr)
      end
c
c  *** This is subroutine pr(nr)
      subroutine pc(nr)
      integer nr
      real pi,xm,ramda
      dimension r(10000)
      double precision dseed
      data pi/3.14159265389793/
      pi2=2.0*pi
c  *** Generating exponential random deviates
      dseed=123457.0d0
      print," ramda=?"
      read, ramda
      xm=1.0/ramda
      call ggexn(dseed,xm,nr,r)
c  ***  Transform random deviates into radius
      do 120 i=2,nr
120   r(i)=r(i-1)+r(i)
      do 121 i=1,nr
121   r(i)=sqrt(r(i)/pi)
c  *** output
      write(06,130) ramda
130   format(//20x,' Polar Coordinates --- Radius ----',//43x,' (parameter
     &lambda=',f5.2,' )  ',//80('*')//)
      nre=nr/4
      ne1=nr/2
      ne2=ne1+nre
      do 116 i=50,nre,50
      i1=i+nre
      i2=i+ne1
      i3=i+ne2
116   write(6,126)i,r(i),i1,r(i1),i2,r(i2),i3,r(i3)
```

```
126   format(2x,i7,f11.2,i7,f11.2,i7,f11.2,i7,f11.2)
      write(06,115)
115   format(//80('*')//)
      return
      end
```

```
c                              POLAR.FORTRAN
c
c     This program will calculate the number of storm cells to be generated
c     during a storm day simulation.
c
c     This program was developed by Wang Qinliang during the course of his
c     research at M.I.T. as a Visiting Scholar.
c
c     ....................................................................
c
c                            Definition of Variables
c
c         nr        number of storm cells to be generated
c         ramda     Model I storm cell concentration parameter: lambda
c         r         Maximum simulation distance (km.)
c
c                                Subroutines
c
c         GGEXN     I.M.S.L. exponential random deviate
c
c     ......................................................................
c
c
      integer nr
      print,"nr = ?"
      read,nr
      call pc(nr)
      end
c
c  *** This is subroutine pr(nr)
      subroutine pc(nr)
      integer nr
      real pi,xm,ramda
      dimension r0(1000),ur(1000),r(1000),x(1000),y(1000),n0(1000)
      double precision dseed
      data pi/3.14159265389793/
      pi2=2.0*pi
c
c  ***  Generating Basic Uniform Random Number
      dseed=123457.0d0
      do 100 i=1,nr
      r0(i)=ggubfs(dseed)
100   ur(i)=pi2*r0(i)
c
c  *** Generating exponential random deviates
      dseed=123457.0d0
      print," ramda = ?"
      read, ramda
      xm=1.0/ramda
      call ggexn(dseed,xm,nr,r)
c
c  ***  Transform random deviates into radius
c
      do 120 i=2,nr
120   r(i)=r(i-1)+r(i)
      do 121 i=1,nr
121   r(i)=sqrt(r(i)/pi)
c
c  *** output
      write(06,130) ramda
```

```
130   format(//15x,' Polar Coordinates --- Radius and Position Angle ---',
     &//43x,' (parameter lambda=',f5.2,' )  ',//80('*'),//8x, 'No',5x,'Radi
     &us  ',6x, 'Angle',12x,'No',5x,'Radius',6x,'Angle',/)
      nre=nr/2
      write(06,131)  (i,r(i),ur(i),i+nre,r(i+nre),ur(i+nre),i=1,nre)
131   format(2x,i8,f11.5,f11.5,6x,i8,f11.5,f11.5)
      write(06,115)
115   format(//80('*')//)
      return
      end
```

```
c                          SIMWET.FORTRAN
c
c     This program will simulate the storm day total rainfall random
c     field using a spatial Poisson process
c     The program will also create three sampling domains: Level I,
c     Level II, and Level III.  This program uses the fine resolution
c     grid mesh (100 meters) and samples each Level simulation field
c     the spatial distribution of total storm area
c
c
c     This program was developed by Neil M. Fennessey at M.I.T. during
c     the course of his Master's Degree research about the Areal Distribution
c     of Rainfall
c      ..................................................................
c
c                             Definition of Variable
c
c                                Storm Day Date
c
c     iyear             year
c     iyear             year
c     iiyr              year
c     imonth            month
c     amonth            month
c     iiday             day
c     idate             day
c
c                              Input Variables
c
c     iwat           water shed i.d. number
c     igage          raingage i.d. number
c     zd             raingage total storm day depth (mm)
c     xd             raingage spatial coordinate
c     yd             raingage spatial coordinate
c
c                              Program Variables
c
c     depth          Level III simulated random field total depth (mm)
c     zl             Level II simulated random field total depth (mm)
c     zi             Level II simulated random field total depth (mm)
c
c     d             Level I simulated random field 0.01 km. sq. finite element
c     d2            Level II simulated random field 0.01 km. sq. finite element
c     d3            Level III simulated random field 0.01 km. sq. finite element
c
c     r0             dummy variable, exponential random variate
c     ur             dummy variable, spatial Poisson point, storm cell
c     x7             simulation storm cell spatial coordinate
c     y7             simulation storm cell spatial coordinate
c     nr             number of simulated storm cells
c     rad            radial distance from the origin to a simulation storm cell
c     xgrd           dimension of simulation grid mesh (x direction)
c     ygrd           dimension of simulation grid mesh (y direction)
c
c     ze             "simulation raingage" depth (mm)
c     nxi            number of columns in the grid mesh matrix
c     nyi            number of rows in the grid mesh matrix
c     a              dummy variable
c     count          Level I finite element counting variable
c     count2         Level II finite element counting variable
```

```
c      count3          Level III finite element counting variable
c      totcount        Level I finite element counting variable
c      totcnt2         Level II finite element counting variable
c      totcnt3         Level III finite element counting variable
c
c
c                                Model parameters
c
c      am                Model i spread function parameters: ai
c
c      ealpha            Model i cell center depth parameter: E(alpha i)
c
c      ramda             Model i cell density parameter: lambda i
c
c                                   Subroutines
c
c      GGUBFS            I.M.S.L. uniform number generator
c      GGEXN             I.M.S.L. exponential variate number generator
c      IQHSCV            I.M.S.L. bivariate surface interpolator
c
c      ......................................................................
c
c
       dimension    iwat(93),igage(93),iyear(93),imonth(93)
       dimension    iday(93),zd(93),xd(93),yd(93),ze(93)
       dimension    count(-2:200),count2(-2:200),count3(-2:200),a(5),c(5)
       dimension    xi(294),yi(140),x(200),y(200),x2(200),y2(200),x3(200),y3(200)
       dimension    totcnt2(-1:200),totcnt1(-1:200)
       dimension    amonth(6),totcnt3(-1:200)
       dimension    r0(5020),ur(5000),r(5000),x7(5000),y7(5000)
       dimension    rad(5000),ur1(5000)
c
       common /element1/d(25000)
       common /element2/d2(42000)
       common /element3/d3(91000)
       common /rectang/z1(294,140)
       common /square/zi(294,140)
       common /deepmod1/depth1(301,301)
c
       double precision dseed,dseed1
c
       data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
       data pi/3.14159265389793/
           pi2=2*pi
c
           nd=93
c
c      *********************************************************************
c
c      this section reads in the storm day data from file84 in order to
c      match the gages,and provide coordinates for the bivariate surface
c      interpolator
c
c      *********************************************************************
c
       do 30 i=1,nd
       read(84,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i),
      &xd(i),yd(i)
30     continue
       rewind(84)
```

```
1000  format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
c
          iiday=iday(1)
          iiyear=1900+iyear(1)
          iii=imonth(1)-5
      print,'..........................  '
      print1005,amonth(iii),iiday,iiyear
      print,'..........................  '
1005  format('storm day  ',a4,i3,i5)
c
c     *******************************************************************
c
c     preset the common bolck variables to zero
c
c     *******************************************************************
c
      do 670 i=1,301
      do 680 j=1,301
          depth1(i,j)=0.0
680   continue
670   continue
      do 675 i=1,294
      do 685 j=1,140
          z1(i,j)=0.0
          zi(i,j)=0.0
685   continue
675   continue
c
      do 690 i=1,25000
          d(i)=0.0
690   continue
c
      do 692 i=1,42000
          d2(i)=0.0
692   continue
c
      do 694 i=1,91000
          d3(i)=0.0
694   continue
c
c      ******************************************************************
c
c     begin the simulation at this point
c
c      ******************************************************************
c
      print,'enter Model number: 1, 2 or 3'
      input,im
          model=im
c
      print,'enter number or random points (nr<=1000):nr'
      input,nr
c
c      print,'enter mesh length (30 km.):nz'
c      input,nz
          nz=30
      print,'enter cell concentration parameter:lambda'
      input,ramda
c
      print,'enter rainfall parameter: E(alpha) '
```

```
      input,ealpha
c
      print,'enter spread function parameter: al'
      input,am
c
c      print,'enter mesh resolution (0.1 km.)'
c      input,grd
          grd=0.1
          istor=iii*10
c
      print,'ready to go, enter 1, to try again, enter a 2'
      input,f1
c
      if(f1.eq.1) then
          go to 3
      else
          go to 4
      end if
c
3     continue
c
c     *****************************************************************
c
c     generate spatial Poisson point angle uniformly distributed random
c     number using imsl routine GGUBFS
c
c     *****************************************************************
c
          dseed=123457.0d0
c
      do 510 i=1,nr
          r0(i)=ggubfs(dseed)
          ur(i)=pi2*r0(i)
510   continue
c
c     *****************************************************************
c
c     generate spatial Poisson "disk" area differences using exponential
c     random deviates, using imsl GGEXN
c
c     *****************************************************************
c
          dseed=123457.0d0
          nub=2
          xm=1.0/ramda
          nex=nr+20
          call ggexn(dseed,xm,nex,r0)
          r(1)=r0(1)
c
      do 520 i=2,nr
          r(i)=r(i-1)+r0(i)
520   continue
c
      do 530 i=1,nr
          rad(i)=sqrt(r(i)/pi)
530   continue
c
c     ************************************************************************
c
c     transform polar to cartesian coordinates
```

```
c
c     ************************************************************************
c
      do 540 i=1,nr
          x7(i)=rad(i)*cos(ur(i))
          y7(i)=rad(i)*sin(ur(i))
540   continue
c
c     ************************************************************************
c
c     generate rainfall depth alpha at the raincell centers: exponentially
c     distributed independent of the overlapping disks using imsl GGEXN
c
c     ************************************************************************
c
          xm=ealpha
          dseed1=1969.0d0
      call ggexn(dseed1,xm,nr,ur1)
c
          aa=am**2.
          dm=ealpha
          arx0=nz
          ary0=arx0
          length=arx0
          ng0=ifix(length/grd)
          xgrd=-length/2.0
          ygrd=xgrd
          ng1=ng0+1
c
      do 560 i=1,ng1
          r0(i)=xgrd+(i-1.0)*grd
560   continue
c
c
c     select the model of interest
c
      if(im.eq.1) then
          go to 570
      else if (im.eq.2) then
          go to 610
      else if (im.eq.3) then
          go to 710
      end if
c
c      *****************************************************************
c
c      Model 1
c
c      *****************************************************************
c
570   continue
          sum=ng1**2.0
          sum1=sum-1.0
          aa2=2.0*aa
      do 580 k=1,nr
          k0=k/10
          k0=k0*10
          step=alog(ur1(k))+4.6052
          rrg1=step/(2.0*aa)
      do 580 i=1,ng1
```

```
          xd1=x7(k)-r0(i)
          xd2=xd1**2.0
      do 590 j=1,ng1
          yd1=y7(k)-r0(j)
          rrg=xd2+(yd1**2.0)
      if(rrg.ge.rrg1) then
          go to 590
      end if
          depth1(i,j)=depth1(i,j)+ur1(k)*exp(-aa2*rrg)
590   continue
580   continue
          go to 640
c
610   continue
c
c     ****************************************************************
c
c     Model 2
c
c     ****************************************************************
c
          sum=ng1**2.0
          sum1=sum-1.0
      do 620 k=1,nr
          k0=k/10
          k0=k0*10
          sp=alog(ur1(k))+4.6052
          rrg1=sp/am
      do 620 i=1,ng1
          xd1=x7(k)-r0(i)
          xd2=xd1**2.0
      do 630 j=1,ng1
          yd1=y7(k)-r0(j)
          rrg=xd2+(yd1**2.0)
          rg=sqrt(rrg)
      if(rg.ge.rrg1) then
          go to 630
      end if
          depth1(i,j)=depth1(i,j)+ur1(k)*exp(-am*rg)
630   continue
620   continue
c
          go to 640
c
c     ****************************************************************
c
c     Model 3
c
c     ****************************************************************
c
710   continue
          sum=ng1**2.
          sum1=sum-1.0
      do 720 k=1,nr
          k0=k/10
          k0=k0*10
          rrg1=ur1(k)/am
      do 720 i=1,ng1
          xd1=x7(k)-r0(i)
          xd2=xd1**2.0
```

```
      do 730 j=1,ng1
          yd1=y7(k)-r0(j)
          rrg=xd2+(yd1**2.0)
          rrg=sqrt(rrg)
      if(rrg.gt.rrg1) then
          go to 730
      end if
          depth1(i,j)=depth1(i,j)+ur1(k)-(am*rrg)
730   continue
720   continue
c
640   continue
c
      print,'done with simulation'
c
c     ******************************************************************
c
c     Create the Level II and Level I random fields from the origional
c     Level III simulated random field
c
c     ******************************************************************
c
c
      do 650 i=1,ng1
      do 660 j=1,ng1
      if((j.ge.80).and.(j.le.219).and.(i.le.294)) then
          l=j-79
          zi(i,l)=depth1(i,j)
          z1(i,l)=depth1(i,j)
          yi(l)=l
          xi(i)=i
c
c      ******************************************************************
c
c      match the gage coordinates with those of the simulated field
c       (create the "simulation raingage depths")
c
c     ******************************************************************
c
      do 665 k=1,nd
      if((yi(l).eq.yd(k)).and.(xi(i).eq.xd(k))) then
          ze(k)=depth1(i,j)
      write(83,1000)iwat(k),igage(k),iyear(k),imonth(k),iday(k),ze(k),
     &xd(k),yd(k)
      end if
665   continue
      end if
660   continue
650   continue
c
c
      nxi=294
      nyi=140
      izi=nxi
      nxi1=293
      nyi1=139
c
c     ******************************************************************
c
c     this section of the program overlays the Level I random field
```

```
c     with depth values of -1. mm corresponding to the Walnut Gulch
c     watershed basin boundary  The values generated by
c     the subroutine are left intact for the integration routine
c
c     ******************************************************************
c
c     nxi refers to the number of columns in the matrix
c     nyi refers to the number of rows in the matrix
c
      do 10 i=1,nyi
      do 20 j=1,nxi
c
      if(j.eq.nxi+1) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.1)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.2)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.3)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.4)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.5)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.6)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.7)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.8)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.9)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.10).and.(((j.ge.127).and.(j.le.129)).or.((j.ge.149).and.
     &(j.le.150)))) then
          go to 25
      else if(i.eq.10) then
          zi(j,i)=-1.
      else if((i.eq.11).and.(((j.ge.127).and.(j.le.132)).or.((j.ge.148).and.
     &(j.le.151)))) then
          go to 25
      else if(i.eq.11) then
          zi(j,i)=-1.
      else if((i.eq.12).and.(((j.ge.89).and.(j.le.90)).or.((j.ge.127)
     &.and.(j.le.133)).or.((j.ge.147).and.(j.le.151)))) then
          go to 25
      else if(i.eq.12) then
          zi(j,i)=-1.
      else if((i.eq.13).and.(((j.ge.89).and.(j.le.91)).or.((j.ge.128)
     &.and.(j.le.135)).or.((j.ge.140).and.(j.le.141)).or.
     &((j.ge.143).and.(j.le.152)))) then
```

```
     go to 25
 else if(i.eq.13) then
     zi(j,i)=-1.
 else if((i.eq.14).and.(((j.ge.89).and.(j.le.94)).or.((j.ge.127)
&.and.(j.le.138)).or.((j.ge.139).and.(j.le.153)))) then
     go to 25
 else if(i.eq.14) then
     zi(j,i)=-1.
 else if((i.eq.15).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126).and.
&(j.le.158)))) then
     go to 25
 else if(i.eq.15) then
     zi(j,i)=-1.
 else if((i.eq.16).and.(((j.ge.88).and.(j.le.95)).or.((j.ge.116).and.
&(j.le.161)))) then
     go to 25
 else if(i.eq.16) then
     zi(j,i)=-1.
 else if((i.eq.17).and.(((j.ge.88).and.(j.le.99)).or.((j.ge.116).and.
&(j.le.161)))) then
     go to 25
 else if(i.eq.17) then
     zi(j,i)=-1.
 else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
&(j.le.162)))) then
     go to 25
 else if(i.eq.18) then
     zi(j,i)=-1.
 else if((i.eq.19).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.117).and.
&(j.le.162)))) then
     go to 25
 else if(i.eq.19) then
     zi(j,i)=-1.
 else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.117).and.
&(j.le.164)))) then
     go to 25
 else if(i.eq.20) then
     zi(j,i)=-1.
 else if((i.eq.21).and.(((j.ge.81).and.(j.le.82)).or.((j.ge.86).and.
&(j.le.102)).or.((j.ge.119).and.(j.le.165)))) then
     go to 25
 else if(i.eq.21) then
     zi(j,i)=-1.
 else if((i.eq.22).and.(((j.ge.81).and.(j.le.83)).or.((j.ge.85).and.
&(j.le.103)).or.((j.ge.119).and.(j.le.166)))) then
     go to 25
 else if(i.eq.22) then
     zi(j,i)=-1.
 else if((i.eq.23).and.(((j.ge.81).and.(j.le.106)).or.((j.ge.119).and.
&(j.le.166)))) then
     go to 25
 else if(i.eq.23) then
     zi(j,i)=-1.
 else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.119).and.
&(j.le.166)))) then
     go to 25
 else if(i.eq.24) then
     zi(j,i)=-1.
 else if((i.eq.25).and.(((j.ge.83).and.(j.le.107)).or.((j.ge.118).and.
&(j.le.166)))) then
```

```
     go to 25
 else if(i.eq.25) then
     zi(j,i)=-1.
 else if((i.eq.26).and.(((j.ge.83).and.(j.le.107)).or.((j.ge.118).and.
&(j.le.166)))) then
     go to 25
 else if(i.eq.26) then
     zi(j,i)=-1.
 else if((i.eq.27).and.(((j.ge.83).and.(j.le.107)).or.((j.ge.118).and.
&(j.le.168)))) then
     go to 25
 else if(i.eq.27) then
     zi(j,i)=-1.
 else if((i.eq.28).and.(((j.ge.84).and.(j.le.107)).or.((j.ge.118).and.
&(j.le.171)))) then
     go to 25
 else if(i.eq.28) then
     zi(j,i)=-1.
 else if((i.eq.29).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.118).and.
&(j.le.172)))) then
     go to 25
 else if(i.eq.29) then
     zi(j,i)=-1.
 else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.111).and.
&(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
     go to 25
 else if(i.eq.30) then
     zi(j,i)=-1.
 else if((i.eq.31).and.(j.ge.83).and.(j.le.172)) then
     go to 25
 else if(i.eq.31) then
     zi(j,i)=-1.
 else if((i.eq.32).and.(j.ge.83).and.(j.le.171)) then
     go to 25
 else if(i.eq.32) then
     zi(j,i)=-1.
 else if((i.eq.33).and.(j.ge.82).and.(j.le.173)) then
     go to 25
 else if(i.eq.33) then
     zi(j,i)=-1.
 else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
     go to 25
 else if(i.eq.34) then
     zi(j,i)=-1.
 else if((i.eq.35).and.(j.ge.82).and.(j.le.174)) then
     go to 25
 else if(i.eq.35) then
     zi(j,i)=-1.
 else if((i.eq.36).and.(j.ge.83).and.(j.le.174)) then
     go to 25
 else if(i.eq.36) then
     zi(j,i)=-1.
 else if((i.eq.37).and.(j.ge.85).and.(j.le.174)) then
     go to 25
 else if(i.eq.37) then
     zi(j,i)=-1.
 else if((i.eq.38).and.(j.ge.85).and.(j.le.174)) then
     go to 25
 else if(i.eq.38) then
     zi(j,i)=-1.
```

```
 else if((i.eq.39).and.(j.ge.84).and.(j.le.174)) then
     go to 25
 else if(i.eq.39) then
     zi(j,i)=-1.
 else if((i.eq.40).and.(j.ge.82).and.(j.le.173)) then
     go to 25
 else if(i.eq.40) then
     zi(j,i)=-1.
 else if((i.eq.41).and.(j.ge.79).and.(j.le.173)) then
     go to 25
 else if(i.eq.41) then
     zi(j,i)=-1.
 else if((i.eq.42).and.(((j.ge.53).and.(j.le.54)).or.((j.ge.71).and.
&(j.le.75)).or.((j.ge.78).and.(j.le.172)))) then
     go to 25
 else if(i.eq.42) then
     zi(j,i)=-1.
 else if((i.eq.43).and.(((j.ge.53).and.(j.le.55)).or.((j.ge.60).and.
&(j.le.61)).or.((j.ge.71).and.(j.le.172)))) then
     go to 25
 else if(i.eq.43) then
     zi(j,i)=-1.
 else if((i.eq.44).and.(((j.ge.53).and.(j.le.57)).or.((j.ge.59).and.
&(j.le.172)))) then
     go to 25
 else if(i.eq.44) then
     zi(j,i)=-1.
 else if((i.eq.45).and.(j.ge.53).and.(j.le.173)) then
     go to 25
 else if(i.eq.45) then
     zi(j,i)=-1.
 else if((i.eq.46).and.(j.ge.53).and.(j.le.174)) then
     go to 25
 else if(i.eq.46) then
     zi(j,i)=-1.
 else if((i.eq.47).and.(j.ge.53).and.(j.le.174)) then
     go to 25
 else if(i.eq.47) then
     zi(j,i)=-1.
 else if((i.eq.48).and.(j.ge.55).and.(j.le.174)) then
     go to 25
 else if(i.eq.48) then
     zi(j,i)=-1.
 else if((i.eq.49).and.(j.ge.58).and.(j.le.174)) then
     go to 25
 else if(i.eq.49) then
     zi(j,i)=-1.
 else if((i.eq.50).and.(((j.ge.58).and.(j.le.175)).or.((j.ge.180).and.
&(j.le.182)))) then
     go to 25
 else if(i.eq.50) then
     zi(j,i)=-1.
 else if((i.eq.51).and.(((j.ge.58).and.(j.le.176)).or.((j.ge.180).and.
&(j.le.183)))) then
     go to 25
 else if(i.eq.51) then
     zi(j,i)=-1.
 else if((i.eq.52).and.(j.ge.58).and.(j.le.183)) then
     go to 25
 else if(i.eq.52) then
```

```
     zi(j,i)=-1.
 else if((i.eq.53).and.(j.ge.57).and.(j.le.183)) then
     go to 25
 else if(i.eq.53) then
     zi(j,i)=-1.
 else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
     go to 25
 else if(i.eq.54) then
     zi(j,i)=-1.
 else if((i.eq.55).and.(j.ge.57).and.(j.le.192)) then
     go to 25
 else if(i.eq.55) then
     zi(j,i)=-1.
 else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
     go to 25
 else if(i.eq.56) then
     zi(j,i)=-1.
 else if((i.eq.57).and.(j.ge.52).and.(j.le.196)) then
     go to 25
 else if(i.eq.57) then
     zi(j,i)=-1.
 else if((i.eq.58).and.(((j.ge.37).and.(j.le.39)).or.((j.ge.52).and.
&(j.le.196)))) then
     go to 25
 else if(i.eq.58) then
     zi(j,i)=-1.
 else if((i.eq.59).and.(((j.ge.35).and.(j.le.45)).or.((j.ge.47).and.
&(j.le.197)))) then
     go to 25
 else if(i.eq.59) then
     zi(j,i)=-1.
 else if((i.eq.60).and.(j.ge.35).and.(j.le.199)) then
     go to 25
 else if(i.eq.60) then
     zi(j,i)=-1.
 else if((i.eq.61).and.(j.ge.34).and.(j.le.203)) then
     go to 25
 else if(i.eq.61) then
     zi(j,i)=-1.
 else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
     go to 25
 else if(i.eq.62) then
     zi(j,i)=-1.
 else if((i.eq.63).and.(((j.ge.34).and.(j.le.207)).or.((j.ge.219).and.
&(j.le.220)))) then
     go to 25
 else if(i.eq.63) then
     zi(j,i)=-1.
 else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
&(j.le.221)))) then
     go to 25
 else if(i.eq.64) then
     zi(j,i)=-1.
 else if((i.eq.65).and.(((j.ge.33).and.(j.le.210)).or.((j.ge.217).and.
&(j.le.221)))) then
     go to 25
 else if(i.eq.65) then
     zi(j,i)=-1.
 else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
     go to 25
```

```
else if(i.eq.66) then
    zi(j,i)=-1.
else if((i.eq.67).and.(j.ge.30).and.(j.le.223)) then
    go to 25
else if(i.eq.67) then
    zi(j,i)=-1.
else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
    go to 25
else if(i.eq.68) then
    zi(j,i)=-1.
else if((i.eq.69).and.(j.ge.29).and.(j.le.226)) then
    go to 25
else if(i.eq.69) then
    zi(j,i)=-1.
else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
    go to 25
else if(i.eq.70) then
    zi(j,i)=-1.
else if((i.eq.71).and.(j.ge.27).and.(j.le.226)) then
    go to 25
else if(i.eq.71) then
    zi(j,i)=-1.
else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
    go to 25
else if(i.eq.72) then
    zi(j,i)=-1.
else if((i.eq.73).and.(j.ge.23).and.(j.le.226)) then
    go to 25
else if(i.eq.73) then
    zi(j,i)=-1.
else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
    go to 25
else if(i.eq.74) then
    zi(j,i)=-1.
else if((i.eq.75).and.(j.ge.19).and.(j.le.226)) then
    go to 25
else if(i.eq.75) then
    zi(j,i)=-1.
else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
    go to 25
else if(i.eq.76) then
    zi(j,i)=-1.
else if((i.eq.77).and.(j.ge.15).and.(j.le.226)) then
    go to 25
else if(i.eq.77) then
    zi(j,i)=-1.
else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
    go to 25
else if(i.eq.78) then
    zi(j,i)=-1.
else if((i.eq.79).and.(j.ge.12).and.(j.le.227)) then
    go to 25
else if(i.eq.79) then
    zi(j,i)=-1.
else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
    go to 25
else if(i.eq.80) then
    zi(j,i)=-1.
else if((i.eq.81).and.(j.ge.10).and.(j.le.229)) then
    go to 25
```

```
else if(i.eq.81) then
    zi(j,i)=-1.
else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
    go to 25
else if(i.eq.82) then
    zi(j,i)=-1.
else if((i.eq.83).and.(j.ge.11).and.(j.le.231)) then
    go to 25
else if(i.eq.83) then
    zi(j,i)=-1.
else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
    go to 25
else if(i.eq.84) then
    zi(j,i)=-1.
else if((i.eq.85).and.(j.ge.13).and.(j.le.234)) then
    go to 25
else if(i.eq.85) then
    zi(j,i)=-1.
else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
    go to 25
else if(i.eq.86) then
    zi(j,i)=-1.
else if((i.eq.87).and.(j.ge.15).and.(j.le.235)) then
    go to 25
else if(i.eq.87) then
    zi(j,i)=-1.
else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
    go to 25
else if(i.eq.88) then
    zi(j,i)=-1.
else if((i.eq.89).and.(j.ge.16).and.(j.le.236)) then
    go to 25
else if(i.eq.89) then
    zi(j,i)=-1.
else if((i.eq.90).and.(j.ge.17).and.(j.le.236)) then
    go to 25
else if(i.eq.90) then
    zi(j,i)=-1.
else if((i.eq.91).and.(j.ge.17).and.(j.le.238)) then
    go to 25
else if(i.eq.91) then
    zi(j,i)=-1.
else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
    go to 25
else if(i.eq.92) then
    zi(j,i)=-1.
else if((i.eq.93).and.(j.ge.19).and.(j.le.238)) then
    go to 25
else if(i.eq.93) then
    zi(j,i)=-1.
else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
    go to 25
else if(i.eq.94) then
    zi(j,i)=-1.
else if((i.eq.95).and.(j.ge.23).and.(j.le.238)) then
    go to 25
else if(i.eq.95) then
    zi(j,i)=-1.
else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
    go to 25
```

