RC 20525 (90933) 8/5/96
Mathematics 157 pages

Research Report

Fortran routines for use with the method of L-moments

Version 3

J. R. M. Hosking

IBM Research Division
P. O. Box 218
Yorktown Heights, NY 10598

IBM **Research Division**
Almaden • T.J. Watson • Tokyo • Zurich

Fortran routines for use with the method of L-moments Version 3

J. R. M. Hosking

IBM Research Division
P. O. Box 218
Yorktown Heights, NY 10598

Abstract. Hosking (*J. R. Statist. Soc. B*, 1990) and Hosking and Wallis (*Regional frequency analysis: an approach based on L-moments*, Cambridge University Press, 1997) have described L-moments and L-moment ratios, quantities useful in the summarization and estimation of probability distributions. This report contains details of Fortran routines that should facilitate the use of L-moment-based metho

Contents

Index of routines

Introduction

Probability weighted moments of a random variable X with cumulative distribution function F were defined by Greenwood et al. (1979) to be the quantities

$$M_{p,r,s} = E\left[X^p\{F(X)\}^r\{1-F(X)\}^s\right] .$$

Particularly useful special cases are the probability weighted moments

$$\alpha_r = M_{1,0,r} = E[X\{1-F(X)\}^r]$$

and

$$\beta_r = M_{1,r,0} = E[X\{F(X)\}^r] .$$

Hosking (1990) defined "L-moments" to be the quantities

$$\lambda_r = E[XP^*_{r-1}\{F(X)\}]$$

where $P^*_r(\,.\,)$ is the rth shifted Legendre polynomial. L-moments and probability weighted moments are related by

$$\lambda_{r+1} = \sum_{k=0}^{r} p^*_{r,k}\beta_k \tag{1}$$

where

$$p^*_{r,k} = (-1)^{r-k}\binom{r}{k}\binom{r+k}{k} .$$

L-moment ratios are the quantities

$$\tau_r = \lambda_r/\lambda_2 .$$

The foregoing quantities are defined for a probability distribution, but in practice must often be estimated from a finite sample. Let $x_{1:n} \le x_{2:n} \le \ldots \le x_{n:n}$ be the ordered sample. Let

$$a_r = n^{-1}\sum_{j=1}^{n}\frac{(n-j)(n-j-1)\ldots(n-j-r+1)}{(n-1)(n-2)\ldots(n-r)}\,x_{j:n} ,$$

$$b_r = n^{-1}\sum_{j=1}^{n}\frac{(j-1)(j-2)\ldots(j-r)}{(n-1)(n-2)\ldots(n-r)}\,x_{j:n} , \tag{2}$$

$$\ell_{r+1} = \sum_{k=0}^{r} p^*_{r,k}b_k . \tag{3}$$

Then a_r, b_r and ℓ_r are unbiased estimators of α_r, β_r and λ_r respectively. The estimator $t_r = \ell_r/\ell_2$ of τ_r is consistent but not unbiased. Landwehr et al. (1979a, 1979b) also considered "plotting-position" estimators of probability weighted moments. Let $p_{j:n}$ be a plotting position, i.e. a distribution-free estimator of $F(x_{j:n})$. Then α_r, β_r, λ_r and τ_r are estimated respectively by

$$\tilde{\alpha}_r = n^{-1} \sum_{j=1}^{n} (1 - p_{j:n})^r \, x_{j:n} \,,$$

$$\tilde{\beta}_r = n^{-1} \sum_{j=1}^{n} p_{j:n}^r \, x_{j:n} \,, \tag{4}$$

$$\tilde{\lambda}_r = n^{-1} \sum_{j=1}^{n} P_{r-1}^*(p_{j:n}) \, x_{j:n} \,, \tag{5}$$

$$\tilde{\tau}_r = \tilde{\lambda}_r / \tilde{\lambda}_2 \,.$$

Plotting-position estimators are consistent, but can have large bias and cannot be generally recommended (Hosking and Wallis, 1995; Hosking and Wallis, 1997, sec. 2.8).

Hosking (1990) gives an exposition of the theory of L-moments and L-moment ratios, and gives, for many distributions in common use, expressions for the L-moments of the distributions and algorithms for estimating the parameters of the distributions by equating sample and population L-moments (the "method of L-moments"). This report contains details of Fortran routines that should facilitate the use of L-moment-based methods. The following routines are included.

- For each of the eleven distributions listed in Table 1 on page 5, routines to evaluate the cumulative distribution function and quantile function of the distribution, to calculate the L-moments given the parameters and to calculate the parameters given the low-order L-moments.
- Given a sample of data, routines to calculate estimates of the probability weighted moments and L-moments of the distribution from which the sample was drawn.
- Routines useful in connection with index-flood procedures for regional frequency analysis (Hosking and Wallis, 1997). Three routines are concerned with cluster analysis, and do not explicitly involve L-moments. One routine calculates a weighted average of L-moment ratios estimated from data measured at different sites, an essential part of L-moment-based index-flood procedures such as the regional L-moment algorithm described in Hosking and Wallis (1997, sec. 6.2). One other routine calculates diagnostic statistics useful in deciding whether a set of samples may be regarded as having similar frequency distributions: these statistics are described more fully in Hosking and Wallis (1997, chaps. 3–5).
- Programs to apply the regional L-moment algorithm and to calculate the diagnostic statistics for a regional data set.
- Auxiliary routines called by some of the earlier routines: mathematical functions, a pseudo-random number generator and a routine to sort a data array into ascending order.

The routines are written in Fortran-77. The routines aim for accuracy of about 8 significant digits. For this reason double-precision arithmetic is used throughout. Single precision may be adequate on machines that use 60 bits to represent a single-precision floating-point number.

Caveat: Statements about the accuracy of individual routines are not unconditional guarantees, but are valid over wide ranges of values of the input parameters including those most frequently encountered in practice. Decreased accuracy is particularly likely when the shape parameter of distribution is close to zero (less than 10^{-4} in absolute value), when the shape parameter is close to the end of the range of its permissible values, or is close to a value at which the mean of the distribution does not exist, or when the shape parameter is large and positive.

Routines for specific distributions

Routines are provided for the ten distributions listed in Table 1. The following routines are provided for each distribution.

`FUNCTION CDFxxx`	The cumulative distribution function of the distribution.
`FUNCTION QUAxxx`	The quantile function (inverse cumulative distribution function) of the distribution.
`SUBROUTINE LMRxxx`	Calculates the L-moment ratios of the distribution given its parameters.
`SUBROUTINE PELxxx`	Calculates the parameters of the distribution given its L-moments. When the L-moments are the sample L-moments of a set of data, then the resulting parameters are of course the "method of L-moments" estimates of the parameters.

Here `xxx` is the 3-letter code used to identify the distribution, as given in Table 1. For example the cumulative distribution function of the gamma distribution is `FUNCTION CDFGAM`.

Most of these distributions are standard and are as defined in Hosking (1990, Table 1).

The generalized Normal distribution combines lognormal distributions (which have positive skewness), reflected lognormal distributions (which have negative skewness) and the Normal distribution (which has zero skewness). The three-parameter lognormal distribution, with cumulative distribution function

$$F(x) = \Phi[\{\log(x-\zeta) - \mu\}/\sigma] \,,$$

is a generalized Normal distribution with parameters $k = -\sigma$, $\alpha = \sigma e^{\mu}$, $\xi = \zeta + e^{\mu}$.

The Pearson type III distribution combines gamma distributions (which have positive skewness), reflected gamma distributions (which have negative skewness) and the Normal distribution (which has zero skewness). We parametrize the Pearson type III distribution by its first three (conventional) moments: the mean μ, the standard deviation σ and the skewness γ. The relationship between these parameters and those of the gamma distribution is as follows. Let X be a random variable with a Pearson type III distribution with parameters μ, σ and γ. If $\gamma > 0$, then $X - \mu + 2\sigma/\gamma$ has a gamma distribution with parameters $\alpha = 4/\gamma^2$, $\beta = \sigma\gamma/2$. If $\gamma = 0$, then X has a Normal distribution with mean μ and standard deviation σ. If $\gamma < 0$, then $-X + \mu - 2\sigma/\gamma$ has a gamma distribution with parameters $\alpha = 4/\gamma^2$, $\beta = |\sigma\gamma/2|$.

Table 1: Distributions for which L-moment subroutines are provided.

Distribution	Code	Number of parameters	Parameters	$F(x)$, $x(F)$
Exponential	EXP	2	$\xi\ \alpha$	$F = 1 - \exp\{-(x-\xi)/\alpha\}$ $x = \xi - \alpha\log(1-F)$
Gamma	GAM	2	$\alpha\ \beta$	$F = G(x/\beta, \alpha)$ $x(F)$ not explicitly defined
Generalized extreme-value	GEV	3	$\xi\ \alpha\ k$	$F = \exp[-\{1-k(x-\xi)/\alpha\}^{1/k}]$ $x = \xi + \alpha\{1-(-\log F)^k\}/k$
Generalized logistic	GLO	3	$\xi\ \alpha\ k$	$F = 1/[1+\{1-k(x-\xi)/\alpha\}^{1/k}]$ $x = \xi + \alpha[1-\{(1-F)/F\}^k]/k$
Generalized Normal	GNO	3	$\xi\ \alpha\ k$	$F = \Phi[-k^{-1}\log\{1-k(x-\xi)/\alpha\}]$ $x(F)$ not explicitly defined
Generalized Pareto	GPA	3	$\xi\ \alpha\ k$	$F = 1-\{1-k(x-\xi)/\alpha\}^{1/k}$ $x = \xi + \alpha\{1-(1-F)^k\}/k$
Gumbel	GUM	2	$\xi\ \alpha$	$F = \exp[-\exp\{-(x-\xi)/\alpha\}]$ $x = \xi + \alpha\log(-\log F)$
Kappa	KAP	4	$\xi\ \alpha\ k\ h$	$F = [1-h\{1-k(x-\xi)/\alpha\}^{1/k}]^{1/h}$ $x = \xi + \alpha[1-\{(1-F)^h/h\}^k]/k$
Normal	NOR	2	$\mu\ \sigma$	$F = \Phi\{(x-\mu)/\sigma\}$ $x(F)$ not explicitly defined
Pearson type III	PE3	3	$\mu\ \sigma\ \gamma$	$F = G\big((x-\mu+2\sigma/\gamma)/\lvert\frac{1}{2}\sigma\gamma\rvert, 4/\gamma^2\big),\quad \gamma>0$ $F = 1-G\big(-(x-\mu+2\sigma/\gamma)/\lvert\frac{1}{2}\sigma\gamma\rvert, 4/\gamma^2\big),\ \gamma<0$ $x(F)$ not explicitly defined
Wakeby	WAK	5	$\xi\ \alpha\ \beta\ \gamma\ \delta$	$F(x)$ not explicitly defined $x = \xi + \dfrac{\alpha}{\beta}\{1-(1-F)^\beta\} - \dfrac{\gamma}{\delta}\{1-(1-F)^{-\delta}\}$

$G(x,\alpha) = \{\Gamma(\alpha)\}^{-1}\int_0^x t^{\alpha-1}e^{-t}\,dt$ is the incomplete gamma integral.

$\Phi(x) = (2\pi)^{-1/2}\int_{-\infty}^x \exp(-t^2/2)\,dt$ is the standard Normal cumulative distribution function.

Cumulative distribution functions

```
DOUBLE PRECISION FUNCTION CDFEXP(X,PARA)
DOUBLE PRECISION FUNCTION CDFGAM(X,PARA)
DOUBLE PRECISION FUNCTION CDFGEV(X,PARA)
DOUBLE PRECISION FUNCTION CDFGLO(X,PARA)
DOUBLE PRECISION FUNCTION CDFGNO(X,PARA)
DOUBLE PRECISION FUNCTION CDFGPA(X,PARA)
DOUBLE PRECISION FUNCTION CDFGUM(X,PARA)
DOUBLE PRECISION FUNCTION CDFKAP(X,PARA)
DOUBLE PRECISION FUNCTION CDFNOR(X,PARA)
DOUBLE PRECISION FUNCTION CDFPE3(X,PARA)
DOUBLE PRECISION FUNCTION CDFWAK(X,PARA)
```

Formal parameters

`X` [input; double precision] The argument of the function.

`PARA` [input; double-precision array of length equal to the number of parameters of the distribution] The parameters of the distribution.

Numerical method

`CDFGAM` and `CDFPE3` are calculated from the incomplete gamma integral, auxiliary routine `GAMIND`. `CDFGNO` and `CDFNOR` are calculated from the error function, supplied as the auxiliary routine `DERF`. Routine `CDFWAK`, given an argument x_0 of the Wakeby cumulative distribution function., solves the equation $x(F) = x_0$ for F using Halley's method, the second-order analogue of Newton-Raphson iteration (Churchhouse, 1981, pp. 151-152; Huh, 1986). The other cumulative distribution functions have explicit analytical expressions, given in Table 1: the routines straightforwardly evaluate these expressions.

Accuracy

In principle (but see the caveat on page 3) all routines are accurate to at least 8 decimal places.

Quantile functions

```
DOUBLE PRECISION FUNCTION QUAEXP(F,PARA)
DOUBLE PRECISION FUNCTION QUAGAM(F,PARA)
DOUBLE PRECISION FUNCTION QUAGEV(F,PARA)
DOUBLE PRECISION FUNCTION QUAGLO(F,PARA)
DOUBLE PRECISION FUNCTION QUAGNO(F,PARA)
DOUBLE PRECISION FUNCTION QUAGPA(F,PARA)
DOUBLE PRECISION FUNCTION QUAGUM(F,PARA)
DOUBLE PRECISION FUNCTION QUAKAP(F,PARA)
DOUBLE PRECISION FUNCTION QUANOR(F,PARA)
DOUBLE PRECISION FUNCTION QUAPE3(F,PARA)
DOUBLE PRECISION FUNCTION QUAWAK(F,PARA)
```

Formal parameters

`F` [input; double precision] The argument of the function.

`PARA` [input; double-precision array of length equal to the number of parameters of the distribution] The parameters of the distribution.

Numerical method

Routine `QUAGAM`, given an argument F_0 of the Gamma quantile function, solves the equation $G(x,\alpha)=F_0$ for x using Newton-Raphson iteration. Routines `QUAGNO` and `QUANOR` use the quantile function of the standard Normal distribution, supplied as the auxiliary routine `QUASTN`, and transform it to get the required quantile function. The other quantile functions have explicit analytical expressions, given in Table 1: the routines straightforwardly evaluate these expressions.

Accuracy

Let F be the argument of the quantile function. In principle (but see the caveat on page 3) each routine returns a quantile $x(F_0)$ corresponding to an argument F_0 that satisfies $|F-F_0|\leq 10^{-8}$. In other words, the error of the approximation is equivalent to a perturbation of less than 10^{-8} in the argument of the function.

L-moment ratios

```
SUBROUTINE LMREXP(PARA,XMOM,NMOM)
SUBROUTINE LMRGAM(PARA,XMOM,NMOM)
SUBROUTINE LMRGEV(PARA,XMOM,NMOM)
SUBROUTINE LMRGLO(PARA,XMOM,NMOM)
SUBROUTINE LMRGNO(PARA,XMOM,NMOM)
SUBROUTINE LMRGPA(PARA,XMOM,NMOM)
SUBROUTINE LMRGUM(PARA,XMOM,NMOM)
SUBROUTINE LMRKAP(PARA,XMOM,NMOM)
SUBROUTINE LMRNOR(PARA,XMOM,NMOM)
SUBROUTINE LMRPE3(PARA,XMOM,NMOM)
SUBROUTINE LMRWAK(PARA,XMOM,NMOM)
```

Formal parameters

`PARA` [input; double-precision array of length] equal to the number of parameters of the distribution] The parameters of the distribution.

`XMOM` [output; double-precision array of length `NMOM`] The L-moment ratios of the distribution, in the order λ_1, λ_2, τ_3, τ_4, … .

`NMOM` [input; integer] The number of L-moments to be found. In routines `LMRGAM` and `LMRPE3`, `NMOM` may be at most 4; in the other routines it may be at most 20.

Numerical method

`LMREXP` L-moments and L-moment ratios are calculated using the expressions in Hosking and Wallis (1997, Appendix A.2).

`LMRGAM` L-moments and L-moment ratios are calculated using expressions in Hosking and Wallis (1997, Appendix A.9): analytic expressions for λ_1 and λ_2, rational-function approximations for τ_3 and τ_4.

`LMRGEV` Probability weighted moments β_k are calculated using eq. (9) of Hosking et al. (1985). Each λ_k is then calculated from β_k and the λ_j, $j=1,...,k-1$, using the formula

$$\lambda_{k+1} = \binom{2k}{k}\beta_k - \sum_{j=1}^{k} \frac{(2j-1)(2k)!}{(k+j)!\,(k+1-j)!}\,\lambda_j\,,$$

easily proved by induction using (1).

`LMRGLO` The first two L-moments are calculated using the expressions in Hosking and Wallis (1997, Appendix A.7). Higher-order moments are expressed as polynomials in the shape parameter k.

`LMRGNO` The first two L-moments are calculated using the expressions in Hosking and Wallis (1997, Appendix A.8). For $k=0$, higher-order moments are those of the

Normal distribution. For $k \neq 0$, higher-order moments are found by numerical evaluation of the integral

$$\int_{-\infty}^{\infty} \exp\{-(x+k/\sqrt{2})^2\} P_{r-1}(\operatorname{erf} x)\, dx\,,$$

where $P_{r-1}(\,.\,)$ is the $(r-1)$th Legendre polynomial and erf is the error function; this integral is, apart from a linear transformation, the rth L-moment of the distribution.

`LMRGPA` The routine evaluates the explicit formulas given by Hosking and Wallis (1997, Appendix A.5). Higher-order L-moments (λ_r, $r \geq 5$) are computed from the general expression

$$\lambda_r = \frac{\alpha\Gamma(1+k)\Gamma(r-1-k)}{\Gamma(1-k)\Gamma(r+1+k)}\,,$$

which can be derived from the expression for α_r given by Hosking and Wallis (1987, p. 341).

`LMRGUM` The first two L-moments are calculated using the expressions in Hosking and Wallis (1997, Appendix A.3). Higher-order moments are the same for all Gumbel distributions. They are initialized in a data statement and copied into array `XMOM` as required.

`LMRKAP` Probability weighted moments β_k are calculated using eq. (12) of Hosking (1994). The λ_k are calculated from the β_k as in routine `LMRGEV`.

`LMRNOR` The first two L-moments are calculated using the expressions in Hosking and Wallis (1997, Appendix A.4). Higher-order moments are the same for all Normal distributions. They are initialized in a data statement and copied into array `XMOM` as required.

`LMRPE3` L-moments are calculated from those of the gamma distribution, of which the Pearson type III is a reparametrization. The numerical method is the same as that of routine `LMRGAM`.

`LMRWAK` The routine evaluates the explicit formulas given by Hosking and Wallis (1997, Appendix A.11). Higher-order L-moments (λ_r, $r \geq 5$) are computed from the general expression

$$\lambda_r = \frac{\alpha\Gamma(1+\beta)\Gamma(r-1-\beta)}{\Gamma(1-\beta)\Gamma(r+1+\beta)} + \frac{\gamma\Gamma(1-\delta)\Gamma(r-1+\delta)}{\Gamma(1+\delta)\Gamma(r+1-\delta)}\,,$$

which can be derived from the expression for α_r given by Kotz et al. (1988, p. 514).

Accuracy

In principle (but see the caveat on page 3) all routines are accurate to at least 8 significant digits for λ_1 and λ_2. For higher-order moments, routines `LMREXP`, `LMRGLO`, `LMRGPA`, `LMRGUM`, `LMRNOR` and `LMRWAK` are accurate to at least 8 decimal places. Routines `LMRGAM`, `LMRGNO` and `LMRPE3` are accurate to 6 decimal places. Routines `LMRGEV` and `LMRKAP` become less and less accurate as the order of the moment increases; their accuracy for τ_r is at least 8 decimal places if $r \leq 12$, at least 4 decimal places if $r \leq 18$.

Parameters

```
SUBROUTINE PELEXP(XMOM,PARA)
SUBROUTINE PELGAM(XMOM,PARA)
SUBROUTINE PELGEV(XMOM,PARA)
SUBROUTINE PELGLO(XMOM,PARA)
SUBROUTINE PELGNO(XMOM,PARA)
SUBROUTINE PELGPA(XMOM,PARA)
SUBROUTINE PELGUM(XMOM,PARA)
SUBROUTINE PELKAP(XMOM,PARA,IFAIL)
SUBROUTINE PELNOR(XMOM,PARA)
SUBROUTINE PELPE3(XMOM,PARA)
SUBROUTINE PELWAK(XMOM,PARA,IFAIL)
```

Formal parameters

`XMOM` [input; double-precision array of length equal to the number of parameters of the distribution] The *L*-moment ratios λ_1, λ_2, τ_3, τ_4, ...

`PARA` [input; double-precision array of length equal to the number of parameters of the distribution] The parameters of the distribution.

`IFAIL` [output; integer] Used only by routines `PELKAP` and `PELWAK`. For explanation see "Numerical method" below.

Numerical method

Except where indicated below, the routines evaluate expressions for the parameters in terms of the *L*-moments given in the appendix of Hosking and Wallis (1997).

`PELGAM` Rational approximation is used to express α, the shape parameter of the gamma distribution, as a function of the *L*-CV $\tau = \lambda_2/\lambda_1$.

`PELGEV` For $-0.8 \le \tau_3 < 1$, the shape parameter k is estimated from rational-function approximations given by Donaldson (1996). This approximation has absolute accuracy better than 3×10^{-7}. If $\tau_3 < -0.8$, Donaldson's approximation is further refined by Newton-Raphson iteration until similar accuracy is achieved. Parameters α and ξ are then calculated as in Hosking and Wallis (1997, Appendix A.6).

`PELKAP` The method is as described by Hosking (1994). Newton-Raphson iteration is used to solve the equations that express τ_3 and τ_4 as functions of k and h, and α and ξ are calculated as functions of λ_1, λ_2, k and h. To ensure a 1-1 relationship between parameters and *L*-moments, the parameter space is restricted as in Hosking (1994, eq. (13)):

$$k > -1; \qquad \text{if } h < 0 \text{ then } hk > -1; \qquad h > -1; \qquad k + 0.725h > -1\,.$$

On exit, parameter IFAIL is set as follows.

0 Successful exit.

1 L-moments invalid: $\lambda_2 \leq 0$, or τ_3 and τ_4 do not lie within the set of feasible values derived by Hosking (1990, eq. (2.7)).

2 Routine was called with $\tau_4 \geq (1+5\tau_3^2)/6$. This implies that the L-moments are inconsistent with a kappa distribution with parameters restricted as above.

3 Newton-Raphson iteration did not converge.

4 Newton-Raphson iteration reached a point from which no further progress could be made.

5 Newton-Raphson iteration encountered numerical difficulties: overflow would have been likely to occur.

6 Iteration for h and k converged, but overflow would have occurred while calculating ξ and α.

PELWAK The basic method is that of Landwehr et al. (1979b), but the distribution is differently parametrized and the calculations are expressed in terms of L-moments rather than probability weighted moments. The computations are given by Hosking and Wallis (1997, Appendix A.11). First a solution is sought in which all five parameters are estimated, as functions of the first five L-moments. If no solution is found, ξ is set equal to zero and a solution is sought in which the other four parameters are estimated as functions of the first four L-moments. If this too is unsuccessful, then a generalized Pareto distribution is fitted instead, using the first three L-moments and the same method as routine PELGPA.

On exit, parameter IFAIL is set as follows.

0 Successful exit; all five parameters estimated.

1 Solution could only be found by setting $\xi = 0$.

2 Solution could only be found by fitting a generalized Pareto distribution.

3 L-moments invalid: $\lambda_2 \leq 0$, or $|\tau_r| \geq 1$ for some $r \geq 3$.

Accuracy

For shape parameters, routines PELGLO and PELGPA are accurate to at least 8 decimal places; PELGEV is accurate to within 3×10^{-7}; PELGNO has relative accuracy better than 2.5×10^{-6}; PELKAP is accurate to within 5×10^{-6}; PELGAM and PELPE3 have relative accuracy better than 5×10^{-5}. For any of these routines, let ϵ be the relative error of estimation of the shape parameter; for routines PELGUM and PELNOR, let $\epsilon = 10^{-8}$; then the estimate of the scale parameter α (or β, or σ) has relative error approximately ϵ, and the estimate of the location parameter ξ has absolute error approximately $\epsilon\alpha$.

Routine PELWAK is usually accurate to within 10^{-6}, but can be much less accurate for certain combinations of input parameters. In such cases, however, the quantiles of the fitted distribution are usually still reliable to within an error equivalent to a perturbation of less than 10^{-6} in the argument of the quantile function.

Routines for computing sample L-moments

Three routines are provided for computing sample L-moments and probability weighted moments of a data sample.

`SUBROUTINE SAMLMR`	Calculates the sample L-moment ratios of a data set, indirectly via the probability weighted moments.
`SUBROUTINE SAMLMU`	Calculates the "unbiased" sample L-moment ratios of a data set by a more direct method.
`SUBROUTINE SAMPWM`	Calculates the sample probability weighted moments ratios of a data set.

SUBROUTINE SAMLMR(X,N,XMOM,NMOM,A,B)

Purpose

Calculates the sample L-moment ratios of a data set.

Formal parameters

X	[input; double-precision array of length N] The data, in ascending order.
N	[input; integer] Sample size.
XMOM	[output; double-precision array of length NMOM] The sample L-moment ratios, in the order ℓ_1, ℓ_2, t_3, t_4, … .
NMOM	[input; integer] The number of L-moments to be found. At most min(N, 20).
A,B	[input; double precision] Parameters of plotting position. If A and B are both set to zero, "unbiased" estimates of L-moments are found. Otherwise plotting-position estimates are found, based on the plotting position $p_{j:n} = (j + \mathtt{A})/(n + \mathtt{B})$ for the jth smallest of n observations. For example, A=-0.35D0 and B=0.0D0 yields the estimates recommended by Hosking et al. (1985) for use with the generalized extreme-value distribution.

Numerical method

The routine explicitly computes probability weighted moments using (2) or (4), and from them the sample L-moments using (3) or (5).

```
SUBROUTINE SAMLMU(X,N,XMOM,NMOM)
```

Purpose

Calculates the "unbiased" sample L-moment ratios of a data set.

Formal parameters

`X`	[input; double-precision array of length `N`] The data, in ascending order.
`N`	[input; integer] Sample size.
`XMOM`	[output; double-precision array of length `NMOM`] The sample L-moment ratios, in the order ℓ_1, ℓ_2, t_3, t_4, $\dots$.
`NMOM`	[input; integer] The number of L-moments to be found. At most min(`N`, 100).

Numerical method

From (2) and (3), ℓ_r is a linear combination of the ordered sample values $x_{1:n}, \dots, x_{n:n}$, and we can write

$$\ell_r = n^{-1} \sum_{j=1}^{n} w_{j:n}^{(r)} x_{j:n} .$$

The weights $w_{j:n}^{(r)}$ are related to the "discrete Legendre polynomials" defined by Neuman and Schonbach (1974). In Neuman and Schonbach's notation, the weight $w_{j:n}^{(r)}$ is the discrete Legendre polynomial $(-1)^r P_{r-1}(j-1, n-1)$. The weights can be computed as $w_{j:n}^{(1)} = 1$, $w_{j:n}^{(2)} = 2(j-1)/(n-1) - 1$, and, for $r \geq 3$,

$$(r-1)(n-r+1) w_{j:n}^{(r)} = (2r-3)(2j-n-1) w_{j:n}^{(r-1)} - (r-2)(n+r-2) w_{j:n}^{(r-2)};$$

this recurrence relation is equivalent to equation (9) of Neuman and Schonbach (1974). Routine `SAMLMU` evaluates sample L-moments using this recurrence, and achieves additional efficiency by making use of the symmetry in the weights, $w_{j:n}^{(r)} = (-1)^{r-1} w_{n+1-j:n}^{(r)}$.

This routine is generally equivalent to `SAMLMR`, but is particularly suitable when many moments are to be calculated. Routine `SAMLMU` is accurate to 8 decimal places for about the first 50 L-moment ratios, whereas `SAMLMR` achieves similar accuracy for only the first 12 L-moment ratios. The routines require approximately the same amount of computing time, but `SAMLMU` can be a little slower than `SAMLMR` when `NMOM` is large or odd.

SUBROUTINE SAMPWM(X,N,XMOM,NMOM,A,B,KIND)

Purpose

Calculates the sample probability weighted moments of a data set.

Formal parameters

X [input; double-precision array of length N] The data, in ascending order.

N [input; integer] Sample size.

XMOM [output; double-precision array of length NMOM] The sample probability weighted moments, in the order (if KIND=1) a_0, a_1, ..., or (if KIND=2) b_0, b_1, ...

NMOM [input; integer] The number of probability weighted moments to be found. At most min(N, 20).

A,B [input; double precision] Parameters of plotting position. If A and B are both set to zero, unbiased estimates of probability weighted moments are found. Otherwise plotting-position estimates are found, based on the plotting position $p_{j:n} = (j + \mathtt{A})/(n + \mathtt{B})$ for the jth smallest of n observations. For example, A=-0.35D0 and B=0.0D0 yields the estimates recommended by Hosking et al. (1985) for use with the generalized extreme-value distribution.

KIND [input; integer] Indicates the kind of probability weighted moments to be found. If KIND=1, the routine estimates the probability weighted moments $\alpha_r = E[X\{1 - F(X)\}^r]$. If KIND=2, the routine estimates the probability weighted moments $\beta_r = E[X\{F(X)\}^r]$.

Routines for regional frequency analysis

Three routines are provided for computing sample L-moments and probability weighted moments of a data sample.

`SUBROUTINE CLUAGG` Performs cluster analysis by one of several agglomerative hierarchical methods: single-link, complete-link, and Ward's procedure.

`SUBROUTINE CLUINF` Obtains information about clusters arising from agglomerative hierarchical clustering.

`SUBROUTINE CLUKM` Performs cluster analysis by the K-means algorithm.

`SUBROUTINE REGLMR` Calculates regional weighted averages of L-moment ratios.

`SUBROUTINE REGTST` Calculates statistics useful in regional frequency analysis. The statistics are based on the at-site and regional average sample L-moments.

- Discordancy measures for individual sites in a region. These are intended to assess whether the sample L-moments at a site are markedly different from the regional average.
- Heterogeneity measures for a region. These are intended to assess whether the between-site variation in the sample L-moments is consistent with what would be expected of a homogeneous region.
- Goodness-of-fit measures, based on the regional average L-moment ratios. These are intended to assess whether any of five candidate distributions gives an adequate fit to the data.

The statistics are described in more detail by Hosking and Wallis (1997, chaps. 3–5).