```
 else if(i.eq.96) then
     zi(j,i)=-1.
 else if((i.eq.97).and.(j.ge.24).and.(j.le.238)) then
     go to 25
 else if(i.eq.97) then
     zi(j,i)=-1.
 else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
     go to 25
 else if(i.eq.98) then
     zi(j,i)=-1.
 else if((i.eq.99).and.(j.ge.30).and.(j.le.240)) then
     go to 25
 else if(i.eq.99) then
     zi(j,i)=-1.
 else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
     go to 25
 else if(i.eq.100) then
     zi(j,i)=-1.
 else if((i.eq.101).and.(j.ge.40).and.(j.le.240)) then
     go to 25
 else if(i.eq.101) then
     zi(j,i)=-1.
 else if((i.eq.102).and.(j.ge.45).and.(j.le.239)) then
     go to 25
 else if(i.eq.102) then
     zi(j,i)=-1.
 else if((i.eq.103).and.(j.ge.50).and.(j.le.241)) then
     go to 25
 else if(i.eq.103) then
     zi(j,i)=-1.
 else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
     go to 25
 else if(i.eq.104) then
     zi(j,i)=-1.
 else if((i.eq.105).and.(j.ge.79).and.(j.le.245)) then
     go to 25
 else if(i.eq.105) then
     zi(j,i)=-1.
 else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
     go to 25
 else if(i.eq.106) then
     zi(j,i)=-1.
 else if((i.eq.107).and.(j.ge.95).and.(j.le.251)) then
     go to 25
 else if(i.eq.107) then
     zi(j,i)=-1.
 else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
     go to 25
 else if(i.eq.108) then
     zi(j,i)=-1.
 else if((i.eq.109).and.(((j.ge.106).and.(j.le.112)).or.((j.ge.115).and.
&(j.le.117)).or.((j.ge.120).and.(j.le.253)))) then
     go to 25
 else if(i.eq.109) then
     zi(j,i)=-1.
 else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
     go to 25
 else if(i.eq.110) then
     zi(j,i)=-1.
 else if((i.eq.111).and.(j.ge.127).and.(j.le.255)) then
```

```
     go to 25
 else if(i.eq.111) then
      zi(j,i)=-1.
 else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
     go to 25
 else if(i.eq.112) then
      zi(j,i)=-1.
 else if((i.eq.113).and.(j.ge.131).and.(j.le.255)) then
     go to 25
 else if(i.eq.113) then
      zi(j,i)=-1.
 else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
&.and.(j.le.256)))) then
     go to 25
 else if(i.eq.114) then
      zi(j,i)=-1.
 else if((i.eq.115).and.(j.ge.147).and.(j.le.257)) then
     go to 25
 else if(i.eq.115) then
      zi(j,i)=-1.
 else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
&.and.(j.le.257)))) then
     go to 25
 else if(i.eq.116) then
      zi(j,i)=-1.
 else if((i.eq.117).and.(((j.ge.152).and.(j.le.196)).or.((j.ge.212)
&.and.(j.le.257)))) then
     go to 25
 else if(i.eq.117) then
      zi(j,i)=-1.
 else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
&.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
     go to 25
 else if(i.eq.118) then
      zi(j,i)=-1.
 else if((i.eq.119).and.(((j.ge.155).and.(j.le.191)).or.((j.ge.226)
&.and.(j.le.230)).or.((j.ge.247).and.(j.le.258))
&.or.((j.ge.277).and.(j.le.279))
&.or.((j.ge.281).and.(j.le.283)))) then
     go to 25
 else if(i.eq.119) then
      zi(j,i)=-1.
 else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
&.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
     go to 25
 else if(i.eq.120) then
      zi(j,i)=-1.
 else if((i.eq.121).and.(((j.ge.170).and.(j.le.183)).or.((j.ge.238)
&.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
     go to 25
 else if(i.eq.121) then
      zi(j,i)=-1.
 else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.275)
&.and.(j.le.283)))) then
     go to 25
 else if(i.eq.122) then
      zi(j,i)=-1.
 else if((i.eq.123).and.(((j.ge.240).and.(j.le.258)).or.((j.ge.273)
&.and.(j.le.283)))) then
     go to 25
```

```
 else if(i.eq.123) then
      zi(j,i)=-1.
 else if((i.eq.124).and.(((j.ge.243).and.(j.le.259)).or.((j.ge.272)
&.and.(j.le.281)))) then
     go to 25
 else if(i.eq.124) then
      zi(j,i)=-1.
 else if((i.eq.125).and.(((j.ge.245).and.(j.le.260)).or.((j.ge.269)
&.and.(j.le.280)))) then
     go to 25
 else if(i.eq.125) then
      zi(j,i)=-1.
 else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
&.and.(j.le.279)))) then
     go to 25
 else if(i.eq.126) then
      zi(j,i)=-1.
 else if((i.eq.127).and.(j.ge.251).and.(j.le.276)) then
     go to 25
 else if(i.eq.127) then
      zi(j,i)=-1.
 else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
     go to 25
 else if(i.eq.128) then
      zi(j,i)=-1.
 else if((i.eq.129).and.(j.ge.254).and.(j.le.276)) then
     go to 25
 else if(i.eq.129) then
      zi(j,i)=-1.
 else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
     go to 25
 else if(i.eq.130) then
      zi(j,i)=-1.
 else if((i.eq.131).and.(j.ge.273).and.(j.le.275)) then
     go to 25
 else if(i.eq.131) then
      zi(j,i)=-1.
 else if(i.eq.132) then
      zi(j,i)=-1.
     go to 20
 else if(i.eq.133) then
     zi(j,i)=-1.
     go to 20
 else if(i.eq.134) then
     zi(j,i)=-1.
     go to 20
 else if(i.eq.135) then
     zi(j,i)=-1.
     go to 20
 else if(i.eq.136) then
     zi(j,i)=-1.
     go to 20
 else if(i.eq.137) then
     zi(j,i)=-1.
     go to 20
 else if(i.eq.138) then
     zi(j,i)=-1.
     go to 20
 else if(i.eq.139) then
     zi(j,i)=-1.
```

```
          go to 20
      else if(i.eq.140) then
          zi(j,i)=-1.
          go to 20
      end if
c
25    continue
20    continue
10    continue
c
c     this portion of the program looks for undulations which may have
c     been created by the surface fitting subroutine.  The values
c     less than 0.01 mm. are considered zero
c
      do 60 i=1,nxi
      do 70 j=1,nyi
      if(zl(i,j).le.0.01) then
          zl(i,j)=0.0
      end if
c
      if(zi(i,j).eq.-1.) then
          go to 70
      else if(zi(i,j).le.0.01) then
          zi(i,j)=0.0
      end if
70    continue
60    continue
      do 65 i=1,ngl
      do 75 j=1,ngl
      if(depthl(i,j).le.0.01) then
          depthl(i,j)=0.0
      end if
75    continue
65    continue
c
c     **************************************************************************
c
c     write the entire coarse (square simulated field (whole field)
c     to file29 ,and parts to file30
c
c     **************************************************************************
c
      write(29,1280)im
1280  format(3x,'model number ',i2,' param al ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
      write(29,1290)iyear(1),imonth(1),iday(1)
      write(30,1290)iyear(1),imonth(1),iday(1)
      write(30,1280)im,am,ramda,ealpha
      do 667 ii=1,301,2
      do 677 jj=1,301,2
          i=(ii+1)/2.0
          j=(jj+1)/2.0
c      write(29,1300)depthl(ii,jj)
      write(29,1300)i,j,depthl(ii,jj)
677   continue
667   continue
1290  format(3x,i3,'/',i3,'/',i3)
1300  format(2x,i4,i4,f8.3)
c
c     *******************************************************************
```

```
c
c     Boundary area integration : Level I
c
c     this portion of the program averages the 4 node depth values
c     to arrive at an average depth for the element.
c     the depth of the element is an average depth of the point depths
c     at each corner of the element
c
c     *****************************************************************
c
c     initialize counting variable k
c
          k=0
c
      do 130 j=1,nyi1
          l5=0
      do 140 i=1,nxi1
c
      if((zi(i,j).eq.-1.).or.(zi(i+1,j).eq.-1.).or.(zi(i,j+1).eq.-1.).or.
     &(zi(i+1,j+1).eq.-1.)) then
          go to 140
      end if
          k=k+1
          d(k)=(zi(i,j)+zi(i+1,j)+zi(i,j+1)+zi(i+1,j+1))/4.
140    continue
130    continue
c
          e=k
      print,'done with the Level I integration'
c
c     this portion of the program will sift through all the element
c     depths and find the maximum value (truncated as an integer)
c
      do 150 i=1,k-1
          a(2)=d(i)
          a(3)=d(i+1)
      if(a(2).gt.a(3)) then
          a(4)=a(2)
      else if (a(3).gt.a(2)) then
          a(4)=a(3)
      end if
      if(a(4).gt.a(1)) then
          a(1)=a(4)
      end if
          max=a(1)
          amax=max+1
150   continue
      do 160 i=-2,max
      do 170 j=1,k
      if((d(j).lt.0.).and.(i.eq.-2)) then
          count(-2)=count(-2)+1
      else if((d(j).eq.0.).and.(i.eq.-1)) then
          count(-1)=count(-1)+1
      else if((d(j).gt.0.).and.(d(j).lt.1.).and.(i.eq.0).and.
     &(max.eq.0)) then
          count(0)=count(0)+1
      else if((d(j).gt.0.).and.(d(j).lt.1.).and.(i.eq.0)) then
          count(0)=count(0)+1
      else if ((d(j).ge.i).and.(d(j).lt.(i+1))) then
      if(i.eq.0) then
```

```
            go to 170
        end if
            count(i)=count(i)+1
        end if
170     continue
160     continue
c
c       the following is cumulative frequency (top down)
c
        totcount=0.0
        do 180 i=max+1,-2,-1
        if(i.eq.-2)then
            go to 180
        else if(i.eq.-1) then
            go to 180
        else if((count(0).gt.0.).and.(max.eq.0).and.(i.eq.0))then
            totcount=totcount+count(0)
            totcnt1(i)=totcount
        else if((count(0).gt.0.).and.(i.eq.0)) then
            totcount=totcount+count(0)
            totcnt1(i)=totcount
        else if(max.eq.0) then
        totcount=0.0
            totcnt1(i)=totcount
        else
        totcount=totcount+count(i)
            totcnt1(i)=totcount
        end if
180     continue
c
c       the following is cumulative frequency (bottom up)
c
            e=k
            kk=amax+2
        write(62,1130)kk
        write(62,1140)iyear(1),imonth(1),iday(1)
        do 190 i=-1,amax
        if(i.eq.-1) then
        write(62,1090)count(-1)/e
            go to 190
        else if(i.eq.0) then
        write(62,1110)totcnt1(0)/e
        else
        write(62,1120)i,totcnt1(i)/e
        end if
            x(i+1)=i
            y(i+1)=totcnt1(i)/e
190     continue
c
c       ****************************************************************
c
c
c       Reactangular area integration : Level II
c
c       this portion of the program averages the 4 node depth values
c       to arrive at an average depth for the element.
c       the depth of the element is an average depth of the point depths
c       at each corner of the element
c
c       ****************************************************************
```

```
c
c     initialize counting variable k
c
          k=0
c
      do 132 j=1,nyi1
          l5=0
      do 142 i=1,nxi1
c
          k=k+1
          d2(k)=(z1(i,j)+z1(i+1,j)+z1(i,j+1)+z1(i+1,j+1))/4.
142    continue
132    continue
c
          e=k
      print,'done with Level II integration'
c
c     this portion of the program will sift through all the element
c     depths and find the maximum value (truncated as an integer)
c
      do 152 i=1,k-1
          a(2)=d2(i)
          a(3)=d2(i+1)
      if(a(2).gt.a(3)) then
          a(4)=a(2)
      else if (a(3).gt.a(2)) then
          a(4)=a(3)
      end if
      if(a(4).gt.a(1)) then
          a(1)=a(4)
      end if
          max2=a(1)
          amax2=max2+1
152   continue
      do 162 i=-2,max2
      do 172 j=1,k
      if((d2(j).lt.0.).and.(i.eq.-2)) then
          count2(-2)=count2(-2)+1
      else if((d2(j).eq.0.).and.(i.eq.-1)) then
          count2(-1)=count2(-1)+1
      else if((d2(j).gt.0.).and.(d2(j).lt.1.).and.(i.eq.0).and.
     &(max2.eq.0)) then
          count2(0)=count2(0)+1
      else if((d2(j).gt.0.).and.(d2(j).lt.1.).and.(i.eq.0)) then
          count2(0)=count2(0)+1
      else if ((d2(j).ge.i).and.(d2(j).lt.(i+1))) then
      if(i.eq.0) then
          go to 172
      end if
          count2(i)=count2(i)+1
      end if
172   continue
162   continue
c
c     the following is cumulative frequency (top down)
c
      totcount=0.0
      do 182 i=max2+1,-2,-1
      if(i.eq.-2)then
          go to 182
```

```
      else if(i.eq.-1) then
          go to 182
      else if((count2(0).gt.0.).and.(max2.eq.0).and.(i.eq.0))then
          totcount=totcount+count2(0)
          totcnt2(i)=totcount
      else if((count2(0).gt.0.).and.(i.eq.0)) then
          totcount=totcount+count2(0)
          totcnt2(i)=totcount
      else if(max2.eq.0) then
          totcount=0.0
          totcnt2(i)=totcount
      else
      totcount=totcount+count2(i)
          totcnt2(i)=totcount
      end if
182   continue
c
c     the following is cumulative frequency (bottom up)
c
          e=k
          kk=amax2+2
      write(62,1130)kk
      write(62,1140)iyear(1),imonth(1),iday(1)
      do 192 i=-1,amax2
      if(i.eq.-1) then
      write(62,1090)count2(-1)/e
          go to 192
      else if(i.eq.0) then
      write(62,1110)totcnt2(0)/e
      else
      write(62,1120)i,totcnt2(i)/e
      end if
          x2(i+1)=i
          y2(i+1)=totcnt2(i)/e
192   continue
c
c     *****************************************************************
c
c     Square area integration : Level III
c
c     this portion of the program averages the 4 node depth values
c     to arrive at an average depth for the element.
c     the depth of the element is an average depth of the point depths
c     at each corner of the element
c
c     *****************************************************************
c
c     initialize counting variable k
c
          k=0
c
      do 134 j=1,ngl-1
          l5=0
      do 144 i=1,ngl-1
c
          k=k+1
          d3(k)=(depth1(i,j)+depth1(i+1,j)+depth1(i,j+1)+depth1(i+1,j+1))/4.
144    continue
134    continue
c
```

```
      e=k
      print,'done with Level III integration'
c
c     this portion of the program will sift through all the element
c     depths and find the maximum value (truncated as an integer)
c
      do 154 i=1,k-1
         a(2)=d3(i)
         a(3)=d3(i+1)
      if(a(2).gt.a(3)) then
         a(4)=a(2)
      else if (a(3).gt.a(2)) then
         a(4)=a(3)
      end if
      if(a(4).gt.a(1)) then
         a(1)=a(4)
      end if
         max3=a(1)
         amax3=max3+1
154   continue
      do 164 i=-2,max3
      do 174 j=1,k
      if((d3(j).lt.0.).and.(i.eq.-2)) then
         count3(-2)=count3(-2)+1
      else if((d3(j).eq.0.).and.(i.eq.-1)) then
         count3(-1)=count3(-1)+1
      else if((d3(j).gt.0.).and.(d3(j).lt.1.).and.(i.eq.0).and.
     &(max3.eq.0)) then
         count3(0)=count3(0)+1
      else if((d3(j).gt.0.).and.(d3(j).lt.1.).and.(i.eq.0)) then
         count3(0)=count3(0)+1
      else if ((d3(j).ge.i).and.(d3(j).lt.(i+1))) then
      if(i.eq.0) then
         go to 174
      end if
         count3(i)=count3(i)+1
      end if
174   continue
164   continue
c
c     the following is cumulative frequency (top down)
c
      totcount=0.0
      do 184 i=max3+1,-2,-1
      if(i.eq.-2)then
         go to 184
      else if(i.eq.-1) then
         go to 184
      else if((count3(0).gt.0.).and.(max3.eq.0).and.(i.eq.0))then
         totcount=totcount+count3(0)
         totcnt3(i)=totcount
      else if((count3(0).gt.0.).and.(i.eq.0)) then
         totcount=totcount+count3(0)
         totcnt3(i)=totcount
      else if(max3.eq.0) then
         totcount=0.0
         totcnt3(i)=totcount
      else
      totcount=totcount+count3(i)
         totcnt3(i)=totcount
```

```
      end if
184   continue
c
c     the following is cumulative frequency (bottom up)
c
          e=k
          kk=amax3+2
      write(62,1130)kk
      write(62,1140)iyear(1),imonth(1),iday(1)
      do 194 i=-1,amax3
      if(i.eq.-1) then
      write(62,1090)count3(-1)/e
          go to 194
      else if(i.eq.0) then
      write(62,1110)totcnt3(0)/e
      else
      write(62,1120)i,totcnt3(i)/e
      end if
          x3(i+1)=i
          y3(i+1)=totcnt3(i)/e
194   continue
c
          go to 35
1050  format('there were ',f9.0,' neg. elements in the integration routine')
1060  format('total unwetted area (0 mm. rainfall)=',f9.2,' sq.km.'/)
1080  format('area wetted by at least ',i3,' mm. or more was ',f9.2,' sq.km.')
1070  format('area wetted by more than 0 mm. was',f9.2,' sq.km.')
1090  format('dry fract. total area (0 mm. rainfall)=',f9.3)
1120  format('fract. of area wetted by at least ',i3,' mm. or more was ',f9.3)
1110  format('fract. of area wetted by more than 0 mm. was',f9.3)
1130  format('there are',i3,' lines in this file')
1140  format(3(i2))
1010  format('storm day ',i2,'/',i2,'/',i2)
c
4     continue
      print,'the run has been aborted'
35     continue
      end
```

```
c                              SIMCRGAM.FORTRAN
c
c      The purpose of this program is to sample the coarse grid mesh
c      simulation field: Level I, Level II, and Level III, for the
c      for the spatial correlation and variance function
c
c
c      This program was developed by Neil M. Fennessey at M.I.T. during
c      the course of his Master's Degree research about the Areal Distribution
c      of Rainfall
c
c       ..................................................................
c
c                              Definition of Variable
c
c                                 Storm Day Date
c
c      iyr               year
c      amnth             month
c      iiday             day
c
c
c                               Program Variables
c
c      iwat              watershed i.d. number
c      igage             raingage number
c      zd                raingage depth (mm)
c      xd                raingage spatial coordinate
c      yd                raingage spatial coordinate
c      im                model number
c      am                model i spread function parameter a i
c      ramda             model i storm cell concentration parameter
c      ealpha            Model i cell center depth parameter: E(alpha i)
c      depth             Level III simulation field
c      zi                Level I simulation field
c      zl                Level II simulation field
c      nxi               number of columns in Level II and Level I random field
c      nyi               number of rows in Level II and Level I random field
c      xarea             Level I variance function element area
c      xarea2            Level II variance function element area
c      xarea3            Level III variance function element area
c      gamma             Level I variance function
c      gamma2            Level II variance function
c      gamma3            Level III variance function
c      ygamma            Level I variance function
c      ygamma2           Level II variance function
c      ygamma3           Level III variance function
c      eledpth           Level I variance function element depth
c      aveeledpth        average  Level I variance function element depth
c      vareldpth         variance of  Level I variance function element depth
c      eledpth2          Level II variance function element depth
c      aveeledpth2       average  Level II variance function element depth
c      vareldpth2        variance of  Level II variance function element depth
c      eledpth3          Level III variance function element depth
c      aveeledpth3       average  Level III variance function element depth
c      vareldpth3        variance of  Level III variance function element depth
c
c      lag1              Level I spatial lag distance
c      lag2              Level II spatial lag distance
c      lag3              Level III spatial lag distance
```

```
c      corr1              Level I spatial correlation
c      corr2              Level II spatial correlation
c      corr3              Level III spatial correlation
c
c      ...................................................................
c
       dimension   iwat(93),igage(93),iyear(93),imonth(93)
       dimension   iday(93),zd(93),xd(93),yd(93)
       dimension   xi(147),yi(70),x(100),y(100)
       dimension   amonth(6),zi(147,70)
c
       dimension   l1(31),a(31)
       dimension   xarea(31),ygamma(31),area(0:30),gamma(0:30)
       dimension   xarea2(31),ygamma2(31),area2(0:30),gamma2(0:30)
       dimension   xarea3(31),ygamma3(31),area3(0:30),gamma3(0:30)
       dimension   eledpth(5000),aveeldph(0:30),vareldph(0:30)
       dimension   corr1(31),lag1(31),xzero(31),yzero(31),alpha(100)
       dimension   aveeldp2(0:30),vareldp2(0:30)
       dimension   corr2(31),lag2(31),corr3(31),lag3(31)
       dimension   aveeldp3(0:30),vareldp3(0:30)
c
       common/blank/zj(147,70)
       common work(40000)
       common /dpmod1/depth1(151,151)
       common/ rectang /z1(147,70)
       common /recelemt/eledpth2(24000)
       common /sqelemt/eledpth3(24000)
c
       double precision dseed
c
       real length,lag1,lag,lag2,lag3
c
c
       data (amonth(i),i=1,5)/'June', 'July', 'Aug', 'Sept', 'Oct'/
       data pi/3.14159265389793/
           pi2=2*pi
           api=pi
c
c
c
           f1=1.0
c
        if(f1.eq.1.0) then
           go to 3
        else
           go to 4
        end if
c
c
3      continue
c
c       the section reads in the storm data from file83
c
           nd3=83
c
        do 30 i=1,93
        read(83,1000) iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i),
       &xd(i),yd(i)
30     continue
        rewind(nd3)
```

```
1000  format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
1110  format(f6.2,2(1x,f4.0))
c
          iiday=iday(1)
          iiyear=1900+iyear(1)
          iii=imonth(1)-5
      print,'..........................  '
      print1005,amonth(iii),iiday,iiyear
      print,'..........................  '
1005  format('storm day ',a4,i3,i5)
      print,' '
c
c    ************************************************************************
c
c     preset the common block variables equal to zero
c
c    ************************************************************************
c
      do 670 i=1,151
      do 680 j=1,151
          depth1(i,j)=0.0
680   continue
670   continue
      do 675 i=1,147
      do 685 j=1,70
          z1(i,j)=0.0
685   continue
675   continue
c
      do 695 i=1,24000
          eledpth2(i)=0.0
          eledpth3(i)=0.0
695   continue
c
c     ************************************************************************
c
c     read in the entire coarse (square simulated field (whole field)
c     from file29 generated by simwet.fortran
c
c     ************************************************************************
c
      read(29,1280) im
1280  format(3x,'model number ',i2,' param a1 ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
      read(29,1290) iyear(1),imonth(1),iday(1)
      read(30,1290) iyear(1),imonth(1),iday(1)
      read(30,1280) im,am,ramda,ealpha
      write(30,1290) iyear(1),imonth(1),iday(1)
      write(30,1280) im,am,ramda,ealpha
      do 667 i=1,151
      do 677 j=1,151
      read(29,1300) i,j,depth1(i,j)
677   continue
667   continue
1290  format(3x,i3,'/',i3,'/',i3)
1300  format(2x,i4,i4,f8.3)
c
c     ************************************************************************
c
c     retain the appropriate values for inclusion within the basin outline
```

```
c     from the simnulataion and create the Level I and Level II random fields
c
c     **********************************************************************
c
      do 650 i=1,147
      do 660 j=40,109
          l=j-39
          zi(i,l)=depth1(i,j)
          zl(i,l)=depth1(i,j)
660   continue
650   continue
c
      nxi=147
      nyi=70
      nxi1=nxi-1
      nyi1=nyi-1
      nxi2=2*nxi
      nyi2=2*nyi
c
c     ******************************************************************
c
c     this section of the program overlays the outside of the walnut
c     gulch basin with depth values of -1. mm.
c
c     ******************************************************************
c
c     nxi refers to the number of columns in the matrix
c     nyi refers to the number of rows in the matrix
c
      do 10 i=2,nyi2,2
      do 20 j=2,nxi2,2
          l=j/2
          m=i/2
c
      if((i.eq.2)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.4)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.6)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.8)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.10).and.(((j.ge.126).and.(j.le.128)).or.((j.ge.148).and.
     &(j.le.150)))) then
          go to 25
      else if(i.eq.10) then
          zi(l,m)=-1.
      else if((i.eq.12).and.(((j.ge.88).and.(j.le.90)).or.((j.ge.126)
     &.and.(j.le.132)).or.((j.ge.146).and.(j.le.150)))) then
          go to 25
      else if(i.eq.12) then
          zi(l,m)=-1.
      else if((i.eq.14).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126)
     &.and.(j.le.138)).or.((j.ge.140).and.(j.le.152)))) then
          go to 25
      else if(i.eq.14) then
```

```
     zi(l,m)=-1.
 else if((i.eq.16).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.116).and.
&(j.le.160)))) then
     go to 25
 else if(i.eq.16) then
     zi(l,m)=-1.
 else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
&(j.le.162)))) then
     go to 25
 else if(i.eq.18) then
     zi(l,m)=-1.
 else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.116).and.
&(j.le.164)))) then
     go to 25
 else if(i.eq.20) then
     zi(l,m)=-1.
 else if((i.eq.22).and.(((j.ge.80).and.(j.le.82)).or.((j.ge.84).and.
&(j.le.102)).or.((j.ge.118).and.(j.le.166)))) then
     go to 25
 else if(i.eq.22) then
     zi(l,m)=-1.
 else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
     go to 25
 else if(i.eq.24) then
     zi(l,m)=-1.
 else if((i.eq.26).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
     go to 25
 else if(i.eq.26) then
     zi(l,m)=-1.
 else if((i.eq.28).and.(((j.ge.84).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.170)))) then
     go to 25
 else if(i.eq.28) then
     zi(l,m)=-1.
 else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.110).and.
&(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
     go to 25
 else if(i.eq.30) then
     zi(l,m)=-1.
     go to 25
 else if((i.eq.32).and.(j.ge.82).and.(j.le.170)) then
     go to 25
 else if(i.eq.32) then
     zi(l,m)=-1.
 else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
     go to 25
 else if(i.eq.34) then
     zi(l,m)=-1.
     go to 25
 else if((i.eq.36).and.(j.ge.82).and.(j.le.174)) then
     go to 25
 else if(i.eq.36) then
     zi(l,m)=-1.
 else if((i.eq.38).and.(j.ge.84).and.(j.le.174)) then
     go to 25
 else if(i.eq.38) then
     zi(l,m)=-1.
 else if((i.eq.40).and.(j.ge.82).and.(j.le.172)) then
```

```
      go to 25
   else if(i.eq.40) then
      zi(l,m)=-1.
   else if((i.eq.42).and.(((j.ge.52).and.(j.le.54)).or.((j.ge.70).and.
  &(j.le.74)).or.((j.ge.78).and.(j.le.172)))) then
      go to 25
   else if(i.eq.42) then
      zi(l,m)=-1.
   else if((i.eq.44).and.(((j.ge.52).and.(j.le.56)).or.((j.ge.58).and.
  &(j.le.172)))) then
      go to 25
   else if(i.eq.44) then
      zi(l,m)=-1.
   else if((i.eq.46).and.(j.ge.52).and.(j.le.174)) then
      go to 25
   else if(i.eq.46) then
      zi(l,m)=-1.
   else if((i.eq.48).and.(j.ge.54).and.(j.le.174)) then
      go to 25
   else if(i.eq.48) then
      zi(l,m)=-1.
   else if((i.eq.50).and.(((j.ge.58).and.(j.le.174)).or.((j.ge.180).and.
  &(j.le.182)))) then
      go to 25
   else if(i.eq.50) then
      zi(l,m)=-1.
   else if((i.eq.52).and.(j.ge.58).and.(j.le.182)) then
      go to 25
   else if(i.eq.52) then
      zi(l,m)=-1.
   else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
      go to 25
   else if(i.eq.54) then
      zi(l,m)=-1.
   else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
      go to 25
   else if(i.eq.56) then
      zi(l,m)=-1.
   else if((i.eq.58).and.(((j.ge.36).and.(j.le.38)).or.((j.ge.52).and.
  &(j.le.196)))) then
      go to 25
   else if(i.eq.58) then
      zi(l,m)=-1.
   else if((i.eq.60).and.(j.ge.34).and.(j.le.198)) then
      go to 25
   else if(i.eq.60) then
      zi(l,m)=-1.
   else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
      go to 25
   else if(i.eq.62) then
      zi(l,m)=-1.
   else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
  &(j.le.221)))) then
      go to 25
   else if(i.eq.64) then
      zi(l,m)=-1.
   else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
      go to 25
   else if(i.eq.66) then
      zi(l,m)=-1.
```