`PROGRAM XCLUST` Program to illustrate the use of the cluster analysis routines.

`PROGRAM XFIT` Program to illustrate the use of routine `REGLMR`. The program performs frequency analysis of a regional data set using an index-flood procedure and the method of L-moments.

`PROGRAM XTEST` Program to illustrate the use of routine `REGTST`.

`PROGRAM XSIM` Program to illustrate the use of simulation to find the accuracy of estimates obtained from regional frequency analysis.

SUBROUTINE CLUAGG(METHOD,X,NX,N,NATT,MERGE,WGSS,IWORK,WORK,NW)

Purpose

Performs cluster analysis by one of several agglomerative hierarchical methods: single-link, complete-link, and Ward's procedure.

Formal parameters

METHOD [input; integer] Defines which clustering method is used. Should be set to 1 for single-link clustering, 2 for complete-link clustering, 3 for Ward's procedure.

X [input; double-precision array of dimension (NX,NATT)] X(I,J) should contain the Jth attribute for the Ith data point.

NX [input; integer] The first dimension of array X, as declared in the calling program.

N [input; integer] Number of data points.

NATT [input; integer] Number of attributes for each data point.

MERGE [output; integer array of dimension (2,N)] MERGE(1,I) and MERGE(2,I) are the labels of the clusters merged at the Ith stage. MERGE(1,N) and MERGE(2,N) are not used.

WGSS [output; double-precision array of length N] The Ith element is the within-cluster sum of squares after the Ith merge.

IWORK [local; integer array of length N] Work array.

WORK [local; double-precision array of length NW] Work array.

NW [input; integer] Length of array W. Must be at least N*(N-1)/2.

Method

In agglomerative hierarchical clustering, there are initially N clusters, each containing one data point, labeled 1 through N in the same order as the data points. At each stage of clustering, two clusters are merged. Their labels are saved in the MERGE array. The smaller of the two labels is used as the label of the merged cluster. After the Mth stage of clustering there are N − M clusters. To find which data points belong to which clusters, use routine CLUINF.

Different methods correspond to different criteria for which clusters should be merged at each stage. The following methods are implemented.

- Single-link clustering: the distance between two clusters A and B is defined to be the minimum of the Euclidean distances between pairs of points with one point in A and one in B. At each stage, the two clusters separated by the smallest distance are merged.

- Complete-link clustering: the distance between two clusters A and B is defined to be the maximum of the Euclidean distances between pairs of points with one point in A and one in B. At each stage, the two clusters separated by the smallest distance are merged.

- Ward's method: at each stage, the clusters that are merged are chosen to minimize the total within-cluster sum of squared deviations of each attribute about the cluster mean.

This routine makes no claim to efficiency in either computing time or storage requirements. It is computationally practical for small and moderate-sized problems. For large problems, more efficient clustering programs may be preferred, e.g. the `hcon2` program of Murtagh (1985), available from StatLib at `http://lib.stat.cmu.edu/multi/hcon2`.

`SUBROUTINE CLUINF(NCLUST,N,MERGE,IASSGN,LIST,NUM)`

Purpose

Obtains information about clusters arising from agglomerative hierarchical clustering.

Agglomerative hierarchical clustering procedures typically produce a list of the clusters merged at each stage of the clustering. This routine uses this list to construct arrays that explicitly show which cluster a given data point belongs to, and which data points belong to a given cluster.

Formal parameters

`NCLUST` [input; integer] Number of clusters.

`N` [input; integer] Number of data points.

`MERGE` [input; integer array of dimension `(2,N)`] `MERGE(1,I)` and `MERGE(2,I)` identify the clusters merged at the `I`th step. This is the array `MERGE` returned by routine `CLUAGG`, and should be left unchanged after exit from that routine.

`IASSGN` [output; integer array of length `N`] Its `I`th element is the number of the cluster to which the `I`th data point belongs.

`LIST` [output; integer array of length `N`] Contains the data points in cluster 1, followed by the data points in cluster 2, etc. Data points in each cluster are listed in increasing order. The last data point in each cluster is indicated by a negative number. See the example below.

`NUM` [output; integer array of length `NCLUST`] Contains the number of data points in each cluster.

Cluster numbers used in arrays `IASSGN`, `LIST` and `NUM` range from 1 to `NCLUST`. They are arbitrary, but are uniquely defined: cluster 1 contains data point 1, cluster `M` (`M` ≥ 2) contains data point `J`, where `J=MERGE(2,N-M)`.

Example of the `LIST` array. Suppose that there are 8 data points and 3 clusters, and that the elements of the `LIST` array are 1, −4, 3, 6, −8, 2, 5, −7. Then the clusters are as follows: cluster 1 contains points 1 and 4; cluster 2 contains points 3, 6 and 8; cluster 3 contains points 2, 5 and 7.

```
SUBROUTINE CLUKM(X,NX,N,NATT,NCLUST,IASSGN,LIST,NUM,SS,MAXIT,IWORK,RW,NW)
```

Purpose

Performs cluster analysis by the K-means algorithm. The aim of the K-means algorithm is to partition a set of points into K clusters so that the within-cluster sum of squared distances from the points to their cluster centers is minimized. Except for very small data sets, it is not practical to require that the sum of squares is minimized over all possible partitions. Instead, a local optimum is sought such that no movement of a point from one cluster to another will reduce the within-cluster sum of squares.

Formal parameters

`X`	[input; double-precision array of dimension `(NX,NATT)`] `X(I,J)` should contain the `J`th attribute for the `I`th data point.
`NX`	[input; integer] The first dimension of array `X`, as declared in the calling program.
`N`	[input; integer] Number of data points.
`NATT`	[input; integer] Number of attributes for each data point.
`NCLUST`	[input; integer] Number of clusters.
`IASSGN`	[input and output; integer array of length `N`] On entry, should contain the initial assignment of sites to clusters. On exit, contains the final assignment. The `I`th element of the array contains the label of the cluster to which the `I`th data point belongs. Labels must be between 1 and `NCLUST`, and each of the values 1 through `NCLUST` must occur at least once.
`LIST`	[output; integer array of length `N`] Contains the data points in cluster 1, followed by the data points in cluster 2, etc. Data points in each cluster are listed in increasing order. The last data point in each cluster is indicated by a negative number.
`NUM`	[output; integer array of length `NCLUST`] Contains the number of data points in each cluster.
`SS`	[output; double precision] Within-group sum of squares of the final clusters.
`MAXIT`	[input; integer] Maximum number of iterations for the K-means clustering algorithm
`IWORK`	[local; integer array of length `NCLUST*3`] Work array.
`RW`	[local; real array of length `NW`] Work array. N.B. this array is of type real, not double precision!
`NW`	[input; integer] Length of array `RW`. Must be at least `(N+NCLUST)*(NATT+1)+2*NCLUST`.

Method

This routine is a front end to Applied Statistics Algorithm AS136 (Hartigan and Wong, 1979). Algorithm AS136 is not included in this package, but is available from StatLib at `http://stat.lib.cmu.edu/apstat/136`.

`SUBROUTINE REGLMR(NSITE,NMOM,NXMOM,XMOM,WEIGHT,RMOM)`

Purpose

Calculates regional weighted averages of L-moment ratios, as in Hosking and Wallis (1997, sec. 6.2).

Formal parameters

`NSITE` [input; integer] The number of sites in the region.

`NMOM` [input; integer] The number of L-moments to be found.

`NXMOM` [input; integer] The first dimension of array `XMOM`, as declared in the calling program.

`XMOM` [input; double-precision array of dimension `(NXMOM,NSITE)`] `XMOM(K,I)` should contain the `K`th L-moment ratio for site `I`, as returned by a previous call to routine `SAMLMR`.

`WEIGHT` [input; double-precision array of length `NSITE`] `WEIGHT(I)` should contain the relative weight of site `I` in the regional weighted average. It is recommended that, as in Hosking and Wallis (1997, sec. 6.2), the weights be proportional to the record lengths at each site.

`RMOM` [output; double-precision array of length `NMOM`] The regional weighted average L-moment ratios ℓ_1^{R}, ℓ_2^{R}, t_3^{R}, t_4^{R}, ... , as defined below.

Numerical method

Let the number of sites be N and let the L-moment ratios at site i be $\ell_1^{(i)}$, $\ell_2^{(i)}$, $t_3^{(i)}$, $t_4^{(i)}$, ... Let the weight allotted to site i be w_i. When the data are scaled by dividing by the mean, the L-moment ratios become 1, $t^{(i)}$, $t_3^{(i)}$, $t_4^{(i)}$, ..., where $t^{(i)} = \ell_2^{(i)}/\ell_1^{(i)}$ is the at-site L-CV. The regional weighted average mean, L-CV and L-moment ratios are the weighted averages of these at-site values, and are defined by

$$\ell_1^{\mathrm{R}} = 1, \qquad t^{\mathrm{R}} = \sum_{i=1}^{N} w_i t^{(i)} \Big/ \sum_{i=1}^{N} w_i \,, \qquad t_r^{\mathrm{R}} = \sum_{i=1}^{N} w_i t_r^{(i)} \Big/ \sum_{i=1}^{N} w_i \,, \quad r = 3, 4, \ldots$$

(cf. Hosking and Wallis, 1997, sec. 6.2, who set w_i to be the record length at site i).

```
SUBROUTINE REGTST(NSITES,NAMES,LEN,XMOM,A,B,SEED,NSIM,NPROB,PROB,
KPRINT,KOUT,RMOM,D,VOBS,VBAR,VSD,H,Z,PARA)
```

Purpose

Calculates the discordancy, heterogeneity and goodness-of-fit measures described by Hosking and Wallis (1997, chaps. 3–5).

Formal parameters

`NSITES` [input; integer] Number of sites in region.

`NAMES` [input; `CHARACTER*12` array of length `NSITES`] Site names.

`LEN` [input; double-precision array of length `NSITES`] Record lengths at each site.

`XMOM` [input; double-precision array of dimension `(5,NSITES)`] Contains the first 5 sample L-moment ratios for each site, in the order mean, L-CV, t_3, t_4, t_5. `XMOM(K,I)` should contain the `K`th L-moment ratio for site `I`. Elements `XMOM(1,.)` are not used by the routine. Note that `XMOM(2,.)` should contain L-CV, not the usual ℓ_2!

`A,B` [input; double precision] Parameters of plotting position. Should be the same as the values used to calculate the L-moments in the `XMOM` array.

`SEED` [input; double precision] Seed for random number generator. Should be a whole number in the range `2D0` to `2147483647D0`.

`NSIM` [input; integer] Number of simulated worlds used in calculation of heterogeneity and goodness-of-fit measures. If `NSIM` is set to zero, the routine will calculate only the discordancy measures.

`NPROB` [input; integer] Number of quantiles to be calculated for each distribution that gives a good fit to the data.

`PROB` [input; double-precision array of length `NPROB`] Probabilities for which quantiles are to be calculated.

`KPRINT` [input; integer] Output flag. Should be set to zero to suppress printed output, to `1` to produce printed output.

`KOUT` [input; integer] Channel to which output is directed.

`RMOM` [output; double-precision array of length 5] Regional weighted average L-moment ratios.

`D` [output; double-precision array of length `NSITES`] Discordancy measure for each site.

`VOBS` [output; double-precision array of length 3] The observed values of three heterogeneity statistics. `V(1)` is the weighted standard deviation of the at-site L-CVs. `V(2)` is the weighted average distance from a site to the regional average on a graph of L-CV vs. L-skewness. `V(3)` is the weighted average distance from a site to the regional average on a graph of L-skewness vs. L-kurtosis.

VBAR [output; double-precision array of length 3] The means of the NSIM simulated values of the V statistics.

VSD [output; double-precision array of length 3] The standard deviations of the NSIM simulated values of the V statistics.

H [output; double-precision array of length 3] Heterogeneity measures for the region: H(J)=(VOBS(J)-VBAR(J))/VSD(J).

Z [output; double-precision array of length 3] Goodness-of-fit measures for five distributions: (1) generalized logistic, (2) generalized extreme-value, (3) generalized Normal (lognormal), (4) Pearson type III (3-parameter gamma), and (5) generalized Pareto.

PARA [output; double-precision array of dimension (5,6)] Parameters of regional distributions fitted by the above five distributions, plus (6) Wakeby. PARA(I,J) contains the Ith parameter of the Jth distribution.

Numerical method

The routine straightforwardly carries out the calculations described in the "Formal description" sections of Hosking and Wallis (1997, chaps. 3–5).

PROGRAM XCLUST

Purpose

This program illustrates the use of the cluster analysis routines. An example data set and the results obtained by applying PROGRAM XCLUST it are given in Tables 2 and 3. This cluster analysis is described in Hosking and Wallis (1997, sec. 9.2).

The data are site characteristics for 104 streamflow gaging stations in central Appalachia. The attributes for each site are drainage basin area, gage elevation, gage latitude and gage longitude. Nonlinear transformations are applied to two of the attributes: a logarithmic transformation to drainage basin area and a square root transform to gage elevation. These transformations give a more symmetric distribution of the values of the site characteristics at the 104 sites, reducing the likelihood that a few sites will have site characteristics so far from the other sites that they will always be assigned to a cluster by themselves, and, in Hosking and Wallis's judgement, give a better correspondence between differences in site characteristics and the degree of hydrologic dissimilarity between different basins. All four attributes are then standardized by dividing by the standard deviation of their values at the 104 sites. Finally, the drainage basin area variable is multiplied by 3 to give it an importance in the clustering procedure equal to that of the other variables together. The transformation and weighting of the variables involves subjective decisions whose ultimate justification is the physical plausibility of the regions that are ultimately obtained from the clustering procedure.

Clusters are formed by Ward's method, using routine CLUAGG. Hosking and Wallis (1997) judged that seven clusters is an appropriate number; information about these clusters is printed using routine CLUINF. The clusters obtained by Ward's method are adjusted using the K-means procedure (routine CLUKM). This yields clusters that are a little more compact in the space of transformed site characteristics. The cluster numbers in the output file are not those used by Hosking and Wallis (1997), who renumbered the clusters according to the average drainage area of the sites in each cluster.

PROGRAM XFIT

Purpose

This program is an example of how the routine REGLMR and others can be used to perform frequency analysis on a regional data set using an index-flood procedure which fits a regional distribution using the method of L-moments. The program fits a Wakeby distribution to the regional L-moments using the regional L-moment algorithm described by Hosking and Wallis (1997, sec. 6.2). It should be easy to change the program to fit other distributions.

An example data set and the results obtained by applying PROGRAM XFIT to it are given in Tables 4 and 5. The data are annual maximum wind speeds at 12 sites in the U.S., and are taken from Simiu et al. (1979). These 12 sites, all in the south-east of the U.S. and close to the coast, were judged to constitute a homogeneous region suitable for the application of an index-flood procedure (Wallis, 1993).

`PROGRAM XTEST`

Purpose

This program is an example of the use of routine `REGTST`. The program reads a file of at-site sample L-moments for one or more regions and calls routine `REGTST` to calculate test statistics for the regions.

An example data set and the results obtained by applying `PROGRAM XTEST` to it are given in Tables 6 and 7. The data set is given in Hosking and Wallis (1997, Table 3.4) and the used in the "Example" subsections of Hosking and Wallis (1997, chaps. 3–5). The data are annual precipitations at 19 sites in the northwest U.S., the "North Cascades" region of Plantico et al. (1990). Data were obtained from the Historical Climatology Network (Karl et al., 1990). The results in Table 5 indicate that the region is acceptably close to homogeneous and can be well described by a lognormal or Pearson type III distribution.

`PROGRAM XSIM`

Purpose

This program illustrates the use of Monte Carlo simulation to derive the properties of estimated quantiles in regional frequency analysis.

The region is set up as in Hosking and Wallis (1997, sec. 6.5). The region contains 19 sites with record lengths as for the North Cascades data. The sites have lognormal frequency distributions with L-CV varying linearly from 0.0978 at site 1 to 0.1228 at site 19, and each site has L-skewness $\tau_3 = 0.0279$. $10,000$ realizations of this region are made and the regional L-moment algorithm described by Hosking and Wallis (1997, sec. 6.2) is used to fit a lognormal distribution to the data generated at each realization. The regional average relative RMSE of the estimated growth curve is calculated from the simulations, and quantiles of the distribution of the ratio of the regional average of the estimated to the true at-site growth curve are computed from a histogram accumulated during the simulations. From these quantiles, 90% error bounds for the growth curve are computed as in Hosking and Wallis (1997, eq. (6.19)). These results are contained in the output from `PROGRAM XSIM`, given in Table 8.

`PROGRAM XSIM` can also be used to find the average value taken by heterogeneity measures over different simulations of a given region. This is useful when assessing the accuracy of estimated quantiles in regional frequency analysis, as discussed by Hosking and Wallis (1997, sec. 6.4). Table 9 contains the output from running `PROGRAM XSIM` with parameter values `NREP=100`, `NSIM=500`, and `KPRINT=1`, and omitting the section between dashed lines near the end of the program.

Table 2: Example data set: site characteristics for 104 streamflow gaging stations in central Appalachia.

*siteid	lat	long	area	elev	n	l1	t	t3	t4	t5
01578500	39.6900	76.1286	193.0	73	41	7379.95	0.4958	0.4844	0.3348	0.2101
01580000	39.6303	76.4036	94.4	250	65	4273.89	0.2878	0.3443	0.2431	0.0951
01582000	39.6044	76.6211	52.9	305	47	2725.96	0.3040	0.3009	0.2183	0.1027
01583000	39.4944	76.7958	2.0	420	34	186.09	0.3147	0.3548	0.2512	0.0736
01583500	39.5105	76.6769	59.8	262	47	3512.02	0.4872	0.5545	0.4564	0.4115
01584500	39.5050	76.4322	36.1	261	58	3559.83	0.3812	0.2951	0.1519	0.0307
01585100	39.3708	76.4461	7.6	38	30	1779.00	0.3992	0.3746	0.2912	0.1843
01585200	39.3736	76.5930	2.1	285	30	747.23	0.3555	0.2613	0.0717	-0.0712
01585300	39.3411	76.4880	4.4	21	29	1523.79	0.3218	0.2961	0.2835	0.2744
01585400	39.3336	76.4730	1.9	10	29	507.66	0.4746	0.5598	0.4086	0.3284
01585500	39.5930	76.9680	3.2	670	43	392.95	0.5376	0.4915	0.2311	0.1043
01586000	39.5000	76.8833	56.6	425	46	3401.70	0.4283	0.6113	0.5218	0.4068
01587500	39.3514	76.9139	64.4	289	32	4291.56	0.4874	0.5857	0.4366	0.2878
01588000	39.3819	76.9667	11.4	450	43	1185.53	0.5377	0.5678	0.4470	0.3535
01589100	39.2400	76.6925	2.4	45	32	742.44	0.2970	0.4479	0.3373	0.1841
01589300	39.3458	76.7336	32.5	361	33	2458.24	0.4945	0.5777	0.4162	0.3109
01589330	39.3111	76.7172	5.5	310	28	1940.04	0.3861	0.5167	0.3361	0.2000
01589440	39.3917	76.6617	25.2	240	31	1873.29	0.5978	0.7086	0.5486	0.3834
01591000	39.2383	77.0564	34.8	364	47	2827.72	0.5539	0.5942	0.4309	0.2827
01593500	39.1678	76.8519	38.0	260	59	1923.37	0.4134	0.4896	0.3047	0.1862
01613900	39.2144	78.2883	15.0	668	31	1015.77	0.3665	0.1977	0.1804	0.0736
01615000	39.1778	78.0722	57.4	503	48	2901.19	0.3939	0.4013	0.3245	0.1985
01616000	39.1778	78.0861	16.5	526	24	579.42	0.3297	0.2297	0.1262	0.0348
01620500	38.3375	79.2403	17.2	2054	45	1204.36	0.5448	0.5875	0.4250	0.3274
01621000	38.5028	79.0539	72.6	1606	30	3088.00	0.3934	0.5723	0.5101	0.4325
01621200	38.4744	78.9872	9.4	1349	28	700.89	0.3689	0.3675	0.3268	0.2441
01622000	38.3403	78.9139	379.0	1103	62	10619.03	0.4360	0.5363	0.4384	0.3285
01622400	38.1986	79.2194	0.4	1622	24	63.38	0.3511	0.4045	0.2726	0.2309
01624300	38.2433	79.0355	178.0	1260	19	6979.47	0.4645	0.5876	0.5897	0.5952
01624800	38.1283	78.9947	70.1	1230	23	2686.74	0.2224	-0.1796	0.1440	0.0904
01625000	38.2617	78.8622	375.0	1061	64	8202.77	0.4135	0.3727	0.2564	0.1686
01626000	38.0575	78.9083	127.0	1296	39	4110.49	0.4978	0.4996	0.2836	0.1281
01627500	38.2186	78.8369	212.0	1129	49	7392.90	0.4180	0.2837	0.1574	0.1244
01628500	38.3225	78.7550	1084.0	1013	61	22378.03	0.4097	0.4074	0.2402	0.0995
01629500	38.6461	78.5350	1377.0	721	31	26762.58	0.4360	0.4547	0.2933	0.1526
01629945	38.5753	78.4589	3.1	1300	32	255.06	0.4341	0.2485	0.0648	0.0130
01631000	38.9139	78.2111	1642.0	469	67	29396.12	0.4256	0.4074	0.2243	0.1085
01632000	38.6369	78.8530	210.0	1051	66	11194.24	0.4009	0.3659	0.2574	0.1884
01632300	38.5786	78.7611	8.1	1080	25	215.92	0.5392	0.5024	0.3746	0.3360
01632900	38.6933	78.6431	93.2	881	30	2796.37	0.3986	0.3117	0.1946	0.1254
01632970	38.7622	78.6850	6.4	962	19	914.11	0.3566	0.2560	0.1623	0.1335
01633000	38.7455	78.6392	506.0	838	49	14044.08	0.3896	0.4329	0.3692	0.2897
01633500	38.8653	78.6292	79.4	895	30	2480.67	0.3635	0.2907	0.1257	0.0978
01633650	38.9300	78.5453	3.6	1027	21	163.29	0.4449	0.4081	0.2392	0.1345
01634000	38.9767	78.3364	768.0	494	65	16093.69	0.4195	0.4786	0.3925	0.2912
01634500	39.0811	78.3297	103.0	647	54	4367.30	0.4468	0.4553	0.3555	0.2323
01635500	38.9580	78.2669	87.8	525	59	3358.56	0.4523	0.5353	0.4270	0.2902
01636210	38.9055	78.1861	14.0	610	29	906.24	0.3339	0.3143	0.1221	0.0053
01638480	39.2544	77.5767	89.6	249	21	5736.00	0.4486	0.4663	0.3873	0.2263
01638500	39.2736	77.5430	9651.0	200	97	122134.02	0.2987	0.3128	0.2710	0.1699
01639500	39.6125	77.2361	102.0	340	44	4590.68	0.3787	0.5601	0.5219	0.3942
01640000	39.5611	77.0439	8.1	525	31	621.65	0.4639	0.4501	0.2809	0.1568
01640500	39.6767	77.4639	5.9	965	53	641.00	0.5189	0.4502	0.1929	0.0804
01641000	39.5944	77.3972	18.4	355	42	1006.10	0.3186	0.1920	0.1306	0.1128
01641500	39.5264	77.4667	7.2	735	37	247.27	0.5571	0.5920	0.4191	0.3084
01643000	39.3878	77.3800	817.0	231	62	21248.87	0.2781	0.3349	0.2842	0.1830
01643500	39.2944	77.4083	62.8	240	42	3724.74	0.4705	0.6372	0.5152	0.3930
01643700	38.9864	77.7969	123.0	329	24	5157.25	0.4228	0.3778	0.2394	0.2055
01644000	39.0194	77.5778	332.0	248	64	10403.44	0.4591	0.5334	0.4074	0.2670
01645000	39.1280	77.3369	101.0	214	61	4509.33	0.4771	0.5796	0.3946	0.2225
01645200	39.0842	77.1772	3.7	330	30	872.20	0.4237	0.4276	0.3410	0.2119
01646000	38.9758	77.2461	57.9	151	57	2574.70	0.4691	0.6327	0.5324	0.4450
01646550	38.9575	77.1086	4.1	169	40	1065.60	0.3412	0.1810	0.1516	0.0280
01649500	38.9603	76.9261	72.8	12	53	3728.66	0.3136	0.2991	0.2275	0.1363

Table 2, continued.

01650500	39.0653	77.0300	21.1	264	60	1769.68	0.4085	0.5452	0.4020	0.2434
01652500	38.8433	77.0794	13.8	22	39	3883.18	0.3870	0.3647	0.2721	0.1401
01653000	38.8055	77.1022	33.7	36	36	5010.42	0.3262	0.3626	0.3744	0.2026
01654000	38.8128	77.2286	23.5	190	44	2859.00	0.4157	0.3968	0.2338	0.1548
01655500	38.7403	77.7878	12.3	419	41	1439.44	0.4781	0.3900	0.2366	0.2036
01656000	38.6367	77.6253	93.4	199	41	4870.68	0.3969	0.4726	0.4330	0.3672
01656200	38.8069	77.8130	2.9	610	36	97.00	0.2901	0.1422	0.2122	0.1338
01656500	38.7805	77.6728	50.5	284	35	2741.31	0.4763	0.5630	0.4612	0.3216
01657000	38.7978	77.4578	148.0	138	31	9862.58	0.4044	0.5411	0.5494	0.4845
01658500	38.5872	77.4289	7.6	238	40	764.05	0.4752	0.5254	0.4407	0.2982
01662000	38.6847	77.9042	195.0	312	49	5730.37	0.4085	0.4872	0.3984	0.2509
01662800	38.6555	78.0742	27.6	374	33	1723.58	0.4700	0.5064	0.2706	0.1513
01663500	38.5917	77.9653	287.0	288	50	11096.00	0.4314	0.4888	0.2732	0.1203
01664000	38.5305	77.8139	620.0	252	49	16107.76	0.3810	0.5102	0.3965	0.2214
01665000	38.4805	78.0528	15.9	389	42	1038.69	0.4888	0.5025	0.2407	0.0765
01665050	38.4511	77.9566	0.3	400	32	78.90	0.3791	0.2629	0.1391	-0.0445
01665500	38.2805	78.3403	114.0	439	48	6371.77	0.4684	0.4451	0.2725	0.1554
01666500	38.3250	78.0958	179.0	283	49	7722.45	0.4271	0.4216	0.2849	0.1961
01667000	38.3130	78.0639	446.0	266	34	11775.00	0.4505	0.5083	0.3879	0.2222
01667500	38.3503	77.9753	472.0	241	61	13100.66	0.4133	0.4492	0.3323	0.1909
01668000	38.3222	77.5181	1596.0	55	84	33591.79	0.3053	0.3357	0.3028	0.1920
02027500	37.7022	79.0278	47.6	633	42	3675.31	0.6022	0.6822	0.5765	0.4666
02027800	37.6055	78.9236	147.0	444	31	8679.45	0.5326	0.4896	0.3184	0.2015
02028500	37.8694	78.8236	94.6	530	49	6344.12	0.5356	0.6149	0.5058	0.4396
02028700	37.8683	78.7255	4.0	640	25	583.84	0.4948	0.4193	0.3024	0.2542
02029200	37.9675	78.6178	11.0	577	25	883.96	0.7136	0.7417	0.5698	0.3669
02030000	37.8125	78.4556	116.0	294	53	5210.28	0.5301	0.5801	0.4513	0.3787
02030500	37.7028	78.3778	226.0	238	65	6141.08	0.4006	0.5397	0.4081	0.2882
02031500	38.1403	78.7514	11.4	999	26	843.50	0.3924	0.3462	0.1352	-0.0028
02032700	38.0422	78.4750	1.3	371	37	516.49	0.3202	0.3432	0.1907	0.1270
02034000	37.8578	78.2661	664.0	210	58	21033.45	0.4168	0.4148	0.2611	0.1676
02034500	37.6667	78.1667	262.0	178	64	3793.92	0.4121	0.4376	0.2250	0.0952
02036500	37.5978	77.8200	22.1	156	47	856.28	0.5401	0.4488	0.1967	0.0910
02038800	37.3819	78.7900	5.7	650	22	702.18	0.4763	0.5626	0.4120	0.3843
02038850	37.4153	78.6361	8.5	472	25	1239.76	0.6638	0.6815	0.4985	0.3273
02039000	37.2569	78.4866	69.7	339	45	2105.42	0.4164	0.3314	0.2290	0.1703
02039500	37.3069	78.3889	303.0	281	66	6260.23	0.3978	0.3525	0.2188	0.1317
02040000	37.4214	77.8591	726.0	174	71	7748.87	0.3636	0.4120	0.2598	0.1723
02041000	37.2831	77.8700	158.0	177	45	4363.09	0.4230	0.3251	0.1392	0.0330
02041500	37.2258	77.5389	1334.0	116	40	8905.50	0.2306	0.4244	0.3813	0.1828

Table 3: Output when program XCLUST is applied to the data of Table 2.