```
else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
    go to 25
else if(i.eq.68) then
    zi(l,m)=-1.
else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
    go to 25
else if(i.eq.70) then
    zi(l,m)=-1.
else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
    go to 25
else if(i.eq.72) then
    zi(l,m)=-1.
else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
    go to 25
else if(i.eq.74) then
    zi(l,m)=-1.
else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
    go to 25
else if(i.eq.76) then
    zi(l,m)=-1.
else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
    go to 25
else if(i.eq.78) then
    zi(l,m)=-1.
else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
    go to 25
else if(i.eq.80) then
    zi(l,m)=-1.
else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
    go to 25
else if(i.eq.82) then
    zi(l,m)=-1.
else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
    go to 25
else if(i.eq.84) then
    zi(l,m)=-1.
else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
    go to 25
else if(i.eq.86) then
    zi(l,m)=-1.
else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
    go to 25
else if(i.eq.88) then
    zi(l,m)=-1.
else if((i.eq.90).and.(j.ge.17).and.(j.le.238)) then
    go to 25
else if(i.eq.90) then
    zi(l,m)=-1.
else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
    go to 25
else if(i.eq.92) then
    zi(l,m)=-1.
else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
    go to 25
else if(i.eq.94) then
    zi(l,m)=-1.
else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
    go to 25
else if(i.eq.96) then
    zi(l,m)=-1.
```

```
 else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
     go to 25
 else if(i.eq.98) then
     zi(l,m)=-1.
 else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
     go to 25
 else if(i.eq.100) then
     zi(l,m)=-1.
 else if((i.eq.102).and.(j.ge.44).and.(j.le.239)) then
     go to 25
 else if(i.eq.102) then
     zi(l,m)=-1.
 else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
     go to 25
 else if(i.eq.104) then
     zi(l,m)=-1.
 else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
     go to 25
 else if(i.eq.106) then
     zi(l,m)=-1.
 else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
     go to 25
 else if(i.eq.108) then
     zi(l,m)=-1.
 else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
     go to 25
 else if(i.eq.110) then
      zi(l,m)=-1.
 else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
     go to 25
 else if(i.eq.112) then
      zi(l,m)=-1.
 else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
&.and.(j.le.256)))) then
     go to 25
 else if(i.eq.114) then
      zi(l,m)=-1.
 else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
&.and.(j.le.257)))) then
     go to 25
 else if(i.eq.116) then
      zi(l,m)=-1.
 else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
&.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
     go to 25
 else if(i.eq.118) then
      zi(l,m)=-1.
 else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
&.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
     go to 25
 else if(i.eq.120) then
      zi(l,m)=-1.
 else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.273)
&.and.(j.le.283)))) then
     go to 25
 else if(i.eq.122) then
      zi(l,m)=-1.
 else if((i.eq.124).and.(((j.ge.243).and.(j.le.258)).or.((j.ge.272)
&.and.(j.le.281)))) then
     go to 25
```

```
      else if(i.eq.124) then
           zi(1,m)=-1.
      else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
     &.and.(j.le.279)))) then
          go to 25
      else if(i.eq.126) then
           zi(1,m)=-1.
      else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
          go to 25
      else if(i.eq.128) then
           zi(1,m)=-1.
      else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
          go to 25
      else if(i.eq.130) then
           zi(1,m)=-1.
      else if(i.eq.132) then
           zi(1,m)=-1.
          go to 20
      else if(i.eq.134) then
          zi(1,m)=-1.
          go to 20
      else if(i.eq.136) then
          zi(1,m)=-1.
          go to 20
      else if(i.eq.138) then
          zi(1,m)=-1.
          go to 20
      else if(i.eq.140) then
          zi(1,m)=-1.
          go to 20
      end if
c
25    continue
20    continue
10    continue
c
c
c     this portion of the program looks for undulations which may have
c     been created by the surface fitting subroutine, anything less than
c     0.01 mm. equals zero.  This filter is also applied to the rectangular
c     field as well as the full square field.
c
      do 80 i=1,nxi
      do 90 j=1,nyi
      if(zl(i,j).le.0.01) then
          zl(i,j)=0.0
      end if
      if(zi(i,j).eq.-1.) then
          go to 90
      else if(zi(i,j).le.0.01) then
          zi(i,j)=0.0
      end if
90    continue
80    continue
c
      do 85 i=1,151
      do 95 j=1,151
      if(depth1(i,j).le.0.01) then
          depth1(i,j)=0.0
      end if
```

```
95    continue
85    continue
c
c     calculate the average point depth for the field within the basin limits
c
      do 110 i=1,nxi
      do 120 j=1,nyi
          a12=z1(i,j)+a12
          d12=d12+1
c
      if(zi(i,j).eq.-1.) then
          go to 120
      else
          a1=zi(i,j)+a1
          d1=d1+1
      end if
120   continue
110   continue
          avedepth=a1/d1
          avdprec=a12/d12
c
c     calculate the average point depth for the square simulated field
c     (entire)
c
      do 115 i=1,151
      do 125 j=1,151
          a14=depth1(i,j)+a14
          d14=d14+1
125   continue
115   continue
          avdpsq=a14/d14
c

c
c
c     this portion will calculate the variance of the point depth within the
c     basin limits as well as the variance of the point depth of the
c     rectangular field.  Calculate the coefficient of skewness for both
c     as well
c
c
          l=0
          l12=0
          l14=0
          a1=0.0
          a2=0.0
          a12=0.0
          a14=0.0
c
      do 130 i=1,nxi
      do 140 j=1,nyi
c
          l12=l12+1
          a12=((z1(i,j)-avdprec)**2.)+a12
          a122=((z1(i,j)-avdprec)**3.)+a122
      if(zi(i,j).eq.-1.) then
          go to 140
      else
          l=l+1
          a2=((zi(i,j)-avedepth)**2.)+a2
```

```
          a222=((zi(i,j)-avedepth)**3.)+a222
      end if
140   continue
130   continue
c
c      calculate the variance of the point depth for the entire square
c     simulated field.  Calculate the coefficient of skewness as well.
c
      do 135 i=1,151
      do 145 j=1,151
c
          l14=l14+1
          a14=((depth1(i,j)-avdpsq)**2.)+a14
          a144=((depth1(i,j)-avdpsq)**3.)+a144
145   continue
135   continue
          vardepth=a2/(l-1)
          vardprec=a12/(l12-1)
          vardpsq=a14/(l14-1)
          stdevdph=sqrt(vardepth)
          stddvrec=sqrt(vardprec)
          stddvsq=sqrt(vardpsq)
          skwdpth=a222/((l-1)*(stdevdph**3.0))
          skwdprec=a122/((l12-1)*(stddvrec**3.0))
          skwdpsq=a144/((l14-1)*(stddvsq**3.0))
      print 1030,avedepth
      print 1040,vardepth
1030  format('the average point depth=',f10.3,' mm.')
1040  format('the variance of the point depth=',f10.3,' mm ')
      print 1032,avdprec
      print 1042,vardprec
1032  format('the average rectang. point depth=',f10.3,' mm.')
1042  format('the variance rectang. of the point depth=',f10.3,' mm ')
      print 1034,avdpsq
      print 1044,vardpsq
1034  format('the average square point depth=',f10.3,' mm.')
1044  format('the variance of the square point depth=',f10.3,' mm ')
c
c     reset variables equal to zero
c
          a1=0.0
          a2=0.0
          a12=0.0
          a14=0.0
          a122=0.0
          a222=0.0
          a144=0.0
c
c     ************************************************************************
c
c     this portion of the program will determine the two dimensional
c     spatial correlation.  The correlation will be determined for point
c     depth spacings from 0.0 km. to 6 km. This section is for the boundary
c     filtered field
c
c     ************************************************************************
c
c     entering the incremental (changes spatial distance) loop
c
      do 150 k1=0,30
```

```
c
            t1=0.
            b1=0.
            b2=0.
            d2=0.
            nxi3=nxi-k1
            nyi3=nyi-k1
c
c     this part of the correlation determination sweeps from left to
c     right across the generated surface fitting matrix.  Pairs of point depths
c     which are not -1.0 are the only values included in each determination.
c
      do 160 j=1,nyi
      do 170 i=1,nxi3
      if((zi(i,j).eq.-1.).or.(zi(i+k1,j).eq.-1.)) then
            go to 170
      else
            t1=((zi(i,j)-avedepth)*(zi(i+k1,j)-avedepth))+t1
            b1=((zi(i,j)-avedepth)**2.)+b1
            b2=((zi(i+k1,j)-avedepth)**2.)+b2
            d2=d2+1
      end if
170   continue
160   continue
c
c     this part of the correlation determination sweeps from the bottom
c     to the top of the generated surface fitting matrix
c
      do 180 i=1,nxi
      do 190 j=1,nyi3
      if((zi(i,j).eq.-1.).or.(zi(i,j+k1).eq.-1.)) then
            go to 190
      else
            t1=((zi(i,j)-avedepth)*(zi(i,j+k1)-avedepth))+t1
            b1=((zi(i,j)-avedepth)**2.)+b1
            b2=((zi(i,j+k1)-avedepth)**2.)+b2
            d2=d2+1
      end if
190   continue
180   continue
c
c     calculate the correlation coefficient for this spacing
c
            e1=0
            e1=t1/(sqrt(b1)*sqrt(b2))
            corr1(k1+1)=e1
            r1=k1
            r2=r1/5
            lag1(k1+1)=r2
c
150   continue
c
c     reset variables equal to zero
c
            t1=0.0
            b1=0.0
            b2=0.0
            d2=0.0
            e1=0.0
            r1=0.0
```

```
      r2=0.0
      k1=0
c
c     ************************************************************************
c
c     this portion of the program will determine the two dimensional
c     spatial correlation.  The correlation will be determined for point
c     depth spacings from 0.0 km. to 6 km. This section is for the rectangular
c     simulated field
c
c     ************************************************************************
c
c     entering the incremental (changes spatial distance) loop
c
      do 152 k1=0,30
c
         t1=0.
         b1=0.
         b2=0.
         d2=0.
         nxi3=nxi-k1
         nyi3=nyi-k1
c
c     this part of the correlation determination sweeps from left to
c     right across the generated surface fitting matrix.
c
      do 162 j=1,nyi
      do 172 i=1,nxi3
         t1=((z1(i,j)-avdprec)*(z1(i+k1,j)-avdprec))+t1
         b1=((z1(i,j)-avdprec)**2.)+b1
         b2=((z1(i+k1,j)-avdprec)**2.)+b2
         d2=d2+1
172   continue
162   continue
c
c     this part of the correlation determination sweeps from the bottom
c     to the top of the generated surface fitting matrix
c
      do 182 i=1,nxi
      do 192 j=1,nyi3
         t1=((z1(i,j)-avdprec)*(z1(i,j+k1)-avdprec))+t1
         b1=((z1(i,j)-avdprec)**2.)+b1
         b2=((z1(i,j+k1)-avdprec)**2.)+b2
         d2=d2+1
c      end if
192   continue
182   continue
c
c     calculate the correlation coefficient for this spacing
c
         e1=0
         e1=t1/(sqrt(b1)*sqrt(b2))
         corr2(k1+1)=e1
         r1=k1
         r2=r1/5
         lag2(k1+1)=r2
c
152   continue
c
c     reset variables equal to zero
```

```
c
          t1=0.0
          b1=0.0
          b2=0.0
          d2=0.0
          e1=0.0
          r1=0.0
          r2=0.0
          k1=0
c
c
c     ************************************************************************
c
c     this portion of the program will determine the two dimensional
c     spatial correlation.  The correlation will be determined for point
c     depth spacings from 0.0 km. to 6 km. This section is for the square
c     (entire) simulated field
c
c     ************************************************************************
c
c     entering the incremental (changes spatial distance) loop
c
      do 154 k1=0,30
c
          t1=0.
          b1=0.
          b2=0.
          d2=0.
          nxi3=151-k1
          nyi3=151-k1
c
c     this part of the correlation determination sweeps from left to
c     right across the generated surface fitting matrix.
c
      do 164 j=1,151
      do 174 i=1,nxi3
          t1=((depth1(i,j)-avdpsq)*(depth1(i+k1,j)-avdpsq))+t1
          b1=((depth1(i,j)-avdpsq)**2.)+b1
          b2=((depth1(i+k1,j)-avdpsq)**2.)+b2
          d2=d2+1
174   continue
164   continue
c
c     this part of the correlation determination sweeps from the bottom
c     to the top of the generated surface fitting matrix
c
      do 184 i=1,151
      do 194 j=1,nyi3
          t1=((depth1(i,j)-avdpsq)*(depth1(i,j+k1)-avdpsq))+t1
          b1=((depth1(i,j)-avdpsq)**2.)+b1
          b2=((depth1(i,j+k1)-avdpsq)**2.)+b2
          d2=d2+1
c      end if
194   continue
184   continue
c
c     calculate the correlation coefficient for this spacing
c
          e1=0
          e1=t1/(sqrt(b1)*sqrt(b2))
```

```
          corr3(k1+1)=e1
          r1=k1
          r2=r1/5
          lag3(k1+1)=r2
c
154   continue
c
c     reset variables equal to zero
c
          t1=0.0
          b1=0.0
          b2=0.0
          d2=0.0
          e1=0.0
          r1=0.0
          r2=0.0
          k1=0
c
c
c     ************************************************************************
c
c     this portion of the program determines the values of the variance
c     function gamma(area) for the boundary filtered area
c
c     ************************************************************************
c
c     begin the loop which varies the length scale of the element
c
      do 220 m=0,30
          area(m)=((m*0.2)**2.)
c
c     enter the mesh at this scale.  Sweep the mesh left to right and from
c     the bottom to the top simultaneously
c
      do 230 j=1,nyi-m
      do 240 i=1,nxi-m
c
c     trip the element dimension counter
          k1=k1+1
c
c     determine the depth of the individual element now
c
      do 250 l=j,j+m
      do 260 k=i,i+m
c
c     check to see if any of the point depths are -1.
      if(zi(k,l).eq.-1.) then
          numpts=0
          totdpth=0.
          k1=k1-1
          go to 240
      else
          numpts=numpts+1
          totdpth=totdpth+zi(k,l)
     end if
260  continue
250  continue
c
c     the element depth is the average value of the point depths
c     which comprise the element
```

```
c
          eledpth(k1)=totdpth/numpts
          toteldph=toteldph+eledpth(k1)
          numpts=0
          totdpth=0.0
240    continue
230    continue
c
c      determine the average element depth and the moving average
c
          aveeldph(m)=toteldph/k1
c
c      determine the variance of the element depth at this length scale
c
       do 270 k2=1,k1
          totvardp=totvardp+((eledpth(k2)-aveeldph(m))**2.)
270    continue
          vareldph(m)=totvardp/(k1-1)
c
c      determine the value of gamma(A)
c
          gamma(m)=vareldph(m)/vardepth
          xarea(m+1)=area(m)
          ygamma(m+1)=gamma(m)
          toteldph=0.
          totvardp=0.
          k1=0
       print221,xarea(m+1),ygamma(m+1)
220    continue
221    format('boundaried. area area=',f6.3,' gamma(A)=',f6.3)
c
c      reset variables equal to zero
c
          toteldph=0.0
          totvardp=0.0
          b1=0.0
          b2=0.0
          b3=0.0
c
c      **************************************************************************
c
c      this portion of the program determines the values of the variance
c      function gamma(area) for the simulated rectangular area : Level II
c
c      **************************************************************************
c
c      begin the loop which varies the length scale of the element
c
       do 222 m=0,30
          area2(m)=((m*0.2)**2.)
c
c      enter the mesh at this scale.  Sweep the mesh left to right and from
c      the bottom to the top simultaneously
c
       do 232 j=1,nyi-m
       do 242 i=1,nxi-m
c
c      trip the element dimension counter
          k1=k1+1
c
```

```
c     determine the depth of the individual element now
c
      do 252 l=j,j+m
      do 262 k=i,i+m
c
          numpts=numpts+1
          totdpth=totdpth+z1(k,l)
c     end if
262   continue
252   continue
c
c     the element depth is the average value of the point depths
c     which comprise the element
c
          eledpth2(k1)=totdpth/numpts
          toteldph=toteldph+eledpth2(k1)
          numpts=0
          totdpth=0.0
242    continue
232    continue
c
c     determine the average element depth and the moving average
c
          aveeldp2(m)=toteldph/k1
c
c     determine the variance of the element depth at this length scale
c
      do 272 k2=1,k1
          totvardp=totvardp+((eledpth2(k2)-aveeldp2(m))**2.)
272    continue
          vareldp2(m)=totvardp/(k1-1)
c
c     determine the value of gamma(A)
c
          gamma2(m)=vareldp2(m)/vareldp2(0)
          xarea2(m+1)=area2(m)
          ygamma2(m+1)=gamma2(m)
          toteldph=0.
          totvardp=0.
          k1=0
      print223,xarea2(m+1),ygamma2(m+1)
222   continue
223   format('rectang. area area=',f6.3,' gamma(A)=',f6.3)
c
c     reset variables equal to zero
c
          toteldph=0.0
          totvardp=0.0
          b1=0.0
          b2=0.0
          b3=0.0
c
c
c
c
c     ***********************************************************************
c
c     this portion of the program determines the values of the variance
c     function gamma(area) for the simulated square (entire field) area
c     Level III
```

```
c
c     ************************************************************************
c
c     begin the loop which varies the length scale of the element
c
      do 224 m=0,30
          area3(m)=((m*0.2)**2.)
c
c     enter the mesh at this scale.  Sweep the mesh left to right and from
c     the bottom to the top simultaneously
c
      do 234 j=1,151-m
      do 244 i=1,151-m
c
c     trip the element dimension counter
          k1=k1+1
c
c     determine the depth of the individual element now
c
      do 254 l=j,j+m
      do 264 k=i,i+m
c
          numpts=numpts+1
          totdpth=totdpth+depth1(k,l)
c     end if
264   continue
254   continue
c
c     the element depth is the average value of the point depths
c     which comprise the element
c
          eledpth3(k1)=totdpth/numpts
          toteldph=toteldph+eledpth3(k1)
          numpts=0
          totdpth=0.0
244    continue
234    continue
c
c     determine the average element depth and the moving average
c
          aveeldp3(m)=toteldph/k1
c
c     determine the variance of the element depth at this length scale
c
      do 274 k2=1,k1
          totvardp=totvardp+((eledpth3(k2)-aveeldp3(m))**2.)
274    continue
          vareldp3(m)=totvardp/(k1-1)
c
c     determine the value of gamma(A)
c
          gamma3(m)=vareldp3(m)/vareldp3(0)
          xarea3(m+1)=area3(m)
          ygamma3(m+1)=gamma3(m)
          toteldph=0.
          totvardp=0.
          k1=0
      print225,xarea3(m+1),ygamma3(m+1)
224   continue
225   format('square area area=',f6.3,' gamma(A)=',f6.3)
```

```
c
c     reset variables equal to zero
c
          toteldph=0.0
          totvardp=0.0
          b1=0.0
          b2=0.0
          b3=0.0
c

c
c     **********************************************************************
c
c     write the moments,correlation and variance function to file30
c
c     **********************************************************************
c
      write(30,1310)avedepth,vardepth,skwdpth
      write(30,1320)avdprec,vardprec,skwdprec
      write(30,1330)avdpsq,vardpsq,skwdpsq
1310  format(3x,'boundary filtered avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
1320  format(3x,'rectang. simulat. avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
1330  format(3x,'square simulat. avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
c
      do 687 i=1,31
      write(30,1340)lag1(i),corr1(i),lag2(i),corr2(i),lag3(i),corr3(i)
687   continue
1340  format(3x,3(2x,f5.2,f6.3))
      do 690 i=1,31
      write(30,1350)xarea(i),ygamma(i),xarea2(i),ygamma2(i),xarea3(i),
     &ygamma3(i)
690   continue
1350  format(3x,3(2x,f5.2,f7.3))
c
          go to 35
c
4     continue
      print,'the run has been aborted'
35     continue
      print,'**********************************************************************'
      print,'**********************************************************************'
      print,'**********************************************************************'
      print,'**********************************************************************'
      end
```

```
c                           TABLE5.FORTRAN
c
c        The purpose of this program is to create nicely formatted
c        data tables.  One table for each simulation random field.
c        Level I, Level II and level III
c
c       This program was developed by Neil M. Fennessey at M.I.T. during
c       the course of his Master's Degree research about the Areal Distribution
c       of Rainfall
c
c        ..................................................................
c
c                              Definition of Variable
c
c                                 Storm Day Date
c
c       iyr              year
c       amnth            month
c       iiday            day
c
c                               Program Variables
c
c       wetfrct1         Level I spatial distribution curve
c       wetfrct2         Level II spatial distribution curve
c       wetfrct3         Level III spatial distribution curve
c       xarea            Level I variance function element area
c       xarea2           Level II variance function element area
c       xarea3           Level III variance function element area
c       ygamma           Observed variance function
c       gamlev1          Level I variance function
c       gamlev2          Level II variance function
c       gamlev3          Level III variance function
c
c       lag1             Observed spatial lag distance
c       laglev1          Level I spatial lag distance
c       laglev2          Level II spatial lag distance
c       laglev3          Level III spatial lag distance
c       corrlev1         Level I spatial correlation
c       corrlev2         Level II spatial correlation
c       corrlev3         Level III spatial correlation
c
c       ......................................................................
c
        dimension   amonth(6),iyear(1),iday(1),imonth(1)
        dimension   wetfrct1(-2:200),wetfrct2(-2:200),wetfrct3(-2:200)
        dimension   xarea(31),ygamma(31),corr1(31),lag1(31),alpha(100)
        dimension   laglev1(31),corrlev1(31),laglev2(31),corrlev2(31)
        dimension   corrlev3(31),gamlev1(31)
        dimension   gamlev2(31),gamlev3(31),laglev3(31)
        dimension   xlev1(200),xlev2(200),xlev3(200)
c
        real lag1,lag,l1,lagmod,laglev1,laglev2,laglev3
c
        data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
c
c       ************************************************************************
c
c       read in the results from file30: the first three moments
c       the variance function and correlation function curves for the
c       three levels of sampling
```

```
c
c     ******************************************************************************
c
      read (30,1051) iyear (1) , imonth (1) , iday (1)
      read (30,1061) model,parama,lamda,ealpha
1061  format(3x,'model number ',i2,' param a1 ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
1051  format(3x,i3,'/',i3,'/',i3)
      read (30,1001) avedepth,vardepth,skwdpth
      read (30,1016) avdprec,vardprec,skwdprec
      read (30,1021) avdpsq,vardpsq,skwdpsq
1001  format(3x,'boundary filtered avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
1016  format(3x,'rectang. simulat. avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
1021  format(3x,'square simulat. avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
c
      do 20 i=1,31
      read (30,1031) laglev1 (i) ,corrlev1 (i) ,laglev2 (i) ,corrlev2 (i) ,laglev3 (i)
     &,corrlev3 (i)
20    continue
1031  format(3x,3(2x,f5.2,f6.3))
      do 30 i=1,31
      read (30,1041) xlev1 (i) ,gamlev1 (i) ,xlev2 (i) ,gamlev2 (i) ,xlev3 (i) ,
     &gamlev3 (i)
30    continue
1041  format(3x,3(2x,f5.2,f7.3))
c
c
c     ******************************************************************************
c
c     read in the results from file 62, the spatial distribution of total storm
c     depth for the three levels of sampling
c
c     ******************************************************************************
c
c
c     Level I wetted area curves
c
      read (62,1132) k1
          amax1=k1-2
      read (62,1142) iyear (1) ,imonth (1) ,iday (1)
      do 190 i=-1,amax1
      if(i.eq.-1) then
      read (62,1092) wetfrct1 (-1)
          go to 190
      else if(i.eq.0) then
      read (62,1112) wetfrct1 (0)
      else
      read (62,1122) i,wetfrct1 (i)
      end if
190   continue
c
c     Level II wetted area curves
c
      read (62,1132) k2
          amax2=k2-2
      read (62,1142) iyear (1) ,imonth (1) ,iday (1)
      do 210 i=-1,amax2
```

```
      if(i.eq.-1) then
      read(62,1092)wetfrct2(-1)
          go to 210
      else if(i.eq.0) then
      read(62,1112)wetfrct2(0)
      else
      read(62,1122)i,wetfrct2(i)
      end if
210   continue
c
c     Square area integration : Level III
c
      read(62,1132)k3
          amax3=k3-2
      read(62,1142)iyear(1),imonth(1),iday(1)
      do 220 i=-1,amax3
      if(i.eq.-1) then
      read(62,1092)wetfrct3(-1)
          go to 220
      else if(i.eq.0) then
      read(62,1112)wetfrct3(0)
      else
      read(62,1122)i,wetfrct3(i)
      end if
220   continue
c
1062  format('total unwetted area (0 mm. rainfall)=',f9.2,' sq.km.'/)
1082  format('area wetted by at least ',i3,' mm. or more was ',f9.2,' sq.km.')
1072  format('area wetted by more than 0 mm. was',f9.2,' sq.km.')
1092  format('dry fract. total area (0 mm. rainfall)=',f9.3)
1122  format('fract. of area wetted by at least ',i3,' mm. or more was ',f9.3)
1112  format('fract. of area wetted by more than 0 mm. was',f9.3)
1132  format('there are',i3,' lines in this file')
1142  format(3(i2))
1012  format('storm day ',i2,'/',i2,'/',i2)
          iiday=iday(1)
          iiyear=1900+iyear(1)
          iii=imonth(1)-5
c
c     ***********************************************************************
c
c     write the Level I sampling table to file95
c
c     ***********************************************************************
c
c
      write(95,1005)amonth(iii),iiday,iiyear
      write(95,1551)model
1551  format('                Model ',i1,' Simulation: Level I Sampling'/)
      write(95,1010)wetfrct1(-1)
      write(95,1050)wetfrct1(0)
      write(95,1290)avedepth
      write(95,1280)vardepth
      write(95,1285)skwdpth
      write(95,1055)
      write(95,1060)
      write(95,1070)
      write(95,1220)
c
c     check to see is the cumulative wetted area is larger or smaller
```

```
c     than the fixed number of observations of area correlation lag
c
      if(k1.gt.31) then
          k1=k1-2
          b=1
      else
          k1=31
          b=2
      end if
      do 40 i=1,k1
c
c     cumulative wetted fraction is greater than 31 mm.
c
      if ((b.eq.1).and.(i.le.31)) then
      write(95,1230)i,wetfrct1(i),laglev1(i),corrlev1(i),xlev1(i),gamlev1(i)
      else if((b.eq.1).and.(i.gt.31)) then
      write(95,1240)i,wetfrct1(i)
c
c     cumulative wetted fraction is less than 31 mm.
c
      else if ((b.eq.2).and.(i.le.(amax1))) then
      write(95,1230)i,wetfrct1(i),laglev1(i),corrlev1(i),xlev1(i),gamlev1(i)
      else
      write(95,1250)laglev1(i),corrlev1(i),xlev1(i),gamlev1(i)
      end if
40    continue
c
c     ************************************************************************
c
c     write the Level II table to file96
c
c     ************************************************************************
c
      write(96,1005)amonth(iii),iiday,iiyear
      write(96,1552)model
1552  format('                Model ',i1,' Simulation: Level II Sampling'/)
      write(96,1010)wetfrct2(-1)
      write(96,1050)wetfrct2(0)
      write(96,1290)avdprec
      write(96,1280)vardprec
      write(96,1285)skwdprec
      write(96,1055)
      write(96,1060)
      write(96,1070)
      write(96,1220)
c
c     check to see is the cumulative wetted area is larger or smaller
c     than the fixed number of observations of area correlation lag
c
      if(k2.gt.31) then
          k1=k2-2
          b=1
      else
          k1=31
          b=2
      end if
      do 50 i=1,k1
c
c     cumulative wetted fraction is greater than 31 mm.
c
```

```
      if ((b.eq.1).and.(i.le.31)) then
      write(96,1230)i,wetfrct2(i),laglev2(i),corrlev2(i),xlev2(i),gamlev2(i)
      else if((b.eq.1).and.(i.gt.31)) then
      write(96,1240)i,wetfrct2(i)
c
c     cumulative wetted fraction is less than 31 mm.
c
      else if ((b.eq.2).and.(i.le.(amax2))) then
      write(96,1230)i,wetfrct2(i),laglev2(i),corrlev2(i),xlev2(i),gamlev2(i)
      else
      write(96,1250)laglev2(i),corrlev2(i),xlev2(i),gamlev2(i)
      end if
50    continue
c
c     ***********************************************************************
c
c     write Level III sampling table to file97
c
c     ***********************************************************************
c
      write(97,1005)amonth(iii),iiday,iiyear
      write(97,1553)model
1553  format('                Model ',i1,' Simulation: Level III Sampling'/)
      write(97,1010)wetfrct3(-1)
      write(97,1050)wetfrct3(0)
      write(97,1290)avdpsq
      write(97,1280)vardpsq
      write(97,1285)skwdpsq
      write(97,1055)
      write(97,1060)
      write(97,1070)
      write(97,1220)
c
c     check to see is the cumulative wetted area is larger or smaller
c     than the fixed number of observations of area correlation lag
c
      if(k3.gt.31) then
          k1=k3-2
          b=1
      else
          k1=31
          b=2
      end if
      do 60 i=1,k1
c
c     cumulative wetted fraction is greater than 31 mm.
c
      if ((b.eq.1).and.(i.le.31)) then
      write(97,1230)i,wetfrct3(i),laglev3(i),corrlev3(i),xlev3(i),gamlev3(i)
      else if((b.eq.1).and.(i.gt.31)) then
      write(97,1240)i,wetfrct3(i)
c
c     cumulative wetted fraction is less than 31 mm.
c
      else if ((b.eq.2).and.(i.le.(amax3))) then
      write(97,1230)i,wetfrct3(i),laglev3(i),corrlev3(i),xlev3(i),gamlev3(i)
      else
      write(97,1250)laglev3(i),corrlev3(i),xlev3(i),gamlev3(i)
      end if
60    continue
```

```
c
          go to 4
c
c      format statements
c
1370  format(' there are ',i5,' storm days in this file for this year:'i4)
1360  format(7x,'Y  (mm.) ',2x,'Acw/Ac  (Y>y) ',10x,'v  (km.) ',1x,'rho(v) ',8x,'A
     &(km.sq.) ',1x,'Gamma(A) ')
1350  format(3x,'Cumulative Wetted Fraction',5x,'Spatial Correlation',5x,
     &'Variance Function')
1340  format(' Wetted Fraction of Total Basin Area: (Acw/Ac)=',f5.3)
1330  format(' Dry Fraction of Total Basin Area: (Acd/Ac)=',f5.3)
1320  format(' point depth E(Y)=',f7.3,' Var(Y)=',f7.3,' S.C.(Y)=',f7.3 )
1310  format(' Storm Day ',a4,i3,i5)
1300  format(' there are ',i3,' wetted area curve pts, there are ',i3,
     &' total data points this day')
c
c     file95 format statements
c
1285  format(10x,'Coef. of Skewness of Point Depth: S.C.(Y)=',f7.3//)
1280  format(10x,'Variance of Point Depth (mm. sq.): Var(Y)=',f7.3/)
1290  format(10x,'Expected Value of Point Depth (mm.): E(Y)=',f7.3/)
1250  format(37x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
1240  format(7x,i3,7x,f5.3)
1230  format(7x,i3,7x,f5.3,15x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
1220  format('____________________________________________________________
     &______________'/)
1070  format(6x,'y(mm.) ',4x,'Acw/Ac(Y>y) ',10x,'v(km.) ',2x,'rho(v) ',6x,'A
     &(km.sq.) ',2x,'Gamma(A) ')
1060  format(6x,'of Total Storm Depth',8x,'Spatial Correlation',5x,
     &'Variance Function')
1055  format(6x,'Spatial Distribution')
1050  format(9x,'Wetted Fraction of Total Basin Area: (Acw/Ac)=',f5.3//)
1010  format(10x,'Dry Fraction of Total Basin Area: (Acd/Ac)=',f5.3/)
1005  format(25x,'Storm Day ',a4,i3,i5/)
1090  format('dry fract. total area (0 mm. rainfall)=',f9.3)
1120  format('fract. of area wetted by at least ',i3,' mm. or more was ',f9.3)
1270  format('fract. of area wetted by more than 0 mm. was',f9.3)
1130  format('there are',i3,' lines in this file')
1140  format(3(i2))
c
4     continue
      end
```

```
c                            SIMPLOT.FORTRAN
C
c     this program will read in the total depth field of the
c     simulation model that has been written to file 29.  The
c     entire square simulation field (30 km. by 30 km.) will be
c     plotted.  Superimposed on this plot will be the rectangular
c     area boundary (14 km. by 29.4 km.).  Superimposed within this
c     area will be the basin boundary
c
c     This program was developed by Neil M. Fennessey at M.I.T. during
c     the course of his Master's Degree research about the Areal Distribution
c     of Rainfall
c
c      .............................................................
c
c
      dimension zi(147,70),iday(1),iyear(1),imonth(1),amonth(5)
      dimension xrecbot(294),xrectop(294),yrectop(294),yrecbot(294)
      dimension xreclft(140),xrecrht(140),yreclft(140),yrecrht(140)
c
      common work(40000)
      common /dpmodl/depth1(151,151)
      common /bodrblk/zj(151,151)
c
      real length,lag1,lag,lag2,lag3
c
c
      data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
c
c     ***********************************************************************
c
c     read the coarse grid mesh simulation field from file 29
c
c     ***********************************************************************
c
      read(29,1280)model
1280  format(3x,'model number ',i2,' param a1 ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
      read(29,1290)iyear(1),imonth(1),iday(1)
c
          iiday=iday(1)
          iiyear=1900+iyear(1)
          iii=imonth(1)-5
      print,'........................... '
      print1005,amonth(iii),iiday,iiyear
      print,'........................... '
1005  format('storm day ',a4,i3,i5)
c
      do 40 i=1,151
      do 50 j=1,151
      read(29,1300)i,j,depth1(i,j)
50    continue
40    continue
1290  format(3x,i3,'/',i3,'/',i3)
          1300  format(2x,i4,i4,f8.3)
c
c     ***********************************************************************
c
c     retain the appropriate values for inclusion within the basin outline
c     from the simnulataion
```

```
c
c     ***************************************************************************
c
      do 60 i=1,147
      do 70 j=40,109
          l=j-39
          zi(i,l)=depth1(i,j)
70    continue
60    continue
c
      nxi=147
      nyi=70
      nxi2=2*nxi
      nyi2=2*nyi
c
c     *************************************************************************
c
c     this section of the program overlays the outside of the walnut
c     gulch basin with depth values of -1. mm. The values generated by
c     the subroutine are left intact for the integration routine and
c     also the contour plotting.
c
c     nxi refers to the number of columns in the matrix
c     nyi refers to the number of rows in the matrix
c
c     ************************************************************************
c
      do 10 i=2,nyi2,2
      do 20 j=2,nxi2,2
          l=j/2
          m=i/2
c
      if((i.eq.2)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.4)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.6)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.8)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.10).and.(((j.ge.126).and.(j.le.128)).or.((j.ge.148).and.
     &(j.le.150)))) then
          go to 25
      else if(i.eq.10) then
          zi(l,m)=-1.
      else if((i.eq.12).and.(((j.ge.88).and.(j.le.90)).or.((j.ge.126)
     &.and.(j.le.132)).or.((j.ge.146).and.(j.le.150)))) then
          go to 25
      else if(i.eq.12) then
          zi(l,m)=-1.
      else if((i.eq.14).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126)
     &.and.(j.le.138)).or.((j.ge.140).and.(j.le.152)))) then
          go to 25
      else if(i.eq.14) then
          zi(l,m)=-1.
      else if((i.eq.16).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.116).and.
```

```
&(j.le.160)))) then
     go to 25
 else if(i.eq.16) then
     zi(l,m)=-1.
 else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
&(j.le.162)))) then
     go to 25
 else if(i.eq.18) then
     zi(l,m)=-1.
 else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.116).and.
&(j.le.164)))) then
     go to 25
 else if(i.eq.20) then
     zi(l,m)=-1.
 else if((i.eq.22).and.(((j.ge.80).and.(j.le.82)).or.((j.ge.84).and.
&(j.le.102)).or.((j.ge.118).and.(j.le.166)))) then
     go to 25
 else if(i.eq.22) then
     zi(l,m)=-1.
 else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
     go to 25
 else if(i.eq.24) then
     zi(l,m)=-1.
 else if((i.eq.26).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
     go to 25
 else if(i.eq.26) then
     zi(l,m)=-1.
 else if((i.eq.28).and.(((j.ge.84).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.170)))) then
     go to 25
 else if(i.eq.28) then
     zi(l,m)=-1.
 else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.110).and.
&(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
     go to 25
 else if(i.eq.30) then
     zi(l,m)=-1.
     go to 25
 else if((i.eq.32).and.(j.ge.82).and.(j.le.170)) then
     go to 25
 else if(i.eq.32) then
     zi(l,m)=-1.
 else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
     go to 25
 else if(i.eq.34) then
     zi(l,m)=-1.
     go to 25
 else if((i.eq.36).and.(j.ge.82).and.(j.le.174)) then
     go to 25
 else if(i.eq.36) then
     zi(l,m)=-1.
 else if((i.eq.38).and.(j.ge.84).and.(j.le.174)) then
     go to 25
 else if(i.eq.38) then
     zi(l,m)=-1.
 else if((i.eq.40).and.(j.ge.82).and.(j.le.172)) then
     go to 25
 else if(i.eq.40) then
```

```
     zi(1,m)=-1.
 else if((i.eq.42).and.(((j.ge.52).and.(j.le.54)).or.((j.ge.70).and.
&(j.le.74)).or.((j.ge.78).and.(j.le.172)))) then
     go to 25
 else if(i.eq.42) then
     zi(1,m)=-1.
 else if((i.eq.44).and.(((j.ge.52).and.(j.le.56)).or.((j.ge.58).and.
&(j.le.172)))) then
     go to 25
 else if(i.eq.44) then
     zi(1,m)=-1.
 else if((i.eq.46).and.(j.ge.52).and.(j.le.174)) then
     go to 25
 else if(i.eq.46) then
     zi(1,m)=-1.
 else if((i.eq.48).and.(j.ge.54).and.(j.le.174)) then
     go to 25
 else if(i.eq.48) then
     zi(1,m)=-1.
 else if((i.eq.50).and.(((j.ge.58).and.(j.le.174)).or.((j.ge.180).and.
&(j.le.182)))) then
     go to 25
 else if(i.eq.50) then
     zi(1,m)=-1.
 else if((i.eq.52).and.(j.ge.58).and.(j.le.182)) then
     go to 25
 else if(i.eq.52) then
     zi(1,m)=-1.
 else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
     go to 25
 else if(i.eq.54) then
     zi(1,m)=-1.
 else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
     go to 25
 else if(i.eq.56) then
     zi(1,m)=-1.
 else if((i.eq.58).and.(((j.ge.36).and.(j.le.38)).or.((j.ge.52).and.
&(j.le.196)))) then
     go to 25
 else if(i.eq.58) then
     zi(1,m)=-1.
 else if((i.eq.60).and.(j.ge.34).and.(j.le.198)) then
     go to 25
 else if(i.eq.60) then
     zi(1,m)=-1.
 else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
     go to 25
 else if(i.eq.62) then
     zi(1,m)=-1.
 else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
&(j.le.221)))) then
     go to 25
 else if(i.eq.64) then
     zi(1,m)=-1.
 else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
     go to 25
 else if(i.eq.66) then
     zi(1,m)=-1.
 else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
     go to 25
```

```
else if(i.eq.68) then
    zi(l,m)=-1.
else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
    go to 25
else if(i.eq.70) then
    zi(l,m)=-1.
else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
    go to 25
else if(i.eq.72) then
    zi(l,m)=-1.
else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
    go to 25
else if(i.eq.74) then
    zi(l,m)=-1.
else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
    go to 25
else if(i.eq.76) then
    zi(l,m)=-1.
else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
    go to 25
else if(i.eq.78) then
    zi(l,m)=-1.
else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
    go to 25
else if(i.eq.80) then
    zi(l,m)=-1.
else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
    go to 25
else if(i.eq.82) then
    zi(l,m)=-1.
else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
    go to 25
else if(i.eq.84) then
    zi(l,m)=-1.
else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
    go to 25
else if(i.eq.86) then
    zi(l,m)=-1.
else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
    go to 25
else if(i.eq.88) then
    zi(l,m)=-1.
else if((i.eq.90).and.(j.ge.17).and.(j.le.238)) then
    go to 25
else if(i.eq.90) then
    zi(l,m)=-1.
else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
    go to 25
else if(i.eq.92) then
    zi(l,m)=-1.
else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
    go to 25
else if(i.eq.94) then
    zi(l,m)=-1.
else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
    go to 25
else if(i.eq.96) then
    zi(l,m)=-1.
else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
    go to 25
```

```
 else if(i.eq.98) then
     zi(1,m)=-1.
 else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
     go to 25
 else if(i.eq.100) then
     zi(1,m)=-1.
 else if((i.eq.102).and.(j.ge.44).and.(j.le.239)) then
     go to 25
 else if(i.eq.102) then
     zi(1,m)=-1.
 else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
     go to 25
 else if(i.eq.104) then
     zi(1,m)=-1.
 else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
     go to 25
 else if(i.eq.106) then
     zi(1,m)=-1.
 else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
     go to 25
 else if(i.eq.108) then
     zi(1,m)=-1.
 else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
     go to 25
 else if(i.eq.110) then
      zi(1,m)=-1.
 else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
     go to 25
 else if(i.eq.112) then
      zi(1,m)=-1.
 else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
&.and.(j.le.256)))) then
     go to 25
 else if(i.eq.114) then
      zi(1,m)=-1.
 else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
&.and.(j.le.257)))) then
     go to 25
 else if(i.eq.116) then
      zi(1,m)=-1.
 else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
&.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
     go to 25
 else if(i.eq.118) then
      zi(1,m)=-1.
 else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
&.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
     go to 25
 else if(i.eq.120) then
      zi(1,m)=-1.
 else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.273)
&.and.(j.le.283)))) then
     go to 25
 else if(i.eq.122) then
      zi(1,m)=-1.
 else if((i.eq.124).and.(((j.ge.243).and.(j.le.258)).or.((j.ge.272)
&.and.(j.le.281)))) then
     go to 25
 else if(i.eq.124) then
      zi(1,m)=-1.
```

```
      else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
     &.and.(j.le.279)))) then
          go to 25
      else if(i.eq.126) then
           zi(1,m)=-1.
      else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
          go to 25
      else if(i.eq.128) then
           zi(1,m)=-1.
      else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
          go to 25
      else if(i.eq.130) then
           zi(1,m)=-1.
      else if(i.eq.132) then
           zi(1,m)=-1.
          go to 20
      else if(i.eq.134) then
          zi(1,m)=-1.
          go to 20
      else if(i.eq.136) then
          zi(1,m)=-1.
          go to 20
      else if(i.eq.138) then
          zi(1,m)=-1.
          go to 20
      else if(i.eq.140) then
          zi(1,m)=-1.
          go to 20
      end if
c
25    continue
20    continue
10    continue
c
c     create basin outline boundary
c
      do 80 i=1,151
      do 90 j=1,151
      if(j.le.39) then
          zj(i,j)=-1.0
      else if(j.ge.110) then
          zj(i,j)=-1.0
      else if(i.gt.147) then
          zj(i,j)=-1.0
      else
          l=j-39
          zj(i,j)=zi(i,l)
      end if
90    continue
80    continue
c
c
c     ****************************************************************************
c
c     This portion of the program will blank out the area outside of the
c     watershed basin boundary
c
c     ***************************************************************************
c
c     this portion of the program overlays the prior blocked out matrix
```

```
c     for the purpose of drawing a single countour line which represents
c     the basin boundary
c
      do 110 i=1,151
      do 120 j=1,151
      if(zj(i,j).eq.-1.0) then
          zj(i,j)=0.1
      else
          zj(i,j)=-0.99
      end if
120   continue
110   continue
c
c     this portion of the program is a special filter and is only for
c     for plotting purposes
c
      do 420 i=1,151
      do 430 j=1,151
      if(depth1(i,j).lt.0.01) then
          depth1(i,j)=-0.1
      end if
430   continue
420   continue
c
c     Create rectangular boundary (14 km. by 29.4 km.)
c
      do 130 i=1,294
          r10=i
          r11=r10/10.0
          xrecbot(i)=r11
          xrectop(i)=r11
          yrecbot(i)=8.0
          yrectop(i)=22.0
130   continue
      do 140 j=80,219
          i=j-79
          r10=j
          r11=r10/10
          xreclft(i)=0.0
          xrecrht(i)=29.4
          yreclft(i)=r11
          yrecrht(i)=r11
140   continue
c
c     this section of the program starts the disspla plotting routines
c     for the correlation function
c
      call comprs
      call blowup(0.6063)
      call page(14.0196,18.1428)
c     entering the disspla routine for the contour plot
c
      print,'enter the countour plot plotting routine'
c
      call physor(2.9869,5.9869)
      call area2d(10.0,10.0)
      call basalf('standard')
      call xname('Kilometers$',100)
      call yname('Kilometers$',100)
c
```

```
      if (model.eq.1) then
      call headin('Model 1 Simulation$',100,1.5,4)
      call headin('Walnut Gulch, Arizona$',100,1.5,4)
      call headin('Ac=154.21 sq.km.$',100,1.0,4)
      call headin('Contour Interval: 2 mm.$',100,1.0,4)
      else if (model.eq.2) then
      call headin('Model 2 Simulation$',100,1.5,4)
      call headin('Walnut Gulch, Arizona$',100,1.5,4)
      call headin('Ac=154.21 sq.km.$',100,1.0,4)
      call headin('Contour Interval: 2 mm.$',100,1.0,4)
      else if(model.eq.3) then
      call headin('Model 3 Simulation$',100,1.5,4)
      call headin('Walnut Gulch, Arizona$',100,1.5,4)
      call headin('Ac=154.21 sq.km.$',100,1.0,4)
      call headin('Contour Interval: 2 mm.$',100,1.0,4)
      end if
c
      call intaxs
      call xintax
      call yintax
      call messag('Storm Day$',9,8.1,10.74)
      call messag(amonth(iii),4,7.9,10.44)
      call messag(',$',1,8.8,10.44)
      call intno(iiday,8.5,10.44)
      call intno(iiyear,8.9,10.44)
      call messag(' SAMPLING AREA LEGEND ',22,3.9,-1.2)
      call messag('Level III Area: (30.0 km. by 30.0 km.) ',38,3.1,-1.6)
      call messag('Level  II Area: (29.4 km. by 14.0 km.) ',38,3.1,-1.9)
      call messag('Level   I Area: (154.21 sq. km.) ',32,3.1,-2.2)
      call graf(0.,2.0,30.0,0.,2.0,30.0)
      call thkfrm(0.01)
      call frame
c
      call reset('dot')
      call chndot
      call thkcrv(2)
      call curve(xrecbot,yrecbot,294,0)
      call chndot
      call thkcrv(2)
      call curve(xrectop,yrectop,294,0)
      call chndot
      call thkcrv(2)
      call curve(xreclft,yreclft,140,0)
      call chndot
      call thkcrv(2)
      call curve(xrecrht,yrecrht,140,0)
      call chndot
      call thkcrv(2)
c
      call bcomon(40000)
      call conthn(0.04)
c
c     plot the contours
c
      call conmak(depth1,151,151,2.)
      call conlin(0,'solid','label',1,7)
      call conlin(1,'dash','labels',1,4)
      call conang(90.)
      call raspln(0.25)
      print,'entering routine first call contur(2,labels,draw) '
```

```
      call contur(2,'labels','draw')
c
c     scribe the outline of the basin boundary as a separate contour line
c
      call bcomon(40000)
      call reset('conang')
      call reset('conthn')
      call conmak(zj,151,151,2.)
      call conlin(0,'dot','no labels',5,10)
      call raspln(0.25)
      print,'entering routine second call contur(1,no labels,draw) '
      print,' *** the code will fail here, simply enter "start"'
      print,' but without the quotation marks'
      call contur(1,'nolabels','draw')
      call endpl(1)
      call donepl
c
      end
```

```
c                       SIM_STORMDAY.FORTRAN
c
c       this program will read in the simulated field from file29
c       and the level I sampling results for the correlation and
c       variance function from file30
c       written by SIMCRGAM.FORTRAN and plot the correlation and the variance
c       function all on one page
c
c      This program was developed by Neil M. Fennessey at M.I.T. during
c      the course of his Master's Degree research about the Areal Distribution
c      of Rainfall
c
c       .............................................................
      dimension   iwat(93),igage(93),iyear(93),imonth(93)
      dimension   iday(93),zd(93),xd(93),yd(93),a(31)
      dimension   xi(147),yi(70),wk(558),zi(147,70)
      dimension   iwk(13173),amonth(6),zj(147,70)
      dimension   xarea(31),ygamma(31),area(0:30),gamma(0:30)
      dimension   eledpth(5000),aveeldph(0:30),vareldph(0:30)
      dimension   corr1(31),lag1(31),xzero(31),yzero(31),alpha(100)
      dimension   gammod1(31),gammod2(31),corrmod1(31),corrmod2(31)
      dimension   l1(1),xkey1(26),ykey1(26),xkey2(26),ykey2(26)
      dimension   xkey3(26),ykey3(26),laglev1(31),laglev2(31),laglev3(31)
      dimension   corrlev1(31),corrlev2(31),corrlev3(31),arealev1(31)
      dimension   arealev2(31),arealev3(31),gamlev1(31),gamlev2(31)
      dimension   gamlev3(31)
c
      common work(40000)
      common /dep1/depth1(151,151)
      real lag1,mmbsk0,mmbsk1,lag,l1,laglev1,laglev2,laglev3,lambda
c
      data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
      data pi/3.14159265389793/
          api=pi
c
      print,'ready to go, enter 1, to try again, enter a 2'
      input,f1
c
      if(f1.eq.1) then
          go to 3
      else
          go to 4
      end if
c
3     continue
c
c     empty out the common blocks
c
      do 652 i=1,151
      do 654 j=1,151
          depth1(i,j)=0.0
654   continue
652   continue
c
c     ************************************************************************
c
c     read in the correlation and variance function results for the three
c     levels of sampling from the simulation model from file30
c
c     ************************************************************************
```

```
c
      read(30,1050)iyear(1),imonth(1),iday(1)
      read(30,1060)model,parama,lamda,ealpha
1060  format(3x,'model number ',i2,' param a1 ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
1050  format(3x,i3,'/',i3,'/',i3)
      read(30,1000)avedepth,vardepth,skwdpth
      read(30,1010)avdprec,vardprec,skwdprec
      read(30,1020)avdpsq,vardpsq,skwdpsq
1000  format(3x,'boundary filtered avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
1010  format(3x,'rectang. simulat. avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
1020  format(3x,'square simulat. avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
c
      do 37 i=1,31
      read(30,1035)laglev1(i),corrlev1(i),laglev2(i),corrlev2(i),laglev3(i)
     &,corrlev3(i)
          lag1(i)=laglev1(i)
          corr1(i)=corrlev1(i)
          xzero(i)=lag1(i)
          yzero(i)=0.0
37    continue
1035  format(3x,3(2x,f5.2,f6.3))
      do 30 i=1,31
      read(30,1047)arealev1(i),gamlev1(i),arealev2(i),gamlev2(i),arealev3(i),
     &gamlev3(i)
          xarea(i)=arealev1(i)
          ygamma(i)=gamlev1(i)
30    continue
1047  format(3x,3(2x,f5.2,f7.3))
c
c     ************************************************************************
c
c
c     read the results from file29 (square simulated field (whole field)
c
c
c     ************************************************************************
c
      read(29,1280)model
1280  format(3x,'model number ',i2,' param a1 ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
      read(29,1290)iyear(1),imonth(1),iday(1)
          rewind(30)
      read(30,1280)im,am,ramda,ealpha
      do 667 i=1,151
      do 677 j=1,151
c      read(29,1300)depth1(i,j)
      read(29,1300)i,j,depth1(i,j)
677   continue
667   continue
1290  format(3x,i3,'/',i3,'/',i3)
1300  format(2x,i4,i4,f8.3)
c         1300  format(f8.3)
c
          iiday=iday(1)
          iiyear=1900+iyear(1)
          iii=imonth(1)-5
```

```
      print,'...........................  '
      print1005,amonth(iii),iiday,iiyear
      print,'...........................  '
1005  format('storm day ',a4,i3,i5)
      print,' '
c
          a5=0.0
          d5=0.0
          a4=0.0
c
c     retain the appropriate values for inclusion within the basin outline
c     from the simnulataion
c
      do 650 i=1,147
      do 660 j=40,109
          l=j-39
          zi(i,l)=depth1(i,j)
660   continue
650   continue
c
c     this portion generates the x and y values for the grid mesh for
c     which the program will interpolate
c
      nxi=147
      nyi=70
      izi=147
      nxi1=146
      nyi1=69
      nxi2=nxi*2
      nyi2=nyi*2
      do 60 j=1,nxi
          xi(j)=j
60    continue
      do 70 i=1,nyi
          yi(i)=i
70    continue
c
c     invoke the subroutine call to generate the interpolated three
c     dimensional surface from the gage data onto the rectangular
c     grid mesh
c
      call iqhscv(xd,yd,zd,nd,xi,nxi,yi,nyi,zi,izi,iwk,wk,ier)
c
c     this section of the program overlays the outside of the walnut
c     gulch basin with depth values of -1. mm. The values generated by
c     the subroutine are left intact for the integration routine and
c     also the contour plotting.
c
c     nxi refers to the number of columns in the matrix
c     nyi refers to the number of rows in the matrix
c
      do 10 i=2,nyi2,2
      do 20 j=2,nxi2,2
          l=j/2
          m=i/2
c
c      this is the new logic for the proper usda scale map
c
       if((i.eq.2)) then
          zi(l,m)=-1.
```

```
        go to 20
     else if((i.eq.4)) then
        zi(1,m)=-1.
        go to 20
     else if((i.eq.6)) then
        zi(1,m)=-1.
        go to 20
     else if((i.eq.8)) then
        zi(1,m)=-1.
        go to 20
     else if((i.eq.10).and.(((j.ge.126).and.(j.le.128)).or.((j.ge.148).and.
    &(j.le.150)))) then
        go to 25
     else if(i.eq.10) then
        zi(1,m)=-1.
     else if((i.eq.12).and.(((j.ge.88).and.(j.le.90)).or.((j.ge.126)
    &.and.(j.le.132)).or.((j.ge.146).and.(j.le.150)))) then
        go to 25
     else if(i.eq.12) then
        zi(1,m)=-1.
     else if((i.eq.14).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126)
    &.and.(j.le.138)).or.((j.ge.140).and.(j.le.152)))) then
        go to 25
     else if(i.eq.14) then
        zi(1,m)=-1.
     else if((i.eq.16).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.116).and.
    &(j.le.160)))) then
        go to 25
     else if(i.eq.16) then
        zi(1,m)=-1.
     else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
    &(j.le.162)))) then
        go to 25
     else if(i.eq.18) then
        zi(1,m)=-1.
     else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.116).and.
    &(j.le.164)))) then
        go to 25
     else if(i.eq.20) then
        zi(1,m)=-1.
     else if((i.eq.22).and.(((j.ge.80).and.(j.le.82)).or.((j.ge.84).and.
    &(j.le.102)).or.((j.ge.118).and.(j.le.166)))) then
        go to 25
     else if(i.eq.22) then
        zi(1,m)=-1.
     else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
    &(j.le.166)))) then
        go to 25
     else if(i.eq.24) then
        zi(1,m)=-1.
     else if((i.eq.26).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
    &(j.le.166)))) then
        go to 25
     else if(i.eq.26) then
        zi(1,m)=-1.
     else if((i.eq.28).and.(((j.ge.84).and.(j.le.106)).or.((j.ge.118).and.
    &(j.le.170)))) then
        go to 25
     else if(i.eq.28) then
        zi(1,m)=-1.
```

```
 else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.110).and.
&(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
      go to 25
 else if(i.eq.30) then
      zi(1,m)=-1.
      go to 25
 else if((i.eq.32).and.(j.ge.82).and.(j.le.170)) then
      go to 25
 else if(i.eq.32) then
      zi(1,m)=-1.
 else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
      go to 25
 else if(i.eq.34) then
      zi(1,m)=-1.
      go to 25
 else if((i.eq.36).and.(j.ge.82).and.(j.le.174)) then
      go to 25
 else if(i.eq.36) then
      zi(1,m)=-1.
 else if((i.eq.38).and.(j.ge.84).and.(j.le.174)) then
      go to 25
 else if(i.eq.38) then
      zi(1,m)=-1.
 else if((i.eq.40).and.(j.ge.82).and.(j.le.172)) then
      go to 25
 else if(i.eq.40) then
      zi(1,m)=-1.
 else if((i.eq.42).and.(((j.ge.52).and.(j.le.54)).or.((j.ge.70).and.
&(j.le.74)).or.((j.ge.78).and.(j.le.172)))) then
      go to 25
 else if(i.eq.42) then
      zi(1,m)=-1.
 else if((i.eq.44).and.(((j.ge.52).and.(j.le.56)).or.((j.ge.58).and.
&(j.le.172)))) then
      go to 25
 else if(i.eq.44) then
      zi(1,m)=-1.
 else if((i.eq.46).and.(j.ge.52).and.(j.le.174)) then
      go to 25
 else if(i.eq.46) then
      zi(1,m)=-1.
 else if((i.eq.48).and.(j.ge.54).and.(j.le.174)) then
      go to 25
 else if(i.eq.48) then
      zi(1,m)=-1.
 else if((i.eq.50).and.(((j.ge.58).and.(j.le.174)).or.((j.ge.180).and.
&(j.le.182)))) then
      go to 25
 else if(i.eq.50) then
      zi(1,m)=-1.
 else if((i.eq.52).and.(j.ge.58).and.(j.le.182)) then
      go to 25
 else if(i.eq.52) then
      zi(1,m)=-1.
 else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
      go to 25
 else if(i.eq.54) then
      zi(1,m)=-1.
 else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
      go to 25
```