MERGING SEQUENCE FROM WARD'S ALGORITHM

STAGE	NUMBER OF CLUSTERS	MERGED CLUSTERS		SUM OF SQUARES
1	103	27	31	0.01
2	102	83	84	0.03
3	101	3	5	0.07
4	100	40	43	0.13
5	99	49	60	0.22
6	98	92	96	0.31
7	97	29	33	0.40
8	96	46	47	0.49
9	95	16	19	0.59
10	94	21	23	0.69
11	93	16	20	0.82
12	92	39	41	0.95
13	91	12	13	1.08
14	90	4	8	1.22
15	89	21	48	1.37
16	88	65	68	1.53
17	87	53	55	1.71
18	86	75	77	1.90
19	85	10	15	2.09
20	84	14	52	2.29
21	83	3	12	2.49
22	82	6	18	2.70
23	81	61	63	2.91
24	80	69	79	3.12
25	79	57	62	3.34
26	78	78	83	3.58
27	77	36	44	3.87
28	76	98	99	4.16
29	75	75	82	4.47
30	74	29	38	4.78
31	73	70	73	5.11
32	72	25	30	5.45
33	71	87	88	5.79
34	70	92	101	6.15
35	69	49	58	6.51
36	68	81	91	6.87
37	67	7	9	7.24
38	66	95	102	7.60
39	65	35	37	7.98
40	64	26	93	8.38
41	63	57	72	8.82
42	62	6	16	9.28
43	61	22	46	9.79
44	60	17	61	10.33
45	59	29	32	10.89
46	58	2	51	11.48
47	57	42	45	12.11
48	56	49	70	12.79
49	55	81	87	13.48
50	54	26	39	14.19
51	53	69	76	14.94
52	52	64	67	15.72
53	51	92	103	16.55
54	50	90	98	17.39
55	49	4	11	18.27
56	48	2	3	19.18
57	47	14	54	20.10
58	46	86	100	21.03
59	45	59	75	21.98
60	44	89	90	23.01
61	43	34	35	24.16
62	42	22	40	25.34
63	41	17	74	26.58
64	40	95	104	27.87
65	39	36	71	29.17
66	38	65	66	30.49
67	37	21	69	31.89
68	36	27	29	33.67
69	35	34	42	35.49
70	34	49	57	37.39
71	33	81	86	39.50
72	32	56	85	41.76
73	31	14	53	44.05
74	30	7	10	46.38
75	29	59	78	48.90
76	28	24	26	51.67
77	27	80	94	54.56
78	26	6	65	57.83
79	25	1	2	61.47
80	24	49	64	65.42
81	23	4	17	69.49
82	22	21	97	74.03
83	21	22	25	78.87
84	20	28	80	84.00
85	19	56	95	89.73
86	18	4	7	96.38
87	17	81	92	103.55
88	16	1	49	110.90
89	15	24	36	118.51
90	14	24	89	127.08
91	13	56	59	137.40
92	12	14	21	148.06
93	11	22	27	159.62
94	10	34	50	173.02
95	9	6	14	187.12
96	8	34	56	205.57
97	7	22	81	229.50
98	6	24	28	264.91
99	5	4	6	313.60
100	4	1	22	395.86
101	3	4	24	488.47
102	2	1	34	598.12
103	1	1	4	1236.00

ASSIGNMENT OF SITES TO CLUSTERS

1	1	1	2	1	6	2	2	2	2
2	1	1	6	2	6	2	6	6	6
6	5	6	4	5	4	5	7	5	5
5	5	5	3	3	4	3	5	4	5
4	3	5	4	3	5	5	6	1	3
1	6	6	6	6	3	1	1	3	1
2	1	2	1	6	6	1	6	6	1
4	1	1	2	3	6	3	3	6	7
5	3	3	3	3	5	5	5	4	4
5	5	4	7	3	5	6	4	4	5
5	3	5	3						

Table 3, continued.

CLUSTER MEMBERSHIP

CLUSTER 1 HAS 17 MEMBERS:
1 2 3 5 12 13 49 51 57 58
60 62 64 67 70 72 73

CLUSTER 2 HAS 11 MEMBERS:
4 7 8 9 10 11 15 17 61 63
74

CLUSTER 3 HAS 18 MEMBERS:
34 35 37 42 45 50 56 59 75 77
78 82 83 84 85 95 102 104

CLUSTER 4 HAS 12 MEMBERS:
24 26 36 39 41 44 71 89 90 93
98 99

CLUSTER 5 HAS 23 MEMBERS:
22 25 27 29 30 31 32 33 38 40
43 46 47 81 86 87 88 91 92 96
100 101 103

CLUSTER 6 HAS 20 MEMBERS:
6 14 16 18 19 20 21 23 48 52
53 54 55 65 66 68 69 76 79 97

CLUSTER 7 HAS 3 MEMBERS:
28 80 94

ADJUSTED CLUSTERS FROM K-MEANS ALGORITHM
(SUM OF SQUARES = 215.86)

ASSIGNMENT OF SITES TO CLUSTERS

1	1	1	2	1	1	2	2	2	2
2	1	1	6	2	1	2	1	1	1
6	1	6	4	5	4	5	7	5	5
5	5	5	3	3	4	3	5	4	5
4	3	5	4	3	5	5	6	1	3
1	6	6	6	6	3	1	1	3	1
2	1	2	1	6	6	1	6	6	1
2	1	1	6	5	6	3	3	6	7
5	5	3	3	3	5	5	5	4	4
5	5	4	7	3	5	6	4	4	5
5	3	5	3						

CLUSTER MEMBERSHIP

CLUSTER 1 HAS 23 MEMBERS:
1 2 3 5 6 12 13 16 18 19
20 22 49 51 57 58 60 62 64 67
70 72 73

CLUSTER 2 HAS 11 MEMBERS:
4 7 8 9 10 11 15 17 61 63
71

CLUSTER 3 HAS 16 MEMBERS:
34 35 37 42 45 50 56 59 77 78
83 84 85 95 102 104

CLUSTER 4 HAS 11 MEMBERS:
24 26 36 39 41 44 89 90 93 98
99

CLUSTER 5 HAS 24 MEMBERS:
25 27 29 30 31 32 33 38 40 43
46 47 75 81 82 86 87 88 91 92
96 100 101 103

CLUSTER 6 HAS 16 MEMBERS:
14 21 23 48 52 53 54 55 65 66
68 69 74 76 79 97

CLUSTER 7 HAS 3 MEMBERS:
28 80 94

TYPICAL MEMBERS OF EACH CLUSTER

	AREA	ELEV	LAT	LONG
1	63.88	232.68	39.23	77.08
2	3.29	204.62	39.26	76.84
3	863.68	322.08	38.49	78.00
4	7.09	967.65	38.22	78.75
5	139.48	636.47	38.15	78.59
6	13.71	392.02	38.99	77.59
7	0.54	702.88	38.23	78.55

Table 4: Example data set: annual maximum wind speeds at 12 sites in the U.S., from Simiu et al. (1979). Numbers in parentheses are those used by Simiu et al. (1979, Table 3.4.1) to identify the sites.

```
12
(  2) Montgomery AL
28
  43  43  60  51  51  48  46  52  43  34
  51  46  37  40  46  47  40  38  40  36
  40  40  47  36  49  77  43  46
( 17) Jacksonville FL
28
  67  39  51  47  42  42  44  42  38  34
  42  44  49  56  74  52  44  69  47  53
  40  51  48  52  48  67  46  36
( 18) Key West FL
19
  58  36  64  48  36  36  78  86  90  38
  52  43  43  42  46  48  35  35  55
( 19) Tampa FL
10
  45  47  65  50  56  55  37  53  44  42
( 21) Macon GA
28
  50  42  58  44  53  32  51  51  42  42
  32  40  34  40  37  37  60  45  45  53
  40  43  48  51  58  49  46  38
( 22) Savannah GA
32
  58  79  60  49  54  51  58  66  58  46
  43  44  41  56  44  47  45  40  39  42
  40  31  39  39  46  42  45  44  43  39
  46  51
( 77) Cape Hatteras NC
45
  51  66  52  51  51  52  71  44  44  44
  51  51  56  51  51  58  61  49  53  55
  64  85  68  51  85  67  62  72  62  52
  55  46 103  52  53  49  61  69  43  44
  47  73  68  63  50
( 80) Wilmington NC
26
  52  62  62  69  63  42  84  43  51  46
  56  47  55  41  51  49  39  42  48  44
  41  46  42  39  42  41
(107) Brownsville TX
35
  35  44  53  43  53  39  34  43  37  41
  40  48  39  41  32  40  33  48  42  37
  38  38  53  36  66  63  56  34  43  51
  46  44  46  42  49
(108) Corpus Christi TX
34
  60  49  71  50  66  47  48  57  55  50
  45  46  46  44  48  44  44  45  67  60
  51  48  45  48  77  48  45 128  70  50
  46  58  52  44
(111) Port Arthur TX
25
  63  45  46  60  66  39  47  51  51  51
  61  67  55  55  45  51  57  45  81  57
  54  44  49  44  43
(116) Norfolk VA
20
  53  69  48  56  52  42  49  64  40  46
  50  56  58  40  37  46  35  46  46  42
```

Table 5: Output when program XFIT is applied to the data of Table 4.

SITE 1 (2) Montgomery AL	N= 28	L-MOMENT RATIOS	45.36	4.6747	0.2170	0.2957	0.1501
SITE 2 (17) Jacksonville FL	N= 28	L-MOMENT RATIOS	48.71	5.8204	0.2195	0.2403	0.0487
SITE 3 (18) Key West FL	N= 19	L-MOMENT RATIOS	51.00	9.6030	0.3226	0.1846	0.0505
SITE 4 (19) Tampa FL	N= 10	L-MOMENT RATIOS	49.40	5.8020	0.0918	0.3013	-0.0216
SITE 5 (21) Macon GA	N= 28	L-MOMENT RATIOS	45.04	4.7797	0.0584	0.1617	0.0138
SITE 6 (22) Savannah GA	N= 32	L-MOMENT RATIOS	47.66	5.3657	0.2449	0.2441	0.0484
SITE 7 (77) Cape Hatteras NC	N= 45	L-MOMENT RATIOS	57.91	6.7387	0.2712	0.2088	0.0672
SITE 8 (80) Wilmington NC	N= 26	L-MOMENT RATIOS	49.88	6.0357	0.3127	0.2222	0.0786
SITE 9 (107) Brownsville TX	N= 35	L-MOMENT RATIOS	43.63	4.7397	0.1892	0.2036	0.0595
SITE 10 (108) Corpus Christi TX	N= 34	L-MOMENT RATIOS	54.47	6.9789	0.4698	0.3379	0.1856
SITE 11 (111) Port Arthur TX	N= 25	L-MOMENT RATIOS	53.08	5.6770	0.1969	0.2196	0.0699
SITE 12 (116) Norfolk VA	N= 20	L-MOMENT RATIOS	48.75	5.5037	0.1344	0.2347	0.0403

```
REGIONAL AVERAGE L-MOMENT RATIOS   1.0000   0.1183   0.2402   0.2359   0.0735

REGIONAL WAKEBY PARAMETERS      0.5702      3.6499     14.9281      0.1838      0.0839
```

SITE NUMBER	0.1000	0.2000	QUANTILES 0.5000	0.8000	0.9000	0.9500	0.9800	0.9900	0.9990	0.9999
REGION	0.78	0.85	0.95	1.13	1.28	1.44	1.67	1.85	2.54	3.37
1	35.53	38.43	42.90	51.32	58.13	65.35	75.56	83.83	115.00	152.81
2	38.16	41.28	46.08	55.12	62.43	70.19	81.16	90.03	123.51	164.12
3	39.95	43.21	48.24	57.70	65.36	73.48	84.96	94.26	129.30	171.82
4	38.70	41.86	46.73	55.89	63.31	71.18	82.30	91.30	125.25	166.43
5	35.28	38.16	42.60	50.96	57.72	64.89	75.03	83.23	114.18	151.73
6	37.33	40.38	45.08	53.92	61.08	68.66	79.39	88.08	120.83	160.56
7	45.37	49.07	54.78	65.52	74.22	83.44	96.48	107.03	146.83	195.11
8	39.08	42.27	47.18	56.44	63.93	71.87	83.11	92.19	126.48	168.06
9	34.18	36.97	41.27	49.36	55.92	62.86	72.68	80.63	110.61	146.99
10	42.67	46.15	51.52	61.63	69.81	78.48	90.75	100.67	138.10	183.51
11	41.58	44.98	50.21	60.06	68.03	76.48	88.43	98.10	134.58	178.83
12	38.19	41.31	46.11	55.16	62.48	70.24	81.22	90.10	123.60	164.24

Table 6: Example data set: L-moments of annual precipitation at 19 sites in the U.S. The columns in the table are the site identifier used by the Historical Climatology Network, the record length, and the at-site sample L-moments ℓ_1, L-CV, t_3, t_4, t_5.

```
  19 NORTH CASCADES
350304        98  19.685  0.1209  0.0488  0.1433 -0.0004
351433        59  62.580  0.0915  0.0105  0.1569  0.0020
351862        90  40.852  0.1124  0.0614  0.1541 -0.0058
351897        61  46.045  0.1032  0.0417  0.1429 -0.0022
352997        65  45.021  0.0967 -0.0134  0.1568  0.0173
353445        86  31.042  0.1328 -0.0176  0.1206  0.0235
353770        78  80.143  0.1008  0.0943  0.1967  0.0856
356907        72  41.305  0.1143  0.0555  0.1210  0.0487
357169        67  30.585  0.1107  0.0478  0.1371  0.0316
357331        99  32.932  0.1179  0.0492  0.0900  0.0225
357354        49  17.560  0.1308  0.0940  0.1273  0.0352
358466        61  69.518  0.1119 -0.0429  0.0927 -0.0061
450945        69  47.653  0.1018  0.0435  0.1446 -0.0056
451233        73 102.501  0.1025  0.0182  0.1047 -0.0221
453284        70  52.413  0.1054 -0.0224  0.1664  0.0035
454764        66  79.696  0.1174  0.0124  0.1317 -0.0176
454769        59  44.643  0.1115 -0.0346  0.1032  0.0083
457773        74  58.655  0.1003  0.0446  0.1450 -0.0379
458773        82  39.024  0.1046  0.0128  0.1583  0.0443
```

Table 7: Output when program `XTEST` is applied to the data of Table 6.

```
  NORTH CASCADES                                            19 SITES

SITE    N      NAME       L-CV   L-SKEW  L-KURT   D(I)
   1   98  350304       0.1209  0.0488  0.1433   0.60
   2   59  351433       0.0915  0.0105  0.1569   1.02
   3   90  351862       0.1124  0.0614  0.1541   0.38
   4   61  351897       0.1032  0.0417  0.1429   0.23
   5   65  352997       0.0967 -0.0134  0.1568   0.93
   6   86  353445       0.1328 -0.0176  0.1206   2.63
   7   78  353770       0.1008  0.0943  0.1967   2.12
   8   72  356907       0.1143  0.0555  0.1210   0.45
   9   67  357169       0.1107  0.0478  0.1371   0.11
  10   99  357331       0.1179  0.0492  0.0900   1.61
  11   49  357354       0.1308  0.0940  0.1273   2.08
  12   61  358466       0.1119 -0.0429  0.0927   1.52
  13   69  450945       0.1018  0.0435  0.1446   0.31
  14   73  451233       0.1025  0.0182  0.1047   1.30
  15   70  453284       0.1054 -0.0224  0.1664   1.58
  16   66  454764       0.1174  0.0124  0.1317   0.29
  17   59  454769       0.1115 -0.0346  0.1032   1.04
  18   74  457773       0.1003  0.0446  0.1450   0.43
  19   82  458773       0.1046  0.0128  0.1583   0.38

    WEIGHTED MEANS      0.1103  0.0279  0.1366

PARAMETERS OF REGIONAL KAPPA DISTRIBUTION   0.9542  0.1533  0.1236 -0.2955

***** HETEROGENEITY MEASURES *****
(NUMBER OF SIMULATIONS  =   500)

OBSERVED     S.D. OF GROUP L-CV          =  0.0104
SIM. MEAN OF S.D. OF GROUP L-CV          =  0.0095
SIM. S.D. OF S.D. OF GROUP L-CV          =  0.0016
STANDARDIZED TEST VALUE H(1)             =  0.62

OBSERVED AVE.  OF L-CV / L-SKEW DISTANCE =  0.0339
SIM. MEAN OF AVE. L-CV / L-SKEW DISTANCE =  0.0447
SIM. S.D. OF AVE. L-CV / L-SKEW DISTANCE =  0.0072
STANDARDIZED TEST VALUE H(2)             = -1.49

OBSERVED AVE.  OF L-SKEW/L-KURT DISTANCE =  0.0405
SIM. MEAN OF AVE. L-SKEW/L-KURT DISTANCE =  0.0578
SIM. S.D. OF AVE. L-SKEW/L-KURT DISTANCE =  0.0073
STANDARDIZED TEST VALUE H(3)             = -2.37


***** GOODNESS-OF-FIT MEASURES *****
(NUMBER OF SIMULATIONS  =   500)

GEN. LOGISTIC        L-KURTOSIS= 0.167   Z VALUE=  3.46
GEN. EXTREME VALUE   L-KURTOSIS= 0.111   Z VALUE= -2.94
GEN. NORMAL          L-KURTOSIS= 0.123   Z VALUE= -1.51 *
PEARSON TYPE III     L-KURTOSIS= 0.123   Z VALUE= -1.60 *
GEN. PARETO          L-KURTOSIS= 0.006   Z VALUE=-14.75


PARAMETER ESTIMATES FOR DISTRIBUTIONS ACCEPTED AT THE 90% LEVEL
GEN. NORMAL          0.994  0.195 -0.057
PEARSON TYPE III     1.000  0.196  0.171
WAKEBY               0.568  2.003  7.330  0.244 -0.270

QUANTILE ESTIMATES
                     0.010  0.020  0.050  0.100  0.200  0.500  0.900  0.950  0.990  0.999
GEN. NORMAL          0.569  0.616  0.688  0.753  0.834  0.994  1.254  1.331  1.480  1.654
PEARSON TYPE III     0.570  0.616  0.688  0.753  0.834  0.994  1.254  1.331  1.480  1.653
WAKEBY               0.590  0.610  0.666  0.740  0.840  0.993  1.259  1.341  1.483  1.603


ALL DATA PROCESSED
```

Table 8: Output from program XSIM.

```
REGIONAL LOGNORMAL SIMULATIONS    NREP= 10000  SEED=  619145091.


SITE     XI   ALPHA       K    L-CV  L-SKEW   N    QUANTILES .....
                                                  0.0100  0.1000  0.5000  0.9000  0.9900  0.9990
  1  0.9951  0.1731 -0.0571  0.0978  0.0279  98   0.618   0.781   0.995   1.225   1.426   1.580
  2  0.9950  0.1756 -0.0571  0.0992  0.0279  59   0.612   0.778   0.995   1.228   1.432   1.588
  3  0.9949  0.1781 -0.0571  0.1006  0.0279  90   0.607   0.775   0.995   1.232   1.438   1.597
  4  0.9948  0.1805 -0.0571  0.1020  0.0279  61   0.602   0.772   0.995   1.235   1.444   1.605
  5  0.9948  0.1830 -0.0571  0.1034  0.0279  65   0.596   0.769   0.995   1.238   1.450   1.613
  6  0.9947  0.1853 -0.0571  0.1047  0.0279  86   0.591   0.766   0.995   1.241   1.456   1.621
  7  0.9946  0.1878 -0.0571  0.1061  0.0279  78   0.586   0.763   0.995   1.244   1.462   1.629
  8  0.9946  0.1903 -0.0571  0.1075  0.0279  72   0.580   0.759   0.995   1.248   1.468   1.638
  9  0.9945  0.1928 -0.0571  0.1089  0.0279  67   0.575   0.756   0.994   1.251   1.474   1.646
 10  0.9944  0.1952 -0.0571  0.1103  0.0279  99   0.569   0.753   0.994   1.254   1.480   1.654
 11  0.9943  0.1977 -0.0571  0.1117  0.0279  49   0.564   0.750   0.994   1.257   1.486   1.663
 12  0.9943  0.2002 -0.0571  0.1131  0.0279  61   0.558   0.747   0.994   1.260   1.492   1.671
 13  0.9942  0.2027 -0.0571  0.1145  0.0279  69   0.553   0.744   0.994   1.264   1.498   1.679
 14  0.9941  0.2051 -0.0571  0.1159  0.0279  73   0.547   0.741   0.994   1.267   1.505   1.687
 15  0.9941  0.2074 -0.0571  0.1172  0.0279  70   0.542   0.738   0.994   1.270   1.510   1.695
 16  0.9940  0.2099 -0.0571  0.1186  0.0279  66   0.537   0.735   0.994   1.273   1.516   1.703
 17  0.9939  0.2124 -0.0571  0.1200  0.0279  59   0.531   0.731   0.994   1.276   1.522   1.712
 18  0.9939  0.2149 -0.0571  0.1214  0.0279  74   0.526   0.728   0.994   1.280   1.528   1.720
 19  0.9938  0.2174 -0.0571  0.1228  0.0279  82   0.520   0.725   0.994   1.283   1.535   1.728
        AVERAGE FOR ALL SITES (ARITHMETIC MEAN)   0.569   0.753   0.994   1.254   1.480   1.654
        REGIONAL GROWTH CURVE (HARMONIC MEAN)     0.568   0.753   0.994   1.254   1.479   1.653


INTER-SITE CORRELATION=  0.64


L-MOMENT ESTIMATION OF LOGNORMAL DISTRIBUTION           FAILURES:     0        WARNINGS:     0

        F      0.0100 0.1000 0.5000 0.9000 0.9900 0.9990

                    SITE  1 QUANTILES
     BIAS      -0.078 -0.035  0.000  0.023  0.037  0.046
     RMSE       0.095  0.043  0.018  0.030  0.048  0.063
0.050 PT.       0.832  0.922  0.969  0.990  0.990  0.982
0.950 PT.       1.005  1.008  1.030  1.056  1.089  1.121

                    SITE  2 QUANTILES
     BIAS      -0.070 -0.031  0.000  0.020  0.033  0.041
     RMSE       0.089  0.043  0.024  0.032  0.047  0.061
0.050 PT.       0.836  0.920  0.961  0.980  0.981  0.973
0.950 PT.       1.019  1.019  1.039  1.061  1.090  1.118

                    SITE  3 QUANTILES
     BIAS      -0.061 -0.027  0.000  0.018  0.029  0.036
     RMSE       0.082  0.038  0.020  0.027  0.042  0.056
0.050 PT.       0.845  0.928  0.967  0.984  0.981  0.971
0.950 PT.       1.025  1.018  1.032  1.053  1.082  1.111

                    SITE  4 QUANTILES
     BIAS      -0.053 -0.023  0.000  0.015  0.024  0.030
     RMSE       0.078  0.038  0.024  0.029  0.041  0.054
0.050 PT.       0.850  0.927  0.961  0.975  0.972  0.962
0.950 PT.       1.038  1.028  1.039  1.056  1.080  1.108
```

Table 8, continued.

	SITE 5 QUANTILES					
BIAS	-0.044	-0.019	0.000	0.012	0.020	0.025
RMSE	0.072	0.036	0.023	0.027	0.038	0.051
0.050 PT.	0.858	0.931	0.961	0.972	0.968	0.958
0.950 PT.	1.046	1.031	1.038	1.053	1.076	1.102
	SITE 6 QUANTILES					
BIAS	-0.036	-0.015	0.000	0.010	0.016	0.020
RMSE	0.067	0.032	0.021	0.024	0.035	0.047
0.050 PT.	0.867	0.938	0.966	0.974	0.967	0.956
0.950 PT.	1.054	1.032	1.035	1.046	1.069	1.094
	SITE 7 QUANTILES					
BIAS	-0.027	-0.011	0.000	0.007	0.012	0.015
RMSE	0.064	0.031	0.022	0.024	0.034	0.045
0.050 PT.	0.875	0.941	0.964	0.970	0.963	0.950
0.950 PT.	1.064	1.037	1.037	1.045	1.066	1.090
	SITE 8 QUANTILES					
BIAS	-0.018	-0.007	0.000	0.004	0.007	0.010
RMSE	0.062	0.032	0.023	0.024	0.033	0.044
0.050 PT.	0.882	0.942	0.962	0.966	0.956	0.943
0.950 PT.	1.077	1.043	1.039	1.044	1.062	1.085
	SITE 9 QUANTILES					
BIAS	-0.009	-0.003	0.000	0.002	0.003	0.005
RMSE	0.060	0.032	0.024	0.025	0.033	0.044
0.050 PT.	0.891	0.945	0.961	0.962	0.952	0.939
0.950 PT.	1.086	1.048	1.040	1.043	1.059	1.081
	SITE 10 QUANTILES					
BIAS	0.001	0.001	0.000	-0.001	-0.001	0.000
RMSE	0.058	0.029	0.020	0.021	0.030	0.041
0.050 PT.	0.903	0.954	0.966	0.965	0.953	0.937
0.950 PT.	1.093	1.048	1.034	1.035	1.051	1.073
	SITE 11 QUANTILES					
BIAS	0.011	0.005	0.000	-0.003	-0.005	-0.005
RMSE	0.064	0.036	0.029	0.029	0.036	0.046
0.050 PT.	0.904	0.946	0.953	0.950	0.938	0.924
0.950 PT.	1.111	1.063	1.048	1.046	1.056	1.074
	SITE 12 QUANTILES					
BIAS	0.020	0.010	0.000	-0.006	-0.009	-0.010
RMSE	0.065	0.035	0.026	0.027	0.035	0.045
0.050 PT.	0.916	0.954	0.958	0.951	0.938	0.923
0.950 PT.	1.119	1.064	1.043	1.038	1.049	1.067
	SITE 13 QUANTILES					
BIAS	0.031	0.014	0.001	-0.008	-0.013	-0.015
RMSE	0.070	0.036	0.025	0.027	0.035	0.046
0.050 PT.	0.925	0.960	0.959	0.951	0.936	0.919
0.950 PT.	1.131	1.068	1.042	1.033	1.042	1.059
	SITE 14 QUANTILES					
BIAS	0.041	0.018	0.001	-0.011	-0.017	-0.020
RMSE	0.075	0.037	0.025	0.027	0.037	0.047
0.050 PT.	0.934	0.965	0.960	0.949	0.932	0.915
0.950 PT.	1.141	1.072	1.042	1.030	1.038	1.054

Table 8, continued.

	SITE 15 QUANTILES					
BIAS	0.051	0.022	0.001	-0.013	-0.021	-0.025
RMSE	0.082	0.040	0.026	0.029	0.039	0.049
0.050 PT.	0.943	0.967	0.959	0.945	0.927	0.909
0.950 PT.	1.154	1.078	1.043	1.030	1.036	1.050
	SITE 16 QUANTILES					
BIAS	0.061	0.027	0.001	-0.016	-0.025	-0.029
RMSE	0.089	0.043	0.026	0.031	0.041	0.052
0.050 PT.	0.951	0.970	0.957	0.941	0.923	0.904
0.950 PT.	1.166	1.084	1.045	1.029	1.032	1.046
	SITE 17 QUANTILES					
BIAS	0.072	0.031	0.001	-0.018	-0.028	-0.034
RMSE	0.099	0.048	0.028	0.033	0.045	0.056
0.050 PT.	0.961	0.972	0.954	0.936	0.917	0.899
0.950 PT.	1.180	1.091	1.048	1.029	1.031	1.042
	SITE 18 QUANTILES					
BIAS	0.084	0.036	0.001	-0.020	-0.032	-0.039
RMSE	0.107	0.049	0.026	0.033	0.046	0.057
0.050 PT.	0.973	0.980	0.958	0.938	0.917	0.898
0.950 PT.	1.190	1.092	1.044	1.022	1.023	1.035
	SITE 19 QUANTILES					
BIAS	0.095	0.040	0.001	-0.023	-0.036	-0.043
RMSE	0.116	0.052	0.025	0.033	0.048	0.060
0.050 PT.	0.984	0.985	0.960	0.937	0.913	0.893
0.950 PT.	1.200	1.094	1.041	1.017	1.017	1.029
	AVERAGE FOR ALL SITES					
ABS.BIAS	0.045	0.020	0.000	0.012	0.019	0.024
BIAS	0.004	0.002	0.000	0.000	0.000	0.000
RMSE	0.079	0.038	0.024	0.028	0.039	0.051
0.050 PT.	0.905	0.955	0.969	0.968	0.955	0.939
0.950 PT.	1.097	1.047	1.032	1.032	1.049	1.072
	GROWTH CURVE COMPONENT					
ABS.BIAS	0.045	0.020	0.000	0.012	0.019	0.024
BIAS	0.003	0.002	0.000	-0.001	0.000	0.000
RMSE	0.073	0.028	0.005	0.017	0.033	0.047
0.050 PT.	0.912	0.971	0.991	0.981	0.961	0.942
0.950 PT.	1.086	1.030	1.009	1.020	1.045	1.068

Hosking and Wallis (1997), Table 6.2

F	qhat(F)	RMSE	Error bounds	
0.010	0.569	0.073	0.524	0.624
0.100	0.753	0.028	0.731	0.776
0.500	0.994	0.005	0.985	1.004
0.900	1.254	0.017	1.230	1.278
0.990	1.480	0.033	1.417	1.541
0.999	1.654	0.047	1.548	1.756

Table 9: Output from program `XSIM` with parameters `NREP=100`, `NSIM=500`, and other modifications as described in text.

```
REGIONAL LOGNORMAL SIMULATIONS    NREP=   100  SEED=  619145091.



SITE     XI   ALPHA       K    L-CV  L-SKEW   N    QUANTILES .....
                                                 0.0100  0.1000  0.5000  0.9000  0.9900  0.9990
  1  0.9951  0.1731 -0.0571  0.0978  0.0279  98   0.618   0.781   0.995   1.225   1.426   1.580
  2  0.9950  0.1756 -0.0571  0.0992  0.0279  59   0.612   0.778   0.995   1.228   1.432   1.588
  3  0.9949  0.1781 -0.0571  0.1006  0.0279  90   0.607   0.775   0.995   1.232   1.438   1.597
  4  0.9948  0.1805 -0.0571  0.1020  0.0279  61   0.602   0.772   0.995   1.235   1.444   1.605
  5  0.9948  0.1830 -0.0571  0.1034  0.0279  65   0.596   0.769   0.995   1.238   1.450   1.613
  6  0.9947  0.1853 -0.0571  0.1047  0.0279  86   0.591   0.766   0.995   1.241   1.456   1.621
  7  0.9946  0.1878 -0.0571  0.1061  0.0279  78   0.586   0.763   0.995   1.244   1.462   1.629
  8  0.9946  0.1903 -0.0571  0.1075  0.0279  72   0.580   0.759   0.995   1.248   1.468   1.638
  9  0.9945  0.1928 -0.0571  0.1089  0.0279  67   0.575   0.756   0.994   1.251   1.474   1.646
 10  0.9944  0.1952 -0.0571  0.1103  0.0279  99   0.569   0.753   0.994   1.254   1.480   1.654
 11  0.9943  0.1977 -0.0571  0.1117  0.0279  49   0.564   0.750   0.994   1.257   1.486   1.663
 12  0.9943  0.2002 -0.0571  0.1131  0.0279  61   0.558   0.747   0.994   1.260   1.492   1.671
 13  0.9942  0.2027 -0.0571  0.1145  0.0279  69   0.553   0.744   0.994   1.264   1.498   1.679
 14  0.9941  0.2051 -0.0571  0.1159  0.0279  73   0.547   0.741   0.994   1.267   1.505   1.687
 15  0.9941  0.2074 -0.0571  0.1172  0.0279  70   0.542   0.738   0.994   1.270   1.510   1.695
 16  0.9940  0.2099 -0.0571  0.1186  0.0279  66   0.537   0.735   0.994   1.273   1.516   1.703
 17  0.9939  0.2124 -0.0571  0.1200  0.0279  59   0.531   0.731   0.994   1.276   1.522   1.712
 18  0.9939  0.2149 -0.0571  0.1214  0.0279  74   0.526   0.728   0.994   1.280   1.528   1.720
 19  0.9938  0.2174 -0.0571  0.1228  0.0279  82   0.520   0.725   0.994   1.283   1.535   1.728
        AVERAGE FOR ALL SITES (ARITHMETIC MEAN)   0.569   0.753   0.994   1.254   1.480   1.654
        REGIONAL GROWTH CURVE (HARMONIC MEAN)     0.568   0.753   0.994   1.254   1.479   1.653


INTER-SITE CORRELATION=  0.64


AVERAGE HETEROGENEITY MEASURES    1.08   -0.53   -0.71   (BASED ON   500 SIMULATIONS)
DISTRIBUTIONS ACCEPTED: GLO 0.0100, GEV 0.5900, LN3 0.6600, PE3 0.6900
DISTRIBUTIONS CHOSEN  : GLO 0.0600, GEV 0.3300, LN3 0.3600, PE3 0.2500


L-MOMENT ESTIMATION OF LOGNORMAL DISTRIBUTION          FAILURES:     0        WARNINGS:     0

        F      0.0100 0.1000 0.5000 0.9000 0.9900 0.9990

                  AVERAGE FOR ALL SITES
 ABS.BIAS       0.045  0.019  0.001  0.013  0.020  0.024
     BIAS      -0.002  0.000  0.000 -0.001 -0.002 -0.002
     RMSE       0.079  0.038  0.024  0.028  0.038  0.048
0.050 PT.       0.895  0.955  0.967  0.965  0.950  0.932
0.950 PT.       1.097  1.043  1.030  1.028  1.035  1.057

                  GROWTH CURVE COMPONENT
 ABS.BIAS       0.045  0.020  0.001  0.012  0.019  0.024
     BIAS      -0.001  0.000  0.000  0.000 -0.001 -0.001
     RMSE       0.074  0.028  0.005  0.017  0.032  0.044
0.050 PT.       0.910  0.965  0.992  0.982  0.960  0.946
0.950 PT.       1.090  1.029  1.009  1.020  1.039  1.059
```

Auxiliary routines

`FUNCTION DERF` Error function, double precision. The numerical method is the same as that of Algorithm 5666 of Hart et al. (1968).