```
 else if(i.eq.56) then
      zi(l,m)=-1.
 else if((i.eq.58).and.(((j.ge.36).and.(j.le.38)).or.((j.ge.52).and.
&(j.le.196)))) then
      go to 25
 else if(i.eq.58) then
      zi(l,m)=-1.
 else if((i.eq.60).and.(j.ge.34).and.(j.le.198)) then
      go to 25
 else if(i.eq.60) then
      zi(l,m)=-1.
 else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
      go to 25
 else if(i.eq.62) then
      zi(l,m)=-1.
 else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
&(j.le.221)))) then
      go to 25
 else if(i.eq.64) then
      zi(l,m)=-1.
 else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
      go to 25
 else if(i.eq.66) then
      zi(l,m)=-1.
 else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
      go to 25
 else if(i.eq.68) then
      zi(l,m)=-1.
 else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
      go to 25
 else if(i.eq.70) then
      zi(l,m)=-1.
 else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
      go to 25
 else if(i.eq.72) then
      zi(l,m)=-1.
 else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
      go to 25
 else if(i.eq.74) then
      zi(l,m)=-1.
 else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
      go to 25
 else if(i.eq.76) then
      zi(l,m)=-1.
 else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
      go to 25
 else if(i.eq.78) then
      zi(l,m)=-1.
 else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
      go to 25
 else if(i.eq.80) then
      zi(l,m)=-1.
 else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
      go to 25
 else if(i.eq.82) then
      zi(l,m)=-1.
 else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
      go to 25
 else if(i.eq.84) then
      zi(l,m)=-1.
```

```
 else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
     go to 25
 else if(i.eq.86) then
     zi(l,m)=-1.
 else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
     go to 25
 else if(i.eq.88) then
     zi(l,m)=-1.
 else if((i.eq.90).and.(j.ge.17).and.(j.le.238)) then
     go to 25
 else if(i.eq.90) then
     zi(l,m)=-1.
 else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
     go to 25
 else if(i.eq.92) then
     zi(l,m)=-1.
 else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
     go to 25
 else if(i.eq.94) then
     zi(l,m)=-1.
 else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
     go to 25
 else if(i.eq.96) then
     zi(l,m)=-1.
 else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
     go to 25
 else if(i.eq.98) then
     zi(l,m)=-1.
 else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
     go to 25
 else if(i.eq.100) then
     zi(l,m)=-1.
 else if((i.eq.102).and.(j.ge.44).and.(j.le.239)) then
     go to 25
 else if(i.eq.102) then
     zi(l,m)=-1.
 else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
     go to 25
 else if(i.eq.104) then
     zi(l,m)=-1.
 else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
     go to 25
 else if(i.eq.106) then
     zi(l,m)=-1.
 else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
     go to 25
 else if(i.eq.108) then
     zi(l,m)=-1.
 else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
     go to 25
 else if(i.eq.110) then
      zi(l,m)=-1.
 else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
     go to 25
 else if(i.eq.112) then
      zi(l,m)=-1.
 else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
&.and.(j.le.256)))) then
     go to 25
 else if(i.eq.114) then
```

```
          zi(l,m)=-1.
       else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
     &.and.(j.le.257)))) then
          go to 25
       else if(i.eq.116) then
          zi(l,m)=-1.
       else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
     &.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
          go to 25
       else if(i.eq.118) then
          zi(l,m)=-1.
       else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
     &.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
          go to 25
       else if(i.eq.120) then
          zi(l,m)=-1.
       else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.273)
     &.and.(j.le.283)))) then
          go to 25
       else if(i.eq.122) then
          zi(l,m)=-1.
       else if((i.eq.124).and.(((j.ge.243).and.(j.le.258)).or.((j.ge.272)
     &.and.(j.le.281)))) then
          go to 25
       else if(i.eq.124) then
          zi(l,m)=-1.
       else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
     &.and.(j.le.279)))) then
          go to 25
       else if(i.eq.126) then
          zi(l,m)=-1.
       else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
          go to 25
       else if(i.eq.128) then
          zi(l,m)=-1.
       else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
          go to 25
       else if(i.eq.130) then
          zi(l,m)=-1.
       else if(i.eq.132) then
          zi(l,m)=-1.
          go to 20
       else if(i.eq.134) then
          zi(l,m)=-1.
          go to 20
       else if(i.eq.136) then
          zi(l,m)=-1.
          go to 20
       else if(i.eq.138) then
          zi(l,m)=-1.
          go to 20
       else if(i.eq.140) then
          zi(l,m)=-1.
          go to 20
       end if
c
25     continue
20     continue
10     continue
c
```

```fortran
c
c     this portion of the program looks for undulations which may have
c     been created by the surface fitting subroutine, anything less than
c     0.01 mm. equals zero
c
      do 80 i=1,nxi
      do 90 j=1,nyi
      if(zi(i,j).eq.-1.) then
          go to 90
      else if(zi(i,j).le.0.01) then
          zi(i,j)=0.0
      end if
90    continue
80    continue
c
c     calculate the average point depth for the field within the basin limits
c
      do 110 i=1,nxi
      do 120 j=1,nyi
c
      if(zi(i,j).eq.-1.) then
          go to 120
      end if
          a1=zi(i,j)+a1
          d1=d1+1
120   continue
110   continue
          avedepth=a1/d1
c
c
c     this portion will calculate the variance of the point depth within the
c     basin limits and the coefficient of skewness
c
          l=0
c
      do 130 i=1,nxi
      do 140 j=1,nyi
c
      if(zi(i,j).eq.-1.) then
          go to 140
      end if
          l=l+1
          a2=((zi(i,j)-avedepth)**2.)+a2
          a3=((zi(i,j)-avedepth)**3.)+a3
140   continue
130   continue
          vardepth=a2/(l-1)
          stdevdph=sqrt(vardepth)
          skwdpth=a3/((l-1)*(stdevdph**3.0))
      print 1030,avedepth
      print 1040,vardepth
      print 1045,skwdpth
1030  format('the average point depth=',f10.4,' mm.')
1040  format('the variance of the point depth=',f10.4,' mm ')
1045  format('the coefficeint of skewness of the point depth=',f10.4)
c
c     reset variables equal to zero
c
          a1=0.0
          a2=0.0
```

```
          a3=0.0
c
c      ***************************************************************************
c
c      This portion of the program will blank out the area outside of the
c      watershed basin boundary
c
c      ***************************************************************************
c
c      this portion of the program overlays the prior blocked out matrix
c      for the purpose of drawing a single countour line which represents
c      the basin boundary
c
c      ***************************************************************************
c
       do 390 i=1,nxi
       do 410 j=1,nyi
       if(zi(i,j).eq.-1.0) then
          zj(i,j)=0.1
       else
          zj(i,j)=-0.99
       end if
410    continue
390    continue
c
c
c      this portion of the program looks for undulations which may have
c      been created by the surface fitting subroutine, this second filter
c      is only for plotting purposes
c
       do 420 i=1,nxi
       do 430 j=1,nyi
       if((zi(i,j).lt.0.01).and.(zi(i,j).gt.-1.0)) then
          zi(i,j)=-0.1
       else if((zi(i,j).eq.-1.)) then
          zi(i,j)=0.0
       end if
430    continue
420    continue
c
c      this portion of the code draws straight line segments to be
c      plotted on the legend (coutour lines and basin boundary)
c
       do 490 j=208,233
          i=j-207
          r10=j
          r11=r10/10.0
          xkey1(i)=r11
          ykey1(i)=3.4
          xkey2(i)=r11
          ykey2(i)=2.8
          xkey3(i)=r11
          ykey3(i)=1.8
490    continue
c
c      this section of the program starts the disspla plotting routines
c      for the correlation function
c
       print,'entering the plotting routine for the correlation function'
       call comprs
```

```
      call blowup(0.6063)
      call page(14.0196,18.1428)
      call physor(2.98688,3.1338)
      call basalf('stand')
      call mx2alf('greek',1h*)
      call yaxang(0.0)
      call xname('Spatial Lag *n)  (kilometers)$',100)
      call yname('Correlation Coefficient: *r(n))$',100)
      call area2d(4.286,4.286)
      call headin('Spatial Correlation$',100,1.5,1)
      call thkfrm(0.01)
      call frame
      call graf(0.,1.,6.,-1.,0.2,1.0)
      call poly3
      call curve(lag1,corr1,31,0)
      call thkcrv(1)
      call curve(xzero,yzero,31,0)
      call endgr(1)
c
c     generate the results for plotting of the variance function
c
c
c     this section of the program starts the disspla plotting routines
c     for the variance function
c
      print,'entering the variance function plotting routine'
      call physor(8.7009,3.1338)
      call xname('Area  (square kilometers)$',100)
      call yname('Variance Function: *g)(Area)$',100)
      call area2d(4.286,4.286)
      call headin('Variance Function$',100,1.5,1)
      call thkfrm(0.01)
      call frame
      call graf(0.,4.,36.,0.,0.1,1.0)
      call poly3
      call curve(xarea,ygamma,31,0)
      call endgr(2)
c
c     entering the disspla routine for the contour plot
c
      print,'enter the countour plot plotting routine'
c
      call physor(2.9869,9.8136)
      call area2d(10.0,4.762)
      call reset('yaxang')
      call xname('Kilometers$',100)
      call yname('Kilometers$',100)
c
      if (model.eq.1) then
      call headin('Model 1 Simulation$',100,1.5,3)
      call headin('Walnut Gulch, Arizona$',100,1.5,3)
      call headin('Ac=154.21 sq.km.$',100,1.0,3)
      else if(model.eq.2) then
      call headin('Model 2 Simulation$',100,1.5,3)
      call headin('Walnut Gulch, Arizona$',100,1.5,3)
      call headin('Ac=154.21 sq.km.$',100,1.0,3)
      else if(model.eq.3) then
      call headin('Model 3 Simulation$',100,1.5,3)
      call headin('Walnut Gulch, Arizona$',100,1.5,3)
      call headin('Ac=154.21 sq.km.$',100,1.0,3)
```

```
      end if
c
      call intaxs
      call xintax
      call yintax
      call messag('Contour Interval: 2 mm.$',100,0.3,4.25)
      call messag('Storm Day$',9,8.1,5.5)
      call messag(amonth(iii),4,7.9,5.2)
      call messag(',$',1,8.8,5.2)
      call intno(iiday,8.5,5.2)
      call intno(iiyear,8.9,5.2)
      call messag('Legend',6,7.9,1.5)
      call messag('Isohyet Line',12,8.0,1.1)
      call messag('Catchment',9,8.0,0.7)
      call messag('Boundary',8,8.0,0.5)
      call graf(0.,2.0,30.0,0.,1.0,14.0)
      call thkfrm(0.01)
      call frame
      call curve(xkey1,ykey1,26,0)
      call dash
      call curve(xkey2,ykey2,26,0)
      call reset('dash')
      call dot
      call thkcrv(5)
      call curve(xkey3,ykey3,26,0)
      call reset('dot')
      call bcomon(40000)
      call conthn(0.04)
c
c     plot the contours
c
      call conmak(zi,nxi,nyi,2.)
      call conlin(0,'solid','label',1,7)
      call conlin(1,'dash','labels',1,4)
      call conang(90.)
      call raspln(0.25)
      print,'entering routine     call contur(2,labels,draw) '
      call contur(2,'labels','draw')
c
c     scribe the outline of the basin boundary as a separate contour line
c
      call bcomon(40000)
      call reset('conang')
      call reset('conthn')
      call conmak(zj,nxi,nyi,2.)
      call conlin(0,'dot','no labels',5,10)
      call raspln(0.25)
      print,'entering routine second call contur(1,no labels,draw) '
      print,' *** the code will fail here, simply enter "start"'
      print,' but without the quotation marks'
      call contur(1,'nolabels','draw')
      call endpl(1)
      call donepl
c
          go to 35
c
4     continue
      print,'the run has been aborted'
35     continue
      end
```

```
c                              WETPLOT.FORTRAN
c
c     this program will read in the observed wetted area curves from file 02
c     and the simulation wetted area curves in from file62.  The theoretical
c     curves will be created here
c
c     This program was developed by Neil M. Fennessey at M.I.T. during
c     the course of his Master's Degree research about the Areal Distribution
c     of Rainfall
c
c      .................................................................
c
      dimension iyear(1),imonth(1),iday(1),totcnt1(-1:200),totcnt2(-1:200)
      dimension totcnt3(-1:200),deplev1(200),deplev2(200),deplev3(200)
      dimension amonth(6),wetlev1(200),wetlev2(200),wetlev3(200)
      dimension   xarea(100),ygamma(100),lag1(100),corr1(100)
      dimension   wetfrct(-1:200),wetobs(200),depobs(200),bigmax(10)
      dimension   a1(6),wet(5)
      dimension   depmod(200),drymod(3),delta(3),theta(3),wetmod(200)
      dimension   xthry(200),ythry(200),xobs(200),yobs(200),xlev1(200)
      dimension   ylev1(200),xlev2(200),ylev2(200),xlev3(200),ylev3(200)
c
c
      external f
      common /pass2/ealpha,parama,ramda
      common/pass1/flash
c
c
      data pi/3.14159265389793/
      api=pi
      data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
c
c     ****************************************************************************
c
c     read file30 for the model number
c
c     ****************************************************************************
c
c
      read(30,1290)iyear(1),imonth(1),iday(1)
      read(30,1280)im,parama,ramda,ealpha
          model=im
1280  format(3x,'model number ',i2,' param a1 ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
1290  format(3x,i3,'/',i3,'/',i3)
c
c
c     ****************************************************************************
c
c     Read in the wetted area curves from file 62 for the three levels of
c     sampling from the simulation  Level I: boundary area,
c     Level II: rectangular, Level III: square (entire )area
c
c     ****************************************************************************
c
c     read in the wetted area curve data from the level I sampling (basin
c     boundary)
c
      read(62,1130)kk1
          amax1=kk1-2
```

```
      read(62,1140)iyear(1),imonth(1),iday(1)
      do 190 i=-1,amax1
      if(i.eq.-1) then
      read(62,1090)dry1
          go to 190
      else if(i.eq.0) then
      read(62,1110)totcnt1(0)
      else
      read(62,1120)i,totcnt1(i)
      end if
          deplev1(i+1)=i
          wetlev1(i+1)=totcnt1(i)
190   continue
c
c     read in the wetted area data for Level II (rectangular area)
c
      read(62,1130)kk2
          amax2=kk2-2
      read(62,1140)iyear(1),imonth(1),iday(1)
      do 192 i=-1,amax2
      if(i.eq.-1) then
      read(62,1090)dry2
          go to 192
      else if(i.eq.0) then
      read(62,1110)totcnt2(0)
      else
      read(62,1120)i,totcnt2(i)
      end if
          deplev2(i+1)=i
          wetlev2(i+1)=totcnt2(i)
192   continue
c
c     read in the data for Level III sampling (square area)
c
      read(62,1130)kk3
          amax3=kk3-2
      read(62,1140)iyear(1),imonth(1),iday(1)
      do 194 i=-1,amax3
      if(i.eq.-1) then
      read(62,1090)dry3
          go to 194
      else if(i.eq.0) then
      read(62,1110)totcnt3(0)
      else
      read(62,1120)i,totcnt3(i)
      end if
          deplev3(i+1)=i
          wetlev3(i+1)=totcnt3(i)
194   continue
c
c     file 62 format statements
c
1090  format('dry fract. total area (0 mm. rainfall)=',f9.3)
1120  format('fract. of area wetted by at least ',i3,' mm. or more was ',f9.3)
1110  format('fract. of area wetted by more than 0 mm. was',f9.3)
1130  format('there are',i3,' lines in this file')
1140  format(3(i2))
c
          rewind(62)
c
```

```
c     ************************************************************************
c
c     read in the observed storm day observations from file02
c
c     ************************************************************************
c
      read(02,1370)n,iiyear
      read(02,1310)amnth,iiday,iiyear
      read(02,1320)avedepth,vardepth,skwdpth
c
          expecty=avedepth
          vary=vardepth
c
      read(02,1330)dryobs
      read(02,1340)wetfrct(0)
      read(02,1300)kk,k1
      if(kk.gt.31) then
          k1=kk
          b=1
      else
          k1=31
          b=2
      end if
      read(02,1350)
      read(02,1360)
      do 50 i=1,k1
c
c     cumulative wetted fraction is greater than 31 mm.
c
      if ((b.eq.1).and.(i.le.31)) then
      read(02,1230)i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
      else if((b.eq.1).and.(i.gt.31)) then
      read(02,1240)i,wetfrct(i)
c
c     cumulative wetted fraction is less than 31 mm.
c
      else if ((b.eq.2).and.(i.le.(kk))) then
      read(02,1230)i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
      else
      read(02,1250)lag1(i),corr1(i),xarea(i),ygamma(i)
      end if
50    continue
c
c     this is a special modification only for wetplot.fortran
c
          amaxobs=kk+1
      do 60 i=0,amaxobs
          depobs(i+1)=i
          wetobs(i+1)=wetfrct(i)
60    continue
c
c     file02 format statements
c
1370  format(' there are ',i5,' storm days in this file for this year:'i4)
1360  format(7x,'Y (mm.) ',2x,'Acw/Ac (Y>y) ',10x,'v (km.) ',1x,'rho(v) ',8x,'A
     &(km.sq.) ',1x,'Gamma(A) ')
1350  format(3x,'Cumulative Wetted Fraction',5x,'Spatial Correlation',5x,
     &'Variance Function')
1340  format(' Wetted Fraction of Total Basin Area: (Acw/Ac)=',f5.3)
1330  format(' Dry Fraction of Total Basin Area: (Acd/Ac)=',f5.3)
```

```
1320  format(' point depth E(Y)=',f7.3,' Var(Y)=',f7.3,' S.C.(Y)=',f7.3 )
1310  format(' Storm Day ',a4,i3,i5)
1300  format(' there are ',i3,' wetted area curve pts, there are ',i3,
     &' total data points this day')
1250  format(37x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
1240  format(7x,i3,7x,f5.3)
1230  format(7x,i3,7x,f5.3,15x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
c
          rewind(02)
c
c     ************************************************************************
c
c     Develop the theoretical spatial distribution curves for Model I
c     Model II and Model III
c
c     ************************************************************************
c
          amonth(iii)=amnth
c
c
c     this portion of the program will determine the values of other
c     model parameters: delta and theta for P(dry) and p(Y>=yi) calculations
c
c
c
c     Model 1
c
          delta(1)=1.0/ealpha
          theta(1)=(pi*ramda)/(2*(parama**2.0))
c
c     Model 2
c
          delta(2)=2.0/ealpha
          theta(2)=(4*pi*ramda)/(parama**2.0)
c
c      For each of the three models, calculate p(Dry)
c
          i=model
          state=1.0
      if(i.eq.1) then
          a=0.0
          b=0.01*delta(1)
          c=theta(1)
          call mdgam(b,c,soln,ier)
          drymod(i)=soln
          wetmod(1)=1.0-soln
      else if(i.eq.2) then
          a=0.0
          b=0.01*delta(2)
          c=theta(2)
          call mdgam(b,c,soln,ier)
          drymod(i)=soln
          wetmod(1)=1.0-soln
      else if((model.eq.3).and.(state.eq.1.)) then
          precip=0.0
          a=precip
c
c      call external function probgt
c
          ff=probgt(precip)
```

```
c
c     check for exponential underflow
c
      if((flash.eq.1).or.(flash.eq.2)) then
      print,'flash??, depth=',i
          dry3=1.0
          go to 91
      end if
          drymod(i)=1.0-ff
          wetmod(1)=1.0-drymod(i)
      end if
91    continue
          depmod(1)=0.0
c
          drythery=drymod(i)
c
          kmod=1
c
c      calculate p(Y>=yi) for each of the three models
c
          j=model
          ff=100
      if ((j.eq.1).or.(j.eq.2)) then
      do 80 i=1,200
      if(ff.lt.0.001) then
          go to 90
      end if
          a=i
          b=i*delta(j)
          c=theta(j)
c
c     call the incomplete gamma density function from the imsl library
c
          call mdgam(b,c,soln,ier)
          ff=1-soln
          kmod=kmod+1
          wetmod(i+1)=ff
          depmod(i+1)=a
80    continue
      end if
c
c      Model 3
c
      if(j.eq.3) then
c
          model=3
c
      do 95 i=1,200
          precip=i
          a=precip
c
c      call external function probgt
c
          ff=probgt(precip)
      if(ff.lt.0.001) then
          go to 90
      end if
          wetmod(i+1)=ff
          depmod(i+1)=a
95    continue
```

```
      end if
c
c
90    continue
c
          amaxthry=kmod
c
c     determine whether theory,observation or Level I, Level II, or
c     Level III have the driest area, this will be used for the vertical
c     axis on the spatial distribution of total storm depth plot
c
          wet(1)=1.0-dry1
          wet(2)=1.0-dry2
          wet(3)=1.0-dry3
          wet(4)=1.0-dryobs
          wet(5)=1.0-drythery
c
c     this portion of the program will sift through the
c     depths and find the maximum value (truncated as an integer)
c
      do 120 i=1,4
          a1(2)=wet(i)
          a1(3)=wet(i+1)
      if(a1(2).gt.a1(3)) then
          a1(4)=a1(2)
      else if (a1(3).gt.a1(2)) then
          a1(4)=a1(3)
      end if
      if(a1(4).gt.a1(1)) then
          a1(1)=a1(4)
      end if
          mxwet=a1(1)*10.
          amxwet=mxwet/10.0
120   continue
c
      if(amxwet.ge.0.9) then
          amaxwet=1.0
      else
          amaxwet=amxwet+0.1
      end if
c
c     Search for the largest maximum depth from the group
c
          bigmax(1)=amax1
          max1=amax1
          bigmax(2)=amax2
          max2=amax2
          bigmax(3)=amax3
          max3=amax3
          bigmax(4)=amaxobs
          maxobs=amaxobs
          bigmax(5)=amaxthry
          maxmod=amaxthry
c
c     this portion of the program will sift through all the element
c     depths and find the maximum value (truncated as an integer)
c
      do 70 i=1,4
          a1(2)=bigmax(i)
          a1(3)=bigmax(i+1)
```

```
      if(a1(2).gt.a1(3)) then
          a1(4)=a1(2)
      else if (a1(3).gt.a1(2)) then
          a1(4)=a1(3)
      end if
      if(a1(4).gt.a1(1)) then
          a1(1)=a1(4)
      end if
          max=a1(1)
          amax=max+1
70    continue
c

      if(amax.le.10.) then
          aincr=1
      else if(amax.le.20) then
          aincr=2
      else if(amax.le.30) then
          aincr=3
      else if(amax.le.40) then
          aincr=4
      else if(amax.le.50) then
          aincr=5
      else if(amax.le.60) then
          aincr=6
      else if(amax.le.70) then
          aincr=7
      else if(amax.le.80) then
          aincr=8
      else if(amax.le.90) then
          aincr=9
      else if(amax.le.100) then
          aincr=10
      else if(amax.le.110) then
          aincr=11
      else if(amax.le.12) then
          aincr=12
      else if(amax.le.130) then
          aincr=13
      else if(amax.le.140) then
          aincr=14
      end if
          incr=aincr
c
c      istart=amax/2.0
      istart=(amax+aincr)/2.0
      iend=istart+incr
      do 110 j=istart,iend
          i=j-istart+1
          xthry(i)=j
          ythry(i)=0.95*amaxwet
          xobs(i)=j
          yobs(i)=0.89*amaxwet
          xlev1(i)=j
          ylev1(i)=0.83*amaxwet
          xlev2(i)=j
          ylev2(i)=0.77*amaxwet
          xlev3(i)=j
          ylev3(i)=0.71*amaxwet
110   continue
```

```
c
c     this section of the program starts the disspla plotting routines
c
      print,'entering the plotting routines'
      call comprs
      call page(11.,8.5)
      call physor (1.75,1.0)
      call xname('Wetted to a Depth Greater Than (mm.)$',100)
      call yname('Fraction of Total Area$',100)
      call grace(0.0)
      call area2d(8.0,5.0)
          xax=8.0
          yax=5.0
c
      if(model.eq.1) then
      call headin('Spatial Distribution of Total Storm Depth$',100,1.3,4)
      call headin('Model 1 Simulation$',100,1.0,4)
      call headin('Theory$',100,1.0,4)
      call headin('Observation$',100,1.0,4)
      else if(model.eq.2) then
      call headin('Spatial Distribution of Total Storm Depth$',100,1.3,4)
      call headin('Model 2 Simulation$',100,1.0,4)
      call headin('Theory$',100,1.0,4)
      call headin('Observation$',100,1.0,4)
      else if(model.eq.3) then
      call headin('Spatial Distribution of Total Storm Depth$',100,1.3,4)
      call headin('Model 3 Simulation$',100,1.0,4)
      call headin('Theory$',100,1.0,4)
      call headin('Observation$',100,1.0,4)
      end if
c
      call thkfrm(0.01)
      call frame
      call xintax
      call yaxang(0.0)
      call messag('Storm Day$',9,6.1,5.5)
      call messag(amnth,4,6.0,5.2)
      call messag(',$',1,6.9,5.2)
      call intno(iiday,6.6,5.2)
      call intno(iiyear,7.0,5.2)
      call messag('Theoretical Model$',17,5.5,4.7)
      call messag('Observation$',11,5.5,4.4)
      call messag('Level I Sampling$',16,5.5,4.1)
      call messag('Level II Sampling$',17,5.5,3.8)
      call messag('Level III Sampling$',18,5.5,3.5)
      call graf(0.,aincr,amax,0.,0.1,amaxwet)
      call poly5
c
c     theory
c
      call curve(depmod,wetmod,maxmod,0)
      call curve(xthry,ythry,incr,0)
c
c     Observation
c
      call dash
      call curve(depobs,wetobs,maxobs,0)
      call curve(xobs,yobs,incr,0)
      call reset ('dash')
c
```

```
c
c     simulation (level 3:full square)
c
      call dot
      call curve(deplev3,wetlev3,max3,0)
      call curve(xlev3,ylev3,incr,0)
      call reset('dot')
c
c     simulation (level 2:rectangular area)
c
      call chndot
      call curve(deplev2,wetlev2,max2,0)
      call curve(xlev2,ylev2,incr,0)
      call reset('chndot')
c
c     simulation (level 1:basin boundary area)
c
      call chndsh
      call curve(deplev1,wetlev1,max1,0)
      call curve(xlev1,ylev1,incr,0)
      call reset('chndsh')
      call endgr(0)
      call endpl(0)
      call donepl
          go to 35
4     continue
      print,'the run has been aborted'
35     continue
      end
c
c ************************************************************
c
c  Function probgt integrates f(y) over  .000001<y<precip and
c  calculates the probability that the value of precip is
c  exceeded.  Routine calls the function f(y) which is
c  the analytical solution of the probability density of
c  storm depth from Model 3.
c
c  dolajf -  NAG integration routine by quadrature
c
      real function probgt(precip)
      integer ier, iw(102),ifail
      real dcadre,precip,error,c,expecty,vary
      double precision f,a,precip_dp,aerr,rerr,prob,err,w(800)
      common /pass2/ealpha,parama,ramda
      external f
      precip_dp = precip
      rerr =0.0
      aerr = 1.0e-5
      ifail = 1
      lw = 800
      liw = 102
c
c dolajf -- NAG integration routine
c a       -- lower limit of integration
c f       -- probability density function for precip > 0
c
      data pi/3.14159265389793/
      api=pi
      a = .000001
```

```
      call d01ajf(f,a,precip_dp,aerr,rerr,prob,err,w,lw,iw,liw,ifail)
c      print *, 'solution' ,prob
c      print *, 'ifail ' ,ifail
      beta = 1.0/ealpha
      eta =  (2*api*ramda)/((beta**2.0)*(parama**2.0))
      probgt = 1 - exp(-eta) - prob
      return
      end
c *************************************************
c *************************************************
c
c  Function f(y) is the probability density of storm depth
c  obtained from Model 3.  Routine utilizes NAG routine
c  "s18aff" which calculates the first order modified Bessel
c  function of the first kind  - "I1".
c
      double precision function f(y)
      real eta,beta
      double precision arg,s18aff,y
      common /pass1/ flash
      common /pass2/ealpha,parama,ramda
      data pi/3.14159265389793/
      api=pi
      beta = 1.0/ealpha
      eta =  (2*api*ramda)/((beta**2.0)*(parama**2.0))
      arg =  2*sqrt(eta*beta*y)
          flash=0
      if(arg.gt.80.1) then
          flash=1
          go to 10
      end if
          c1= -eta-beta * y
      if(c1.lt.-82) then
          flash=2
          go to 10
      end if
          c= exp(c1)
      f   =  c * sqrt((eta*beta)/y) * s18aff(arg,ifail)
10    continue
      return
      end
```

```
c                        CORR_GAM_PLOT.FORTRAN
c
c       this code will plot the correlation and variance functions
c       the three levels of sampling within the field of simulation
c       plus the theoretical curves and the observed curves
c
c       the observations will be read in from file02, the curves from the
c       three levels of smpling will be read in from file30 and the
c       theoretical curves will be developed here
c       this has the model 1 and the model 2 correlation functions
c       in this file
c
c
c      This program was developed by Neil M. Fennessey at M.I.T. during
c      the course of his Master's Degree research about the Areal Distribution
c      of Rainfall
c
c       ..............................................................
c
      dimension   iyear(1),iday(1),imonth(1)  ,corrmod1(31)
      dimension   amonth(6),wetfrct(-2:200),xzero(31),yzero(31)
      dimension   xarea(31),ygamma(31),corr1(31),lag1(31),alpha(100)
      dimension   gammod(31),corrmod(31),l1(31),lagmod(31),areamod(31)
      dimension   laglev1(31),corrlev1(31),laglev2(31),corrlev2(31)
      dimension   corrlev3(31),arealev1(31),gamlev1(31),arealev2(31)
      dimension   gamlev2(31),arealev3(31),gamlev3(31),laglev3(31)
      dimension   xthry(200),ythry(200),xobs(200),yobs(200),xlev1(200)
      dimension   ylev1(200),xlev2(200),ylev2(200),xlev3(200),ylev3(200)
c
      real lag1,lag,l1,lagmod,laglev1,laglev2,laglev3
c
      double precision mmbsk0,mmbsk1,arg
c
c
      data  (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
      data pi/3.14159265389793/
          api=pi
c
      print,' choose between plotting the Spatial Correlation and the
     &Variance Function'
      print,'Spatial Correlation, enter  1'
      print,'Variance Function, enter a 2'
      input,choice
      print,'upper right plot legend: enter a 2'
      print,'lower left plot legend: enter a 1'
      input,leg
c
c      ***********************************************************************
c
c      read storm day results from file 02 for the observed correlation
c      and variance functions
c
c      ***********************************************************************
c
      read(02,1370)n,iiyear
      read(02,1310)amnth,iiday,iiyear
      read(02,1320)avedepth,vardepth,skwdpth
      read(02,1330)wetfrct(-1)
      read(02,1340)wetfrct(0)
      read(02,1300)kk,k1
```

```
      if(kk.gt.31) then
          k1=kk
          b=1
      else
          k1=31
          b=2
      end if
      read(02,1350)
      read(02,1360)
      do 10 i=1,k1
c
c     cumulative wetted fraction is greater than 31 mm.
c
      if ((b.eq.1).and.(i.le.31)) then
      read(02,1230)i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
      else if((b.eq.1).and.(i.gt.31)) then
      read(02,1240)i,wetfrct(i)
c
c     cumulative wetted fraction is less than 31 mm.
c
      else if ((b.eq.2).and.(i.le.(kk))) then
      read(02,1230)i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
      else
      read(02,1250)lag1(i),corr1(i),xarea(i),ygamma(i)
      end if
10    continue
c
c     file02 format statements
c
1370  format(' there are ',i5,' storm days in this file for this year:'i4)
1360  format(7x,'Y (mm.) ',2x,'Acw/Ac (Y>y) ',10x,'v (km.) ',1x,'rho(v) ',8x,'A
     &(km.sq.) ',1x,'Gamma(A) ')
1350  format(3x,'Cumulative Wetted Fraction',5x,'Spatial Correlation',5x,
     &'Variance Function')
1340  format(' Wetted Fraction of Total Basin Area: (Acw/Ac)=',f5.3)
1330  format(' Dry Fraction of Total Basin Area: (Acd/Ac)=',f5.3)
1320  format(' point depth E(Y)=',f7.3,' Var(Y)=',f7.3,' S.C.(Y)=',f7.3 )
1310  format(' Storm Day ',a4,i3,i5)
1300  format(' there are ',i3,' wetted area curve pts, there are ',i3,
     &' total data points this day')
1250  format(37x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
1240  format(7x,i3,7x,f5.3)
1230  format(7x,i3,7x,f5.3,15x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
c
c     ************************************************************************
c
c     read in the correlation and variance function results for the three
c     levels of sampling from the simulation model from file30
c
c     ************************************************************************
c
      read(30,1050)iyear(1),imonth(1),iday(1)
      read(30,1060)model,parama,lamda,ealpha
1060  format(3x,'model number ',i2,' param a1 ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
1050  format(3x,i3,'/',i3,'/',i3)
      read(30,1000)avedpth,vardpth,skwdph
      read(30,1010)avdprec,vardprec,skwdprec
      read(30,1020)avdpsq,vardpsq,skwdpsq
1000  format(3x,'boundary filtered avedepth=',f9.3,' vardepth=',f9.3,
```

```
     &' coef. of skew=',f9.3)
1010  format(3x,'rectang. simulat. avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
1020  format(3x,'square simulat. avedepth=',f9.3,' vardepth=',f9.3,
     &' coef. of skew=',f9.3)
c
      do 20 i=1,31
      read(30,1030)laglev1(i),corrlev1(i),laglev2(i),corrlev2(i),laglev3(i)
     &,corrlev3(i)
20    continue
1030  format(3x,3(2x,f5.2,f6.3))
      do 30 i=1,31
      read(30,1040)arealev1(i),gamlev1(i),arealev2(i),gamlev2(i),arealev3(i),
     &gamlev3(i)
      if((arealev2(i).eq.0.84).or.(arealev3(i).eq.0.84)) then
          arealev3(i)=7.84
          arealev2(i)=7.84
      end if
30    continue
1040  format(3x,3(2x,f5.2,f7.3))
c
c
      if(choice.eq.2) then
          go to 190
      end if
c
c     ****************************************************************************
c
c     this portion of the program will determine values of the correlation
c     function from the analytical solutions at different lags
c       For models 2 and 3, the IMSL external functions for
c     the modified Bessel function of the second kind zero and first order
c     are called by mmbsk0 and mmbsk1 respectively
c
c     ****************************************************************************
c
c     initialize variables
c
c
      do 50 i=0,30
          ai=i
          lagmod(i+1)=ai/5
          xzero(i+1)=lagmod(i+1)
          yzero(i+1)=0.0
c
c     determine correlation for the model of interest
c
      if(model.eq.1) then
          go to 60
      else if (model.eq.2) then
          go to 70
      else if (model.eq.3) then
          go to 80
      end if
c
c     Model 1 Theoretical Correlation Function
c
60    continue
          corrmod1(i+1)=exp(-(parama*lagmod(i+1))**2.)
          go to 50
```

```
c
c     Model 2 Theoretical Correlation Function
c
70    continue
c
          arg=parama*lagmod(i+1)
      if(i.eq.0) then
          corrmod1(1)=1.
      else
          iopt=1
          c1=mmbsk0(iopt,arg,ier)
          c2=mmbsk1(iopt,arg,ier)
          c3=0.5*(arg**2.)
          c4=arg*c2
          corrmod1(i+1)=(c3*c1)+c4
      end if
          go to 50
c
c     Model 3 Correlation
c
80    continue
          beta=1/ealpha
          arg=parama*lagmod(i+1)*beta/2.0
      if(i.eq.0) then
          corrmod1(1)=1.
      else
          iopt=1
          a2=arg+(sqrt(pi)*(arg**2.0))+(((pi/2.0)-0.25)*(arg**3.0))
          a3=((arg**2.0)/2.0)+(sqrt(pi)*(arg**3.0)/2.0)+(((pi/4.0)-(1/16.0))*
     &(arg**4.0))
          a4=exp(-sqrt(pi)*arg)
          c1=mmbsk0(iopt,arg,ier)
          c2=mmbsk1(iopt,arg,ier)
          corrmod1(i+1)=(a2*c2*a4)+(a3*c1*a4)
      end if
          go to 50
50    continue
c
190   continue
c
c     ****************************************************************************
c
c     This portion of the program will use the approximations of the variance
c     functions at each lag in order to arrive at a theoretical variance
c     function curve
c
c     ****************************************************************************
      if (model.eq.2) then
          go to 120
      end if
c
      do 90 m=0,30
          areamod(m+1)=((m*0.2)**2.0)
c
c
c
      if(model.eq.1) then
          go to 110
      else if (model.eq.3) then
          go to 130
```

```
      end if
c
c     Model 1 Theoretical Variance Function (approximation)
c
110   continue
c
          gammod(m+1)=api/(api+((parama**2.0)*areamod(m+1)))
c
          go to 90
c
c      Model 3 Theoretical Variance Function (approximation)
c
130   continue
c
          beta=1/ealpha
          eta=((parama**2.0)*(beta**2.0)*areamod(m+1))
          a1=1+(0.03392*eta)
          a2=1+(0.04266*eta)
          a3=0.03392*eta
          bb=1+(a3*a2/a1)
          gammod(m+1)=1/(sqrt(a1)*sqrt(bb))

          go to 90
c
c
90    continue
c
          go to 800
c
c     Model 2 Theoretical Variance Function
c
120   continue
c
      do 340 j=0,6
          a2=j
          areamod(j+1)=a2**2.0
c
      if(j.eq.0) then
          gammod(j+1)=1.0
      write(06,1492)areamod(j+1),gammod(j+1)
          go to 340
      end if
c
c      create the zeroth and first order modified Bessel Functions
c      of the Second kind
c
          areamod(j+1)=a2**2.0
          a4=parama*a2
          arg=a4
          iopt=1
          ck0=mmbsk0(iopt,arg,ier)
          ck1=mmbsk1(iopt,arg,ier)
c
c       first order Modified Struve Function
c
      do 310 i=0,100
          aa=(arg/2.0)**((2*i)+2)
          bb=i+1.5
          cc=i+2.5
          gam1=gamma(bb)
```

```
          gam2=gamma(cc)
          ee=alog(aa)-alog(gam1)-alog(gam2)
          gam3=exp(ee)
          soln=soln1+(gam3)
      if(soln.gt.0.0) then
          diff=abs(gam3)
      end if
          soln1=soln
      if((diff.lt.10E-8).or.(aa.gt.10E36)) then
          go to 320
      end if
310   continue
c
320   continue
c
          stru1=soln1
          soln1=0.0
          soln=0.0
c
c      Zero order Modified Struve Function
c
      do 330 i=0,100
          aa=(arg/2.0)**((2*i)+1)
          bb=i+1.5
          gam1=gamma(bb)
          ee=alog(aa)-2*alog(gam1)
          gam1=exp(ee)
          soln=soln0+gam1
          diff=abs(gam1)
          soln0=soln
      if((diff.lt.10E-8).or.(aa.gt.10E36)) then
          go to 350
      end if
330   continue
c
350   continue
c
          stru0=soln0
          soln0=0.0
c
c     Create the variance Function here
c
          part1=3*pi*((ck1*stru0)+(ck0*stru1))/2.0
          part2=(8*ck1/arg)+(4*ck0)
          part3=(8/(arg**2.0))
c         gammod(j+1)=part1+part2-part3
          gammod(j+1)=(part1+part2-part3)**2.0
c
      write(06,1492)areamod(j+1),gammod(j+1)
c
340   continue
c
1492  format('area=',f5.2,' gam=',f9.3)
c
800   continue
c
c     this section of the program starts the disspla plotting routines
c     for the correlation function
c
      if(choice.eq.1) then
```

```
      print,'entering the plotting routine for the correlation function'
      end if
      call comprs
      call page(11.,8.5)
c
c     choose the plot
c
          do 560 i=1,2
      if((choice.eq.1).and.(leg.eq.1)) then
          xthry(1)=0.15
          xthry(2)=0.8
          ythry(i)=-0.42
          xobs(1)=0.15
          xobs(i)=0.8
          yobs(i)=-0.54
          xlev1(1)=0.15
          xlev1(2)=0.8
          ylev1(i)=-0.67
          xlev2(1)=0.15
          xlev2(2)=0.8
          ylev2(i)=-0.78
          xlev3(1)=0.15
          xlev3(2)=0.8
          ylev3(i)=-0.9
      else if((choice.eq.1).and.(leg.eq.2)) then
          xthry(1)=3.3
          xthry(2)=3.95
          ythry(i)=0.9
          xobs(1)=3.3
          xobs(i)=3.95
          yobs(i)=0.79
          xlev1(1)=3.3
          xlev1(2)=3.95
          ylev1(i)=0.67
          xlev2(1)=3.3
          xlev2(2)=3.95
          ylev2(i)=0.54
          xlev3(1)=3.3
          xlev3(2)=3.95
          ylev3(i)=0.42
          end if
560   continue
c
          do 570 i=1,2
      if ((choice.eq.2).and.(leg.eq.2)) then
          xthry(1)=20.0
          xthry(2)=24.0
          ythry(i)=0.95
          xobs(1)=20
          xobs(i)=24
          yobs(i)=0.89
          xlev1(1)=20
          xlev1(2)=24
          ylev1(i)=0.83
          xlev2(1)=20
          xlev2(2)=24
          ylev2(i)=0.77
          xlev3(1)=20
          xlev3(2)=24
          ylev3(i)=0.71
```

```
      else if ((choice.eq.2).and.(leg.eq.1)) then
          xthry(1)=1.5
          xthry(2)=5.5
          ythry(i)=0.29
          xobs(1)=1.5
          xobs(i)=5.5
          yobs(i)=0.23
          xlev1(1)=1.5
          xlev1(2)=5.5
          ylev1(i)=0.17
          xlev2(1)=1.5
          xlev2(2)=5.5
          ylev2(i)=0.11
          xlev3(1)=1.5
          xlev3(2)=5.5
          ylev3(i)=0.05
          end if
570   continue
c
      print,'entering the disspla plotting routines'
c
      call comprs
      call page(11.0,8.5)
      call basalf('stand')
      if(choice.eq.2) then
          go to 600
      end if
c
c
      call physor(1.75,1.0)
      call mx2alf('greek',1h*)
      call xname('Spatial Lag *n) (kilometers)$',100)
      call yname('Correlation Coefficient: *r(n))$',100)
      call yaxang(0.0)
      call grace(0.)
      call area2d(8.0,5.0)
C
      if(model.eq.1) then
      call headin('Spatial Correlation$',100,1.3,4)
      call headin('Model 1 Simulation$',100,1.0,4)
      call headin('Theory$',100,1.0,4)
      call headin('Observation$',100,1.0,4)
      else if (model.eq.2) then
      call headin('Spatial Correlation$',100,1.3,4)
      call headin('Model 2 Simulation$',100,1.0,4)
      call headin('Theory$',100,1.0,4)
      call headin('Observation$',100,1.0,4)
      else if (model.eq.3) then
      call headin('Spatial Correlation$',100,1.3,4)
      call headin('Model 3 Simulation$',100,1.0,4)
      call headin('Theory$',100,1.0,4)
      call headin('Observation$',100,1.0,4)
      end if
c
      call thkfrm(0.01)
      call frame
      call messag('Storm Day$',9,6.1,5.6)
      call messag(amnth,4,6.0,5.3)
      call messag(',$',1,6.9,5.3)
      call intno(iiday,6.6,5.3)
```

```
      call intno(iiyear,7.0,5.3)
      if(leg.eq.1) then
      call messag('Theoretical Model$',17,1.3,1.39)
      call messag('Observation$',11,1.3,1.08)
      call messag('Level I Sampling$',16,1.3,0.78)
      call messag('Level II Sampling$',17,1.3,0.5)
      call messag('Level III Sampling$',18,1.3,0.2)
      else if(leg.eq.2) then
      call messag('Theoretical Model$',17,5.5,4.7)
      call messag('Observation$',11,5.5,4.4)
      call messag('Level I Sampling$',16,5.5,4.1)
      call messag('Level II Sampling$',17,5.5,3.8)
      call messag('Level III Sampling$',18,5.5,3.5)
      end if
      call graf(0.,1.,6.,-1.,0.2,1.0)
c
      call poly3
c
c      Legend
c
      call curve(xthry,ythry,2,0)
      call dash
      call curve(xobs,yobs,2,0)
      call reset ('dash')
      call dot
      call curve(xlev3,ylev3,2,0)
      call reset('dot')
      call chndot
      call curve(xlev2,ylev2,2,0)
      call reset('chndot')
      call chndsh
      call curve(xlev1,ylev1,2,0)
      call reset('chndsh')
      if(leg.eq.1) then
      call blrec(0.0,0.0,3.8,1.6,0.001)
      else
      call blrec(4.1,3.4,3.9,1.6,-0.01)
      end if
c
c     theory
c
      call thkcrv(1)
      call curve(xzero,yzero,31,0)
      call reset ('thkcrv')
      call curve(lagmod,corrmod1,31,0)
c
c     Observation
c
      call dash
      call curve(lag1,corr1,31,0)
      call reset ('dash')
c
c
c     simulation (level 3:full square)
c

      call dot
      call curve(laglev3,corrlev3,31,0)
      call reset('dot')
c
```

```
c     simulation (level 2:rectangular area)
c
      call chndot
      call curve(laglev2,corrlev2,31,0)
      call reset('chndot')
c
c     simulation (level 1:basin boundary area)
c
      call chndsh
      call curve(laglev1,corrlev1,31,0)
      call reset('chndsh')
c
          go to 38
c
c     this section of the program starts the disspla plotting routines
c     for the variance function
c
600   continue
      print,'entering the variance function plotting routine'
      call physor(1.75,1.0)
      call mx2alf('greek',1h*)
      call xname('Area (square kilometers)$',100)
      call yname('Variance Function: *g) (Area)$',100)
      call yaxang(0.0)
      call grace(0.)
      call area2d(8.0,5.0)
c
      if (model.eq.1) then
      call headin('Variance Function$',100,1.3,4)
      call headin('Model 1 Simulation$',100,1.0,4)
      call headin('Theory$',100,1.0,4)
      call headin('Observation$',100,1.0,4)
      else if (model.eq.2) then
      call headin('Variance Function$',100,1.3,4)
      call headin('Model 2 Simulation$',100,1.0,4)
      call headin('Theory$',100,1.0,4)
      call headin('Observation$',100,1.0,4)
      else if (model.eq.3) then
      call headin('Variance Function$',100,1.3,4)
      call headin('Model 3 Simulation$',100,1.0,4)
      call headin('Theory$',100,1.0,4)
      call headin('Observation$',100,1.0,4)
      end if
c
      call thkfrm(0.01)
      call frame
      call messag('Storm Day$',9,6.1,5.6)
      call messag(amnth,4,6.0,5.3)
      call messag(',$',1,6.9,5.3)
      call intno(iiday,6.6,5.3)
      call intno(iiyear,7.0,5.3)
      if(leg.eq.2) then
      call messag('Theoretical Model$',17,5.5,4.7)
      call messag('Observation$',11,5.5,4.4)
      call messag('Level I Sampling$',16,5.5,4.1)
      call messag('Level II Sampling$',17,5.5,3.8)
      call messag('Level III Sampling$',18,5.5,3.5)
      else if(leg.eq.1) then
      call messag('Theoretical Model$',17,1.3,1.39)
      call messag('Observation$',11,1.3,1.08)
```

```
      call messag('Level I Sampling$',16,1.3,0.78)
      call messag('Level II Sampling$',17,1.3,0.5)
      call messag('Level III Sampling$',18,1.3,0.2)
      end if
      call yaxang(0.0)
      call graf(0.,4.,36.,0.,0.1,1.0)
      call poly3
c
c
c      Legend
c
      call curve(xthry,ythry,2,0)
      call dash
      call curve(xobs,yobs,2,0)
      call reset ('dash')
      call dot
      call curve(xlev3,ylev3,2,0)
      call reset('dot')
      call chndot
      call curve(xlev2,ylev2,2,0)
      call reset('chndot')
      call chndsh
      call curve(xlev1,ylev1,2,0)
      call reset('chndsh')
      if(leg.eq.1) then
      call blrec(0.0,0.0,3.8,1.6,0.001)
      else if(leg.eq.2) then
      call blrec(4.1,3.4,3.9,1.6,-0.01)
      end if
c
c     theory
c
      if((model.eq.1).or.(model.eq.3)) then
      call curve(areamod,gammod,31,0)
      else if(model.eq.2) then
      call curve(areamod,gammod,7,0)
      end if
c
c     Observation
c
      call dash
      call curve(xarea,ygamma,31,0)
      call reset ('dash')
c
c
c     simulation (level 3:full square)
c
      call dot
      call curve(arealev3,gamlev3,31,0)
      call reset('dot')
c
c     simulation (level 2:rectangular area)
c
      call chndot
      call curve(arealev2,gamlev2,31,0)
      call reset('chndot')
c
c     simulation (level 1:basin boundary area)
c
      call chndsh
```

```
      call curve(arealevl,gamlevl,31,0)
      call reset('chndsh')
c
38    continue
      call endpl(0)
      call donepl
c
          go to 35
4     continue
      print,'this run was aborted'
35    continue
      end
```

```
c                          SIMINTRP.FORTRAN
c
c     this program will read in the simulation gauges from file 83
c     and create an interpolated versian of sim_stormday.fortran
c
c     This program was developed by Neil M. Fennessey at M.I.T. during
c     the course of his Master's Degree research about the Areal Distribution
c     of Rainfall
c
c      ..............................................................
c
      dimension   iwat(93),igage(93),iyear(93),imonth(93)
      dimension   iday(93),zd(93),xd(93),yd(93),a(31)
      dimension   xi(147),yi(70),wk(558),zi(147,70)
      dimension   iwk(13173),amonth(6),zj(147,70)
      dimension   xarea(31),ygamma(31),area(0:30),gamma(0:30)
      dimension   eledpth(5000),aveeldph(0:30),vareldph(0:30)
      dimension   corr1(31),lag1(31),xzero(31),yzero(31),alpha(100)
      dimension   gammod1(31),gammod2(31),corrmod1(31),corrmod2(31)
      dimension   l1(1),xkey1(26),ykey1(26),xkey2(26),ykey2(26)
      dimension   xkey3(26),ykey3(26)
c
      common work(40000)
      real lag1,mmbsk0,mmbsk1,lag,l1
c
      data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
      data pi/3.14159265389793/
          api=pi
c
      print,'ready to go, enter 1, to try again, enter a 2'
      input,f1
c
      if(f1.eq.1) then
          go to 3
      else
          go to 4
      end if
c
3     continue
c
c     the section reads in the simulation "gauges" from file83 and devides
c     these coordinates in two (both x and y)
c
      rewind(83)
      do 30 i=1,93
      read(83,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i),
     &xd(i),yd(i)
          xd(i)=xd(i)/2.
          yd(i)=yd(i)/2.
30    continue
          nd=93
      rewind(83)
1000  format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
c
c
c     read the results from file29 (square simulated field (whole field)
c
      read(29,1280)model
1280  format(3x,'model number ',i2,' param a1 ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
```

```
c
          iiday=iday(1)
          iiyear=1900+iyear(1)
          iii=imonth(1)-5
      print,'..........................  '
      print1005,amonth(iii),iiday,iiyear
      print,'..........................  '
1005  format('storm day ',a4,i3,i5)
      print,' '
c
          a5=0.0
          d5=0.0
          a4=0.0
c
c     this portion generates the x and y values for the grid mesh for
c     which the program will interpolate
c
      nxi=147
      nyi=70
      izi=147
      nxi1=146
      nyi1=69
      nxi2=nxi*2
      nyi2=nyi*2
      do 60 j=1,nxi
          xi(j)=j
60    continue
      do 70 i=1,nyi
          yi(i)=i
70    continue
c
c     invoke the subroutine call to generate the interpolated three
c     dimensional surface from the gage data onto the rectangular
c     grid mesh
c
      call iqhscv(xd,yd,zd,nd,xi,nxi,yi,nyi,zi,izi,iwk,wk,ier)
c
c     this section of the program overlays the outside of the walnut
c     gulch basin with depth values of -1. mm. The values generated by
c     the subroutine are left intact for the integration routine and
c     also the contour plotting.
c
c     nxi refers to the number of columns in the matrix
c     nyi refers to the number of rows in the matrix
c
      do 10 i=2,nyi2,2
      do 20 j=2,nxi2,2
          l=j/2
          m=i/2
c
c      this is the new logic for the proper usda scale map
c
      if((i.eq.2)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.4)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.6)) then
          zi(l,m)=-1.
```

```
      go to 20
 else if((i.eq.8)) then
      zi(1,m)=-1.
      go to 20
 else if((i.eq.10).and.(((j.ge.126).and.(j.le.128)).or.((j.ge.148).and.
&(j.le.150)))) then
      go to 25
 else if(i.eq.10) then
      zi(1,m)=-1.
 else if((i.eq.12).and.(((j.ge.88).and.(j.le.90)).or.((j.ge.126)
&.and.(j.le.132)).or.((j.ge.146).and.(j.le.150)))) then
      go to 25
 else if(i.eq.12) then
      zi(1,m)=-1.
 else if((i.eq.14).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126)
&.and.(j.le.138)).or.((j.ge.140).and.(j.le.152)))) then
      go to 25
 else if(i.eq.14) then
      zi(1,m)=-1.
 else if((i.eq.16).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.116).and.
&(j.le.160)))) then
      go to 25
 else if(i.eq.16) then
      zi(1,m)=-1.
 else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
&(j.le.162)))) then
      go to 25
 else if(i.eq.18) then
      zi(1,m)=-1.
 else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.116).and.
&(j.le.164)))) then
      go to 25
 else if(i.eq.20) then
      zi(1,m)=-1.
 else if((i.eq.22).and.(((j.ge.80).and.(j.le.82)).or.((j.ge.84).and.
&(j.le.102)).or.((j.ge.118).and.(j.le.166)))) then
      go to 25
 else if(i.eq.22) then
      zi(1,m)=-1.
 else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
      go to 25
 else if(i.eq.24) then
      zi(1,m)=-1.
 else if((i.eq.26).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
      go to 25
 else if(i.eq.26) then
      zi(1,m)=-1.
 else if((i.eq.28).and.(((j.ge.84).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.170)))) then
      go to 25
 else if(i.eq.28) then
      zi(1,m)=-1.
 else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.110).and.
&(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
      go to 25
 else if(i.eq.30) then
      zi(1,m)=-1.
      go to 25
```

```
  else if((i.eq.32).and.(j.ge.82).and.(j.le.170)) then
      go to 25
  else if(i.eq.32) then
      zi(l,m)=-1.
  else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
      go to 25
  else if(i.eq.34) then
      zi(l,m)=-1.
      go to 25
  else if((i.eq.36).and.(j.ge.82).and.(j.le.174)) then
      go to 25
  else if(i.eq.36) then
      zi(l,m)=-1.
  else if((i.eq.38).and.(j.ge.84).and.(j.le.174)) then
      go to 25
  else if(i.eq.38) then
      zi(l,m)=-1.
  else if((i.eq.40).and.(j.ge.82).and.(j.le.172)) then
      go to 25
  else if(i.eq.40) then
      zi(l,m)=-1.
  else if((i.eq.42).and.(((j.ge.52).and.(j.le.54)).or.((j.ge.70).and.
&(j.le.74)).or.((j.ge.78).and.(j.le.172)))) then
      go to 25
  else if(i.eq.42) then
      zi(l,m)=-1.
  else if((i.eq.44).and.(((j.ge.52).and.(j.le.56)).or.((j.ge.58).and.
&(j.le.172)))) then
      go to 25
  else if(i.eq.44) then
      zi(l,m)=-1.
  else if((i.eq.46).and.(j.ge.52).and.(j.le.174)) then
      go to 25
  else if(i.eq.46) then
      zi(l,m)=-1.
  else if((i.eq.48).and.(j.ge.54).and.(j.le.174)) then
      go to 25
  else if(i.eq.48) then
      zi(l,m)=-1.
  else if((i.eq.50).and.(((j.ge.58).and.(j.le.174)).or.((j.ge.180).and.
&(j.le.182)))) then
      go to 25
  else if(i.eq.50) then
      zi(l,m)=-1.
  else if((i.eq.52).and.(j.ge.58).and.(j.le.182)) then
      go to 25
  else if(i.eq.52) then
      zi(l,m)=-1.
  else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
      go to 25
  else if(i.eq.54) then
      zi(l,m)=-1.
  else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
      go to 25
  else if(i.eq.56) then
      zi(l,m)=-1.
  else if((i.eq.58).and.(((j.ge.36).and.(j.le.38)).or.((j.ge.52).and.
&(j.le.196)))) then
      go to 25
  else if(i.eq.58) then
```