`FUNCTION DIGAMD` Euler's psi function, $\psi(x) = d\{\log\Gamma(x)\}/dx$, double precision. The numerical method is the same as that of Algorithm AS103 (Bernardo, 1976).

`FUNCTION DLGAMA` Natural logarithm of the gamma function, double precision. The numerical method is the same as that of Algorithm ACM291 (Pike and Hill, 1966).

`SUBROUTINE DURAND(SEED,N,X)` Fills the double-precision array `X`, of length `N`, with pseudo-random numbers uniformly distributed on the interval $(0,1)$. The double-precision variable `SEED` is the seed for the random-number generator. It should be initialized to an integer value between 2 and 2147483647 ($2^{31}-1$), and left unchanged between successive calls to the routine. The routine uses the algorithm of Lewis et al. (1969).

This routine is supplied for completeness and portability, but on most computers it is more efficient to replace routine `DURAND` by a routine programmed in assembly language. For example, on an IBM 390, the Engineering and Scientific Subroutine Library, Version 2.1, contains a routine, also called `DURAND`, which has an identical specification to the one described here and is about 2.4 times as fast.

`FUNCTION GAMIND(X,A,G)` Incomplete gamma integral, double precision. The function is defined to be

$$G(x,\alpha) = \{\Gamma(\alpha)\}^{-1} \int_0^x t^{\alpha-1} e^{-t}\, dt\,.$$

The three arguments of the routine are x, α and $\log\{\Gamma(\alpha)\}$. The numerical method is the same as that of Algorithm AS239 (Shea, 1988), except that a different approximation to $G(x,\alpha)$—Hill's approximation (Johnson et al., 1994, eq. (18.35))—is used for large α.

`FUNCTION QUASTN` Quantile function of standard Normal distribution, double precision. The numerical method is the same as that of Algorithm AS241 (Wichura, 1988).

`SUBROUTINE SORT(X,N)` Sorts the double-precision array `X`, of length `N`, into ascending order.

Links between routines

The following functions and subroutines call other routines.

CDFGAM calls DERF, DLGAMA, GAMIND.

CDFGNO calls DERF.

CDFNOR calls DERF.

CDFPE3 calls DERF, DLGAMA, GAMIND.

CDFWAK calls QUAWAK.

LMRGAM calls DLGAMA.

LMRGEV calls DLGAMA.

LMRGNO calls DERF.

LMRKAP calls DLGAMA, DIGAMD.

LMRPE3 calls DLGAMA.

PELGAM calls DLGAMA.

PELGEV calls DLGAMA.

PELGNO calls DERF.

PELKAP calls DLGAMA, DIGAMD.

PELPE3 calls DLGAMA.

QUAGAM calls DERF, DLGAMA, GAMIND, QUASTN.

QUAGNO calls QUASTN.

QUANOR calls QUASTN.

QUAPE3 calls DERF, DLGAMA, GAMIND, QUAGAM, QUASTN.

CLUKM calls routines KMNS, OPTRA and QTRAN from Hartigan and Wong (1979).

REGTST calls most of the PELxxx, QUAxxx and auxiliary routines.

GAMIND calls DERF.

Machine dependence

Some constants initialized in data statements in some of the routines may need to be changed to give good results on different machines. The list below explains how the values of these constants should be chosen. The values assigned to these constants in the listings of the routines are appropriate for an IBM 390 computer using the FORTVS2 compiler, for which double-precision constants have precision of approximately 15 decimal digits and maximum magnitude approximately 10^{75}.

Routine	Constant	
`CDFxxx`	`SMALL`	[used by routines `CDFGEV`, `CDFGLO`, `CDFGNO`, `CDFGPA`, `CDFKAP`] Should be as small as possible, so long as underflow or overflow conditions do not occur.
`CDFWAK`	`UFL`	A number x such that the operation e^x just does not cause underflow.
`LMRKAP`	`OFL`	A number x such that the operation e^x just does not cause overflow.
`PELKAP`	`OFLEXP`	A number x such that the operation e^x just does not cause overflow.
`PELKAP`	`OFLGAM`	A number x such that the operation $\Gamma(x)$ just does not cause overflow.
`QUAWAK`	`UFL`	A number x such that the operation e^x just does not cause underflow.
`DERF`	`X1`	A number x just large enough that $\operatorname{erf} x = 1$ to machine accuracy.
`DLGAMA`	`BIG`	A number x such that $$\log\Gamma(x) = x - \tfrac{1}{2}\log x - x + \tfrac{1}{2}\log(2\pi)$$ to machine precision. The value $10^{d/2}$, where d is the number of decimal digits in a double-precision constant, should be adequate.
`DLGAMA`	`TOOBIG`	A number x such that the operation x^2 just does not cause overflow.
`GAMIND`	`UFL`	A number x such that the operation e^x just does not cause underflow.

In routines that use iterative methods (`CDFWAK`, `LMRGNO`, `PELGEV`, `PELKAP`, `QUAGAM`, `GAMIND`), constants `EPS` and `MAXIT` control the test for convergence of the iterations. If an iterative procedure fails frequently, it may be helpful to change the values of these constants.

Change history

Initial release (July 1988)

Routines: CDFGEV, CDFGLO, CDFGNO, CDFGPA, CDFGUM, CDFKAP, CDFWAK, LMRGEV, LMRGLO, LMRGNO, LMRGPA, LMRGUM, LMRKAP, LMRWAK, PELGEV, PELGLO, PELGNO, PELGPA, PELGUM, PELKAP, PELWAK, QUAGEV, QUAGLO, QUAGNO, QUAGPA, QUAGUM, QUAKAP, QUAWAK, SAMLMR, SAMPWM, SAMREG, SAMXMP, ALGAMD, DIGAMD, ERFD, QUASTN, SORT.

Version 2 (August 1991)

Routines added: CDFGAM, CDFNOR, CDFPE3, LMRGAM, LMRNOR, LMRPE3, PELGAM, PELNOR, PELPE3, QUAGAM, QUANOR, QUAPE3, REGLMR, REGTST, XFIT, XTEST, DERF, DLGAMA, GAMIND.

Routines modified: all except CDFGUM, LMRGPA, LMRWAK.

Routines replaced: ALGAMD (by DLGAMA), ERFD (by DERF), SAMREG (by REGLMR), SAMXMP (by XTEST).

Version 3 (August 1996)

Routines added: CDFEXP, LMREXP, PELEXP, QUAEXP, SAMLMU, CLUAGG, CLUINF, CLUKM, XCLUST, XSIM.

Routines modified:

CDFGUM	Removed initialization of an unused variable.
CDFWAK	Iterative procedure modified to reduce the (already small) chance that it will fail to converge. Minor bug fixes.
LMRKAP	Removed initialization of an unused variable.
PELGEV	More accurate numerical approximation.
PELGNO	Simpler and more accurate numerical approximation. Routine QUASTN no longer required. Returns $\alpha = -1$ if τ_3 is too large ($\lvert\tau_3\rvert \geq 0.95$).
PELKAP	Modified to reduce the chance of numeric overflow. In consequence, two new values of the IFAIL parameter are possible.
REGTST	Changed critical values for discordancy measure. Order in which output is printed was changed: table of parameters now printed before table of quantiles. Placement of IMPLICIT statement corrected: this can cause minor changes in the simulation results returned by the routine. Minor bugs and inconsistencies removed. Minor changes to comments.
XFIT	Minor bug fix. Minor changes to comments.
XTEST	Minor bug fixes. Minor changes to comments.
DERF	Corrected error when argument of function was exactly zero. Removed initialization of an unused variable.
DLGAMA	Minor bug fix.
DURAND	Speed increased.

References

BERNARDO, J. M. (1976). Algorithm AS 103: psi (digamma) function. *Applied Statistics*, **25**, 315–317.

CHURCHHOUSE, R. F. (ed.) (1981). *Numerical methods*, Handbook of applicable mathematics, vol. 3. New York: Wiley.

DONALDSON, R. W. (1996). Calculating inverse CV, skew and PWM functions for Pearson-3, log-normal, extreme-value, and log-logistic distributions. *Communications in Statistics—Simulation and Computation*, **25**, 741–747.

GREENWOOD, J. A., LANDWEHR, J. M., MATALAS, N. C., and WALLIS, J. R. (1979). Probability weighted moments: definition and relation to parameters of several distributions expressable in inverse form. *Water Resources Research*, **15**, 1049–1054.

HART, J. F., et al. (1968). *Computer approximations*. New York: Wiley.

HARTIGAN, J. A., and WONG, M. A. (1979). Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics*, **28**, 100–108.

HOSKING, J. R. M. (1990). L-moments: analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society, Series B*, **52**, 105–124.

HOSKING, J. R. M. (1994). The four-parameter kappa distribution. *IBM Journal of Research and Development*, **38**, 251–258.

HOSKING, J. R. M., and WALLIS, J. R. (1987). Parameter and quantile estimation for the generalized Pareto distribution. *Technometrics*, **29**, 339–349.

HOSKING, J. R. M., and WALLIS, J. R. (1995). A comparison of unbiased and plotting-position estimators of L-moments. *Water Resources Research*, **31**, 2019–2025.

HOSKING, J. R. M., and WALLIS, J. R. (1997). *Regional frequency analysis: an approach based on L-moments*. Cambridge, England: Cambridge University Press.

HOSKING, J. R. M., WALLIS, J. R., and WOOD, E. F. (1985). Estimation of the generalized extreme-value distribution by the method of probability-weighted moments. *Technometrics*, **27**, 251–261.

HUH, M. Y. (1986). Computation of percentage points. *Communications in Statistics—Simulation and Computation*, **15**, 1191–1198.

JOHNSON, N. L., KOTZ, S., and BALAKRISHNAN, N. (1994). *Continuous univariate distributions, vol. 1*, 2nd edn. New York: Wiley.

KARL, T. R., WILLIAMS, C. N., QUINLAN, F. T., and BODEN, T. A. (1990). United States historical climatology network (HCN) serial temperature and precipitation data. *ORNL/CDIAC-30 NDP-019/R1*, Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, Oak Ridge, Tenn.

KOTZ, S., JOHNSON, N. L., and READ, C. B. (eds.) (1988). *Encyclopedia of statistical sciences, vol. 9*. New York: Wiley.

LANDWEHR, J. M., MATALAS, N. C., and WALLIS, J. R. (1979a). Probability weighted moments compared with some traditional techniques in estimating Gumbel parameters and quantiles. *Water Resources Research*, **15**, 1055–1064.

LANDWEHR, J. M., MATALAS, N. C., and WALLIS, J. R. (1979b). Estimation of parameters and quantiles of Wakeby distributions. *Water Resources Research*, **15**, 1361–1379. Correction: *Water Resources Research*, **15** (1979), 1672.

LEWIS, P. A. W., GOODMAN, A. E., and MILLER, J. M. (1969). A pseudorandom number generator for the System/360. *IBM Systems Journal*, **8**, 136–146.

MURTAGH, F. (1985). *Multidimensional clustering algorithms.* Berlin: Springer.

NEUMAN, C. P., and SCHONBACH, D. I. (1974). Discrete (Legendre) orthogonal polynomials: a survey. *International Journal for Numerical Methods in Engineering*, **8**, 743–770.

PIKE, M. C., and HILL, I. D. (1966). Algorithm 291: logarithm of the gamma function. *Communications of the Association for Computing Machinery*, **9**, 684.

PLANTICO, M. S., KARL, T. R., KUKLA, G., and GAVIN, J. (1990). Is recent climate change across the United States related to rising levels of anthropogenic greenhouse gases? *Journal of Geophysical Research*, **95**, 16617–16637.

SHEA, B. L. (1988). Algorithm AS 239: chi-squared and incomplete gamma integral. *Applied Statistics*, **37**, 466–473.

SIMIU, E., CHANERY, M. J., and FILLIBEN, J. J. (1979). Extreme wind speeds at 129 stations in the contiguous United States. *Building Science Series 118*, National Bureau of Standards, Washington, D.C.

WALLIS, J. R. (1993). Regional frequency studies using L-moments. In *Concise encyclopedia of environmental systems*, ed. P. C. Young, 468–476. Oxford, U.K.: Pergamon.

WICHURA, M. (1988). Algorithm AS 241: the percentage points of the normal distribution. *Applied Statistics*, **37**, 477–484.

Listings of routines

```
      DOUBLE PRECISION FUNCTION CDFEXP(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE EXPONENTIAL DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2)
      DATA ZERO/0D0/,ONE/1D0/
      U=PARA(1)
      A=PARA(2)
      IF(A.LE.ZERO)GOTO 1000
      Y=(X-U)/A
      CDFEXP=ZERO
      IF(Y.LE.ZERO)RETURN
      CDFEXP=ONE-DEXP(-Y)
      RETURN
C
 1000 WRITE(6,7000)
      CDFEXP=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFEXP : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION CDFGAM(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE GAMMA DISTRIBUTION
C
C  OTHER ROUTINES USED: DERF,DLGAMA,GAMIND
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2)
      DATA ZERO/0D0/
      CDFGAM=ZERO
      ALPHA=PARA(1)
      BETA=PARA(2)
      IF(ALPHA.LE.ZERO.OR.BETA.LE.ZERO)GOTO 1000
      IF(X.LE.ZERO)RETURN
      CDFGAM=GAMIND(X/BETA,ALPHA,DLGAMA(ALPHA))
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFGAM : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION CDFGEV(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE GENERALIZED EXTREME-VALUE DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3)
      DATA ZERO/0D0/,ONE/1D0/
C
C         SMALL IS USED TO TEST WHETHER X IS EFFECTIVELY AT
C         THE ENDPOINT OF THE DISTRIBUTION
C
      DATA SMALL/1D-15/
C
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO)GOTO 1000
      Y=(X-U)/A
      IF(G.EQ.ZERO)GOTO 20
      ARG=ONE-G*Y
      IF(ARG.GT.SMALL)GOTO 10
      IF(G.LT.ZERO)CDFGEV=ZERO
      IF(G.GT.ZERO)CDFGEV=ONE
      RETURN
   10 Y=-DLOG(ARG)/G
   20 CDFGEV=DEXP(-DEXP(-Y))
      RETURN
C
 1000 WRITE(6,7000)
      CDFGEV=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFGEV : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION CDFGLO(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE GENERALIZED LOGISTIC DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3)
      DATA ZERO/0D0/,ONE/1D0/
C
C         SMALL IS USED TO TEST WHETHER X IS EFFECTIVELY AT
C         THE ENDPOINT OF THE DISTRIBUTION
C
      DATA SMALL/1D-15/
C
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO)GOTO 1000
      Y=(X-U)/A
      IF(G.EQ.ZERO)GOTO 20
      ARG=ONE-G*Y
      IF(ARG.GT.SMALL)GOTO 10
      IF(G.LT.ZERO)CDFGLO=ZERO
      IF(G.GT.ZERO)CDFGLO=ONE
      RETURN
   10 Y=-DLOG(ARG)/G
   20 CDFGLO=ONE/(ONE+DEXP(-Y))
      RETURN
C
 1000 WRITE(6,7000)
      CDFGLO=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFGLO : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION CDFGNO(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE GENERALIZED NORMAL DISTRIBUTION
C
C  OTHER ROUTINES USED: DERF
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/
      DATA RTHALF/0.70710 67811 86547 524D0/
C
C         SMALL IS USED TO TEST WHETHER X IS EFFECTIVELY AT
C         THE ENDPOINT OF THE DISTRIBUTION
C
      DATA SMALL/1D-15/
C
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO)GOTO 1000
      Y=(X-U)/A
      IF(G.EQ.ZERO)GOTO 20
      ARG=ONE-G*Y
      IF(ARG.GT.SMALL)GOTO 10
      IF(G.LT.ZERO)CDFGNO=ZERO
      IF(G.GT.ZERO)CDFGNO=ONE
      RETURN
   10 Y=-DLOG(ARG)/G
   20 CDFGNO=HALF+HALF*DERF(Y*RTHALF)
      RETURN
C
 1000 WRITE(6,7000)
      CDFGNO=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFGNO : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION CDFGPA(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE GENERALIZED PARETO DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3)
      DATA ZERO/0D0/,ONE/1D0/
C
C         SMALL IS USED TO TEST WHETHER X IS EFFECTIVELY AT
C         THE ENDPOINT OF THE DISTRIBUTION
C
      DATA SMALL/1D-15/
C
      CDFGPA=ZERO
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO)GOTO 1000
      Y=(X-U)/A
      IF(Y.LE.ZERO)RETURN
      IF(G.EQ.ZERO)GOTO 20
      ARG=ONE-G*Y
      IF(ARG.GT.SMALL)GOTO 10
      CDFGPA=ONE
      RETURN
   10 Y=-DLOG(ARG)/G
   20 CDFGPA=ONE-DEXP(-Y)
      RETURN
C
 1000 WRITE(6,7000)
      CDFGPA=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFGPA : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION CDFGUM(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE GUMBEL DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2)
      DATA ZERO/0D0/
      U=PARA(1)
      A=PARA(2)
      IF(A.LE.ZERO)GOTO 1000
      Y=(X-U)/A
      CDFGUM=DEXP(-DEXP(-Y))
      RETURN
C
 1000 WRITE(6,7000)
      CDFGUM=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFGUM : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION CDFKAP(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE KAPPA DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(4)
      DATA ZERO/0D0/,ONE/1D0/
C
C         SMALL IS A SMALL NUMBER, USED TO TEST WHETHER X IS
C         EFFECTIVELY AT AN ENDPOINT OF THE DISTRIBUTION
C
      DATA SMALL/1D-15/
C
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      H=PARA(4)
      IF(A.LE.ZERO)GOTO 1000
      Y=(X-U)/A
      IF(G.EQ.ZERO)GOTO 20
      ARG=ONE-G*Y
      IF(ARG.GT.SMALL)GOTO 10
      IF(G.LT.ZERO)CDFKAP=ZERO
      IF(G.GT.ZERO)CDFKAP=ONE
      RETURN
   10 Y=-DLOG(ARG)/G
   20 Y=DEXP(-Y)
      IF(H.EQ.ZERO)GOTO 40
      ARG=ONE-H*Y
      IF(ARG.GT.SMALL)GOTO 30
      CDFKAP=ZERO
      RETURN
   30 Y=-DLOG(ARG)/H
   40 CDFKAP=DEXP(-Y)
      RETURN
C
 1000 WRITE(6,7000)
      CDFKAP=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFKAP : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION CDFNOR(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE STANDARD NORMAL DISTRIBUTION
C
C  OTHER ROUTINES USED: DERF
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2)
      DATA HALF/0.5D0/,RTHALF/0.70710 67811 86547 524D0/
      CDFNOR=HALF+HALF*DERF((X-PARA(1))/PARA(2)*RTHALF)
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION CDFPE3(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DISTRIBUTION FUNCTION OF THE PEARSON TYPE 3 DISTRIBUTION
C
C  OTHER ROUTINES USED: DERF,DLGAMA,GAMIND
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/,TWO/2D0/,FOUR/4D0/
      DATA RTHALF/0.70710 67811 86547 524D0/
C
C         SMALL IS USED TO TEST WHETHER SKEWNESS IS EFFECTIVELY ZERO
C
      DATA SMALL/1D-6/
C
      CDFPE3=ZERO
      IF(PARA(2).LE.ZERO)GOTO 1000
      GAMMA=PARA(3)
      IF(DABS(GAMMA).LE.SMALL)GOTO 10
      ALPHA=FOUR/(GAMMA*GAMMA)
      Z=TWO*(X-PARA(1))/(PARA(2)*GAMMA)+ALPHA
      IF(Z.GT.ZERO)CDFPE3=GAMIND(Z,ALPHA,DLGAMA(ALPHA))
      IF(GAMMA.LT.ZERO)CDFPE3=ONE-CDFPE3
      RETURN
C
C         ZERO SKEWNESS
C
   10 Z=(X-PARA(1))/PARA(2)
      CDFPE3=HALF+HALF*DERF(Z*RTHALF)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFPE3 : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION CDFWAK(X,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  CUMULATIVE DISTRIBUTION FUNCTION OF THE WAKEBY DISTRIBUTION
C
C  OTHER ROUTINES USED: QUAWAK
C
C  METHOD: THE EQUATION X=G(Z), WHERE G(Z) IS THE WAKEBY QUANTILE
C  EXPRESSED AS A FUNCTION OF Z=-LOG(1-F), IS SOLVED USING HALLEY'S
C  METHOD (THE 2ND-ORDER ANALOGUE OF NEWTON-RAPHSON ITERATION).
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(5)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/
      DATA P1/0.1D0/,P7/0.7D0/,P99/0.99D0/
C
C         EPS,MAXIT CONTROL THE TEST FOR CONVERGENCE OF THE ITERATION
C         ZINCMX IS THE LARGEST PERMITTED ITERATIVE STEP
C         ZMULT CONTROLS WHAT HAPPENS WHEN THE ITERATION STEPS BELOW ZERO
C         UFL SHOULD BE CHOSEN SO THAT DEXP(UFL) JUST DOES NOT CAUSE
C           UNDERFLOW
C
      DATA EPS/1D-8/,MAXIT/20/,ZINCMX/3D0/,ZMULT/0.2D0/
      DATA UFL/-170D0/
C
      XI=PARA(1)
      A=PARA(2)
      B=PARA(3)
      C=PARA(4)
      D=PARA(5)
C
C         TEST FOR VALID PARAMETERS
C
      IF(B+D.LE.ZERO.AND.(B.NE.ZERO.OR.C.NE.ZERO.OR.D.NE.ZERO))GOTO 1000
      IF(A.EQ.ZERO.AND.B.NE.ZERO)GOTO 1000
      IF(C.EQ.ZERO.AND.D.NE.ZERO)GOTO 1000
      IF(C.LT.ZERO.OR.A+C.LT.ZERO)GOTO 1000
      IF(A.EQ.ZERO.AND.C.EQ.ZERO)GOTO 1000
C
      CDFWAK=ZERO
      IF(X.LE.XI)RETURN
C
C         TEST FOR SPECIAL CASES
C
      IF(B.EQ.ZERO.AND.C.EQ.ZERO.AND.D.EQ.ZERO)GOTO 100
      IF(C.EQ.ZERO)GOTO 110
      IF(A.EQ.ZERO)GOTO 120
C
C         GENERAL CASE
C
      CDFWAK=ONE
      IF(D.LT.ZERO.AND.X.GE.XI+A/B-C/D)RETURN
C
C         INITIAL VALUES FOR ITERATION:
C         IF X IS IN THE LOWEST DECILE OF THE DISTRIBUTION, START AT Z=0
C           (F=0);
C         IF X IS IN THE HIGHEST PERCENTILE OF THE DISTRIBUTION,
C           STARTING VALUE IS OBTAINED FROM ASYMPTOTIC FORM OF THE
```

```
C           DISTRIBUTION FOR LARGE Z (F NEAR 1);
C         OTHERWISE START AT Z=0.7 (CLOSE TO F=0.5).
C
      Z=P7
      IF(X.LT.QUAWAK(P1,PARA))Z=ZERO
      IF(X.LT.QUAWAK(P99,PARA))GOTO 10
      IF(D.LT.ZERO)Z=DLOG((X-XI-A/B)*D/C+ONE)/D
      IF(D.EQ.ZERO)Z=(X-XI-A/B)/C
      IF(D.GT.ZERO)Z=DLOG((X-XI)*D/C+ONE)/D
   10 CONTINUE
C
C         HALLEY'S METHOD, WITH MODIFICATIONS:
C         IF HALLEY ITERATION WOULD MOVE IN WRONG DIRECTION
C           (TEMP.LE.ZERO), USE ORDINARY NEWTON-RAPHSON INSTEAD;
C         IF STEP GOES TOO FAR (ZINC.GT.ZINCMX OR ZNEW.LE.ZERO),
C            LIMIT ITS LENGTH.
C
      DO 30 IT=1,MAXIT
      EB=ZERO
      BZ=-B*Z
      IF(BZ.GE.UFL)EB=DEXP(BZ)
      GB=Z
      IF(DABS(B).GT.EPS)GB=(ONE-EB)/B
      ED=DEXP(D*Z)
      GD=-Z
      IF(DABS(D).GT.EPS)GD=(ONE-ED)/D
      XEST=XI+A*GB-C*GD
      FUNC=X-XEST
      DERIV1=A*EB+C*ED
      DERIV2=-A*B*EB+C*D*ED
      TEMP=DERIV1+HALF*FUNC*DERIV2/DERIV1
      IF(TEMP.LE.ZERO)TEMP=DERIV1
      ZINC=FUNC/TEMP
      IF(ZINC.GT.ZINCMX)ZINC=ZINCMX
      ZNEW=Z+ZINC
      IF(ZNEW.LE.ZERO)GOTO 20
      Z=ZNEW
      IF(DABS(ZINC).LE.EPS)GOTO 200
      GOTO 30
   20 Z=Z*ZMULT
   30 CONTINUE
C
C         NOT CONVERGED
C
      WRITE(6,7010)
      GOTO 200
C
C         SPECIAL CASE B=C=D=0: WAKEBY IS EXPONENTIAL
C
  100 CONTINUE
      Z=(X-XI)/A
      GOTO 200
C
C         SPECIAL CASE C=0: WAKEBY IS GENERALIZED PARETO, BOUNDED ABOVE
C
  110 CONTINUE
      CDFWAK=ONE
      IF(X.GE.XI+A/B)RETURN
      Z=-DLOG(ONE-(X-XI)*B/A)/B
      GOTO 200
C
C         SPECIAL CASE A=0: WAKEBY IS GENERALIZED PARETO, NO UPPER BOUND
C
  120 CONTINUE
      Z=DLOG(ONE+(X-XI)*D/C)/D
      GOTO 200
C
C         CONVERT Z VALUE TO PROBABILITY
C
  200 CDFWAK=ONE
      IF(-Z.LT.UFL)RETURN
```

```
      CDFWAK=ONE-DEXP(-Z)
      RETURN
C
 1000 WRITE(6,7000)
      CDFWAK=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CDFWAK : PARAMETERS INVALID')
 7010 FORMAT(' ** WARNING ** ROUTINE CDFWAK :',
     *  ' ITERATION HAS NOT CONVERGED. RESULT MAY BE UNRELIABLE.')
      END
```

```
      SUBROUTINE LMREXP(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC17097, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE EXPONENTIAL DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 2. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER XI, ALPHA (LOCATION,
C                  SCALE).
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE L-MOMENTS
C                  LAMBDA-1, LAMBDA-2, TAU-3, TAU-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 20
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2),XMOM(NMOM)
      DATA ZERO/0D0/,HALF/0.5D0/,TWO/2D0/
C
      A=PARA(2)
      IF(A.LE.ZERO)GOTO 1000
      IF(NMOM.GT.20)GOTO 1010
      XMOM(1)=PARA(1)+A
      IF(NMOM.EQ.1)RETURN
      XMOM(2)=HALF*A
      IF(NMOM.EQ.2)RETURN
      DO 10 J=3,NMOM
   10 XMOM(J)=TWO/DFLOAT(J*(J-1))
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMREXP : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMREXP : PARAMETER NMOM TOO LARGE')
      END
```

```
      SUBROUTINE LMRGAM(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE GAMMA DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 2. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER ALPHA,BETA (SHAPE,SCALE).
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS UP TO 4 OF
C                  THE L-MOMENTS LAMBDA-1, LAMBDA-2, TAU-3, TAU-4.
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 4.
C
C  OTHER ROUTINES USED: DLGAMA
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2),XMOM(NMOM)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/
C
C         CONST IS 1/SQRT(PI)
C
      DATA CONST/0.56418 95835 47756 287D0/
C
C         COEFFICIENTS OF RATIONAL-FUNCTION APPROXIMATIONS
C         A0 IS 1/SQRT(3*PI)
C         C0 IS TAU-4 FOR THE NORMAL DISTRIBUTION
C
      DATA A0      / 0.32573501D+00/
      DATA A1,A2,A3/ 0.16869150D+00, 0.78327243D-01,-0.29120539D-02/
      DATA B1,B2   / 0.46697102D+00, 0.24255406D+00/
      DATA C0      / 0.12260172D+00/
      DATA C1,C2,C3/ 0.53730130D-01, 0.43384378D-01, 0.11101277D-01/
      DATA D1,D2   / 0.18324466D+00, 0.20166036D+00/
      DATA E1,E2,E3/ 0.23807576D+01, 0.15931792D+01, 0.11618371D+00/
      DATA F1,F2,F3/ 0.51533299D+01, 0.71425260D+01, 0.19745056D+01/
      DATA G1,G2,G3/ 0.21235833D+01, 0.41670213D+01, 0.31925299D+01/
      DATA H1,H2,H3/ 0.90551443D+01, 0.26649995D+02, 0.26193668D+02/
C
      ALPHA=PARA(1)
      BETA=PARA(2)
      IF(ALPHA.LE.ZERO.OR.BETA.LE.ZERO)GOTO 1000
      IF(NMOM.GT.4)GOTO 1010
C
C         LAMBDA-1
C
      XMOM(1)=ALPHA*BETA
      IF(NMOM.EQ.1)RETURN
C
C         LAMBDA-2
C
      XMOM(2)=BETA*CONST*DEXP(DLGAMA(ALPHA+HALF)-DLGAMA(ALPHA))
      IF(NMOM.EQ.2)RETURN
C
C         HIGHER MOMENTS
C
      IF(ALPHA.LT.ONE)GOTO 10
      Z=ONE/ALPHA
      XMOM(3)=DSQRT(Z)*(((A3*Z+A2)*Z+A1)*Z+A0)/((B2*Z+B1)*Z+ONE)
      IF(NMOM.EQ.3)RETURN
```

```
      XMOM(4)=(((C3*Z+C2)*Z+C1)*Z+C0)/((D2*Z+D1)*Z+ONE)
      IF(NMOM.GT.4)WRITE(6,7010)
      RETURN
C
   10 Z=ALPHA
      XMOM(3)=(((E3*Z+E2)*Z+E1)*Z+ONE)/(((F3*Z+F2)*Z+F1)*Z+ONE)
      IF(NMOM.EQ.3)RETURN
      XMOM(4)=(((G3*Z+G2)*Z+G1)*Z+ONE)/(((H3*Z+H2)*Z+H1)*Z+ONE)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRGAM : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRGAM : PARAMETER NMOM TOO LARGE')
      END
```

```
      SUBROUTINE LMRGEV(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE GENERALIZED EXTREME-VALUE DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER XI, ALPHA, K (LOCATION,
C                  SCALE, SHAPE).
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE L-MOMENTS
C                  LAMBDA-1, LAMBDA-2, TAU-3, TAU-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 20.
C
C  OTHER ROUTINES USED: DLGAMA
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3),XMOM(NMOM),ZMOM(20)
      DATA ZERO/0D0/,ONE/1D0/,TWO/2D0/,THREE/3D0/,FOUR/4D0/,SIX/6D0/
C
C         ARRAY ZMOM CONTAINS THE L-MOMENT RATIOS OF THE STANDARD
C         GUMBEL DISTRIBUTION (XI=0, ALPHA=1).
C         ZMOM(1) IS EULER'S CONSTANT, ZMOM(2) IS LOG(2).
C
      DATA ZMOM/
     *  0.57721 56649 01532 861D 0,  0.69314 71805 59945 309D 0,
     *  0.16992 50014 42312 363D 0,  0.15037 49927 88438 185D 0,
     *  0.55868 35005 77583 138D-1,  0.58110 02399 99710 876D-1,
     *  0.27624 25842 97309 125D-1,  0.30556 37665 79053 126D-1,
     *  0.16465 02822 58328 802D-1,  0.18784 66242 98170 912D-1,
     *  0.10932 82150 63027 148D-1,  0.12697 31266 76329 530D-1,
     *  0.77898 28180 57231 804D-2,  0.91483 61796 21999 726D-2,
     *  0.58333 23893 28363 588D-2,  0.69010 42875 90348 154D-2,
     *  0.45326 79701 80679 549D-2,  0.53891 68113 26595 459D-2,
     *  0.36240 77677 72368 790D-2,  0.43238 76086 05538 096D-2/
C
C         SMALL IS USED TO TEST WHETHER K IS EFFECTIVELY ZERO
C
      DATA SMALL/1D-6/
C
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO.OR.G.LE.-ONE)GOTO 1000
      IF(NMOM.GT.20)GOTO 1010
C
C         TEST FOR K=0
C
      IF(DABS(G).GT.SMALL)GOTO 5
      XMOM(1)=U
      IF(NMOM.EQ.1)RETURN
      XMOM(2)=A*ZMOM(2)
      IF(NMOM.EQ.2)RETURN
      DO 2 I=3,NMOM
    2 XMOM(I)=ZMOM(I)
      RETURN
    5 CONTINUE
C
C         FIRST 2 MOMENTS
```

```
C
   10 CONTINUE
      GAM=DEXP(DLGAMA(ONE+G))
      XMOM(1)=U+A*(ONE-GAM)/G
      IF(NMOM.EQ.1)RETURN
      XX2=ONE-TWO**(-G)
      XMOM(2)=A*XX2*GAM/G
      IF(NMOM.EQ.2)RETURN
C
C         HIGHER MOMENTS
C
      Z0=ONE
      DO 50 J=3,NMOM
      DJ=J
      BETA=(ONE-DJ**(-G))/XX2
      Z0=Z0*(FOUR*DJ-SIX)/DJ
      Z=Z0*THREE*(DJ-ONE)/(DJ+ONE)
      SUM=Z0*BETA-Z
      IF(J.EQ.3)GOTO 40
      DO 30 I=2,J-2
      DI=I
      Z=Z*(DI+DI+ONE)*(DJ-DI)/((DI+DI-ONE)*(DJ+DI))
      SUM=SUM-Z*XMOM(I+1)
   30 CONTINUE
   40 XMOM(J)=SUM
   50 CONTINUE
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRGEV : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRGEV : PARAMETER NMOM TOO LARGE')
      END
```

```
      SUBROUTINE LMRGLO(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE GENERALIZED LOGISTIC DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER XI, ALPHA, K (LOCATION,
C                  SCALE, SHAPE).
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE L-MOMENTS
C                  LAMBDA-1, LAMBDA-2, TAU-3, TAU-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 20.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3),XMOM(NMOM),Z(10,20)
      DATA ZERO/0D0/,ONE/1D0/
      DATA PI/3.141592653589793238D0/
C
C         SMALL IS USED TO DECIDE WHETHER TO APPROXIMATE THE FIRST 2
C         L-MOMENTS BY A POWER-SERIES EXPANSION WHEN G IS NEAR ZERO.
C         C1,C2 ARE COEFFICIENTS OF THIS POWER-SERIES EXPANSION.
C         C1 IS PI**2/6, C2 IS 7*PI**4/360.
C
      DATA SMALL/1D-4/
      DATA C1,C2/
     *  0.16449 34066 84822 644D 1,  0.18940 65658 99449 184D 1/
C
C         Z-ARRAY CONTAINS COEFFICIENTS OF THE REPRESENTATIONS OF
C         L-MOMENT RATIOS AS POLYNOMIALS IN THE SHAPE PARAMETER K
C
      DATA Z(1,3)/1D0/
      DATA (Z(I, 4),I=1, 2)/
     *  0.16666 66666 66666 667D 0,  0.83333 33333 33333 333D 0/
      DATA (Z(I, 5),I=1, 2)/
     *  0.41666 66666 66666 667D 0,  0.58333 33333 33333 333D 0/
      DATA (Z(I, 6),I=1, 3)/
     *  0.66666 66666 66666 667D-1,  0.58333 33333 33333 333D 0,
     *  0.35000 00000 00000 000D 0/
      DATA (Z(I, 7),I=1, 3)/
     *  0.23333 33333 33333 333D 0,  0.58333 33333 33333 333D 0,
     *  0.18333 33333 33333 333D 0/
      DATA (Z(I, 8),I=1, 4)/
     *  0.35714 28571 42857 143D-1,  0.42083 33333 33333 333D 0,
     *  0.45833 33333 33333 333D 0,  0.85119 04761 90476 190D-1/
      DATA (Z(I, 9),I=1, 4)/
     *  0.15099 20634 92063 492D 0,  0.51562 50000 00000 000D 0,
     *  0.29791 66666 66666 667D 0,  0.35466 26984 12698 413D-1/
      DATA (Z(I,10),I=1, 5)/
     *  0.22222 22222 22222 222D-1,  0.31889 32980 59964 727D 0,
     *  0.47997 68518 51851 852D 0,  0.16550 92592 59259 259D 0,
     *  0.13398 36860 67019 400D-1/
      DATA (Z(I,11),I=1, 5)/
     *  0.10650 79365 07936 508D 0,  0.44766 31393 29805 996D 0,
     *  0.36081 01851 85185 185D 0,  0.80390 21164 02116 402D-1,
     *  0.46285 27336 86067 019D-2/
      DATA (Z(I,12),I=1, 6)/
     *  0.15151 51515 15151 515D-1,  0.25131 61375 66137 566D 0,
     *  0.46969 52160 49382 716D 0,  0.22765 04629 62962 963D 0,
```

```
     * 0.34713 95502 64550 265D-1,  0.14727 13243 54657 688D-2/
      DATA (Z(I,13),I=1, 6)/
     * 0.79569 50456 95045 695D-1,  0.38976 59465 02057 613D 0,
     * 0.39291 73096 70781 893D 0,  0.12381 31062 61022 928D 0,
     * 0.13499 87139 91769 547D-1,  0.43426 15974 56041 900D-3/
      DATA (Z(I,14),I=1, 7)/
     * 0.10989 01098 90109 890D-1,  0.20413 29966 32996 633D 0,
     * 0.44773 66255 14403 292D 0,  0.27305 34428 27748 383D 0,
     * 0.59191 74382 71604 938D-1,  0.47768 77572 01646 091D-2,
     * 0.11930 26366 63747 775D-3/
      DATA (Z(I,15),I=1, 7)/
     * 0.61934 52050 59490 774D-1,  0.34203 17593 92870 504D 0,
     * 0.40701 37051 73427 396D 0,  0.16218 91928 06752 331D 0,
     * 0.25249 21002 35155 791D-1,  0.15509 34276 62872 107D-2,
     * 0.30677 82085 63922 850D-4/
      DATA (Z(I,16),I=1, 8)/
     * 0.83333 33333 33333 333D-2,  0.16976 83649 02293 474D 0,
     * 0.42219 12828 68366 202D 0,  0.30542 71728 94620 811D 0,
     * 0.84082 79399 72285 210D-1,  0.97243 57914 46208 113D-2,
     * 0.46528 02829 88616 322D-3,  0.74138 06706 96146 887D-5/
      DATA (Z(I,17),I=1, 8)/
     * 0.49716 60284 16028 416D-1,  0.30276 58385 89871 328D 0,
     * 0.41047 33000 89185 506D 0,  0.19483 90265 03251 764D 0,
     * 0.38659 80637 04648 526D-1,  0.34139 94076 42897 226D-2,
     * 0.12974 16173 71825 705D-3,  0.16899 11822 91033 482D-5/
      DATA (Z(I,18),I=1, 9)/
     * 0.65359 47712 41830 065D-2,  0.14387 48475 95085 690D 0,
     * 0.39643 28537 10259 464D 0,  0.32808 41807 20899 471D 0,
     * 0.10797 13931 65194 318D 0,  0.15965 33699 32077 769D-1,
     * 0.11012 77375 69143 819D-2,  0.33798 23645 82066 963D-4,
     * 0.36449 07853 33601 627D-6/
      DATA (Z(I,19),I=1, 9)/
     * 0.40878 45705 49276 431D-1,  0.27024 42907 25441 519D 0,
     * 0.40759 95245 14551 521D 0,  0.22211 14264 89320 008D 0,
     * 0.52846 38846 29533 398D-1,  0.59829 82392 72872 761D-2,
     * 0.32859 39655 65898 436D-3,  0.82617 91134 22830 354D-5,
     * 0.74603 37711 50646 605D-7/
      DATA (Z(I,20),I=1,10)/
     * 0.52631 57894 73684 211D-2,  0.12381 76557 53054 913D 0,
     * 0.37185 92914 44794 917D 0,  0.34356 87476 70189 607D 0,
     * 0.13019 86628 12524 058D 0,  0.23147 43648 99477 023D-1,
     * 0.20519 25194 79869 981D-2,  0.91205 82581 07571 930D-4,
     * 0.19023 86116 43414 884D-5,  0.14528 02606 97757 497D-7/
C
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO.OR.DABS(G).GE.ONE)GOTO 1000
      IF(NMOM.GT.20)GOTO 1010
C
C         FIRST 2 MOMENTS
C
      GG=G*G
      ALAM1=-G*(C1+GG*C2)
      ALAM2=ONE+GG*(C1+GG*C2)
      IF(DABS(G).GT.SMALL)ALAM2=G*PI/DSIN(G*PI)
      IF(DABS(G).GT.SMALL)ALAM1=(ONE-ALAM2)/G
      XMOM(1)=U+A*ALAM1
      IF(NMOM.EQ.1)RETURN
      XMOM(2)=A*ALAM2
      IF(NMOM.EQ.2)RETURN
C
C         HIGHER MOMENTS
C
      DO 20 M=3,NMOM
      KMAX=M/2
      SUM=Z(KMAX,M)
      DO 10 K=KMAX-1,1,-1
   10 SUM=SUM*GG+Z(K,M)
      IF(M.NE.M/2*2)SUM=-G*SUM
      XMOM(M)=SUM
```

```
   20 CONTINUE
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRGLO : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRGLO : PARAMETER NMOM TOO LARGE')
      END
```

```
      SUBROUTINE LMRGNO(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE GENERALIZED NORMAL DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER XI, ALPHA, K (LOCATION,
C                  SCALE, SHAPE).
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE L-MOMENTS
C                  LAMBDA-1, LAMBDA-2, TAU-3, TAU-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 20.
C
C  OTHER ROUTINES USED: DERF
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3),XMOM(NMOM),EST(20),ESTX(20),SUM(20),
     *  ZMOM(20)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/
C
C         ARRAY ZMOM CONTAINS L-MOMENTS OF THE STANDARD NORMAL DIST.
C
      DATA ZMOM/
     *  0D0,   0.56418 95835 47756 287D 0,
     *  0D0,   0.12260 17195 40890 947D 0,
     *  0D0,   0.43661 15389 50024 944D-1,
     *  0D0,   0.21843 13603 32508 776D-1,
     *  0D0,   0.12963 50158 01507 746D-1,
     *  0D0,   0.85296 21241 91705 402D-2,
     *  0D0,   0.60138 90151 79323 333D-2,
     *  0D0,   0.44555 82586 47650 150D-2,
     *  0D0,   0.34264 32435 78076 985D-2,
     *  0D0,   0.27126 79630 48139 365D-2/
C
C         RRT2 IS 1/SQRT(2), RRTPI IS 1/SQRT(PI)
C
      DATA RRT2 /0.70710 67811 86547 524D0/
      DATA RRTPI/0.56418 95835 47756 287D0/
C
C         RANGE,EPS,MAXIT CONTROL THE ITERATIVE PROCEDURE FOR NUMERICAL
C         INTEGRATION
C
      DATA RANGE/5D0/,EPS/1D-8/,MAXIT/10/
C
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO)GOTO 1000
      IF(NMOM.GT.20)GOTO 1010
C
C         TEST FOR K=0
C
      IF(DABS(G).GT.EPS)GOTO 5
      XMOM(1)=U
      IF(NMOM.EQ.1)RETURN
      XMOM(2)=A*ZMOM(2)
      IF(NMOM.EQ.2)RETURN
      DO 2 I=3,NMOM
```

```
    2 XMOM(I)=ZMOM(I)
      RETURN
    5 CONTINUE
C
C         LAMBDA-1
C
      EGG=DEXP(HALF*G*G)
      ALAM1=(ONE-EGG)/G
      XMOM(1)=U+A*ALAM1
      IF(NMOM.EQ.1)RETURN
C
C         LAMBDA-2
C
      ALAM2=EGG*DERF(HALF*G)/G
      XMOM(2)=A*ALAM2
      IF(NMOM.EQ.2)RETURN
C
C         HIGHER MOMENTS. THE INTEGRAL DEFINING LAMBDA-R IS EVALUATED
C         BY ITERATIVE APPLICATION OF THE TRAPEZIUM RULE.
C
C         - INITIAL ESTIMATE, USING 16 ORDINATES  (THE 'DO 20' LOOP
C           CALCULATES LEGENDRE POLYNOMIALS RECURSIVELY)
C
      CC=-G*RRT2
      XMIN=CC-RANGE
      XMAX=CC+RANGE
      DO 10 M=3,NMOM
   10 SUM(M)=ZERO
      N=16
      XINC=(XMAX-XMIN)/N
      DO 30 I=1,N-1
      X=XMIN+I*XINC
      E=DEXP(-((X-CC)**2))
      D=DERF(X)
      P1=ONE
      P=D
      DO 20 M=3,NMOM
      C1=M+M-3
      C2=M-2
      C3=M-1
      P2=P1
      P1=P
      P=(C1*D*P1-C2*P2)/C3
   20 SUM(M)=SUM(M)+E*P
   30 CONTINUE
      DO 40 M=3,NMOM
   40 EST(M)=SUM(M)*XINC
C
C         - DOUBLE THE NUMBER OF ORDINATES UNTIL CONVERGED
C
      DO 90 IT=1,MAXIT
      DO 50 M=3,NMOM
   50 ESTX(M)=EST(M)
      N=N*2
      XINC=(XMAX-XMIN)/N
      DO 70 I=1,N-1,2
      X=XMIN+I*XINC