```
     zi(1,m)=-1.
 else if((i.eq.60).and.(j.ge.34).and.(j.le.198)) then
     go to 25
 else if(i.eq.60) then
     zi(1,m)=-1.
 else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
     go to 25
 else if(i.eq.62) then
     zi(1,m)=-1.
 else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
&(j.le.221)))) then
     go to 25
 else if(i.eq.64) then
     zi(1,m)=-1.
 else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
     go to 25
 else if(i.eq.66) then
     zi(1,m)=-1.
 else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
     go to 25
 else if(i.eq.68) then
     zi(1,m)=-1.
 else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
     go to 25
 else if(i.eq.70) then
     zi(1,m)=-1.
 else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
     go to 25
 else if(i.eq.72) then
     zi(1,m)=-1.
 else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
     go to 25
 else if(i.eq.74) then
     zi(1,m)=-1.
 else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
     go to 25
 else if(i.eq.76) then
     zi(1,m)=-1.
 else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
     go to 25
 else if(i.eq.78) then
     zi(1,m)=-1.
 else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
     go to 25
 else if(i.eq.80) then
     zi(1,m)=-1.
 else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
     go to 25
 else if(i.eq.82) then
     zi(1,m)=-1.
 else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
     go to 25
 else if(i.eq.84) then
     zi(1,m)=-1.
 else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
     go to 25
 else if(i.eq.86) then
     zi(1,m)=-1.
 else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
     go to 25
```

```
 else if(i.eq.88) then
     zi(l,m)=-1.
 else if((i.eq.90).and.(j.ge.17).and.(j.le.238)) then
     go to 25
 else if(i.eq.90) then
     zi(l,m)=-1.
 else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
     go to 25
 else if(i.eq.92) then
     zi(l,m)=-1.
 else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
     go to 25
 else if(i.eq.94) then
     zi(l,m)=-1.
 else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
     go to 25
 else if(i.eq.96) then
     zi(l,m)=-1.
 else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
     go to 25
 else if(i.eq.98) then
     zi(l,m)=-1.
 else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
     go to 25
 else if(i.eq.100) then
     zi(l,m)=-1.
 else if((i.eq.102).and.(j.ge.44).and.(j.le.239)) then
     go to 25
 else if(i.eq.102) then
     zi(l,m)=-1.
 else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
     go to 25
 else if(i.eq.104) then
     zi(l,m)=-1.
 else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
     go to 25
 else if(i.eq.106) then
     zi(l,m)=-1.
 else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
     go to 25
 else if(i.eq.108) then
     zi(l,m)=-1.
 else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
     go to 25
 else if(i.eq.110) then
      zi(l,m)=-1.
 else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
     go to 25
 else if(i.eq.112) then
      zi(l,m)=-1.
 else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
&.and.(j.le.256)))) then
     go to 25
 else if(i.eq.114) then
      zi(l,m)=-1.
 else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
&.and.(j.le.257)))) then
     go to 25
 else if(i.eq.116) then
      zi(l,m)=-1.
```

```
      else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
     &.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
          go to 25
      else if(i.eq.118) then
           zi(l,m)=-1.
      else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
     &.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
          go to 25
      else if(i.eq.120) then
           zi(l,m)=-1.
      else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.273)
     &.and.(j.le.283)))) then
          go to 25
      else if(i.eq.122) then
           zi(l,m)=-1.
      else if((i.eq.124).and.(((j.ge.243).and.(j.le.258)).or.((j.ge.272)
     &.and.(j.le.281)))) then
          go to 25
      else if(i.eq.124) then
           zi(l,m)=-1.
      else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
     &.and.(j.le.279)))) then
          go to 25
      else if(i.eq.126) then
           zi(l,m)=-1.
      else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
          go to 25
      else if(i.eq.128) then
           zi(l,m)=-1.
      else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
          go to 25
      else if(i.eq.130) then
           zi(l,m)=-1.
      else if(i.eq.132) then
           zi(l,m)=-1.
          go to 20
      else if(i.eq.134) then
          zi(l,m)=-1.
          go to 20
      else if(i.eq.136) then
          zi(l,m)=-1.
          go to 20
      else if(i.eq.138) then
          zi(l,m)=-1.
          go to 20
      else if(i.eq.140) then
          zi(l,m)=-1.
          go to 20
      end if
c
25    continue
20    continue
10    continue
c
c
c     this portion of the program looks for undulations which may have
c     been created by the surface fitting subroutine, anything less than
c     0.01 mm equals zero
c
      do 80 i=1,nxi
```

```
      do 90 j=1,nyi
      if(zi(i,j).eq.-1.) then
          go to 90
      else if(zi(i,j).le.0.01) then
          zi(i,j)=0.0
      end if
90    continue
80    continue
c
c     calculate the average point depth for the field within the basin limits
c
      do 110 i=1,nxi
      do 120 j=1,nyi
c
      if(zi(i,j).eq.-1.) then
          go to 120
      end if
          a1=zi(i,j)+a1
          d1=d1+1
120   continue
110   continue
          avedepth=a1/d1
c
c
c     this portion will calculate the variance of the point depth within the
c     basin limits and the coefficient of skewness
c
          l=0
c
      do 130 i=1,nxi
      do 140 j=1,nyi
c
      if(zi(i,j).eq.-1.) then
          go to 140
      end if
          l=l+1
          a2=((zi(i,j)-avedepth)**2.)+a2
          a3=((zi(i,j)-avedepth)**3.)+a3
140   continue
130   continue
          vardepth=a2/(l-1)
          stdevdph=sqrt(vardepth)
          skwdpth=a3/((l-1)*(stdevdph**3.0))
      print 1030,avedepth
      print 1040,vardepth
      print 1045,skwdpth
1030  format('the average point depth=',f10.4,' mm.')
1040  format('the variance of the point depth=',f10.4,' mm ')
1045  format('the coefficeint of skewness of the point depth=',f10.4)
c
c     reset variables equal to zero
c
          a1=0.0
          a2=0.0
          a3=0.0
c     ************************************************************************
c
c     this portion of the program will determine the two dimensional
c     spatial correlation.  The correlation will be determined for point
c     depth spacings from 0.2 km. to 6 km.
```

```
c
c     ************************************************************************
c
c     entering the incremental (changes spatial distance) loop
c
      do 150 k1=0,30
c
          t1=0.
          b1=0.
          b2=0.
          d2=0.
          nxi3=nxi-k1
          nyi3=nyi-k1
c
c     this part of the correlation determination sweeps from left to
c     right across the generated surface fitting matrix.  Pairs of point depths
c     which are not -1.0 are the only values included in each determination.
c
      do 160 j=1,nyi
      do 170 i=1,nxi3
      if((zi(i,j).eq.-1.).or.(zi(i+k1,j).eq.-1.)) then
          go to 170
      else
          t1=((zi(i,j)-avedepth)*(zi(i+k1,j)-avedepth))+t1
          b1=((zi(i,j)-avedepth)**2.)+b1
          b2=((zi(i+k1,j)-avedepth)**2.)+b2
          d2=d2+1
      end if
170   continue
160   continue
c
c     this part of the correlation determination sweeps from the bottom
c     to the top of the generated surface fitting matrix
c
      do 180 i=1,nxi
      do 190 j=1,nyi3
      if((zi(i,j).eq.-1.).or.(zi(i,j+k1).eq.-1.)) then
          go to 190
      else
          t1=((zi(i,j)-avedepth)*(zi(i,j+k1)-avedepth))+t1
          b1=((zi(i,j)-avedepth)**2.)+b1
          b2=((zi(i,j+k1)-avedepth)**2.)+b2
          d2=d2+1
      end if
190   continue
180   continue
c
c     calculate the correlation coefficient for this spacing
c
          e1=0
          e1=t1/(sqrt(b1)*sqrt(b2))
          corr1(k1+1)=e1
          r1=k1
          r2=r1/5
          lag1(k1+1)=r2
c      draw the zero line
          xzero(k1+1)=r2
          yzero(k1+1)=0.
c
150   continue
```

```
c
c     reset variables equal to zero
c
          t1=0.0
          b1=0.0
          b2=0.0
          d2=0.0
          e1=0.0
          r1=0.0
          r2=0.0
          k1=0
c
c
c     ***********************************************************************
c
c     this portion of the program determines the values of the variance
c     function gamma(area)
c
c     ***********************************************************************

c     begin the loop which varies the length scale of the element
c
      do 220 m=0,30
          area(m)=((m*0.2)**2.)
c
c     enter the mesh at this scale.  Sweep the mesh left to right and from
c     the bottom to the top simultaneously
c
      do 230 j=1,nyi-m
      do 240 i=1,nxi-m
c
c     trip the element dimension counter
          k1=k1+1
c
c     determine the depth of the individual element now
c
      do 250 l=j,j+m
      do 260 k=i,i+m
c
c     check to see if any of the point depths are -1.
      if(zi(k,l).eq.-1.) then
          numpts=0
          totdpth=0.
          k1=k1-1
          go to 240
      else
          numpts=numpts+1
          totdpth=totdpth+zi(k,l)
     end if
260  continue
250  continue
c
c     the element depth is the average value of the point depths
c     which comprise the element
c
          eledpth(k1)=totdpth/numpts
          toteldph=toteldph+eledpth(k1)
          numpts=0
          totdpth=0.0
240    continue
```

```
230    continue
c
c      determine the average element depth and the moving average
c
           aveeldph(m)=toteldph/k1
c
c      determine the variance of the element depth at this length scale
c
       do 270 k2=1,k1
           totvardp=totvardp+((eledpth(k2)-aveeldph(m))**2.)
270     continue
           vareldph(m)=totvardp/(k1-1)
c
c      determine the value of gamma(A).  Set all plot values of gamma(A)
c      greater than one to one.
c
           gamma(m)=vareldph(m)/vardepth
           xarea(m+1)=area(m)
       if(gamma(m).gt.1.0)  then
           ygamma(m+1)=1.0
       else
           ygamma(m+1)=gamma(m)
       end if
           toteldph=0.
           totvardp=0.
           k1=0
       print221,area(m),gamma(m)
220    continue
221    format('area=',f5.2,' gamma(A)=',f7.4)
c
c      reset variables equal to zero
c
           toteldph=0.0
           totvardp=0.0
           b1=0.0
           b2=0.0
           b3=0.0
c
c
c      ****************************************************************************
c
c      This portion of the program will blank out the area outside of the
c      watershed basin boundary
c
c      ****************************************************************************
c
c      this portion of the program overlays the prior blocked out matrix
c      for the purpose of drawing a single countour line which represents
c      the basin boundary
c
c      ****************************************************************************
c
       do 390 i=1,nxi
       do 410 j=1,nyi
       if(zi(i,j).eq.-1.0)  then
           zj(i,j)=0.1
       else
           zj(i,j)=-0.99
       end if
410    continue
```

```
390   continue
c
c
c     this portion of the program looks for undulations which may have
c     been created by the surface fitting subroutine, this second filter
c     is only for plotting purposes
c
      do 420 i=1,nxi
      do 430 j=1,nyi
      if((zi(i,j).lt.0.01).and.(zi(i,j).gt.-1.0)) then
          zi(i,j)=-0.1
      else if((zi(i,j).eq.-1.)) then
          zi(i,j)=0.0
      end if
430   continue
420   continue
c
c
c     this portion of the code draws straight line segments to be
c     plotted on the legend (coutour lines and basin boundary)
c
      do 490 j=208,233
          i=j-207
          r10=j
          r11=r10/10.0
          xkey1(i)=r11
          ykey1(i)=3.4
          xkey2(i)=r11
          ykey2(i)=2.8
          xkey3(i)=r11
          ykey3(i)=1.8
490   continue
c
c     this section of the program starts the disspla plotting routines
c     for the correlation function
c
      print,'entering the plotting routine for the correlation function'
c
c
c
      call comprs
      call blowup(0.6063)
      call page(14.0196,18.1428)
      call physor(2.98688,3.1338)
      call basalf('stand')
      call mx2alf('greek',1h*)
      call xname('Spatial Lag *n)  (kilometers)$',100)
      call yname('Correlation Coefficient: *r(n))$',100)
      call area2d(4.286,4.286)
      call headin('Spatial Correlation$',100,1.5,1)
      call thkfrm(0.01)
      call frame
      call yaxang(0.0)
      call graf(0.,1.,6.,-1.,0.2,1.0)
      call poly3
      call curve(lag1,corr1,31,0)
      call thkcrv(1)
      call curve(xzero,yzero,31,0)
      call endgr(1)
c
```

```
c     generate the results for plotting of the variance function
c
c
c     this section of the program starts the disspla plotting routines
c     for the variance function
c
      print,'entering the variance function plotting routine'
      call physor(8.7009,3.1338)
      call xname('Area (square kilometers)$',100)
      call yname('Variance Function: *g) (Area)$',100)
      call area2d(4.286,4.286)
      call headin('Variance Function$',100,1.5,1)
      call thkfrm(0.01)
      call frame
      call graf(0.,4.,36.,0.,0.1,1.0)
      call poly3
      call curve(xarea,ygamma,31,0)
      call endgr(2)
c
c     entering the disspla routine for the contour plot
c
      print,'enter the countour plot plotting routine'
c
      call physor(2.9869,9.8136)
      call area2d(10.0,4.762)
      call reset('yaxang')
      call xname('Kilometers$',100)
      call yname('Kilometers$',100)
c
      if (model.eq.1) then
      call headin('Interpolation of $',100,1.5,4)
      call headin('Model 1 Simulation$',100,1.5,4)
      call headin('Walnut Gulch, Arizona$',100,1.4,4)
      call headin('Ac=154.21 sq.km.$',100,1.0,4)
      else if(model.eq.2) then
      call headin('Interpolation of $',100,1.5,4)
      call headin('Model 2 Simulation$',100,1.5,4)
      call headin('Walnut Gulch, Arizona$',100,1.4,4)
      call headin('Ac=154.21 sq.km.$',100,1.0,4)
      else if(model.eq.3) then
      call headin('Interpolation of $',100,1.5,4)
      call headin('Model 3 Simulation$',100,1.5,4)
      call headin('Walnut Gulch, Arizona$',100,1.4,4)
      call headin('Ac=154.21 sq.km.$',100,1.0,4)
      end if
c
      call intaxs
      call xintax
      call yintax
      call messag('Contour Interval: 2 mm.$',100,0.3,4.25)
      call messag('Storm Day$',9,8.1,5.5)
      call messag(amonth(iii),4,7.9,5.2)
      call messag(',$',1,8.8,5.2)
      call intno(iiday,8.5,5.2)
      call intno(iiyear,8.9,5.2)
      call messag('Legend',6,7.9,1.5)
      call messag('Isohyet Line',12,8.0,1.1)
      call messag('Catchment',9,8.0,0.7)
      call messag('Boundary',8,8.0,0.5)
      call graf(0.,2.0,30.0,0.,1.0,14.0)
```

```
      call thkfrm(0.01)
      call frame
      call curve(xkey1,ykey1,26,0)
      call dash
      call curve(xkey2,ykey2,26,0)
      call reset('dash')
      call dot
      call thkcrv(5)
      call curve(xkey3,ykey3,26,0)
      call reset('dot')
      call bcomon(40000)
      call conthn(0.04)
c
c     plot the contours
c
      call conmak(zi,nxi,nyi,2.)
      call conlin(0,'solid','label',1,7)
      call conlin(1,'dash','labels',1,4)
      call conang(90.)
      call raspln(0.25)
      print,'entering routine     call contur(2,labels,draw) '
      call contur(2,'labels','draw')
c
c     scribe the outline of the basin boundary as a separate contour line
c
      call bcomon(40000)
      call reset('conang')
      call reset('conthn')
      call conmak(zj,nxi,nyi,2.)
      call conlin(0,'dot','no labels',5,10)
      call raspln(0.25)
      print,'entering routine second call contur(1,no labels,draw) '
      print,' *** the code will fail here, simply enter "start"'
      print,' but without the quotation marks'
      call contur(1,'nolabels','draw')
      call endpl(1)
      call donepl
c
          go to 35
c
4     continue
      print,'the run has been aborted'
35     continue
      end
```

```
c                           SIMCORR.FORTYRAN
c
c     This program determines the
c     correlation between the surface created by the surface interpolator
c     using the "simulation" rain gauges from file83 and the intact
c     simulated surface read in from file29.  The resulting value will
c     be placed in file61 .  A double watershed boundary filter is used
c     to accomplish this determination of correlation
c
c     This program was developed by Neil M. Fennessey at M.I.T. during
c     the course of his Master's Degree research about the Areal Distribution
c     of Rainfall
c
c      ..............................................................
c
      dimension   iwat(93),igage(93),iyear(93),imonth(93)
      dimension   iday(93),zd(93),xd(93),yd(93)
      dimension   xi(147),yi(70),x(100),y(100)
      dimension   amonth(6),zi(147,70),wk(558),iwk(13173)
      dimension   zk(147,70)
c
      common/blank/zj(147,70)
      common /dpmod1/depth1(151,151)
c
      double precision dseed
c
c
c
      data  (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
      data pi/3.14159265389793/
          pi2=2*pi
          api=pi
c
c
c     the section reads in the simulation gage data from file83
c
          nd=93
c
      do 30  i=1,93
      read(83,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i),
     &xd(i),yd(i)
          xd(i)=xd(i)/2.0
          yd(i)=yd(i)/2.0
30   continue
      rewind(83)
1000  format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
1110  format(f6.2,2(1x,f4.0))
c
          iiday=iday(1)
          iiyear=1900+iyear(1)
          iii=imonth(1)-5
      print,'..........................   '
      print1005,amonth(iii),iiday,iiyear
      print,'..........................   '
1005  format('storm day ',a4,i3,i5)
      print,' '
c
c     empty out the common block areas
c
      do 670  i=1,151
```

```
      do 680 j=1,151
          depth1(i,j)=0.0
680   continue
670   continue
      do 675 i=1,147
      do 685 j=1,70
          zk(i,j)=0.0
          zj(i,j)=0.0
685   continue
675   continue
c
c     read the simulation field (square simulated field (whole field)
c     from file29
c
      read(29,1280)im
1280  format(3x,'model number ',i2,' param a1 ',f7.4,' cell conc. lambda ',
     & f7.4,' E(alpha) ',f9.4)
      read(29,1290)iyear(1),imonth(1),iday(1)
      do 667 i=1,151
      do 677 j=1,151
c      read(29,1300)depth1(i,j)
      read(29,1300)i,j,depth1(i,j)
677   continue
667   continue
1290  format(3x,i3,'/',i3,'/',i3)
1300  format(2x,i4,i4,f8.3)
c         1300  format(f8.3)
c
c     retain the appropriate values for inclusion within the basin outline
c     from the simnulataion
c
      do 650 i=1,147
      do 660 j=40,109
          l=j-39
          zi(i,l)=depth1(i,j)
660   continue
650   continue
c
c     this portion generates the x and y values for the grid mesh for
c     which the program will interpolate
c
      nxi=147
      nyi=70
      izi=147
      nxi1=146
      nyi1=69
      nxi2=nxi*2
      nyi2=nyi*2
      do 60 i=1,nxi
          xi(i)=i
60    continue
      do 70 j=1,nyi
          yi(j)=j
70    continue
c
c     invoke the subroutine call to generate the interpolated three
c     dimensional surface from the gage data onto the rectangular
c     grid mesh
c
      call iqhscv(xd,yd,zd,nd,xi,nxi,yi,nyi,zk,izi,iwk,wk,ier)
```

```
      print,'iqhscv err=',ier
c
c
c     this section of the program overlays the outside of the walnut
c     gulch basin with depth values of -1. mm. The values generated by
c     the subroutine are left intact for the integration routine and
c     also the contour plotting.
c
c     nxi refers to the number of columns in the matrix
c     nyi refers to the number of rows in the matrix
c
      do 10 i=2,nyi2,2
      do 20 j=2,nxi2,2
          l=j/2
          m=i/2
c
c      this is the new logic for the proper usda scale map and the simulated
c      field
c
      if((i.eq.2)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.4)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.6)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.8)) then
          zi(l,m)=-1.
          go to 20
      else if((i.eq.10).and.(((j.ge.126).and.(j.le.128)).or.((j.ge.148).and.
     &(j.le.150)))) then
          go to 25
      else if(i.eq.10) then
          zi(l,m)=-1.
      else if((i.eq.12).and.(((j.ge.88).and.(j.le.90)).or.((j.ge.126)
     &.and.(j.le.132)).or.((j.ge.146).and.(j.le.150)))) then
          go to 25
      else if(i.eq.12) then
          zi(l,m)=-1.
      else if((i.eq.14).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126)
     &.and.(j.le.138)).or.((j.ge.140).and.(j.le.152)))) then
          go to 25
      else if(i.eq.14) then
          zi(l,m)=-1.
      else if((i.eq.16).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.116).and.
     &(j.le.160)))) then
          go to 25
      else if(i.eq.16) then
          zi(l,m)=-1.
      else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
     &(j.le.162)))) then
          go to 25
      else if(i.eq.18) then
          zi(l,m)=-1.
      else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.116).and.
     &(j.le.164)))) then
          go to 25
      else if(i.eq.20) then
```

```
      zi (l,m)=-1.
 else if((i.eq.22).and.(((j.ge.80).and.(j.le.82)).or.((j.ge.84).and.
&(j.le.102)).or.((j.ge.118).and.(j.le.166)))) then
      go to 25
 else if(i.eq.22) then
      zi (l,m)=-1.
 else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
      go to 25
 else if(i.eq.24) then
      zi (l,m)=-1.
 else if((i.eq.26).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
      go to 25
 else if(i.eq.26) then
      zi (l,m)=-1.
 else if((i.eq.28).and.(((j.ge.84).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.170)))) then
      go to 25
 else if(i.eq.28) then
      zi (l,m)=-1.
 else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.110).and.
&(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
      go to 25
 else if(i.eq.30) then
      zi (l,m)=-1.
      go to 25
 else if((i.eq.32).and.(j.ge.82).and.(j.le.170)) then
      go to 25
 else if(i.eq.32) then
      zi (l,m)=-1.
 else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
      go to 25
 else if(i.eq.34) then
      zi (l,m)=-1.
      go to 25
 else if((i.eq.36).and.(j.ge.82).and.(j.le.174)) then
      go to 25
 else if(i.eq.36) then
      zi (l,m)=-1.
 else if((i.eq.38).and.(j.ge.84).and.(j.le.174)) then
      go to 25
 else if(i.eq.38) then
      zi (l,m)=-1.
 else if((i.eq.40).and.(j.ge.82).and.(j.le.172)) then
      go to 25
 else if(i.eq.40) then
      zi (l,m)=-1.
 else if((i.eq.42).and.(((j.ge.52).and.(j.le.54)).or.((j.ge.70).and.
&(j.le.74)).or.((j.ge.78).and.(j.le.172)))) then
      go to 25
 else if(i.eq.42) then
      zi (l,m)=-1.
 else if((i.eq.44).and.(((j.ge.52).and.(j.le.56)).or.((j.ge.58).and.
&(j.le.172)))) then
      go to 25
 else if(i.eq.44) then
      zi (l,m)=-1.
 else if((i.eq.46).and.(j.ge.52).and.(j.le.174)) then
      go to 25
```

```
 else if(i.eq.46) then
     zi(l,m)=-1.
 else if((i.eq.48).and.(j.ge.54).and.(j.le.174)) then
     go to 25
 else if(i.eq.48) then
     zi(l,m)=-1.
 else if((i.eq.50).and.(((j.ge.58).and.(j.le.174)).or.((j.ge.180).and.
&(j.le.182)))) then
     go to 25
 else if(i.eq.50) then
     zi(l,m)=-1.
 else if((i.eq.52).and.(j.ge.58).and.(j.le.182)) then
     go to 25
 else if(i.eq.52) then
     zi(l,m)=-1.
 else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
     go to 25
 else if(i.eq.54) then
     zi(l,m)=-1.
 else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
     go to 25
 else if(i.eq.56) then
     zi(l,m)=-1.
 else if((i.eq.58).and.(((j.ge.36).and.(j.le.38)).or.((j.ge.52).and.
&(j.le.196)))) then
     go to 25
 else if(i.eq.58) then
     zi(l,m)=-1.
 else if((i.eq.60).and.(j.ge.34).and.(j.le.198)) then
     go to 25
 else if(i.eq.60) then
     zi(l,m)=-1.
 else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
     go to 25
 else if(i.eq.62) then
     zi(l,m)=-1.
 else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
&(j.le.221)))) then
     go to 25
 else if(i.eq.64) then
     zi(l,m)=-1.
 else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
     go to 25
 else if(i.eq.66) then
     zi(l,m)=-1.
 else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
     go to 25
 else if(i.eq.68) then
     zi(l,m)=-1.
 else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
     go to 25
 else if(i.eq.70) then
     zi(l,m)=-1.
 else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
     go to 25
 else if(i.eq.72) then
     zi(l,m)=-1.
 else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
     go to 25
 else if(i.eq.74) then
```

```
     zi(1,m)=-1.
else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
     go to 25
else if(i.eq.76) then
     zi(1,m)=-1.
else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
     go to 25
else if(i.eq.78) then
     zi(1,m)=-1.
else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
     go to 25
else if(i.eq.80) then
     zi(1,m)=-1.
else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
     go to 25
else if(i.eq.82) then
     zi(1,m)=-1.
else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
     go to 25
else if(i.eq.84) then
     zi(1,m)=-1.
else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
     go to 25
else if(i.eq.86) then
     zi(1,m)=-1.
else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
     go to 25
else if(i.eq.88) then
     zi(1,m)=-1.
else if((i.eq.90).and.(j.ge.17).and.(j.le.238)) then
     go to 25
else if(i.eq.90) then
     zi(1,m)=-1.
else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
     go to 25
else if(i.eq.92) then
     zi(1,m)=-1.
else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
     go to 25
else if(i.eq.94) then
     zi(1,m)=-1.
else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
     go to 25
else if(i.eq.96) then
     zi(1,m)=-1.
else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
     go to 25
else if(i.eq.98) then
     zi(1,m)=-1.
else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
     go to 25
else if(i.eq.100) then
     zi(1,m)=-1.
else if((i.eq.102).and.(j.ge.44).and.(j.le.239)) then
     go to 25
else if(i.eq.102) then
     zi(1,m)=-1.
else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
     go to 25
else if(i.eq.104) then
```

```
     zi(1,m)=-1.
 else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
     go to 25
 else if(i.eq.106) then
     zi(1,m)=-1.
 else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
     go to 25
 else if(i.eq.108) then
     zi(1,m)=-1.
 else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
     go to 25
 else if(i.eq.110) then
      zi(1,m)=-1.
 else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
     go to 25
 else if(i.eq.112) then
      zi(1,m)=-1.
 else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
&.and.(j.le.256)))) then
     go to 25
 else if(i.eq.114) then
      zi(1,m)=-1.
 else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
&.and.(j.le.257)))) then
     go to 25
 else if(i.eq.116) then
      zi(1,m)=-1.
 else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
&.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
     go to 25
 else if(i.eq.118) then
      zi(1,m)=-1.
 else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
&.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
     go to 25
 else if(i.eq.120) then
      zi(1,m)=-1.
 else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.273)
&.and.(j.le.283)))) then
     go to 25
 else if(i.eq.122) then
      zi(1,m)=-1.
 else if((i.eq.124).and.(((j.ge.243).and.(j.le.258)).or.((j.ge.272)
&.and.(j.le.281)))) then
     go to 25
 else if(i.eq.124) then
      zi(1,m)=-1.
 else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
&.and.(j.le.279)))) then
     go to 25
 else if(i.eq.126) then
      zi(1,m)=-1.
 else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
     go to 25
 else if(i.eq.128) then
      zi(1,m)=-1.
 else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
     go to 25
 else if(i.eq.130) then
      zi(1,m)=-1.
```

```
      else if(i.eq.132) then
           zi(l,m)=-1.
          go to 20
      else if(i.eq.134) then
          zi(l,m)=-1.
          go to 20
      else if(i.eq.136) then
          zi(l,m)=-1.
          go to 20
      else if(i.eq.138) then
          zi(l,m)=-1.
          go to 20
      else if(i.eq.140) then
          zi(l,m)=-1.
          go to 20
      end if
c
25    continue
20    continue
10    continue
c
c     this over lay is for the interpolated field from the gages stripped
c     from the fine mesh simulation: point depth is zk(l,m)
c
      do 45 i=2,nyi2,2
      do 55 j=2,nxi2,2
          l=j/2
          m=i/2
c
c      this is the new logic for the proper usda scale map and the simulated
c      field
c
      if((i.eq.2)) then
          zk(l,m)=-1.
          go to 205
      else if((i.eq.4)) then
          zk(l,m)=-1.
          go to 205
      else if((i.eq.6)) then
          zk(l,m)=-1.
          go to 205
      else if((i.eq.8)) then
          zk(l,m)=-1.
          go to 205
      else if((i.eq.10).and.(((j.ge.126).and.(j.le.128)).or.((j.ge.148).and.
     &(j.le.150)))) then
          go to 205
      else if(i.eq.10) then
          zk(l,m)=-1.
      else if((i.eq.12).and.(((j.ge.88).and.(j.le.90)).or.((j.ge.126)
     &.and.(j.le.132)).or.((j.ge.146).and.(j.le.150)))) then
          go to 205
      else if(i.eq.12) then
          zk(l,m)=-1.
      else if((i.eq.14).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126)
     &.and.(j.le.138)).or.((j.ge.140).and.(j.le.152)))) then
          go to 205
      else if(i.eq.14) then
          zk(l,m)=-1.
      else if((i.eq.16).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.116).and.
```