      E=DEXP(-((X-CC)**2))
      D=DERF(X)
      P1=ONE
      P=D
      DO 60 M=3,NMOM
      C1=M+M-3
      C2=M-2
      C3=M-1
      P2=P1
      P1=P
      P=(C1*D*P1-C2*P2)/C3
   60 SUM(M)=SUM(M)+E*P
   70 CONTINUE
C
```

```
C         --- TEST FOR CONVERGENCE
C
      NOTCGD=0
      DO 80 M=NMOM,3,-1
      EST(M)=SUM(M)*XINC
      IF(DABS(EST(M)-ESTX(M)).GT.EPS*DABS(EST(M)))NOTCGD=M
   80 CONTINUE
      IF(NOTCGD.EQ.0)GOTO 100
   90 CONTINUE
C
      WRITE(6,7020)NOTCGD-1
  100 CONTINUE
      CONST=-DEXP(CC*CC)*RRTPI/(ALAM2*G)
      DO 110 M=3,NMOM
  110 XMOM(M)=CONST*EST(M)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRGNO : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRGNO : PARAMETER NMOM TOO LARGE')
 7020 FORMAT(' ** WARNING ** ROUTINE LMRGNO :',
     *  ' ITERATION HAS NOT CONVERGED. ONLY THE FIRST',I3,
     *  ' L-MOMENTS ARE RELIABLE.')
      END
```

```
      SUBROUTINE LMRGPA(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE GENERALIZED PARETO DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER XI, ALPHA, K (LOCATION,
C                  SCALE, SHAPE).
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE L-MOMENTS
C                  LAMBDA-1, LAMBDA-2, TAU-3, TAU-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 20.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(5),XMOM(NMOM)
      DATA ZERO/0D0/,ONE/1D0/,TWO/2D0/
C
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO.OR.G.LT.-ONE)GOTO 1000
      IF(NMOM.GT.20)GOTO 1010
C
C         LAMBDA-1
C
      Y=ONE/(ONE+G)
      XMOM(1)=U+A*Y
      IF(NMOM.EQ.1)RETURN
C
C         LAMBDA-2
C
      Y=Y/(TWO+G)
      XMOM(2)=A*Y
      IF(NMOM.EQ.2)RETURN
C
C         HIGHER MOMENTS
C
      Y=ONE
      DO 10 M=3,NMOM
      AM=M-TWO
      Y=Y*(AM-G)/(M+G)
      XMOM(M)=Y
   10 CONTINUE
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRGPA : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRGPA : PARAMETER NMOM TOO LARGE')
      END
```

```
      SUBROUTINE LMRGUM(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE GUMBEL DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 2. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER XI, ALPHA (LOCATION,
C                  SCALE).
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE L-MOMENTS
C                  LAMBDA-1, LAMBDA-2, TAU-3, TAU-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 20.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2),XMOM(NMOM),ZMOM(20)
      DATA ZERO/0D0/
C
C         ARRAY ZMOM CONTAINS THE L-MOMENT RATIOS OF THE STANDARD
C         GUMBEL DISTRIBUTION (XI=0, ALPHA=1).
C         ZMOM(1) IS EULER'S CONSTANT, ZMOM(2) IS LOG(2).
C
      DATA ZMOM/
     *  0.57721 56649 01532 861D 0,  0.69314 71805 59945 309D 0,
     *  0.16992 50014 42312 363D 0,  0.15037 49927 88438 185D 0,
     *  0.55868 35005 77583 138D-1,  0.58110 02399 99710 876D-1,
     *  0.27624 25842 97309 125D-1,  0.30556 37665 79053 126D-1,
     *  0.16465 02822 58328 802D-1,  0.18784 66242 98170 912D-1,
     *  0.10932 82150 63027 148D-1,  0.12697 31266 76329 530D-1,
     *  0.77898 28180 57231 804D-2,  0.91483 61796 21999 726D-2,
     *  0.58333 23893 28363 588D-2,  0.69010 42875 90348 154D-2,
     *  0.45326 79701 80679 549D-2,  0.53891 68113 26595 459D-2,
     *  0.36240 77677 72368 790D-2,  0.43238 76086 05538 096D-2/
C
      A=PARA(2)
      IF(A.LE.ZERO)GOTO 1000
      IF(NMOM.GT.20)GOTO 1010
      XMOM(1)=PARA(1)+A*ZMOM(1)
      IF(NMOM.EQ.1)RETURN
      XMOM(2)=A*ZMOM(2)
      IF(NMOM.EQ.2)RETURN
      DO 10 J=3,NMOM
   10 XMOM(J)=ZMOM(J)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRGUM : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRGUM : PARAMETER NMOM TOO LARGE')
      END
```

```
      SUBROUTINE LMRKAP(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE KAPPA DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 4. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER XI, ALPHA, K, H.
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE L-MOMENTS
C                  LAMBDA-1, LAMBDA-2, TAU-3, TAU-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 20.
C
C  OTHER ROUTINES USED: DLGAMA,DIGAMD
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(4),XMOM(NMOM),BETA(20)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/,THREE/3D0/,FOUR/4D0/,SIX/6D0/
C
C         EU  IS EULER'S CONSTANT
C
      DATA EU/0.577215664901532861D0/
C
C         SMALL IS USED TO TEST WHETHER H IS EFFECTIVELY ZERO
C         OFL SHOULD BE CHOSEN SO THAT EXP(OFL) JUST DOES NOT CAUSE
C         OVERFLOW
C
      DATA SMALL/1D-8/,OFL/170D0/
C
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      H=PARA(4)
C
C         TEST FOR FEASIBLE PARAMETERS
C
      IF(A.LE.ZERO)GOTO 1000
      IF(G.LE.-ONE)GOTO 1000
      IF(H.LT.ZERO.AND.G*H.LE.-ONE)GOTO 1000
      IF(NMOM.GT.20)GOTO 1010
C
C         CALCULATE FUNCTIONS OCCURRING IN THE PWM'S BETA-SUB-R
C
      DLGAM=DLGAMA(ONE+G)
      ICASE=1
      IF(H.GT.ZERO)ICASE=3
      IF(DABS(H).LT.SMALL)ICASE=2
      IF(G.EQ.ZERO)ICASE=ICASE+3
      GOTO(10,30,50,70,90,110),ICASE
C
C         - CASE H<0, G NONZERO
C
   10 DO 20 IR=1,NMOM
      R=IR
      ARG=DLGAM+DLGAMA(-R/H-G)-DLGAMA(-R/H)-G*DLOG(-H)
      IF(DABS(ARG).GT.OFL)GOTO 1020
   20 BETA(IR)=DEXP(ARG)
      GOTO 130
C
```

```
C         - CASE H SMALL, G NONZERO
C
   30 DO 40 IR=1,NMOM
      R=IR
   40 BETA(IR)=DEXP(DLGAM-G*DLOG(R))*(ONE-HALF*H*G*(ONE+G)/R)
      GOTO 130
C
C         - CASE H>0, G NONZERO
C
   50 DO 60 IR=1,NMOM
      R=IR
      ARG=DLGAM+DLGAMA(ONE+R/H)-DLGAMA(ONE+G+R/H)-G*DLOG(H)
      IF(DABS(ARG).GT.OFL)GOTO 1020
   60 BETA(IR)=DEXP(ARG)
      GOTO 130
C
C         - CASE H<0, G=0
C
   70 DO 80 IR=1,NMOM
      R=IR
   80 BETA(IR)=EU+DLOG(-H)+DIGAMD(-R/H)
      GOTO 130
C
C         - CASE H SMALL, G=0
C
   90 DO 100 IR=1,NMOM
      R=IR
  100 BETA(IR)=EU+DLOG(R)
      GOTO 130
C
C         - CASE H>0, G=0
C
  110 DO 120 IR=1,NMOM
      R=IR
  120 BETA(IR)=EU+DLOG(H)+DIGAMD(ONE+R/H)
      GOTO 130
C
C         LAMBDA-1
C
  130 CONTINUE
      IF(G.EQ.ZERO)XMOM(1)=U+A*BETA(1)
      IF(G.NE.ZERO)XMOM(1)=U+A*(ONE-BETA(1))/G
      IF(NMOM.EQ.1)RETURN
C
C         LAMBDA-2
C
      ALAM2=BETA(2)-BETA(1)
      IF(G.EQ.ZERO)XMOM(2)=A*ALAM2
      IF(G.NE.ZERO)XMOM(2)=A*ALAM2/(-G)
      IF(NMOM.EQ.2)RETURN
C
C         HIGHER MOMENTS
C
      Z0=ONE
      DO 170 J=3,NMOM
      DJ=J
      Z0=Z0*(FOUR*DJ-SIX)/DJ
      Z=Z0*THREE*(DJ-ONE)/(DJ+ONE)
      SUM=Z0*(BETA(J)-BETA(1))/ALAM2-Z
      IF(J.EQ.3)GOTO 160
      DO 150 I=2,J-2
      DI=I
      Z=Z*(DI+DI+ONE)*(DJ-DI)/((DI+DI-ONE)*(DJ+DI))
      SUM=SUM-Z*XMOM(I+1)
  150 CONTINUE
  160 XMOM(J)=SUM
  170 CONTINUE
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
```

```
 1010 WRITE(6,7010)
      RETURN
 1020 WRITE(6,7020)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRKAP : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRKAP : PARAMETER NMOM TOO LARGE')
 7020 FORMAT(' *** ERROR *** ROUTINE LMRKAP :',
     *  ' CALCULATIONS OF L-MOMENTS HAVE BROKEN DOWN')
      END
```

```
      SUBROUTINE LMRNOR(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE NORMAL DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 2. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER MU,SIGMA (LOCATION,SCALE).
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE L-MOMENTS
C                  LAMBDA-1, LAMBDA-2, TAU-3, TAU-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 20.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2),XMOM(NMOM),ZMOM(20)
      DATA ZERO/0D0/
C
C         ARRAY ZMOM CONTAINS L-MOMENTS OF THE STANDARD NORMAL DIST.
C
      DATA ZMOM/
     *  0D0,   0.56418 95835 47756 287D 0,
     *  0D0,   0.12260 17195 40890 947D 0,
     *  0D0,   0.43661 15389 50024 944D-1,
     *  0D0,   0.21843 13603 32508 776D-1,
     *  0D0,   0.12963 50158 01507 746D-1,
     *  0D0,   0.85296 21241 91705 402D-2,
     *  0D0,   0.60138 90151 79323 333D-2,
     *  0D0,   0.44555 82586 47650 150D-2,
     *  0D0,   0.34264 32435 78076 985D-2,
     *  0D0,   0.27126 79630 48139 365D-2/
C
      IF(PARA(2).LE.ZERO)GOTO 1000
      IF(NMOM.GT.20)GOTO 1010
      XMOM(1)=PARA(1)
      IF(NMOM.EQ.1)RETURN
      XMOM(2)=PARA(2)*ZMOM(2)
      IF(NMOM.EQ.2)RETURN
      DO 10 M=3,NMOM
   10 XMOM(M)=ZMOM(M)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRNOR : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRNOR : PARAMETER NMOM TOO LARGE')
      END
```

```
      SUBROUTINE LMRPE3(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE PEARSON TYPE 3 DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER MU, SIGMA, GAMMA (MEAN,
C                  S.D., SKEWNESS).
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS UP TO 4 OF
C                  THE L-MOMENTS LAMBDA-1, LAMBDA-2, TAU-3, TAU-4.
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 4.
C
C  OTHER ROUTINES USED: DLGAMA
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3),XMOM(NMOM)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/,FOUR/4D0/
C
C         SMALL IS USED TO TEST WHETHER SKEWNESS IS EFFECTIVELY ZERO
C
      DATA SMALL/1D-6/
C
C         CONST IS 1/SQRT(PI)
C
      DATA CONST/0.56418 95835 47756 287D0/
C
C         COEFFICIENTS OF RATIONAL-FUNCTION APPROXIMATIONS
C         A0 IS 1/SQRT(3*PI)
C         C0 IS TAU-4 FOR THE NORMAL DISTRIBUTION
C
      DATA A0      / 0.32573501D+00/
      DATA A1,A2,A3/ 0.16869150D+00, 0.78327243D-01,-0.29120539D-02/
      DATA B1,B2   / 0.46697102D+00, 0.24255406D+00/
      DATA C0      / 0.12260172D+00/
      DATA C1,C2,C3/ 0.53730130D-01, 0.43384378D-01, 0.11101277D-01/
      DATA D1,D2   / 0.18324466D+00, 0.20166036D+00/
      DATA E1,E2,E3/ 0.23807576D+01, 0.15931792D+01, 0.11618371D+00/
      DATA F1,F2,F3/ 0.51533299D+01, 0.71425260D+01, 0.19745056D+01/
      DATA G1,G2,G3/ 0.21235833D+01, 0.41670213D+01, 0.31925299D+01/
      DATA H1,H2,H3/ 0.90551443D+01, 0.26649995D+02, 0.26193668D+02/
C
      SD=PARA(2)
      IF(SD.LE.ZERO)GOTO 1000
      IF(NMOM.GT.4)GOTO 1010
C
C         LAMBDA-1
C
      XMOM(1)=PARA(1)
      IF(NMOM.EQ.1)RETURN
C
C         LAMBDA-2
C
      GAMMA=PARA(3)
      IF(DABS(GAMMA).LT.SMALL)GOTO 20
      ALPHA=FOUR/(GAMMA*GAMMA)
      BETA=DABS(HALF*SD*GAMMA)
      ALAM2=CONST*DEXP(DLGAMA(ALPHA+HALF)-DLGAMA(ALPHA))
```

```
      XMOM(2)=ALAM2*BETA
      IF(NMOM.EQ.2)RETURN
C
C         HIGHER MOMENTS
C
      IF(ALPHA.LT.ONE)GOTO 10
      Z=ONE/ALPHA
      XMOM(3)=DSQRT(Z)*(((A3*Z+A2)*Z+A1)*Z+A0)/((B2*Z+B1)*Z+ONE)
      IF(GAMMA.LT.ZERO)XMOM(3)=-XMOM(3)
      IF(NMOM.EQ.3)RETURN
      XMOM(4)=(((C3*Z+C2)*Z+C1)*Z+C0)/((D2*Z+D1)*Z+ONE)
      RETURN
C
   10 Z=ALPHA
      XMOM(3)=(((E3*Z+E2)*Z+E1)*Z+ONE)/(((F3*Z+F2)*Z+F1)*Z+ONE)
      IF(GAMMA.LT.ZERO)XMOM(3)=-XMOM(3)
      IF(NMOM.EQ.3)RETURN
      XMOM(4)=(((G3*Z+G2)*Z+G1)*Z+ONE)/(((H3*Z+H2)*Z+H1)*Z+ONE)
      IF(NMOM.GT.4)WRITE(6,7010)
      RETURN
C
C         CASE OF ZERO SKEWNESS
C
   20 XMOM(1)=PARA(1)
      IF(NMOM.EQ.1)RETURN
      XMOM(2)=CONST*PARA(2)
      IF(NMOM.EQ.2)RETURN
      XMOM(3)=0
      IF(NMOM.EQ.3)RETURN
      XMOM(4)=C0
      IF(NMOM.GT.4)WRITE(6,7010)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRPE3 : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRPE3 : PARAMETER NMOM TOO LARGE')
      END
```

```
      SUBROUTINE LMRWAK(PARA,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  L-MOMENT RATIOS FOR THE WAKEBY DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  PARA   * INPUT* ARRAY OF LENGTH 5. CONTAINS THE PARAMETERS OF THE
C                  DISTRIBUTION, IN THE ORDER XI, ALPHA, BETA, GAMMA,
C                  DELTA.
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE L-MOMENTS
C                  LAMBDA-1, LAMBDA-2, TAU-3, TAU-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 20.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(5),XMOM(NMOM)
      DATA ZERO/0D0/,ONE/1D0/,TWO/2D0/
C
      XI=PARA(1)
      A=PARA(2)
      B=PARA(3)
      C=PARA(4)
      D=PARA(5)
C
C         TEST FOR VALID PARAMETERS
C
      IF(D.GE.ONE)GOTO 1000
      IF(B+D.LE.ZERO.AND.(B.NE.ZERO.OR.C.NE.ZERO.OR.D.NE.ZERO))GOTO 1000
      IF(A.EQ.ZERO.AND.B.NE.ZERO)GOTO 1000
      IF(C.EQ.ZERO.AND.D.NE.ZERO)GOTO 1000
      IF(C.LT.ZERO)GOTO 1000
      IF(A+C.LT.ZERO)GOTO 1000
      IF(A.EQ.ZERO.AND.C.EQ.ZERO)GOTO 1000
      IF(NMOM.GT.20)GOTO 1010
C
C         LAMBDA-1
C
      Y=A/(ONE+B)
      Z=C/(ONE-D)
      XMOM(1)=XI+Y+Z
      IF(NMOM.EQ.1)RETURN
C
C         LAMBDA-2
C
      Y=Y/(TWO+B)
      Z=Z/(TWO-D)
      ALAM2=Y+Z
      XMOM(2)=ALAM2
      IF(NMOM.EQ.2)RETURN
C
C         HIGHER MOMENTS
C
      DO 10 M=3,NMOM
      AM=M
      Y=Y*(AM-TWO-B)/(AM+B)
      Z=Z*(AM-TWO+D)/(AM-D)
      XMOM(M)=(Y+Z)/ALAM2
   10 CONTINUE
      RETURN
```

```
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE LMRWAK : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE LMRWAK : PARAMETER NMOM TOO LARGE')
      END
```

```
      SUBROUTINE PELEXP(XMOM,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE EXPONENTIAL DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 2. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2.
C  PARA   *OUTPUT* ARRAY OF LENGTH 2. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER XI, ALPHA (LOCATION, SCALE).
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(2),PARA(2)
      DATA ZERO/0D0/,TWO/2D0/
C
      IF(XMOM(2).LE.ZERO)GOTO 1000
      PARA(2)=TWO*XMOM(2)
      PARA(1)=XMOM(1)-PARA(2)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE PELEXP : L-MOMENTS INVALID')
      END
```

```
      SUBROUTINE PELGAM(XMOM,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE GAMMA DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 2. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2.
C  PARA   *OUTPUT* ARRAY OF LENGTH 2. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER ALPHA, BETA (SHAPE, SCALE).
C
C  OTHER ROUTINES USED: DLGAMA
C
C  METHOD: RATIONAL APPROXIMATION IS USED TO EXPRESS ALPHA AS A FUNCTION
C  OF L-CV. RELATIVE ACCURACY OF THE  APPROXIMATION IS BETTER THAN 5E-5.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(2),PARA(2)
      DATA ZERO/0D0/,HALF/0.5D0/ONE/1D0/
C
C         CONSTANTS USED IN MINIMAX APPROXIMATIONS
C
      DATA A1,A2,A3/-0.3080D0,-0.05812D0,0.01765D0/
      DATA B1,B2,B3,B4/0.7213D0,-0.5947D0,-2.1817D0,1.2113D0/
      DATA PI/3.1415927D0/
C
      IF(XMOM(1).LE.XMOM(2).OR.XMOM(2).LE.ZERO)GOTO 1000
      CV=XMOM(2)/XMOM(1)
      IF(CV.GE.HALF)GOTO 10
      T=PI*CV*CV
      ALPHA=(ONE+A1*T)/(T*(ONE+T*(A2+T*A3)))
      GOTO 20
   10 CONTINUE
      T=ONE-CV
      ALPHA=T*(B1+T*B2)/(ONE+T*(B3+T*B4))
   20 CONTINUE
      PARA(1)=ALPHA
      PARA(2)=XMOM(1)/ALPHA
      RETURN
C
 1000 WRITE(6,7000)
      PARA(1)=ZERO
      PARA(2)=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE PELGAM : L-MOMENTS INVALID')
      END
```

```
      SUBROUTINE PELGEV(XMOM,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE GENERALIZED EXTREME-VALUE
C  DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2, TAU-3.
C  PARA   *OUTPUT* ARRAY OF LENGTH 3. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER XI, ALPHA, K (LOCATION, SCALE, SHAPE).
C
C  OTHER ROUTINES USED: DLGAMA
C
C  METHOD: FOR  -0.8 LE TAU3 LT 1,  K IS APPROXIMATED BY RATIONAL
C  FUNCTIONS AS IN DONALDSON (1996, COMMUN. STATIST. SIMUL. COMPUT.).
C  IF TAU3 IS OUTSIDE THIS RANGE, NEWTON-RAPHSON ITERATION IS USED.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(3),PARA(3)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/,TWO/2D0/,THREE/3D0/
      DATA P8/0.8D0/,P97/0.97D0/
C
C         SMALL IS USED TO TEST WHETHER K IS EFFECTIVELY ZERO
C         EPS,MAXIT CONTROL THE TEST FOR CONVERGENCE OF N-R ITERATION
C
      DATA SMALL/1D-5/,EPS/1D-6/,MAXIT/20/
C
C         EU IS EULER'S CONSTANT
C         DL2 IS LOG(2), DL3 IS LOG(3)
C
      DATA EU/0.57721566D0/,DL2/0.69314718D0/,DL3/1.0986123D0/
C
C         COEFFICIENTS OF RATIONAL-FUNCTION APPROXIMATIONS FOR K
C
      DATA A0,A1,A2/ 0.28377530D0,-1.21096399D0,-2.50728214D0/
      DATA A3,A4   /-1.13455566D0,-0.07138022D0/
      DATA B1,B2,B3/ 2.06189696D0, 1.31912239D0, 0.25077104D0/
      DATA C1,C2,C3/ 1.59921491D0,-0.48832213D0, 0.01573152D0/
      DATA D1,D2   /-0.64363929D0, 0.08985247D0/
C
      T3=XMOM(3)
      IF(XMOM(2).LE.ZERO)GOTO 1000
      IF(DABS(T3).GE.ONE)GOTO 1000
      IF(T3.LE.ZERO)GOTO 10
C
C         RATIONAL-FUNCTION APPROXIMATION FOR TAU3 BETWEEN 0 AND 1
C
      Z=ONE-T3
      G=(-ONE+Z*(C1+Z*(C2+Z*C3)))/(ONE+Z*(D1+Z*D2))
      IF(DABS(G).LT.SMALL)GOTO 50
      GOTO 40
C
C         RATIONAL-FUNCTION APPROXIMATION FOR TAU3 BETWEEN -0.8 AND 0
C
   10 G=(A0+T3*(A1+T3*(A2+T3*(A3+T3*A4))))/(ONE+T3*(B1+T3*(B2+T3*B3)))
      IF(T3.GE.-P8)GOTO 40
C
```

```
C         NEWTON-RAPHSON ITERATION FOR TAU3 LESS THAN -0.8
C
      IF(T3.LE.-P97)G=ONE-DLOG(ONE+T3)/DL2
      T0=(T3+THREE)*HALF
      DO 20 IT=1,MAXIT
      X2=TWO**(-G)
      X3=THREE**(-G)
      XX2=ONE-X2
      XX3=ONE-X3
      T=XX3/XX2
      DERIV=(XX2*X3*DL3-XX3*X2*DL2)/(XX2*XX2)
      GOLD=G
      G=G-(T-T0)/DERIV
      IF(DABS(G-GOLD).LE.EPS*G)GOTO 30
   20 CONTINUE
      WRITE(6,7010)
   30 CONTINUE
C
C         ESTIMATE ALPHA,XI
C
   40 PARA(3)=G
      GAM=DEXP(DLGAMA(ONE+G))
      PARA(2)=XMOM(2)*G/(GAM*(ONE-TWO**(-G)))
      PARA(1)=XMOM(1)-PARA(2)*(ONE-GAM)/G
      RETURN
C
C         ESTIMATED K EFFECTIVELY ZERO
C
   50 PARA(3)=ZERO
      PARA(2)=XMOM(2)/DL2
      PARA(1)=XMOM(1)-EU*PARA(2)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE PELGEV : L-MOMENTS INVALID')
 7010 FORMAT(' ** WARNING ** ROUTINE PELGEV :',
     *  ' ITERATION HAS NOT CONVERGED. RESULTS MAY BE UNRELIABLE.')
      END
```

```
      SUBROUTINE PELGLO(XMOM,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE GENERALIZED LOGISTIC
C  DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2, TAU-3.
C  PARA   *OUTPUT* ARRAY OF LENGTH 3. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER XI, ALPHA, K (LOCATION, SCALE, SHAPE).
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(3),PARA(3)
      DATA ZERO/0D0/,ONE/1D0/
      DATA PI/3.141592653589793238D0/
C
C         SMALL IS USED TO TEST WHETHER K IS EFFECTIVELY ZERO
C
      DATA SMALL/1D-6/
C
C         ESTIMATE K
C
      G=-XMOM(3)
      IF(XMOM(2).LE.ZERO.OR.DABS(G).GE.ONE)GOTO 1000
      IF(DABS(G).LE.SMALL)GOTO 10
C
C         ESTIMATE ALPHA, XI
C
      GG=G*PI/DSIN(G*PI)
      A=XMOM(2)/GG
      PARA(1)=XMOM(1)-A*(ONE-GG)/G
      PARA(2)=A
      PARA(3)=G
      RETURN
C
C         ESTIMATED K EFFECTIVELY ZERO
C
   10 PARA(3)=ZERO
      PARA(2)=XMOM(2)
      PARA(1)=XMOM(1)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE PELGLO : L-MOMENTS INVALID')
      END
```

```
      SUBROUTINE PELGNO(XMOM,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE GENERALIZED NORMAL
C  DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2, TAU-3. ABS(TAU3) MAY NOT EXCEED 0.95.
C  PARA   *OUTPUT* ARRAY OF LENGTH 3. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER XI, ALPHA, K (LOCATION, SCALE, SHAPE).
C
C  OTHER ROUTINES USED: DERF
C
C  METHOD: RATIONAL-FUNCTION APPROXIMATION OF K IN TERMS OF TAU-3
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(3),PARA(3)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/
      DATA P95/0.95D0/
      DATA ROOTPI/1.772453850905516027D0/
C
C         COEFFICIENTS OF RATIONAL-FUNCTION APPROXIMATION
C         A0 IS 0.5*SQRT(3/PI)
C
      DATA A0,A1,A2,A3/
     *  0.20466534D+01,-0.36544371D+01,0.18396733D+01,-0.20360244D+00/
      DATA B1,B2,B3/-0.20182173D+01,0.12420401D+01,-0.21741801D+00/
C
C         SMALL IS USED TO TEST WHETHER SKEWNESS IS EFFECTIVELY ZERO
C
      DATA SMALL/1D-8/
C
      T3=XMOM(3)
      IF(XMOM(2).LE.ZERO.OR.DABS(T3).GE.ONE)GOTO 1000
      IF(DABS(T3).GE.P95)GOTO 1010
      IF(DABS(T3).LE.SMALL)GOTO 30
C
      TT=T3*T3
      G=-T3*(A0+TT*(A1+TT*(A2+TT*A3)))/(ONE+TT*(B1+TT*(B2+TT*B3)))
      E=DEXP(HALF*G*G)
      A=XMOM(2)*G/(E*DERF(HALF*G))
      U=XMOM(1)+A*(E-ONE)/G
      PARA(1)=U
      PARA(2)=A
      PARA(3)=G
      RETURN
C
   30 PARA(1)=XMOM(1)
      PARA(2)=XMOM(2)*ROOTPI
      PARA(3)=ZERO
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      PARA(1)=ZERO
      PARA(2)=-ONE
```

```
      PARA(3)=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE PELGNO : L-MOMENTS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE PELGNO :',
     *  ' TAU-3 TOO LARGE FOR ROUTINE')
      END
```

```
      SUBROUTINE PELGPA(XMOM,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR  THE GENERALIZED PARETO
C  DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2, TAU-3.
C  PARA   *OUTPUT* ARRAY OF LENGTH 3. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER XI, ALPHA, K (LOCATION, SCALE, SHAPE).
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(3),PARA(3)
      DATA ZERO/0D0/,ONE/1D0/,TWO/2D0/,THREE/3D0/
C
      T3=XMOM(3)
      IF(XMOM(2).LE.ZERO)GOTO 1000
      IF(DABS(T3).GE.ONE)GOTO 1000
      G=(ONE-THREE*T3)/(ONE+T3)
      PARA(3)=G
      PARA(2)=(ONE+G)*(TWO+G)*XMOM(2)
      PARA(1)=XMOM(1)-PARA(2)/(ONE+G)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE PELGPA : L-MOMENTS INVALID')
      END
```

```
      SUBROUTINE PELGUM(XMOM,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE GUMBEL DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 2. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2.
C  PARA   *OUTPUT* ARRAY OF LENGTH 2. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER XI, ALPHA (LOCATION, SCALE).
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(2),PARA(2)
      DATA ZERO/0D0/
C
C         EU IS EULER'S CONSTANT, DL2 IS LOG(2)
C
      DATA EU/0.577215664901532861D0/,DL2/0.693147180559945309D0/
C
      IF(XMOM(2).LE.ZERO)GOTO 1000
      PARA(2)=XMOM(2)/DL2
      PARA(1)=XMOM(1)-EU*PARA(2)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE PELGUM : L-MOMENTS INVALID')
      END
```

```
      SUBROUTINE PELKAP(XMOM,PARA,IFAIL)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE KAPPA DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 4. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2, TAU-3, TAU-4.
C  PARA   *OUTPUT* ARRAY OF LENGTH 4. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER XI, ALPHA, K, H.
C  IFAIL  *OUTPUT* FAIL FLAG. ON EXIT, IT IS SET AS FOLLOWS.
C                  0  SUCCESSFUL EXIT
C                  1  L-MOMENTS INVALID
C                  2  (TAU-3, TAU-4) LIES ABOVE THE GENERALIZED-LOGISTIC
C                     LINE (SUGGESTS THAT L-MOMENTS ARE NOT CONSISTENT
C                     WITH ANY KAPPA DISTRIBUTION WITH H.GT.-1)
C                  3  ITERATION FAILED TO CONVERGE
C                  4  UNABLE TO MAKE PROGRESS FROM CURRENT POINT IN
C                     ITERATION
C                  5  ITERATION ENCOUNTERED NUMERICAL DIFFICULTIES -
C                     OVERFLOW WOULD HAVE BEEN LIKELY TO OCCUR
C                  6  ITERATION FOR H AND K CONVERGED, BUT OVERFLOW
C                     WOULD HAVE OCCURRED WHEN CALCULATING XI AND ALPHA
C
C  N.B.  PARAMETERS ARE SOMETIMES NOT UNIQUELY DEFINED BY THE FIRST 4
C  L-MOMENTS. IN SUCH CASES THE ROUTINE RETURNS THE SOLUTION FOR WHICH
C  THE H PARAMETER IS LARGEST.
C
C  OTHER ROUTINES USED: DLGAMA,DIGAMD
C
C  THE SHAPE PARAMETERS K AND H ARE ESTIMATED USING NEWTON-RAPHSON
C  ITERATION ON THE RELATIONSHIP BETWEEN (TAU-3,TAU-4) AND (K,H).
C  THE CONVERGENCE CRITERION IS THAT TAU-3 AND TAU-4 CALCULATED FROM
C  THE ESTIMATED VALUES OF K AND H SHOULD DIFFER BY LESS THAN 'EPS'
C  FROM THE VALUES SUPPLIED IN ARRAY XMOM.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(4),PARA(4)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/,TWO/2D0/,THREE/3D0/,FOUR/4D0/
      DATA FIVE/5D0/,SIX/6D0/,TWELVE/12D0/,TWENTY/20D0/,THIRTY/30D0/
      DATA P725/0.725D0/,P8/0.8D0/
C
C         EPS,MAXIT CONTROL THE TEST FOR CONVERGENCE OF N-R ITERATION
C         MAXSR IS THE MAX. NO. OF STEPLENGTH REDUCTIONS PER ITERATION
C         HSTART IS THE STARTING VALUE FOR H
C         BIG IS USED TO INITIALIZE THE CRITERION FUNCTION
C         OFLEXP IS SUCH THAT DEXP(OFLEXP) JUST DOES NOT CAUSE OVERFLOW
C         OFLGAM IS SUCH THAT DEXP(DLGAMA(OFLGAM)) JUST DOES NOT CAUSE
C           OVERFLOW
C
      DATA EPS/1D-6/,MAXIT/20/,MAXSR/10/,HSTART/1.001D0/,BIG/10D0/
      DATA OFLEXP/170D0/,OFLGAM/53D0/
C
      T3=XMOM(3)
      T4=XMOM(4)
      DO 10 I=1,4
   10 PARA(I)=ZERO
C
```

```
C         TEST FOR FEASIBILITY
C
      IF(XMOM(2).LE.ZERO)GOTO 1000
      IF(DABS(T3).GE.ONE.OR.DABS(T4).GE.ONE)GOTO 1000
      IF(T4.LE.(FIVE*T3*T3-ONE)/FOUR)GOTO 1000
      IF(T4.GE.(FIVE*T3*T3+ONE)/SIX )GOTO 1010
C
C         SET STARTING VALUES FOR N-R ITERATION:
C         G IS CHOSEN TO GIVE THE CORRECT VALUE OF TAU-3 ON THE
C         ASSUMPTION THAT H=1 (I.E. A GENERALIZED PARETO FIT) -
C         BUT H IS ACTUALLY SET TO 1.001 TO AVOID NUMERICAL
C         DIFFICULTIES WHICH CAN SOMETIMES ARISE WHEN H=1 EXACTLY
C
      G=(ONE-THREE*T3)/(ONE+T3)
      H=HSTART
      Z=G+H*P725
      XDIST=BIG
C
C         START OF NEWTON-RAPHSON ITERATION
C
      DO 100 IT=1,MAXIT
C
C         REDUCE STEPLENGTH UNTIL WE ARE NEARER TO THE REQUIRED
C         VALUES OF TAU-3 AND TAU-4 THAN WE WERE AT THE PREVIOUS STEP
C
      DO 40 I=1,MAXSR
C
C         - CALCULATE CURRENT TAU-3 AND TAU-4
C
C           NOTATION:
C           U.    - RATIOS OF GAMMA FUNCTIONS WHICH OCCUR IN THE PWM'S
C                   BETA-SUB-R
C           ALAM. - L-MOMENTS (APART FROM A LOCATION AND SCALE SHIFT)
C           TAU.  - L-MOMENT RATIOS
C
      IF(G.GT.OFLGAM)GOTO 1020
      IF(H.GT.ZERO)GOTO 20
      U1=DEXP(DLGAMA(  -ONE/H-G)-DLGAMA(  -ONE/H+ONE))
      U2=DEXP(DLGAMA(  -TWO/H-G)-DLGAMA(  -TWO/H+ONE))
      U3=DEXP(DLGAMA(-THREE/H-G)-DLGAMA(-THREE/H+ONE))
      U4=DEXP(DLGAMA( -FOUR/H-G)-DLGAMA( -FOUR/H+ONE))
      GOTO 30
   20 U1=DEXP(DLGAMA(  ONE/H)-DLGAMA(  ONE/H+ONE+G))
      U2=DEXP(DLGAMA(  TWO/H)-DLGAMA(  TWO/H+ONE+G))
      U3=DEXP(DLGAMA(THREE/H)-DLGAMA(THREE/H+ONE+G))
      U4=DEXP(DLGAMA( FOUR/H)-DLGAMA( FOUR/H+ONE+G))
   30 CONTINUE
      ALAM2=U1-TWO*U2
      ALAM3=-U1+SIX*U2-SIX*U3
      ALAM4=U1-TWELVE*U2+THIRTY*U3-TWENTY*U4
      IF(ALAM2.EQ.ZERO)GOTO 1020
      TAU3=ALAM3/ALAM2
      TAU4=ALAM4/ALAM2
      E1=TAU3-T3
      E2=TAU4-T4
C
C         - IF NEARER THAN BEFORE, EXIT THIS LOOP
C
      DIST=DMAX1(DABS(E1),DABS(E2))
      IF(DIST.LT.XDIST)GOTO 50
C
C         - OTHERWISE, HALVE THE STEPLENGTH AND TRY AGAIN
C
      DEL1=HALF*DEL1
      DEL2=HALF*DEL2
      G=XG-DEL1
      H=XH-DEL2
   40 CONTINUE
C
C         TOO MANY STEPLENGTH REDUCTIONS
C
```

```
      IFAIL=4
      RETURN
C
C         TEST FOR CONVERGENCE
C
   50 CONTINUE
      IF(DIST.LT.EPS)GOTO 110
C
C         NOT CONVERGED: CALCULATE NEXT STEP
C
C         NOTATION:
C         U1G  - DERIVATIVE OF U1 W.R.T. G
C         DL2G - DERIVATIVE OF ALAM2 W.R.T. G
C         D..  - MATRIX OF DERIVATIVES OF TAU-3 AND TAU-4 W.R.T. G AND H
C         H..  - INVERSE OF DERIVATIVE MATRIX
C         DEL. - STEPLENGTH
C
      XG=G
      XH=H
      XZ=Z
      XDIST=DIST
      RHH=ONE/(H*H)
      IF(H.GT.ZERO)GOTO 60
      U1G=-U1*DIGAMD(  -ONE/H-G)
      U2G=-U2*DIGAMD(  -TWO/H-G)
      U3G=-U3*DIGAMD(-THREE/H-G)
      U4G=-U4*DIGAMD( -FOUR/H-G)
      U1H=      RHH*(-U1G-U1*DIGAMD(  -ONE/H+ONE))
      U2H=  TWO*RHH*(-U2G-U2*DIGAMD(  -TWO/H+ONE))
      U3H=THREE*RHH*(-U3G-U3*DIGAMD(-THREE/H+ONE))
      U4H= FOUR*RHH*(-U4G-U4*DIGAMD( -FOUR/H+ONE))
      GOTO 70
   60 U1G=-U1*DIGAMD(  ONE/H+ONE+G)
      U2G=-U2*DIGAMD(  TWO/H+ONE+G)
      U3G=-U3*DIGAMD(THREE/H+ONE+G)
      U4G=-U4*DIGAMD( FOUR/H+ONE+G)
      U1H=      RHH*(-U1G-U1*DIGAMD(  ONE/H))
      U2H=  TWO*RHH*(-U2G-U2*DIGAMD(  TWO/H))
      U3H=THREE*RHH*(-U3G-U3*DIGAMD(THREE/H))
      U4H= FOUR*RHH*(-U4G-U4*DIGAMD( FOUR/H))
   70 CONTINUE
      DL2G=U1G-TWO*U2G
      DL2H=U1H-TWO*U2H
      DL3G=-U1G+SIX*U2G-SIX*U3G
      DL3H=-U1H+SIX*U2H-SIX*U3H
      DL4G=U1G-TWELVE*U2G+THIRTY*U3G-TWENTY*U4G
      DL4H=U1H-TWELVE*U2H+THIRTY*U3H-TWENTY*U4H
      D11=(DL3G-TAU3*DL2G)/ALAM2
      D12=(DL3H-TAU3*DL2H)/ALAM2
      D21=(DL4G-TAU4*DL2G)/ALAM2
      D22=(DL4H-TAU4*DL2H)/ALAM2
      DET=D11*D22-D12*D21
      H11= D22/DET
      H12=-D12/DET
      H21=-D21/DET
      H22= D11/DET
      DEL1=E1*H11+E2*H12
      DEL2=E1*H21+E2*H22
C
C         TAKE NEXT N-R STEP
C
      G=XG-DEL1
      H=XH-DEL2
      Z=G+H*P725
C
C         REDUCE STEP IF G AND H ARE OUTSIDE THE PARAMETER SPACE
C
      FACTOR=ONE
      IF(G.LE.-ONE)FACTOR=P8*(XG+ONE)/DEL1
      IF(H.LE.-ONE)FACTOR=DMIN1(FACTOR,P8*(XH+ONE)/DEL2)
      IF(Z.LE.-ONE)FACTOR=DMIN1(FACTOR,P8*(XZ+ONE)/(XZ-Z))
```

```
      IF(H.LE.ZERO.AND.G*H.LE.-ONE)
     *  FACTOR=DMIN1(FACTOR,P8*(XG*XH+ONE)/(XG*XH-G*H))
      IF(FACTOR.EQ.ONE)GOTO 80
      DEL1=DEL1*FACTOR
      DEL2=DEL2*FACTOR
      G=XG-DEL1
      H=XH-DEL2
      Z=G+H*P725
   80 CONTINUE
C
C         END OF NEWTON-RAPHSON ITERATION
C
  100 CONTINUE
C
C         NOT CONVERGED
C
      IFAIL=3
      RETURN
C
C         CONVERGED
C
  110 IFAIL=0
      PARA(4)=H
      PARA(3)=G
      TEMP=DLGAMA(ONE+G)
      IF(TEMP.GT.OFLEXP)GOTO 1030
      GAM=DEXP(TEMP)
      TEMP=(ONE+G)*DLOG(DABS(H))
      IF(TEMP.GT.OFLEXP)GOTO 1030
      HH=DEXP(TEMP)
      PARA(2)=XMOM(2)*G*HH/(ALAM2*GAM)
      PARA(1)=XMOM(1)-PARA(2)/G*(ONE-GAM*U1/HH)
      RETURN
C
 1000 IFAIL=1
      RETURN
 1010 IFAIL=2
      RETURN
 1020 IFAIL=5
      RETURN
 1030 IFAIL=6
      RETURN
C
      END
```

```
      SUBROUTINE PELNOR(XMOM,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE NORMAL DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 2. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2.
C  PARA   *OUTPUT* ARRAY OF LENGTH 2. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER MU, SIGMA (LOCATION, SCALE).
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(2),PARA(2)
      DATA ZERO/0D0/
      DATA ROOTPI/1.7724 53850 90551 603D0/
C
      IF(XMOM(2).LE.ZERO)GOTO 1000
      PARA(2)=XMOM(2)*ROOTPI
      PARA(1)=XMOM(1)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE PELNOR : L-MOMENTS INVALID')
      END
```

```
      SUBROUTINE PELPE3(XMOM,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE PEARSON TYPE 3 DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 3. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2 AND TAU-3.
C  PARA   *OUTPUT* ARRAY OF LENGTH 3. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER MU, SIGMA, GAMMA (MEAN, S.D., SKEWNESS).
C
C  OTHER ROUTINES USED: DLGAMA
C
C  METHOD: RATIONAL APPROXIMATION IS USED TO EXPRESS ALPHA, THE SHAPE
C  PARAMETER OF THE GAMMA DISTRIBUTION, AS A FUNCTION OF TAU-3.
C  RELATIVE ACCURACY OF THE APPROXIMATION IS BETTER THAN 3E-5.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(3),PARA(3)
      DATA ZERO/0D0/,THIRD/0.33333333D0/,HALF/0.5D0/,ONE/1D0/,TWO/2D0/
C
C         SMALL IS USED TO TEST WHETHER SKEWNESS IS EFFECTIVELY ZERO
C
      DATA SMALL/1D-6/
C
C         CONSTANTS USED IN MINIMAX APPROXIMATIONS
C
      DATA C1,C2,C3/ 0.2906D0,  0.1882D0,  0.0442D0/
      DATA D1,D2,D3/ 0.36067D0,-0.59567D0, 0.25361D0/
      DATA D4,D5,D6/-2.78861D0, 2.56096D0,-0.77045D0/
      DATA PI3,ROOTPI/9.4247780D0,1.7724539D0/
C
      T3=DABS(XMOM(3))
      IF(XMOM(2).LE.ZERO.OR.T3.GE.ONE)GOTO 1000
      IF(T3.LE.SMALL)GOTO 100
      IF(T3.GE.THIRD)GOTO 10
      T=PI3*T3*T3
      ALPHA=(ONE+C1*T)/(T*(ONE+T*(C2+T*C3)))
      GOTO 20
   10 CONTINUE
      T=ONE-T3
      ALPHA=T*(D1+T*(D2+T*D3))/(ONE+T*(D4+T*(D5+T*D6)))
   20 CONTINUE
      RTALPH=DSQRT(ALPHA)
      BETA=ROOTPI*XMOM(2)*DEXP(DLGAMA(ALPHA)-DLGAMA(ALPHA+HALF))
      PARA(1)=XMOM(1)
      PARA(2)=BETA*RTALPH
      PARA(3)=TWO/RTALPH
      IF(XMOM(3).LT.ZERO)PARA(3)=-PARA(3)
      RETURN
C
C         ZERO SKEWNESS
C
  100 CONTINUE
      PARA(1)=XMOM(1)
      PARA(2)=XMOM(2)*ROOTPI
      PARA(3)=ZERO
      RETURN
```

```
C
 1000 WRITE(6,7000)
      DO 1010 I=1,3
 1010 PARA(I)=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE PELPE3 : L-MOMENTS INVALID')
      END
```

```
      SUBROUTINE PELWAK(XMOM,PARA,IFAIL)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PARAMETER ESTIMATION VIA L-MOMENTS FOR THE WAKEBY DISTRIBUTION
C
C  PARAMETERS OF ROUTINE:
C  XMOM   * INPUT* ARRAY OF LENGTH 5. CONTAINS THE L-MOMENTS LAMBDA-1,
C                  LAMBDA-2, TAU-3, TAU-4, TAU-5.
C  PARA   *OUTPUT* ARRAY OF LENGTH 5. ON EXIT, CONTAINS THE PARAMETERS
C                  IN THE ORDER XI, ALPHA, BETA, GAMMA, DELTA.
C  IFAIL  *OUTPUT* FAIL FLAG. ON EXIT, IT IS SET AS FOLLOWS.
C                  0 SUCCESSFUL EXIT
C                  1 ESTIMATES COULD ONLY BE OBTAINED BY SETTING XI=0
C                  2 ESTIMATES COULD ONLY BE OBTAINED BY FITTING A
C                    GENERALIZED PARETO DISTRIBUTION
C                  3 L-MOMENTS INVALID
C
C  PROCEDURE:
C  1. LOOK FOR A SOLUTION WITH XI UNCONSTRAINED;
C  2. IF NONE FOUND, LOOK FOR A SOLUTION WITH XI=0;
C  3. IF NONE FOUND, FIT A GENERALIZED PARETO DISTRIBUTION TO THE
C     FIRST 3 L-MOMENTS.
C  ESTIMATES ARE CALCULATED USING THE FORMULAS GIVEN BY GREENWOOD ET AL.
C  (1979, WATER RESOUR. RES., TABLE 5), BUT EXPRESSED IN TERMS OF
C  L-MOMENTS RATHER THAN PROBABILITY WEIGHTED MOMENTS.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(5),PARA(5)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/,TWO/2D0/,THREE/3D0/,FOUR/4D0/
      DATA X2/2D0/,X3/3D0/,X4/4D0/,X5/5D0/,X7/7D0/,X8/8D0/,X9/9D0/,
     *  X10/10D0/,X11/11D0/,X16/16D0/,X25/25D0/,X29/29D0/,X32/32D0/,
     *  X35/35D0/,X85/85D0/,X125/125D0/,X203/203D0/
C
      IF(DABS(XMOM(2)).LE.ZERO)GOTO 1000
      IF(DABS(XMOM(3)).GE.ONE)GOTO 1000
      IF(DABS(XMOM(4)).GE.ONE)GOTO 1000
      IF(DABS(XMOM(5)).GE.ONE)GOTO 1000
      IFAIL=0
C
C         CALCULATE THE L-MOMENTS (LAMBDA'S)
C
      ALAM1=XMOM(1)
      ALAM2=XMOM(2)
      ALAM3=XMOM(3)*ALAM2
      ALAM4=XMOM(4)*ALAM2
      ALAM5=XMOM(5)*ALAM2
C
C         ESTIMATE N1,N2,N3,C1,C2,C3 WHEN XI.NE.0
C
      XN1= X3*ALAM2-X25*ALAM3 +X32*ALAM4
      XN2=-X3*ALAM2 +X5*ALAM3  +X8*ALAM4
      XN3= X3*ALAM2 +X5*ALAM3  +X2*ALAM4
      XC1= X7*ALAM2-X85*ALAM3+X203*ALAM4-X125*ALAM5
      XC2=-X7*ALAM2+X25*ALAM3  +X7*ALAM4 -X25*ALAM5
      XC3= X7*ALAM2 +X5*ALAM3  -X7*ALAM4  -X5*ALAM5
C
C         ESTIMATE B AND D
C
```

```
      XA=XN2*XC3-XC2*XN3
      XB=XN1*XC3-XC1*XN3
      XC=XN1*XC2-XC1*XN2
      DISC=XB*XB-FOUR*XA*XC
      IF(DISC.LT.ZERO)GOTO 10
      DISC=DSQRT(DISC)
      ROOT1=HALF*(-XB+DISC)/XA
      ROOT2=HALF*(-XB-DISC)/XA
      B= DMAX1(ROOT1,ROOT2)
      D=-DMIN1(ROOT1,ROOT2)
      IF(D.GE.ONE)GOTO 10
C
C         ESTIMATE A, C AND XI
C
      A=(ONE+B)*(TWO+B)*(THREE+B)/
     *  (FOUR*(B+D))*((ONE+D)*ALAM2-(THREE-D)*ALAM3)
      C=-(ONE-D)*(TWO-D)*(THREE-D)/
     *  (FOUR*(B+D))*((ONE-B)*ALAM2-(THREE+B)*ALAM3)
      XI=ALAM1-A/(ONE+B)-C/(ONE-D)
C
C         CHECK FOR VALID PARAMETERS
C
      IF(C.GE.ZERO.AND.A+C.GE.ZERO)GOTO 30
C
C         CAN'T FIND VALID ESTIMATES FOR XI UNRESTRICTED, SO TRY XI=0
C
C         ESTIMATE B AND D FOR XI=0
C
   10 IFAIL=1
      XI=ZERO
      ZN1=X4*ALAM1-X11*ALAM2+X9*ALAM3
      ZN2=-ALAM2+X3*ALAM3
      ZN3=ALAM2+ALAM3
      ZC1=X10*ALAM1-X29*ALAM2+X35*ALAM3-X16*ALAM4
      ZC2=-ALAM2+X5*ALAM3-X4*ALAM4
      ZC3=ALAM2-ALAM4
      ZA=ZN2*ZC3-ZC2*ZN3
      ZB=ZN1*ZC3-ZC1*ZN3
      ZC=ZN1*ZC2-ZC1*ZN2
      DISC=ZB*ZB-FOUR*ZA*ZC
      IF(DISC.LT.ZERO)GOTO 20
      DISC=DSQRT(DISC)
      ROOT1=HALF*(-ZB+DISC)/ZA
      ROOT2=HALF*(-ZB-DISC)/ZA
      B= DMAX1(ROOT1,ROOT2)
      D=-DMIN1(ROOT1,ROOT2)
      IF(D.GE.ONE)GOTO 20
C
C         ESTIMATE A AND C
C
      A= (ONE+B)*(TWO+B)/(B+D)*(ALAM1-(TWO-D)*ALAM2)
      C=-(ONE-D)*(TWO-D)/(B+D)*(ALAM1-(TWO+B)*ALAM2)
      IF(C.GE.ZERO.AND.A+C.GE.ZERO)GOTO 30
C
C         CAN'T FIND VALID ESTIMATES EVEN WITH XI=0 -
C         FIT GENERALIZED PARETO DISTRIBUTION INSTEAD
C
   20 IFAIL=2
      D=-(ONE-THREE*XMOM(3))/(ONE+XMOM(3))
      C=(ONE-D)*(TWO-D)*XMOM(2)
      B=ZERO
      A=ZERO
      XI=XMOM(1)-C/(ONE-D)
      IF(D.GT.ZERO)GOTO 30
      A=C
      B=-D
      C=ZERO
      D=ZERO
C
C         COPY RESULTS INTO ARRAY PARA
C
```

```
   30 PARA(1)=XI
      PARA(2)=A
      PARA(3)=B
      PARA(4)=C
      PARA(5)=D
      RETURN
C
 1000 IFAIL=3
      DO 1010 I=1,5
 1010 PARA(I)=ZERO
      END
```

```
      DOUBLE PRECISION FUNCTION QUAEXP(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE EXPONENTIAL DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2)
      DATA ZERO/0D0/,ONE/1D0/
      U=PARA(1)
      A=PARA(2)
      IF(A.LE.ZERO)GOTO 1000
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 1010
      QUAEXP=U-A*DLOG(ONE-F)
      RETURN
C
 1000 WRITE(6,7000)
      QUAEXP=ZERO
      RETURN
 1010 WRITE(6,7010)
      QUAEXP=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAEXP : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAEXP :',
     *  ' ARGUMENT OF FUNCTION INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION QUAGAM(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE GAMMA DISTRIBUTION
C
C  OTHER ROUTINES USED: DERF,DLGAMA,GAMIND,QUASTN
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2)
      DATA ZERO/0D0/,P01/0.01D0/,ONE/1D0/,NINE/9D0/
C
C         EPS,MAXIT CONTROL THE TEST FOR CONVERGENCE OF N-R ITERATION
C
      DATA EPS/1D-10/,MAXIT/30/
C
      QUAGAM=ZERO
      ALPHA=PARA(1)
      BETA=PARA(2)
      IF(ALPHA.LE.ZERO.OR.BETA.LE.ZERO)GOTO 1000
      IF(F.LT.ZERO.OR.F.GE.ONE)GOTO 1010
      IF(F.EQ.ZERO)RETURN
      AM1=ALPHA-ONE
      IF(AM1.NE.ZERO)GOTO 10
C
C         CASE ALPHA.EQ.1 - GAMMA IS EXPONENTIAL
C
      QUAGAM=(-DLOG(ONE-F))*BETA
      RETURN
C
C         INITIAL ESTIMATE OF ROOT OF EQUATION GAMIND(X)=F:
C         - IF ALPHA.GT.1, USE WILSON-HILFERTY APPROXIMATION IF IT'S
C           POSITIVE AND NOT TOO CLOSE TO ZERO;
C         - IF ALPHA.LT.1, OR IF W-H APPROX. ISN'T POSITIVE ENOUGH,
C           USE THE SMALL-X APPROXIMATION OF IGNORING THE EXP(-T) TERM
C           IN THE INTEGRAL DEFINING GAMIND(X)
C
   10 DLOGG=DLGAMA(ALPHA)
      IF(AM1.LE.ZERO)GOTO 20
      ROOT=ALPHA*(ONE-ONE/(NINE*ALPHA)+QUASTN(F)/DSQRT(NINE*ALPHA))**3
      IF(ROOT.GT.P01*ALPHA)GOTO 30
   20 ROOT=DEXP((DLOG(ALPHA*F)+DLOGG)/ALPHA)
   30 CONTINUE
C
C         REFINE INITIAL ESTIMATE BY NEWTON-RAPHSON ITERATION
C
      DO 40 IT=1,MAXIT
      FUNC=GAMIND(ROOT,ALPHA,DLOGG)-F
      RINC=FUNC*DEXP(DLOGG+ROOT-AM1*DLOG(ROOT))
      ROOT=ROOT-RINC
      IF(DABS(FUNC).LE.EPS)GOTO 50
   40 CONTINUE
      WRITE(6,7020)
C
C         SCALE SOLUTION
C
   50 QUAGAM=ROOT*BETA
      RETURN
C
```

```
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAGAM : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAGAM :',
     *  ' ARGUMENT OF FUNCTION INVALID')
 7020 FORMAT(' ** WARNING ** ROUTINE QUAGAM :',
     *  ' ITERATION HAS NOT CONVERGED. RESULT MAY BE UNRELIABLE')
      END
```

```
      DOUBLE PRECISION FUNCTION QUAGEV(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE GENERALIZED EXTREME-VALUE DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3)
      DATA ZERO/0D0/,ONE/1D0/
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO)GOTO 1000
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 10
      Y=-DLOG(-DLOG(F))
      IF(G.NE.ZERO)Y=(ONE-DEXP(-G*Y))/G
      QUAGEV=U+A*Y
      RETURN
C
   10 IF(F.EQ.ZERO.AND.G.LT.ZERO)GOTO 20
      IF(F.EQ.ONE .AND.G.GT.ZERO)GOTO 20
      WRITE(6,7000)
      QUAGEV=ZERO
      RETURN
   20 QUAGEV=U+A/G
      RETURN
C
 1000 WRITE(6,7010)
      QUAGEV=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAGEV :',
     *  ' ARGUMENT OF FUNCTION INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAGEV : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION QUAGLO(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE GENERALIZED LOGISTIC DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3)
      DATA ZERO/0D0/,ONE/1D0/
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO)GOTO 1000
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 10
      Y=DLOG(F/(ONE-F))
      IF(G.NE.ZERO)Y=(ONE-DEXP(-G*Y))/G
      QUAGLO=U+A*Y
      RETURN
C
   10 IF(F.EQ.ZERO.AND.G.LT.ZERO)GOTO 20
      IF(F.EQ.ONE .AND.G.GT.ZERO)GOTO 20
      WRITE(6,7000)
      QUAGLO=ZERO
      RETURN
   20 QUAGLO=U+A/G
      RETURN
C
 1000 WRITE(6,7010)
      QUAGLO=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAGLO :',
     *  ' ARGUMENT OF FUNCTION INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAGLO : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION QUAGNO(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE GENERALIZED NORMAL DISTRIBUTION
C
C  OTHER ROUTINES USED: QUASTN
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3)
      DATA ZERO/0D0/,ONE/1D0/
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO)GOTO 1000
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 10
      Y=QUASTN(F)
      IF(G.NE.ZERO)Y=(ONE-DEXP(-G*Y))/G
      QUAGNO=U+A*Y
      RETURN
C
   10 IF(F.EQ.ZERO.AND.G.LT.ZERO)GOTO 20
      IF(F.EQ.ONE .AND.G.GT.ZERO)GOTO 20
      WRITE(6,7000)
      QUAGNO=ZERO
      RETURN
   20 QUAGNO=U+A/G
      RETURN
C
 1000 WRITE(6,7010)
      QUAGNO=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAGNO :',
     *  ' ARGUMENT OF FUNCTION INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAGNO : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION QUAGPA(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE GENERALIZED PARETO DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3)
      DATA ZERO/0D0/,ONE/1D0/
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      IF(A.LE.ZERO)GOTO 1000
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 10
      Y=-DLOG(ONE-F)
      IF(G.NE.ZERO)Y=(ONE-DEXP(-G*Y))/G
      QUAGPA=U+A*Y
      RETURN
C
   10 IF(F.EQ.ZERO)QUAGPA=U
      IF(F.EQ.ZERO)RETURN
      IF(F.EQ.ONE.AND.G.GT.ZERO)QUAGPA=U+A/G
      IF(F.EQ.ONE.AND.G.GT.ZERO)RETURN
      WRITE(6,7000)
      QUAGPA=ZERO
      RETURN
C
 1000 WRITE(6,7010)
      QUAGPA=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAGPA :',
     *  ' ARGUMENT OF FUNCTION INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAGPA : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION QUAGUM(F,PARA)
C**********************************************************************
C*                                                                    *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                    *
C*  J. R. M. HOSKING                                                  *
C*  IBM RESEARCH DIVISION                                             *
C*  T. J. WATSON RESEARCH CENTER                                      *
C*  YORKTOWN HEIGHTS                                                  *
C*  NEW YORK 10598, U.S.A.                                            *
C*                                                                    *
C*  VERSION 3     AUGUST 1996                                         *
C*                                                                    *
C**********************************************************************
C
C  QUANTILE FUNCTION OF THE GUMBEL DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2)
      DATA ZERO/0D0/,ONE/1D0/
      U=PARA(1)
      A=PARA(2)
      IF(A.LE.ZERO)GOTO 1000
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 1010
      QUAGUM=U-A*DLOG(-DLOG(F))
      RETURN
C
 1000 WRITE(6,7000)
      QUAGUM=ZERO
      RETURN
 1010 WRITE(6,7010)
      QUAGUM=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAGUM : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAGUM :',
     *  ' ARGUMENT OF FUNCTION INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION QUAKAP(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE KAPPA DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(4)
      DATA ZERO/0D0/,ONE/1D0/
      U=PARA(1)
      A=PARA(2)
      G=PARA(3)
      H=PARA(4)
      IF(A.LE.ZERO)GOTO 1000
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 10
      Y=-DLOG(F)
      IF(H.NE.ZERO)Y=(ONE-DEXP(-H*Y))/H
      Y=-DLOG(Y)
      IF(G.NE.ZERO)Y=(ONE-DEXP(-G*Y))/G
      QUAKAP=U+A*Y
      RETURN
C
   10 IF(F.EQ.ZERO)GOTO 20
      IF(F.EQ.ONE)GOTO 30
      GOTO 1010
   20 IF(H.LE.ZERO.AND.G.LT.ZERO)QUAKAP=U+A/G
      IF(H.LE.ZERO.AND.G.GE.ZERO)GOTO 1010
      IF(H.GT.ZERO.AND.G.NE.ZERO)QUAKAP=U+A/G*(ONE-H**(-G))
      IF(H.GT.ZERO.AND.G.EQ.ZERO)QUAKAP=U+A*DLOG(H)
      RETURN
   30 IF(G.LE.ZERO)GOTO 1010
      QUAKAP=U+A/G
      RETURN
C
 1000 WRITE(6,7000)
      QUAKAP=ZERO
      RETURN
 1010 WRITE(6,7010)
      QUAKAP=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAKAP : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAKAP :',
     *  ' ARGUMENT OF FUNCTION INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION QUANOR(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE NORMAL DISTRIBUTION
C
C  OTHER ROUTINES USED: QUASTN
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(2)
      DATA ZERO/0D0/,ONE/1D0/
      IF(PARA(2).LE.ZERO)GOTO 1000
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 1010
      QUANOR=PARA(1)+PARA(2)*QUASTN(F)
      RETURN
C
 1000 WRITE(6,7000)
      QUANOR=ZERO
      RETURN
 1010 WRITE(6,7010)
      QUANOR=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUANOR : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUANOR :',
     *  ' ARGUMENT OF FUNCTION INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION QUAPE3(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE PEARSON TYPE 3 DISTRIBUTION
C
C  OTHER ROUTINES USED: DERF,DLGAMA,GAMIND,QUAGAM,QUASTN
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(3),PAR(2)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/,TWO/2D0/,FOUR/4D0/
C
C         SMALL IS USED TO TEST WHETHER SKEWNESS IS EFFECTIVELY ZERO
C
      DATA SMALL/1D-6/
C
      IF(PARA(2).LE.ZERO)GOTO 1000
      GAMMA=PARA(3)
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 20
      IF(DABS(GAMMA).LT.SMALL)GOTO 10
      ALPHA=FOUR/(GAMMA*GAMMA)
      BETA=DABS(HALF*PARA(2)*GAMMA)
      PAR(1)=ALPHA
      PAR(2)=BETA
      IF(GAMMA.GT.ZERO)QUAPE3=PARA(1)-ALPHA*BETA+QUAGAM(F,PAR)
      IF(GAMMA.LT.ZERO)QUAPE3=PARA(1)+ALPHA*BETA-QUAGAM(ONE-F,PAR)
      RETURN
C
C         ZERO SKEWNESS
C
   10 QUAPE3=PARA(1)+PARA(2)*QUASTN(F)
      RETURN
C
   20 IF(F.EQ.ZERO.AND.GAMMA.GT.ZERO)GOTO 30
      IF(F.EQ.ONE .AND.GAMMA.LT.ZERO)GOTO 30
      WRITE(6,7000)
      QUAPE3=ZERO
      RETURN
   30 QUAPE3=PARA(1)-TWO*PARA(2)/GAMMA
      RETURN
C
 1000 WRITE(6,7010)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAPE3 :',
     *  ' ARGUMENT OF FUNCTION INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAPE3 : PARAMETERS INVALID')
      END
```

```
      DOUBLE PRECISION FUNCTION QUAWAK(F,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE WAKEBY DISTRIBUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PARA(5)
      DATA ZERO/0D0/,ONE/1D0/
C
C         UFL SHOULD BE CHOSEN SO THAT EXP(UFL) JUST DOES NOT CAUSE
C         UNDERFLOW
C
      DATA UFL/-170D0/
C
      XI=PARA(1)
      A=PARA(2)
      B=PARA(3)
      C=PARA(4)
      D=PARA(5)
C
C         TEST FOR VALID PARAMETERS
C
      IF(B+D.LE.ZERO.AND.(B.NE.ZERO.OR.C.NE.ZERO.OR.D.NE.ZERO))GOTO 1000
      IF(A.EQ.ZERO.AND.B.NE.ZERO)GOTO 1000
      IF(C.EQ.ZERO.AND.D.NE.ZERO)GOTO 1000
      IF(C.LT.ZERO.OR.A+C.LT.ZERO)GOTO 1000
      IF(A.EQ.ZERO.AND.C.EQ.ZERO)GOTO 1000
C
      IF(F.LE.ZERO.OR.F.GE.ONE)GOTO 10
      Z=-DLOG(ONE-F)
      Y1=Z
      IF(B.EQ.ZERO)GOTO 5
      TEMP=-B*Z
      IF(TEMP.LT.UFL)Y1=ONE/B
      IF(TEMP.GE.UFL)Y1=(ONE-DEXP(TEMP))/B
    5 CONTINUE
      Y2=Z
      IF(D.NE.ZERO)Y2=(ONE-DEXP(D*Y2))/(-D)
      QUAWAK=XI+A*Y1+C*Y2
      RETURN
C
   10 IF(F.EQ.ZERO)GOTO 20
      IF(F.EQ.ONE)GOTO 30
      GOTO 1010
   20 QUAWAK=XI
      RETURN
   30 IF(D.GT.ZERO)GOTO 1010
      IF(D.LT.ZERO)QUAWAK=XI+A/B-C/D
      IF(D.EQ.ZERO.AND.C.GT.ZERO)GOTO 1010
      IF(D.EQ.ZERO.AND.C.EQ.ZERO.AND.B.EQ.ZERO)GOTO 1010
      IF(D.EQ.ZERO.AND.C.EQ.ZERO.AND.B.GT.ZERO)QUAWAK=XI+A/B
      RETURN
C
 1000 WRITE(6,7000)
      QUAWAK=ZERO
      RETURN
 1010 WRITE(6,7010)
      QUAWAK=ZERO
```

```
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUAWAK : PARAMETERS INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE QUAWAK :',
     *  ' ARGUMENT OF FUNCTION INVALID')
      END
```

```
      SUBROUTINE SAMLMR(X,N,XMOM,NMOM,A,B)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  SAMPLE L-MOMENTS OF A DATA ARRAY
C
C  PARAMETERS OF ROUTINE:
C  X      * INPUT* ARRAY OF LENGTH N. CONTAINS THE DATA, IN ASCENDING
C                  ORDER.
C  N      * INPUT* NUMBER OF DATA VALUES
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE SAMPLE
C                  L-MOMENTS L-1, L-2, T-3, T-4, ... .
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST MAX(N,20).
C  A      * INPUT* ) PARAMETERS OF PLOTTING
C  B      * INPUT* ) POSITION (SEE BELOW)
C
C  FOR UNBIASED ESTIMATES (OF THE LAMBDA'S) SET A=B=ZERO. OTHERWISE,
C  PLOTTING-POSITION ESTIMATORS ARE USED, BASED ON THE PLOTTING POSITION
C  (J+A)/(N+B)  FOR THE J'TH SMALLEST OF N OBSERVATIONS. FOR EXAMPLE,
C  A=-0.35D0 AND B=0.0D0 YIELDS THE ESTIMATORS RECOMMENDED BY
C  HOSKING ET AL. (1985, TECHNOMETRICS) FOR THE GEV DISTRIBUTION.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION X(N),XMOM(NMOM),SUM(20)
      DATA ZERO/0D0/,ONE/1D0/
      IF(NMOM.GT.20.OR.NMOM.GT.N)GOTO 1000
      DO 10 J=1,NMOM
   10 SUM(J)=ZERO
      IF(A.EQ.ZERO.AND.B.EQ.ZERO)GOTO 50
      IF(A.LE.-ONE.OR.A.GE.B)GOTO 1010
C
C         PLOTTING-POSITION ESTIMATES OF PWM'S
C
      DO 30 I=1,N
      PPOS=(I+A)/(N+B)
      TERM=X(I)
      SUM(1)=SUM(1)+TERM
      DO 20 J=2,NMOM
      TERM=TERM*PPOS
   20 SUM(J)=SUM(J)+TERM
   30 CONTINUE
      DO 40 J=1,NMOM