```
&(j.le.160)))) then
     go to 205
 else if(i.eq.16) then
     zk(1,m)=-1.
 else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
&(j.le.162)))) then
     go to 205
 else if(i.eq.18) then
     zk(1,m)=-1.
 else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.116).and.
&(j.le.164)))) then
     go to 205
 else if(i.eq.20) then
     zk(1,m)=-1.
 else if((i.eq.22).and.(((j.ge.80).and.(j.le.82)).or.((j.ge.84).and.
&(j.le.102)).or.((j.ge.118).and.(j.le.166)))) then
     go to 205
 else if(i.eq.22) then
     zk(1,m)=-1.
 else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
     go to 205
 else if(i.eq.24) then
     zk(1,m)=-1.
 else if((i.eq.26).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.166)))) then
     go to 205
 else if(i.eq.26) then
     zk(1,m)=-1.
 else if((i.eq.28).and.(((j.ge.84).and.(j.le.106)).or.((j.ge.118).and.
&(j.le.170)))) then
     go to 205
 else if(i.eq.28) then
     zk(1,m)=-1.
 else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.110).and.
&(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
     go to 205
 else if(i.eq.30) then
     zk(1,m)=-1.
 else if((i.eq.32).and.(j.ge.82).and.(j.le.170)) then
     go to 205
 else if(i.eq.32) then
     zk(1,m)=-1.
 else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
     go to 205
 else if(i.eq.34) then
     zk(1,m)=-1.
 else if((i.eq.36).and.(j.ge.82).and.(j.le.174)) then
     go to 205
 else if(i.eq.36) then
     zk(1,m)=-1.
 else if((i.eq.38).and.(j.ge.84).and.(j.le.174)) then
     go to 205
 else if(i.eq.38) then
     zk(1,m)=-1.
 else if((i.eq.40).and.(j.ge.82).and.(j.le.172)) then
     go to 205
 else if(i.eq.40) then
     zk(1,m)=-1.
 else if((i.eq.42).and.(((j.ge.52).and.(j.le.54)).or.((j.ge.70).and.
```

```
&(j.le.74)).or.((j.ge.78).and.(j.le.172)))) then
      go to 205
 else if(i.eq.42) then
      zk(l,m)=-1.
 else if((i.eq.44).and.(((j.ge.52).and.(j.le.56)).or.((j.ge.58).and.
&(j.le.172)))) then
      go to 205
 else if(i.eq.44) then
      zk(l,m)=-1.
 else if((i.eq.46).and.(j.ge.52).and.(j.le.174)) then
      go to 205
 else if(i.eq.46) then
      zk(l,m)=-1.
 else if((i.eq.48).and.(j.ge.54).and.(j.le.174)) then
      go to 205
 else if(i.eq.48) then
      zk(l,m)=-1.
 else if((i.eq.50).and.(((j.ge.58).and.(j.le.174)).or.((j.ge.180).and.
&(j.le.182)))) then
      go to 205
 else if(i.eq.50) then
      zk(l,m)=-1.
 else if((i.eq.52).and.(j.ge.58).and.(j.le.182)) then
      go to 205
 else if(i.eq.52) then
      zk(l,m)=-1.
 else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
      go to 205
 else if(i.eq.54) then
      zk(l,m)=-1.
 else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
      go to 205
 else if(i.eq.56) then
      zk(l,m)=-1.
 else if((i.eq.58).and.(((j.ge.36).and.(j.le.38)).or.((j.ge.52).and.
&(j.le.196)))) then
      go to 205
 else if(i.eq.58) then
      zk(l,m)=-1.
 else if((i.eq.60).and.(j.ge.34).and.(j.le.198)) then
      go to 205
 else if(i.eq.60) then
      zk(l,m)=-1.
 else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
      go to 205
 else if(i.eq.62) then
      zk(l,m)=-1.
 else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
&(j.le.221)))) then
      go to 205
 else if(i.eq.64) then
      zk(l,m)=-1.
 else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
      go to 205
 else if(i.eq.66) then
      zk(l,m)=-1.
 else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
      go to 205
 else if(i.eq.68) then
      zk(l,m)=-1.
```

```
else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
    go to 205
else if(i.eq.70) then
    zk(l,m)=-1.
else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
    go to 205
else if(i.eq.72) then
    zk(l,m)=-1.
else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
    go to 205
else if(i.eq.74) then
    zk(l,m)=-1.
else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
    go to 205
else if(i.eq.76) then
    zk(l,m)=-1.
else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
    go to 205
else if(i.eq.78) then
    zk(l,m)=-1.
else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
    go to 205
else if(i.eq.80) then
    zk(l,m)=-1.
else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
    go to 205
else if(i.eq.82) then
    zk(l,m)=-1.
else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
    go to 205
else if(i.eq.84) then
    zk(l,m)=-1.
else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
    go to 205
else if(i.eq.86) then
    zk(l,m)=-1.
else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
    go to 205
else if(i.eq.88) then
    zk(l,m)=-1.
else if((i.eq.90).and.(j.ge.17).and.(j.le.238)) then
    go to 205
else if(i.eq.90) then
    zk(l,m)=-1.
else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
    go to 205
else if(i.eq.92) then
    zk(l,m)=-1.
else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
    go to 205
else if(i.eq.94) then
    zk(l,m)=-1.
else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
    go to 205
else if(i.eq.96) then
    zk(l,m)=-1.
else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
    go to 205
else if(i.eq.98) then
    zk(l,m)=-1.
```

```
 else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
     go to 205
 else if(i.eq.100) then
     zk(1,m)=-1.
 else if((i.eq.102).and.(j.ge.44).and.(j.le.239)) then
     go to 205
 else if(i.eq.102) then
     zk(1,m)=-1.
 else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
     go to 205
 else if(i.eq.104) then
     zk(1,m)=-1.
 else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
     go to 205
 else if(i.eq.106) then
     zk(1,m)=-1.
 else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
     go to 205
 else if(i.eq.108) then
     zk(1,m)=-1.
 else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
     go to 205
 else if(i.eq.110) then
      zk(1,m)=-1.
 else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
     go to 205
 else if(i.eq.112) then
      zk(1,m)=-1.
 else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
&.and.(j.le.256)))) then
     go to 205
 else if(i.eq.114) then
      zk(1,m)=-1.
 else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
&.and.(j.le.257)))) then
     go to 205
 else if(i.eq.116) then
      zk(1,m)=-1.
 else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
&.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
     go to 205
 else if(i.eq.118) then
      zk(1,m)=-1.
 else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
&.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
     go to 205
 else if(i.eq.120) then
      zk(1,m)=-1.
 else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.273)
&.and.(j.le.283)))) then
     go to 205
 else if(i.eq.122) then
      zk(1,m)=-1.
 else if((i.eq.124).and.(((j.ge.243).and.(j.le.258)).or.((j.ge.272)
&.and.(j.le.281)))) then
     go to 205
 else if(i.eq.124) then
      zk(1,m)=-1.
 else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
&.and.(j.le.279)))) then
```

```
           go to 205
       else if(i.eq.126) then
            zk(l,m)=-1.
       else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
           go to 205
       else if(i.eq.128) then
            zk(l,m)=-1.
       else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
           go to 205
       else if(i.eq.130) then
            zk(l,m)=-1.
       else if(i.eq.132) then
            zk(l,m)=-1.
           go to 205
       else if(i.eq.134) then
           zk(l,m)=-1.
           go to 205
       else if(i.eq.136) then
           zk(l,m)=-1.
           go to 205
       else if(i.eq.138) then
           zk(l,m)=-1.
           go to 205
       else if(i.eq.140) then
           zk(l,m)=-1.
           go to 205
       end if
c
205    continue
55     continue
45     continue
c
c
c      this portion of the program looks for undulations which may have
c      been created by the surface fitting subroutine, anything less than
c      or equal to 0.01 mm. equals zero.  Because of the necessity of filtering
c      especially for those simulations where there isn't much rainfall, we
c      will impose an identical filter on the simulated field as well.
c      Consequently, the plot of the simulated field will no longer be
c      untouched.
c
       do 80 i=1,nxi
       do 90 j=1,nyi
       if((zk(i,j).eq.-1.).and.(zi(i,j).eq.-1.)) then
           go to 90
       else if(zk(i,j).le.0.01) then
           zk(i,j)=0.0
       else if(zi(i,j).le.0.01) then
           zi(i,j)=0.0
       end if
90     continue
80     continue
c
c      calculate the average point depth for the field within the basin limits
c
       do 110 i=1,nxi
       do 120 j=1,nyi
c
       if(zi(i,j).eq.-1.) then
           go to 120
```

```
      end if
          a1=zi(i,j)+a1
          d1=d1+1
120   continue
110   continue
          avedepth=a1/d1
c
c
c     this portion will calculate the variance of the point depth within the
c     basin limits
c
          l=0
c
      do 130 i=1,nxi
      do 140 j=1,nyi
c
      if(zi(i,j).eq.-1.) then
          go to 140
      end if
          l=l+1
          a2=((zi(i,j)-avedepth)**2.)+a2
          a3=((zi(i,j)-avedepth)**3.)+a3
140   continue
130   continue
          vardepth=a2/(l-1)
          stdevdph=sqrt(vardepth)
          skwdpth=a3/((l-1)*(stdevdph**3.0))
      print 1030,avedepth
      print 1040,vardepth
      print 1045,skwdpth
1030  format('the average point depth=',f10.4,' mm.')
1040  format('the variance of the point depth=',f10.4,' mm ')
1045  format('the coefficeint of skewness of the point depth=',f10.4)
c
          a1=0.0
          a2=0.0
          a3=0.0
          d1=0.0
          d2=0.0
c
c     calculate the average point depth for the interpolated
c     field within the basin limits
c
      do 115 i=1,nxi
      do 125 j=1,nyi
c
      if(zk(i,j).eq.-1.) then
          go to 125
      end if
          a1=zk(i,j)+a1
          d1=d1+1
125   continue
115   continue
          aveintrp=a1/d1
c
c
c     this portion will calculate the variance of the interpolated
c     point depth within the basin limits and the coefficient of skewness
c
          l=0
```

```
          d1=0.0
          a1=0.0
          a2=0.0
          a3=0.0
c
      do 135 i=1,nxi
      do 145 j=1,nyi
c
      if(zk(i,j).eq.-1.) then
          go to 145
      end if
          l=l+1
          a2=((zk(i,j)-aveintrp)**2.)+a2
          a3=((zk(i,j)-aveintrp)**3.)+a3
145   continue
135   continue
          varintrp=a2/(l-1)
          stdevint=sqrt(vardepth)
          skwint=a3/((l-1)*(stdevdph**3.0))
      print 1035,aveintrp
      print 1055,varintrp
      print 1065,skwint
1035  format('the average intrp. point depth=',f10.4,' mm.')
1055  format('the variance of the intrp. point depth=',f10.4,' mm ')
1065  format('the coefficeint of skewness of the intrp. point depth=',f10.4)
c
          a1=0.0
          a2=0.0
          a3=0.0
          l=0
c
c     *************************************************************************
c
c     this section calculated the mean square error between the simulated
c     field and the interpolated field and the point correlation between
c     the simulated mesh point depth and the interpolated mesh point depths
c
c     ***********************************************************************
c
          d1=0.0
c
      do 830 i=1,nxi
      do 840 j=1,nyi
      if((zi(i,j).eq.-1.0).and.(zk(i,j).eq.-1.0)) then
          go to 840
      else
          toterr=toterr+((zi(i,j)-zk(i,j))**2.0)
          d1=d1+1
      end if
840   continue
830   continue
c
      do 835 i=1,nxi
      do 845 j=1,nyi
      if((zi(i,j).eq.-1.0).and.(zk(i,j).eq.-1.0)) then
          go to 845
      else if((zi(i,j).eq.-1.0).and.(zk(i,j).ne.-1.0)) then
          print,'***** there is a problem between the two fields,
     & overlay is out of allignment *****'
          print,'this message is at line 896'
```

```
          stop
       else
          totcor=totcor+((zi(i,j)-avedepth)*(zk(i,j)-aveintrp))
       end if
845    continue
835    continue
c
          err=sqrt(toterr/d1)
          correlat=totcor/(d1*sqrt(vardepth*varintrp))
       print1302,err
       print1301,correlat
c
1302   format('the standard error bet.simulat. and interpol.=',E12.6)
1301   format('the correlation bet.simulat. and interpol.=',E12.6)
c
c      reset variables equal to zero
c
          a1=0.0
          a2=0.0
c
c      ************************************************************************
c
c      write the correlation between the generated surface and the simulated
c      surface to file61
c
c      ************************************************************************
c
       write (61,1005)amonth(iii),iiday,iiyear
       write (61,1305)correlat
       print1005,amonth(iii),iiday,iiyear
       print1305,correlat
1305   format(' Correlation bet. simulat. and surf. fitted field=',f7.4//)
1160   format(//3i2)
1170   format(3(1x,f8.4))
1310   format(E16.6)
1180   format(f5.2,1x,f7.3,1x,f6.2,1x,f6.3)
c
c
          go to 35
c
4      continue
       print,'the run has been aborted'
35      continue
       end
```

```
c                         OBS_EVAL.FORTRAN
c
c       evaluate Et(Y>yi) and VARt(Y>yi) for all 8 years of data
c       read in files 50,51,52,...,57  to do this
c       write resluts to file01
c
c      This program was developed by Neil M. Fennessey at M.I.T. during
c      the course of his Master's Degree research about the Areal Distribution
c      of Rainfall
c
c       ...................................................................
c
       dimension   wet(70),tot(70),sumwet(200),sumsqwet(200),count(200)
       dimension   wetarea(70),dryarea(70),wetfrct(70,200),idepth(70,200)
       dimension   avewet(52),varwet(52)
c
       real lag1
c
       integer wet,tot
c
c
       print,'to continue, enter a 1, to abort enter a 2'
       input,f1
       if (f1.eq.1.0) then
           go to 3
       else
           go to 4
       end if
3      continue
c
c      begin to loop through the files (50,51,52,...,57)
c
       do 890 jj9=50,57
           l=jj9
           how=jj9+1920
       print,'year=',how
c
c
c      check to see how many storm days are in the file for this particular
c      year
c
        read(1,1370)n,iyr
c
        do 10 j=1,n
c
c       read file for further analysis for this particular date
c
       read(1,1310)amonth,iday,iyar
           iiday=iday
           iyear=iyar-1900
           iii=imonth-5
       read(1,1320)avedepth,vardepth
       read(1,1330)dryarea(j)
       read(1,1340)wetarea(j)
           sumwet(1)=sumwet(1)+wetarea(j)
           sumsqwet(1)=sumsqwet(1)+(wetarea(j)**2.0)
           count(1)=count(1)+1
       read(1,1300)wet(j),tot(j)
       read(1,1350)
       read(1,1360)
```

```
c
          kk=wet(j)
          k2=tot(j)
c
c
c
c     check to see is the cumulative wetted area is larger or smaller
c     than the fixed number of observations of area correlation lag
c
      if(kk.gt.31) then
          k1=k2
          b=1
      else
          k1=31
          b=2
      end if
      do 50 i=1,k1
c
c     cumulative wetted fraction is greater than 31 mm.
c
      if ((b.eq.1).and.(i.le.31)) then
      read(1,1230) idepth(j,i),wetfrct(j,i)
          sumwet(i+1)=sumwet(i+1)+wetfrct(j,i)
          sumsqwet(i+1)=sumwet(i+1)+(wetfrct(j,1)**2.0)
          count(i+1)=count(i+1)+1
c
      else if((b.eq.1).and.(i.gt.31)) then
      read(1,1240) idepth(j,i),wetfrct(j,i)
          sumwet(i+1)=sumwet(i+1)+wetfrct(j,i)
          sumsqwet(i+1)=sumwet(i+1)+(wetfrct(j,1)**2.0)
          count(i+1)=count(i+1)+1
c
c     cumulative wetted fraction is less than 31 mm.
c
      else if ((b.eq.2).and.(i.le.kk)) then
      read(1,1230) idepth(j,i),wetfrct(j,i)
c
          sumwet(i+1)=sumwet(i+1)+wetfrct(j,i)
          sumsqwet(i+1)=sumwet(i+1)+(wetfrct(j,1)**2.0)
          count(i+1)=count(i+1)+1
      else if ((b.eq.2).and.(i.gt.kk)) then
      read(1,1250)
      end if
50    continue
c
10    continue
c
c
890   continue
c
c     determine the values of Et(Y>yi) and VARt(Y>yi)
c
      do 20 i=1,51
          avewet(i)=sumwet(i)/count(i)
          varwet(i)=(sumsqwet(i)/(count(i)-1))-(avewet(i)**2.0)
20    continue
          k=51
      write (01,1001)k
1001  format(' there are ',i5,'records in this file')
c
```

```
c     write results for file01
c
      do 30 i=1,51
          idep=i-1
     write(01,1986)idep,count(i),avewet(i),varwet(i)
30    continue
c
c
1986  format(2x,i4,1x,f4.0,1x,f7.4,1x,f9.3)
c
1370  format(' there are ',i5,' storm days in this file for this year:'i4)
1360  format(7x,'Y (mm.) ',2x,'Acw/Ac (Y>y) ',10x,'v (km.) ',1x,'rho(v) ',8x,'A
     &(km.sq.) ',1x,'Gamma(A) ')
1350  format(3x,'Cumulative Wetted Fraction',5x,'Spatial Correlation',5x,
     &'Variance Function')
1340  format(' Wetted Fraction of Total Basin Area: (Acw/Ac)=',f5.3)
1330  format(' Dry Fraction of Total Basin Area: (Acd/Ac)=',f5.3)
1320  format(' point depth E(Y)=',f7.3,' Var(Y)=',f7.3)
1310  format(' Storm Day ',a4,i3,i5)
1300  format(' there are ',i3,' wetted area curve pts, there are ',i3,
     &' total data points this day')
c
c     file95 format statements
c
1250  format(37x,f3.1,5x,f5.3,7x,f6.2,7x,f5.3)
1240  format(7x,i3,7x,f5.3)
1230  format(7x,i3,7x,f5.3,15x,f3.1,5x,f5.3,7x,f6.2,7x,f5.3)
          go to 35
c
c
c
4     continue
      print,'this run has been aborted'
35    continue
      end
```

```
c                       MODEL_EVAL.FORTRAN
c
c     This program will read in the unbiased_model_parameters.data
c     file38 and will run the three mathematical models and determine
c     the moments of E(P(Y>y)) and VAR(P(Y>y)
c     written to file37 for each depth, up to a depth of 50 mm. maximum
c     the values of P(dry) will be written to file36 for each of the
c     three models for each storm
c
c     This program was developed by Neil M. Fennessey at M.I.T. during
c     the course of his Master's Degree research about the Areal Distribution
c     of Rainfall
c
c      ....................................................................
c
      dimension   summd1(500),summd2(500),summd3(500),sumsqmd1(500)
      dimension   sumsqmd2(500),sumsqmd3(500),depmd1(500),depmd2(500)
      dimension   depmd3(500),avemod1(500),avemod2(500),avemod3(500)
      dimension   varmod1(500),varmod2(500),varmod3(500)
      dimension   depmod1(500),depmod2(500),depmod3(500)
      dimension   a1(6),wet(5),depmod(500),wetmod1(500)
      dimension   wetmod2(500),wetmod3(500)
      dimension   delta(3),theta(3),drymod(3)
c
c
      external f
      common /pass2/ealpha3,parama3,ramda3
      common/pass1/flash
c
      data pi/3.14159265389793/
      api=pi
c
c     read from file38 and begin the master loop
c
       read(38,*)n
c
       do 10 k99=1,n
       print1492,k99
1492  format('k=',i4)
c
      read(38,1880)amnth,iiday,iiyr,ave,var,skwdp
      write(06,1885)amnth,iiday,iiyr
1885  format(1x,a4,i3,i5)
      read(38,1890)parama1,ealpha1,ramda1,parama2,ealpha2,ramda2,
     &parama3,ealpha3,ramda3
1880  format(1x,a4,i3,i5,3(f7.3))
1890  format(9(f9.4))
c
c
      if(var.eq.0.0) then
          var=0.0005
      end if
c
          avedepth=ave
          vardepth=var
          expecty=avedepth
          vary=vardepth
c
c     check to see if the variance equals zero in the archive file
c
```

```
      if(vardepth.eq.0) then
          go to 10
      end if
c
c
c     ************************************************************************
c
c     Develop the theoretical spatial distribution curves for Model I
c     Model II and Model III
c
c     ************************************************************************
c
c
c     this portion of the program will determine the values of other
c     model parameters: delta and theta for P(dry) and p(Y>=yi) calculations
c
c
      do 90 j=1,3
c
c     Model 1
c
          delta(1)=1.0/ealpha1
          theta(1)=(pi*ramda1)/(2*(parama1**2.0))
c
c     Model 2
c
          delta(2)=2.0/ealpha2
          theta(2)=(4*pi*ramda2)/(parama2**2.0)
c
c
c      For each of the three models, calculate p(Dry)
c
          model=j
          state=1.0
      if(j.eq.1) then
          a=0.0
          b=0.01*delta(1)
          c=theta(1)
          call mdgam(b,c,soln,ier)
          drymod(j)=soln
          dry1=soln
          wetmod1(1)=1.0-soln
          summd1(1)=summd1(1)+wetmod1(1)
          sumsqmd1(1)=sumsqmd1(1)+(wetmod1(1)**2.0)
          depmod1(1)=depmod1(1)+1
      else if(j.eq.2) then
          a=0.0
          b=0.01*delta(2)
          c=theta(2)
          call mdgam(b,c,soln,ier)
          drymod(j)=soln
          dry2=soln
          wetmod2(1)=1.0-soln
          summd2(1)=summd2(1)+wetmod2(1)
          sumsqmd2(1)=sumsqmd2(1)+(wetmod2(1)**2.0)
          depmod2(1)=depmod2(1)+1
      else if(j.eq.3) then
c
          precip=0.0
          a=precip
```

```
c
c      call external function probgt
c
          ff=probgt(precip)
c
c     check for exponential underflow
c
      if((flash.eq.1).or.(flash.eq.2)) then
      print,'flash??, depth=',i
          dry3=1.0
          go to 91
      end if
          drymod(j)=1.0-ff
          dry3=drymod(3)
          wetmod3(1)=1.0-drymod(3)
          summd3(1)=summd3(1)+wetmod3(1)
          sumsqmd3(1)=sumsqmd3(1)+(wetmod3(1)**2.0)
          depmod3(1)=depmod3(1)+1
      end if
c
c
91    continue
c
c      calculate p(Y>=yi) for each of the three models
c
c      Model 1
c
          ff=100
      if (j.eq.1)then
      do 80 i=1,50
      if(ff.lt.0.001) then
          go to 90
      end if
          a=i
          b=i*delta(j)
          c=theta(j)
c
c     call the incomplete gamma density function from the imsl library
c
          call mdgam(b,c,soln,ier)
          ff=1-soln
          kmod1=kmod1+1
          wetmod1(i+1)=ff
          summd1(i+1)=summd1(i+1)+wetmod1(i+1)
          sumsqmd1(i+1)=sumsqmd1(i+1)+(wetmod1(i+1)**2.0)
          depmod1(i+1)=depmod1(i+1)+1
80    continue
      end if
c
c     Model 2
c
          ff=100
      if (j.eq.2) then
      do 86 i=1,50
      if(ff.lt.0.001) then
          go to 90
      end if
          a=i
          b=i*delta(j)
          c=theta(j)
```

```
c
c     call the incomplete gamma density function from the imsl library
c
          call mdgam(b,c,soln,ier)
          ff=1-soln
          kmod2=kmod2+1
          wetmod2(i+1)=ff
          summd2(i+1)=summd2(i+1)+wetmod2(i+1)
          sumsqmd2(i+1)=sumsqmd2(i+1)+(wetmod2(i+1)**2.0)
          depmod2(i+1)=depmod2(i+1)+1
86    continue
      end if
c
c      Model 3
c
      if(j.eq.3) then
c
          model=3
c
          ff=100
      do 95 i=1,50
          precip=i
          a=precip
c
c      call external function probgt
c
          ff=probgt(precip)
c
c     check for exponential underflow
c
      if((flash.eq.1).or.(flash.eq.2)) then
      print,'flash??, depth=',i
          go to 90
      end if
c
      if(ff.lt.0.001) then
          go to 90
      end if
          wetmod3(i+1)=ff
          summd3(i+1)=summd3(i+1)+wetmod3(i+1)
          sumsqmd3(i+1)=sumsqmd3(i+1)+(wetmod3(i+1)**2.0)
          depmod3(i+1)=depmod3(i+1)+1
95    continue
      end if
c
90    continue
c
c     write the P(dry) results for model's I,II and III
c     to file36
c
      write(36,1940)amnth,iiday,iiyr,dry1,dry2,dry3
1940  format(a4,i3,i5,3(f7.3))
10    continue
c
c     Calculate the moments of Acw/Ac
c
      do 120 i=1,51
c
c      Model 1
c
```

```
          avemod1(i)=summd1(i)/depmod1(i)
          varmod1(i)=(sumsqmd1(i)/(depmod1(i)-1))-(avemod1(i)**2.0)
c
c      Model 2
c
          avemod2(i)=summd2(i)/depmod2(i)
          varmod2(i)=(sumsqmd2(i)/(depmod2(i)-1))-(avemod2(i)**2.0)
c
c      Model 3
c
          avemod3(i)=summd3(i)/depmod3(i)
          varmod3(i)=(sumsqmd3(i)/(depmod3(i)-1))-(avemod3(i)**2.0)
c
c     write results to file37
c
          depth=i-1
c
      write(37,1490)depth,avemod1(i),varmod1(i),depmod1(i)  ,
     &avemod2(i),varmod2(i),depmod2(i),avemod3(i),varmod3(i),depmod3(i)
c
120   continue
1490  format(2x,f3.0,3x,3(2(1x,f6.4),1x,f4.0))
c
          go to 35
4     continue
      print,'the run has been aborted'
35     continue
      end
c
c ***********************************************************
c
c  Function probgt integrates f(y) over  .000001<y<precip and
c  calculates the probability that the value of precip is
c  exceeded.  Routine calls the function f(y) which is
c  the analytical solution of the probability density of
c  storm depth from Model 3.
c
c  d01ajf -  NAG integration routine by quadrature
c
      real function probgt(precip)
      integer ier, iw(102),ifail
      real dcadre,precip,error,c,expecty,vary
      double precision f,a,precip_dp,aerr,rerr,prob,err,w(800)
      common /pass2/ealpha3,parama3,ramda3
      external f
      precip_dp = precip
      rerr =0.0
      aerr = 1.0e-5
      ifail = 1
      lw = 800
      liw = 102
c
c d01ajf -- NAG integration routine
c a      -- lower limit of integration
c f      -- probability density function for precip > 0
c
      data pi/3.14159265389793/
      api=pi
      a = .000001
      call d01ajf(f,a,precip_dp,aerr,rerr,prob,err,w,lw,iw,liw,ifail)
```

```
c       print *, 'solution' ,prob
c       print *, 'ifail ' ,ifail
      beta = 1.0/ealpha3
      eta =  (2*api*ramda3)/((beta**2.0)*(parama3**2.0))
      probgt = 1 - exp(-eta) - prob
      return
      end
c ************************************************
c ************************************************
c
c  Function f(y) is the probability density of storm depth
c  obtained from Model 3.  Routine utilizes NAG routine
c  "s18aff" which calculates the first order modified Bessel
c  function of the first kind  - "I1".
c
      double precision function f(y)
      real eta,beta
      double precision arg,s18aff,y
      common /pass1/ flash
      common /pass2/ealpha3,parama3,ramda3
      data pi/3.14159265389793/
      api=pi
      beta = 1.0/ealpha3
      eta =  (2*api*ramda3)/((beta**2.0)*(parama3**2.0))
      arg =  2*sqrt(eta*beta*y)
          flash=0
      if(arg.gt.80.1) then
          flash=1
          go to 10
      end if
          c1= -eta-beta * y
      if(c1.lt.-82) then
          flash=2
          go to 10
      end if
          c= exp(c1)
      f   =  c * sqrt((eta*beta)/y) * s18aff(arg,ifail)
10    continue
      return
      end
```