   40 SUM(J)=SUM(J)/N
      GOTO 100
C
C         UNBIASED ESTIMATES OF PWM'S
C
   50 DO 70 I=1,N
      Z=I
      TERM=X(I)
      SUM(1)=SUM(1)+TERM
      DO 60 J=2,NMOM
      Z=Z-ONE
      TERM=TERM*Z
   60 SUM(J)=SUM(J)+TERM
   70 CONTINUE
      Y=N
      Z=N
      SUM(1)=SUM(1)/Z
```

```
      DO 80 J=2,NMOM
      Y=Y-ONE
      Z=Z*Y
   80 SUM(J)=SUM(J)/Z
C
C         L-MOMENTS
C
  100 K=NMOM
      P0=ONE
      IF(NMOM-NMOM/2*2.EQ.1)P0=-ONE
      DO 120 KK=2,NMOM
      AK=K
      P0=-P0
      P=P0
      TEMP=P*SUM(1)
      DO 110 I=1,K-1
      AI=I
      P=-P*(AK+AI-ONE)*(AK-AI)/(AI*AI)
  110 TEMP=TEMP+P*SUM(I+1)
      SUM(K)=TEMP
  120 K=K-1
      XMOM(1)=SUM(1)
      IF(NMOM.EQ.1)RETURN
      XMOM(2)=SUM(2)
      IF(SUM(2).EQ.ZERO)GOTO 1020
      IF(NMOM.EQ.2)RETURN
      DO 130 K=3,NMOM
  130 XMOM(K)=SUM(K)/SUM(2)
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
 1020 WRITE(6,7020)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE SAMLMR : PARAMETER NMOM INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE SAMLMR :',
     *  ' PLOTTING-POSITION PARAMETERS INVALID')
 7020 FORMAT(' *** ERROR *** ROUTINE SAMLMR : ALL DATA VALUES EQUAL')
      END
```

```
      SUBROUTINE SAMLMU(X,N,XMOM,NMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  SAMPLE L-MOMENTS OF A DATA ARRAY
C
C  PARAMETERS OF ROUTINE:
C  X      * INPUT* ARRAY OF LENGTH N. CONTAINS THE DATA, IN ASCENDING
C                  ORDER.
C  N      * INPUT* NUMBER OF DATA VALUES
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. CONTAINS THE SAMPLE L-MOMENTS,
C                  STORED AS DESCRIBED BELOW.
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND. AT MOST 100.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MAXMOM=100)
      DOUBLE PRECISION X(N),XMOM(NMOM),COEF(2,MAXMOM)
      DATA ZERO/0D0/,ONE/1D0/,TWO/2D0/
C
      IF(NMOM.GT.MAXMOM)GOTO 1000
      DN=N
      DO 10 J=1,NMOM
   10 XMOM(J)=ZERO
      IF(NMOM.LE.2)GOTO 100
C
C         UNBIASED ESTIMATES OF L-MOMENTS -- THE 'DO 30' LOOP
C         RECURSIVELY CALCULATES DISCRETE LEGENDRE POLYNOMIALS, VIA
C         EQ.(9) OF NEUMAN AND SCHONBACH (1974, INT.J.NUM.METH.ENG.)
C
      DO 20 J=3,NMOM
      TEMP=ONE/DFLOAT((J-1)*(N-J+1))
      COEF(1,J)=DFLOAT(J+J-3)*TEMP
      COEF(2,J)=DFLOAT((J-2)*(N+J-2))*TEMP
   20 CONTINUE
      TEMP=-DN-ONE
      CONST=ONE/(DN-ONE)
      NHALF=N/2
      DO 40 I=1,NHALF
      TEMP=TEMP+TWO
      XI=X(I)
      XII=X(N+1-I)
      TERMP=XI+XII
      TERMN=XI-XII
      XMOM(1)=XMOM(1)+TERMP
      S1=ONE
      S=TEMP*CONST
      XMOM(2)=XMOM(2)+S*TERMN
      DO 30 J=3,NMOM,2
      S2=S1
      S1=S
      S=COEF(1,J)*TEMP*S1-COEF(2,J)*S2
      XMOM(J)=XMOM(J)+S*TERMP
      IF(J.EQ.NMOM)GOTO 30
      JJ=J+1
      S2=S1
      S1=S
      S=COEF(1,JJ)*TEMP*S1-COEF(2,JJ)*S2
      XMOM(JJ)=XMOM(JJ)+S*TERMN
   30 CONTINUE
```

```
   40 CONTINUE
      IF(N.EQ.NHALF+NHALF)GOTO 60
      TERM=X(NHALF+1)
      S=ONE
      XMOM(1)=XMOM(1)+TERM
      DO 50 J=3,NMOM,2
      S=-COEF(2,J)*S
      XMOM(J)=XMOM(J)+S*TERM
   50 CONTINUE
C
C         L-MOMENT RATIOS
C
   60 CONTINUE
      XMOM(1)=XMOM(1)/DN
      IF(XMOM(2).EQ.ZERO)GOTO 1010
      DO 70 J=3,NMOM
   70 XMOM(J)=XMOM(J)/XMOM(2)
      XMOM(2)=XMOM(2)/DN
      RETURN
C
C         AT MOST TWO L-MOMENTS
C
  100 CONTINUE
      SUM1=ZERO
      SUM2=ZERO
      TEMP=-DN+ONE
      DO 110 I=1,N
      SUM1=SUM1+X(I)
      SUM2=SUM2+X(I)*TEMP
      TEMP=TEMP+TWO
  110 CONTINUE
      XMOM(1)=SUM1/DN
      IF(NMOM.EQ.1)RETURN
      XMOM(2)=SUM2/(DN*(DN-ONE))
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      DO 1020 J=1,NMOM
 1020 XMOM(J)=ZERO
      RETURN
C
 7000 FORMAT(' ** WARNING ** ROUTINE SAMLMU :',
     *  ' PARAMETER NMOM INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE SAMLMU :',
     *  ' ALL DATA VALUES EQUAL')
      END
```

```
      SUBROUTINE SAMPWM(X,N,XMOM,NMOM,A,B,KIND)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PROBABILITY WEIGHTED MOMENTS OF A DATA ARRAY
C
C  PARAMETERS OF ROUTINE:
C  X      * INPUT* ARRAY OF LENGTH N. CONTAINS THE DATA, IN ASCENDING
C                  ORDER.
C  N      * INPUT* NUMBER OF DATA VALUES
C  XMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE SAMPLE
C                  PROBABILITY WEIGHTED MOMENTS. XMOM(I) CONTAINS
C                  ALPHA-SUB-(I-1) OR BETA-SUB-(I-1).
C  NMOM   * INPUT* NUMBER OF PROBABILITY WEIGHTED MOMENTS TO BE FOUND.
C                  AT MOST MAX(N,20).
C  A      * INPUT* ) PARAMETERS OF PLOTTING
C  B      * INPUT* ) POSITION (SEE BELOW)
C  KIND   * INPUT* SPECIFIES WHICH KIND OF PWM'S ARE TO BE FOUND.
C                  1  ALPHA-SUB-R = E ( X (1-F(X))**R )
C                  2  BETA -SUB-R = E ( X F(X)**R )
C
C  FOR UNBIASED ESTIMATES SET A AND B EQUAL TO ZERO. OTHERWISE,
C  PLOTTING-POSITION ESTIMATORS ARE USED, BASED ON THE PLOTTING POSITION
C  (J+A)/(N+B)  FOR THE J'TH SMALLEST OF N OBSERVATIONS. FOR EXAMPLE,
C  A=-0.35D0 AND B=0.0D0 YIELDS THE ESTIMATORS RECOMMENDED BY
C  HOSKING ET AL. (1985, TECHNOMETRICS) FOR THE GEV DISTRIBUTION.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION X(N),XMOM(NMOM)
      DATA ZERO/0D0/,ONE/1D0/
      IF(NMOM.GT.20.OR.NMOM.GT.N)GOTO 1000
      IF(KIND.NE.1.AND.KIND.NE.2)GOTO 1010
      DO 10 J=1,NMOM
   10 XMOM(J)=ZERO
      DN=N
      IF(A.EQ.ZERO.AND.B.EQ.ZERO)GOTO 50
      IF(A.LE.-ONE.OR.A.GE.B)GOTO 1020
C
C         PLOTTING-POSITION ESTIMATES OF PWM'S
C
      DO 30 I=1,N
      PPOS=(I+A)/(N+B)
      IF(KIND.EQ.1)PPOS=ONE-PPOS
      TERM=X(I)
      XMOM(1)=XMOM(1)+TERM
      DO 20 J=2,NMOM
      TERM=TERM*PPOS
   20 XMOM(J)=XMOM(J)+TERM
   30 CONTINUE
      DO 40 J=1,NMOM
   40 XMOM(J)=XMOM(J)/DN
      RETURN
C
C         UNBIASED ESTIMATES OF PWM'S
C
   50 DO 70 I=1,N
      DI=I
      WEIGHT=ONE/DN
      XMOM(1)=XMOM(1)+WEIGHT*X(I)
```

```
      DO 60 J=2,NMOM
      DJ=J-ONE
      IF(KIND.EQ.1)WEIGHT=WEIGHT*(DN-DI-DJ+ONE)/(DN-DJ)
      IF(KIND.EQ.2)WEIGHT=WEIGHT*(DI-DJ)/(DN-DJ)
   60 XMOM(J)=XMOM(J)+WEIGHT*X(I)
   70 CONTINUE
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)
      RETURN
 1020 WRITE(6,7020)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE SAMPWM : PARAMETER NMOM INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE SAMPWM : PARAMETER KIND INVALID')
 7020 FORMAT(' *** ERROR *** ROUTINE SAMPWM :',
     *  ' PLOTTING-POSITION PARAMETERS INVALID')
      END
```

```
      SUBROUTINE CLUAGG(METHOD,X,NX,N,NATT,MERGE,WGSS,IWORK,WORK,NW)
C
C  CLUSTER ANALYSIS BY WARD'S METHOD
C
C  PARAMETERS OF ROUTINE:
C  METHOD * INPUT* CLUSTERING METHOD. SHOULD BE SET TO:
C                   1 FOR SINGLE-LINK CLUSTERING
C                   2 FOR COMPLETE-LINK CLUSTERING
C                   3 FOR WARD'S PROCEDURE
C  X      * INPUT* ARRAY OF DIMENSION (NX,NATT).  X(I,J) SHOULD
C                  CONTAIN THE J'TH ATTRIBUTE FOR THE I'TH DATA POINT.
C  NX     * INPUT* THE FIRST DIMENSION OF ARRAY X, AS DECLARED IN THE
C                  CALLING PROGRAM.
C  N      * INPUT* NUMBER OF DATA POINTS
C  NATT   * INPUT* NUMBER OF ATTRIBUTES FOR EACH DATA POINT
C  MERGE  *OUTPUT* ARRAY OF DIMENSION (2,N). MERGE(1,I) AND MERGE(2,I)
C                  ARE THE LABELS OF THE CLUSTERS MERGED AT THE I'TH
C                  STAGE.  MERGE(1,N) AND MERGE(2,N) ARE NOT USED.
C  WGSS   *OUTPUT* ARRAY OF LENGTH N. ITS I'TH ELEMENT IS THE WITHIN-
C                  CLUSTER SUM OF SQUARES AFTER THE I'TH MERGE.
C  IWORK  * LOCAL* WORK ARRAY OF LENGTH N
C  WORK   * LOCAL* WORK ARRAY OF LENGTH NW
C  NW     * INPUT* LENGTH OF ARRAY W. MUST BE AT LEAST N*(N-1)/2.
C
C  AGGLOMERATIVE HIERARCHICAL CLUSTERING: GENERAL DESCRIPTION.
C  INITIALLY THERE ARE N CLUSTERS, EACH CONTAINING ONE DATA POINT,
C  LABELED 1 THROUGH N IN THE SAME ORDER AS THE DATA POINTS.  AT EACH
C  STAGE OF CLUSTERING, TWO CLUSTERS ARE MERGED.  THEIR LABELS ARE SAVED
C  IN THE 'MERGE' ARRAY.  THE SMALLER OF THE TWO LABELS IS USED AS THE
C  LABEL OF THE MERGED CLUSTER.  AFTER THE M'TH STAGE OF CLUSTERING
C  THERE ARE N-M CLUSTERS.  TO FIND WHICH DATA POINTS BELONG TO WHICH
C  CLUSTERS, USE ROUTINE CLINFO.
C
C  WARD'S METHOD: AT EACH STAGE, THE CLUSTERS THAT ARE MERGED ARE CHOSEN
C  TO MINIMIZE THE TOTAL WITHIN-CLUSTER SUM OF SQUARED DEVIATIONS OF EACH
C  ATTRIBUTE ABOUT THE CLUSTER MEAN.
C
C-----------------------------------------------------------------------
C  NOTE: CURRENTLY, 'METHOD' IS IGNORED AND ONLY WARD'S PROCEDURE
C        IS IMPLEMENTED
C-----------------------------------------------------------------------
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION X(NX,NATT),WGSS(N),WORK(NW)
      INTEGER MERGE(2,N),IWORK(N)
      DATA ZERO/0D0/,HALF/0.5D0/
C
C         BIG IS A LARGE NUMBER, USED TO INITIALIZE THE SEARCH CRITERION
C
      DATA BIG/1D72/
C
      NWREQ=N*(N-1)/2
      IF(NW.LT.NWREQ)GOTO 1000
C
C         INITIALLY THERE ARE N CLUSTERS, EACH CONTAINING ONE DATA
C         POINT.  COMPUTE THE COST (INCREASE IN THE WITHIN-CLUSTER
C         SUM OF SQUARES) OF MERGING EACH PAIR OF CLUSTERS.
C
      IW=0
      DO 20 J=2,N
      DO 20 I=1,J-1
      SUM=ZERO
      DO 10 IATT=1,NATT
   10 SUM=SUM+(X(I,IATT)-X(J,IATT))**2
      IW=IW+1
      WORK(IW)=HALF*SUM
   20 CONTINUE
      DO 30 I=1,N
   30 IWORK(I)=1
      SS=ZERO
C
```

```
C         START OF MAIN LOOP
C
      DO 100 IMERGE=1,N-1
C
C         FIND THE PAIR OF CLUSTERS WITH THE LOWEST COST OF MERGING
C
      COST=BIG
      DO 50 J=2,N
      IF(IWORK(J).EQ.0)GOTO 50
      IORIG=(J-1)*(J-2)/2
      DO 40 I=1,J-1
      IF(IWORK(I).EQ.0)GOTO 40
      LIJ=IORIG+I
      IF(WORK(LIJ).GE.COST)GOTO 40
      COST=WORK(LIJ)
      II=I
      JJ=J
   40 CONTINUE
   50 CONTINUE
C
C         MERGE THEM
C
      SS=SS+COST
      MERGE(1,IMERGE)=II
      MERGE(2,IMERGE)=JJ
      WGSS(IMERGE)=SS
C
C         COMPUTE THE COST OF MERGING THE NEW CLUSTER WITH EACH OF THE
C         OTHERS
C
      NI=IWORK(II)
      NJ=IWORK(JJ)
      NIJ=NI+NJ
      DO 60 KK=1,N
      NK=IWORK(KK)
      IF(NK.EQ.0)GOTO 60
      IF(KK.EQ.II.OR.KK.EQ.JJ)GOTO 60
      MM=MAX0(II,KK)
      M =MIN0(II,KK)
      IK=(MM-1)*(MM-2)/2+M
      MM=MAX0(JJ,KK)
      M =MIN0(JJ,KK)
      JK=(MM-1)*(MM-2)/2+M
      WORK(IK)=((NI+NK)*WORK(IK)+(NJ+NK)*WORK(JK)-NK*COST)/(NIJ+NK)
   60 CONTINUE
      IWORK(II)=NIJ
      IWORK(JJ)=0
C
C         END OF MAIN LOOP
C
  100 CONTINUE
C
      RETURN
C
 1000 WRITE(6,7000)NWREQ
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CLUAGG : INSUFFICIENT WORKSPACE.',
     *  ' LENGTH OF WORK ARRAY SHOULD BE AT LEAST ',I8)
C
      END
```

```
      SUBROUTINE CLUINF(NCLUST,N,MERGE,IASSGN,LIST,NUM)
C
C  OBTAINS INFORMATION ABOUT CLUSTERS ARISING FROM AGGLOMERATIVE
C  HIERARCHICAL CLUSTERING
C
C  AGGLOMERATIVE HIERARCHICAL CLUSTERING PROCEDURES TYPICALLY PRODUCE A
C  LIST OF THE CLUSTERS MERGED AT EACH STAGE OF THE CLUSTERING.  THIS
C  ROUTINE USES THIS LIST TO CONSTRUCT ARRAYS THAT EXPLICITLY SHOW
C  WHICH CLUSTER A GIVEN DATA POINT BELONGS TO, AND WHICH DATA POINTS
C  BELONG TO A GIVEN CLUSTER.
C
C  PARAMETERS OF ROUTINE:
C  NCLUST * INPUT* NUMBER OF CLUSTERS
C  N      * INPUT* NUMBER OF DATA POINTS
C  MERGE  * INPUT* ARRAY OF DIMENSION (2,N). MERGE(1,I) AND MERGE(2,I)
C                  IDENTIFY THE CLUSTERS MERGED AT THE I'TH STEP.
C                  THIS IS THE ARRAY MERGE RETURNED BY ROUTINE CAWARD,
C                  AND SHOULD BE LEFT UNCHANGED AFTER EXIT FROM THAT
C                  ROUTINE.
C  IASSGN *OUTPUT* ARRAY OF LENGTH N. ITS I'TH ELEMENT IS THE NUMBER
C                  OF THE CLUSTER TO WHICH THE I'TH DATA POINT BELONGS.
C  LIST   *OUTPUT* ARRAY OF LENGTH N. CONTAINS THE DATA POINTS IN
C                  CLUSTER 1, FOLLOWED BY THE DATA POINTS IN CLUSTER 2,
C                  ETC.  DATA POINTS IN EACH CLUSTER ARE LISTED IN
C                  INCREASING ORDER.  THE LAST DATA POINT IN EACH
C                  CLUSTER IS INDICATED BY A NEGATIVE NUMBER.
C                  SEE THE EXAMPLE BELOW.
C  NUM    *OUTPUT* ARRAY OF LENGTH NCLUST.  NUMBER OF DATA POINTS IN
C                  EACH CLUSTER.
C
C  CLUSTER NUMBERS USED IN ARRAYS IASSGN, LIST AND NUM RANGE FROM 1 TO
C  NCLUST.  THEY ARE ARBITRARY, BUT ARE UNIQUELY DEFINED: CLUSTER 1
C  CONTAINS DATA POINT 1, CLUSTER M (M.GE.2) CONTAINS DATA POINT J,
C  WHERE J=MERGE(2,N-M).
C
C  EXAMPLE OF THE LIST ARRAY.  SUPPOSE THAT THERE ARE 8 DATA POINTS
C  AND 3 CLUSTERS, AND THAT THE ELEMENTS OF THE LIST ARRAY ARE
C  1, -4, 3, 6, -8, 2, 5, -7.  THEN THE CLUSTERS ARE AS FOLLOWS:
C  CLUSTER 1 CONTAINS POINTS 1 AND 4; CLUSTER 2 CONTAINS POINTS
C  3, 6 AND 8; CLUSTER 3 CONTAINS POINTS 2, 5 AND 7.
C
      INTEGER MERGE(2,N),IASSGN(N),LIST(N),NUM(NCLUST)
C
C        CONSTRUCT THE IASSGN ARRAY
C
      IASSGN(1)=1
      DO 10 I=1,NCLUST-1
      ITEMP=MERGE(2,N-I)
      IASSGN(ITEMP)=I+1
   10 CONTINUE
      DO 20 I=NCLUST,N-1
      ICL=N-I
      II=MERGE(1,ICL)
      JJ=MERGE(2,ICL)
      IASSGN(JJ)=IASSGN(II)
   20 CONTINUE
C
C        CONSTRUCT THE LIST AND NUM ARRAYS
C
      LASTI=0
      I=0
      DO 70 ICL=1,NCLUST
      DO 60 K=1,N
      IF(IASSGN(K).NE.ICL)GOTO 60
      I=I+1
      LIST(I)=K
   60 CONTINUE
      LIST(I)=-LIST(I)
      NUM(ICL)=I-LASTI
      LASTI=I
   70 CONTINUE
```

```
      RETURN
C
      END
```

```
      SUBROUTINE CLUKM(X,NX,N,NATT,NCLUST,IASSGN,LIST,NUM,SS,MAXIT,
     *  IWORK,RW,NW)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  CLUSTER ANALYSIS BY THE K-MEANS ALGORITHM
C
C  PARAMETERS OF ROUTINE:
C  X      * INPUT* ARRAY OF DIMENSION (NX,NATT).  X(I,J) SHOULD
C                  CONTAIN THE J'TH ATTRIBUTE FOR THE I'TH DATA POINT.
C  NX     * INPUT* THE FIRST DIMENSION OF ARRAY X, AS DECLARED IN THE
C                  CALLING PROGRAM.
C  N      * INPUT* NUMBER OF DATA POINTS
C  NATT   * INPUT* NUMBER OF ATTRIBUTES FOR EACH DATA POINT
C  NCLUST * INPUT* NUMBER OF CLUSTERS
C  IASSGN *IN/OUT* ARRAY OF LENGTH N.  ON ENTRY, SHOULD CONTAIN THE
C                  INITIAL ASSIGNMENT OF SITES TO CLUSTERS.  ON EXIT,
C                  CONTAINS THE FINAL ASSIGNMENT.  THE I'TH ELEMENT OF
C                  THE ARRAY CONTAINS THE LABEL OF THE CLUSTER TO WHICH
C                  THE I'TH DATA POINT BELONGS.  LABELS MUST BE BETWEEN
C                  1 AND NCLUST, AND EACH OF THE VALUES 1 THROUGH NCLUST
C                  MUST OCCUR AT LEAST ONCE.
C  LIST   *OUTPUT* ARRAY OF LENGTH N. CONTAINS THE DATA POINTS IN
C                  CLUSTER 1, FOLLOWED BY THE DATA POINTS IN CLUSTER 2,
C                  ETC.  DATA POINTS IN EACH CLUSTER ARE LISTED IN
C                  INCREASING ORDER.  THE LAST DATA POINT IN EACH
C                  CLUSTER IS INDICATED BY A NEGATIVE NUMBER.
C  NUM    *OUTPUT* ARRAY OF LENGTH NCLUST.  NUMBER OF DATA POINTS IN
C                  EACH CLUSTER.
C  SS     *OUTPUT* WITHIN-GROUP SUM OF SQUARES OF THE FINAL CLUSTERS.
C  MAXIT  * INPUT* MAXIMUM NUMBER OF ITERATIONS FOR THE K-MEANS
C                  CLUSTERING ALGORITHM
C  IWORK  * LOCAL* (INTEGER) WORK ARRAY OF LENGTH NCLUST*3
C  RW     * LOCAL* REAL WORK ARRAY OF LENGTH NW.  N.B. THIS ARRAY IS OF
C                  TYPE REAL, NOT DOUBLE PRECISION!
C  NW     * INPUT* LENGTH OF ARRAY RW.  MUST BE AT LEAST
C                  (N+NCLUST)*(NATT+1)+2*NCLUST
C
C  OTHER ROUTINES USED: APPLIED STATISTICS ALGORITHM AS136 (ROUTINES
C                       KMNS,OPTRA,QTRAN), AVAILABLE FROM
C                       HTTP://STAT.LIB.CMU.EDU/APSTAT/136
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION X(NX,NATT)
      INTEGER IASSGN(N),LIST(N),NUM(NCLUST),IWORK(NCLUST,3)
      REAL RW(NW)
      DATA ZERO/0D0/
C
C         SET ADDRESSES FOR SUBDIVISION OF WORK ARRAY
C
      MC=1
      MA=MC+NCLUST*NATT
      MAN1=MA+N*NATT
      MAN2=MAN1+NCLUST
      MWSS=MAN2+NCLUST
      MD=MWSS+NCLUST
      NWREQ=MD+N-1
      IF(NW.LT.NWREQ)GOTO 1000
      LA=MA-1
```

```
      LWSS=MWSS-1
C
C          COPY ATTRIBUTES TO WORK ARRAY
C
      IW=LA
      DO 5 IATT=1,NATT
      DO 5 I=1,N
      IW=IW+1
    5 RW(IW)=X(I,IATT)
C
C          COMPUTE CLUSTER CENTERS
C
      DO 10 ICL=1,NCLUST
   10 NUM(ICL)=0
      IWMAX=NCLUST*NATT
      DO 20 IW=1,IWMAX
   20 RW(IW)=ZERO
      DO 40 I=1,N
      ICL=IASSGN(I)
      IF(ICL.LE.0.OR.ICL.GT.NCLUST)GOTO 1010
      NUM(ICL)=NUM(ICL)+1
      IW=ICL
      DO 30 IATT=1,NATT
      RW(IW)=RW(IW)+X(I,IATT)
      IW=IW+NCLUST
   30 CONTINUE
   40 CONTINUE
      DO 60 ICL=1,NCLUST
      NSIZE=NUM(ICL)
      IF(NSIZE.EQ.0)GOTO 1020
      IW=ICL
      DO 50 IATT=1,NATT
      RW(IW)=RW(IW)/NSIZE
      IW=IW+NCLUST
   50 CONTINUE
   60 CONTINUE
C
C          CALL ALGORITHM AS136
C
      CALL KMNS(RW(MA),N,NATT,RW(MC),NCLUST,IASSGN,LIST,NUM,RW(MAN1),
     *  RW(MAN2),IWORK(1,1),RW(MD),IWORK(1,2),IWORK(1,3),MAXIT,RW(MWSS),
     *  IFAULT)
      IF(IFAULT.EQ.2)WRITE(6,7030)
C
C          COMPUTE LIST ARRAY AND FINAL SUM OF SQUARES
C
      I=0
      DO 80 ICL=1,NCLUST
      DO 70 K=1,N
      IF(IASSGN(K).NE.ICL)GOTO 70
      I=I+1
      LIST(I)=K
   70 CONTINUE
      LIST(I)=-LIST(I)
   80 CONTINUE
      SS=ZERO
      DO 90 ICL=1,NCLUST
   90 SS=SS+RW(LWSS+ICL)
C
      RETURN
C
 1000 WRITE(6,7000)NWREQ
      RETURN
 1010 WRITE(6,7010)I
      RETURN
 1020 WRITE(6,7020)ICL
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE CLUKM  : INSUFFICIENT WORKSPACE.',
     *  ' LENGTH OF WORK ARRAY SHOULD BE AT LEAST ',I8)
 7010 FORMAT(' *** ERROR *** ROUTINE CLUKM  :',
```

```
     *  ' INVALID INITIAL CLUSTER NUMBER FOR DATA POINT ',I5)
 7020 FORMAT(' *** ERROR *** ROUTINE CLUKM  :',
     *  ' INITIAL CLUSTERS INVALID.  CLUSTER ',I4,' HAS NO MEMBERS.')
 7030 FORMAT(' ** WARNING ** ROUTINE CLUKM  :',
     *  ' ITERATION HAS NOT CONVERGED. RESULTS MAY BE UNRELIABLE.')
C
      END
```

```
      SUBROUTINE REGLMR(NSITE,NMOM,NXMOM,XMOM,WEIGHT,RMOM)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  REGIONAL WEIGHTED AVERAGE OF L-MOMENTS
C
C  PARAMETERS OF ROUTINE:
C  NSITE  * INPUT* NUMBER OF SITES IN REGION
C  NMOM   * INPUT* NUMBER OF L-MOMENTS TO BE FOUND.
C  NXMOM  * INPUT* THE FIRST DIMENSION OF ARRAY XMOM, AS DECLARED IN THE
C                  CALLING PROGRAM.
C  XMOM   * INPUT* ARRAY OF DIMENSION (NXMOM,NSITE). X(I,J) CONTAINS
C                  THE I'TH L-MOMENT RATIO FOR SITE J.
C  WEIGHT * INPUT* ARRAY OF LENGTH NSITE. CONTAINS THE WEIGHTS TO BE
C                  APPLIED TO EACH SITE.
C  RMOM   *OUTPUT* ARRAY OF LENGTH NMOM. ON EXIT, CONTAINS THE REGIONAL
C                  WEIGHTED AVERAGE L-MOMENT RATIOS.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XMOM(NXMOM,NSITE),WEIGHT(NSITE),RMOM(NMOM)
      DATA ZERO/0D0/,ONE/1D0/
      IF(NMOM.LT.2.OR.NMOM.GT.NXMOM)GOTO 1000
      DO 10 J=1,NMOM
   10 RMOM(J)=ZERO
      WSUM=ZERO
      DO 30 ISITE=1,NSITE
      SMEAN=XMOM(1,ISITE)
      IF(SMEAN.EQ.ZERO)GOTO 1010
      W=WEIGHT(ISITE)
      WSUM=WSUM+W
      RMOM(2)=RMOM(2)+W*XMOM(2,ISITE)/SMEAN
      IF(NMOM.EQ.2)GOTO 30
      DO 20 J=3,NMOM
   20 RMOM(J)=RMOM(J)+W*XMOM(J,ISITE)
   30 CONTINUE
      IF(WSUM.LE.ZERO)GOTO 1020
      RMOM(1)=ONE
      RMOM(2)=RMOM(2)/WSUM
      IF(NMOM.EQ.2)RETURN
      DO 40 J=3,NMOM
   40 RMOM(J)=RMOM(J)/WSUM
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
 1010 WRITE(6,7010)ISITE
      RETURN
 1020 WRITE(6,7020)
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE REGLMR : PARAMETER NMOM INVALID')
 7010 FORMAT(' *** ERROR *** ROUTINE REGLMR : ZERO MEAN AT SITE',I4)
 7020 FORMAT(' *** ERROR *** ROUTINE REGLMR :',
     *  ' SUM OF WEIGHTS IS NEGATIVE OR ZERO')
      END
```

```
      SUBROUTINE REGTST(NSITES,NAMES,LEN,XMOM,A,B,SEED,NSIM,NPROB,PROB,
     *  KPRINT,KOUT,RMOM,D,VOBS,VBAR,VSD,H,Z,PARA)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  CALCULATES THREE STATISTICS USEFUL IN REGIONAL FREQUENCY ANALYSIS
C
C  DISCORDANCY MEASURE, D(I), FOR INDIVIDUAL SITES IN A REGION.
C      LARGE VALUES MIGHT BE USED AS A FLAG TO INDICATE POTENTIAL ERRORS
C      IN THE DATA AT THE SITE.  "LARGE" MIGHT BE 3 FOR REGIONS WITH 15
C      OR MORE SITES, BUT LESS (EXACT VALUES IN ARRAY DC1) FOR SMALLER
C      REGIONS.
C
C  HETEROGENEITY MEASURES, H(J), FOR A REGION BASED UPON EITHER:-
C      J=1: THE WEIGHTED S.D. OF THE L-CVS OR
C      J=2: THE AVERAGE DISTANCE FROM THE SITE TO THE REGIONAL AVERAGE
C           ON A GRAPH OF L-CV VS. L-SKEWNESS
C      J=3: THE AVERAGE DISTANCE FROM THE SITE TO THE REGIONAL AVERAGE
C           ON A GRAPH OF L-SKEWNESS VS. L-KURTOSIS
C
C      IN PRACTICE H(1) IS PROBABLY SUFFICIENT.  A VALUE GREATER THAN
C      (SAY) 1.0 SUGGESTS THAT FURTHER SUBDIVISION OF THE REGION SHOULD
C      BE CONSIDERED AS IT MIGHT IMPROVE QUANTILE ESTIMATES.
C
C  GOODNESS-OF-FIT MEASURES, Z(K), FOR 5 CANDIDATE DISTRIBUTIONS:
C      K=1: GENERALIZED LOGISTIC
C      K=2: GENERALIZED EXTREME VALUE
C      K=3: GENERALIZED NORMAL (LOGNORMAL)
C      K=4: PEARSON TYPE III (3-PARAMETER GAMMA)
C      K=5: GENERALIZED PARETO
C
C      PROVIDED THAT THE REGION IS ACCEPTABLY CLOSE TO HOMOGENEOUS,
C      THE FIT MAY BE JUDGED ACCEPTABLE AT 10% SIGNIFICANCE LEVEL
C      IF Z(K) IS LESS THAN 1.645 IN ABSOLUTE VALUE.
C
C  FOR FURTHER DETAILS SEE J.R.M. HOSKING AND J.R. WALLIS (1997),
C  "REGIONAL FREQUENCY ANALYSIS: AN APPROACH BASED ON L-MOMENTS",
C  CAMBRIDGE UNIVERSITY PRESS, CHAPTERS 3-5.
C
C  PARAMETERS OF ROUTINE:
C  NSITES * INPUT* NUMBER OF SITES IN REGION
C  NAMES  * INPUT* CHARACTER*12 ARRAY OF LENGTH NSITES. SITE NAMES.
C  LEN    * INPUT* ARRAY OF LENGTH NSITES. RECORD LENGTHS AT EACH SITE.
C  XMOM   * INPUT* ARRAY OF DIMENSION (5,NSITES). ARRAY CONTAINING
C                  THE FIRST 5 SAMPLE L-MOMENTS FOR EACH SITE, IN THE
C                  ORDER MEAN, L-CV, L-SKEWNESS, L-KURTOSIS, T-5, I.E
C                  XMOM(I,J) CONTAINS THE I'TH L-MOMENT FOR SITE J.
C                    N.B. XMOM(2,.) CONTAINS L-CV, NOT THE USUAL L-2!
C  A      * INPUT* ) PARAMETERS OF
C  B      * INPUT* ) PLOTTING POSITION.
C                  NOTE: A AND B SHOULD BE THE SAME AS THE VALUES USED
C                  TO CALCULATE THE MOMENTS IN THE XMOM ARRAY.
C  SEED   * INPUT* SEED FOR RANDOM NUMBER GENERATOR. SHOULD BE A WHOLE
C                  NUMBER IN THE RANGE 2D0 TO 2147483647D0.
C  NSIM   * INPUT* NUMBER OF SIMULATED WORLDS FOR HETEROGENEITY AND
C                  GOODNESS-OF-FIT TESTS.
C                    NOTE: NSIM=0 WILL FORCE RETURN AT COMPLETION OF
C                  OUTLIER TEST.  NSIM=1 WILL SUPPRESS CALCULATION OF
C                  H AND Z STATISTICS, BUT PARAMETER AND QUANTILE
```

```
C                    ESTIMATES WILL BE FOUND.
C  NPROB  * INPUT* NUMBER OF QUANTILES TO BE CALCULATED
C  PROB   * INPUT* ARRAY OF LENGTH NPROB.  PROBABILITIES FOR WHICH
C                  QUANTILES ARE TO BE CALCULATED.
C  KPRINT * INPUT* OUTPUT FLAG. SHOULD BE SET TO
C                  0  TO SUPPRESS OUTPUT
C                  1  TO PRINT OUTPUT
C  KOUT   * INPUT* CHANNEL TO WHICH OUTPUT IS DIRECTED
C  RMOM   *OUTPUT* ARRAY OF LENGTH 5. ON EXIT, CONTAINS THE REGIONAL
C                  WEIGHTED AVERAGE L-MOMENT RATIOS.
C  D      *OUTPUT* ARRAY OF LENGTH NSITES. ON EXIT, CONTAINS THE
C                  DISCORDANCY MEASURE (D STATISTIC) FOR EACH SITE.
C  VOBS   *OUTPUT* ARRAY OF LENGTH 3. ON EXIT, CONTAINS THE REGIONAL
C                  OBSERVED VALUES OF 3 HETEROGENEITY STATISTICS:
C                  (1) WEIGHTED S.D. OF L-CVS;
C                  (2) AVERAGE OF L-CV/L-SKEW DISTANCES;
C                  (3) AVERAGE OF L-SKEW/L-KURTOSIS DISTANCES.
C  VBAR   *OUTPUT* ARRAY OF LENGTH 3. ON EXIT, CONTAINS THE MEAN OF THE
C                  SIMULATED VALUES OF THE 3 HETEROGENEITY STATISTICS.
C  VSD    *OUTPUT* ARRAY OF LENGTH 3. ON EXIT, CONTAINS THE S.D. OF THE
C                  SIMULATED VALUES OF THE 3 HETEROGENEITY STATISTICS.
C  H      *OUTPUT* ARRAY OF LENGTH 3. ON EXIT, CONTAINS HETEROGENEITY
C                  MEASURES (H STATISTICS), I.E. H=(VOBS-VBAR)/VSD.
C  Z      *OUTPUT* ARRAY OF LENGTH 5. ON EXIT, CONTAINS GOODNESS-OF-FIT
C                  MEASURES (Z STATISTICS) FOR 5 DISTRIBUTIONS:
C                  (1) GEN. LOGISTIC, (2) GEN. EXTREME VALUE,
C                  (3) GEN. NORMAL, (4) PEARSON TYPE III,
C                  (5) GEN. PARETO.
C  PARA   *OUTPUT* ARRAY OF DIMENSION (5,6). ON EXIT, IF NSIM.GE.1,
C                  CONTAINS PARAMETERS OF GROWTH CURVES FITTED BY THE
C                  ABOVE 5 DISTRIBUTIONS, PLUS WAKEBY.
C
C  OTHER ROUTINES USED: DERF,DIGAMD,DLGAMA,DURAND,GAMIND,PELGEV,PELGLO,
C    PELGNO,PELGPA,PELKAP,PELPE3,PELWAK,QUAGAM,QUAGEV,QUAGLO,QUAGNO,
C    QUAGPA,QUAKAP,QUAPE3,QUASTN,QUAWAK,SAMLMR,SORT
C
C  QUANTITIES DEFINED IN PARAMETER STATEMENT:
C  MAXNS  - MUST BE AT LEAST AS LARGE AS NSITES
C  MAXREC - MUST BE AT LEAST AS LARGE AS EACH ELEMENT OF ARRAY LEN
C  MAXQ   - MUST BE AT LEAST AS LARGE AS NPROB
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MAXNS=200,MAXQ=30,MAXREC=200)
C
      CHARACTER*1 BLANK,STAR,LOOK1,LOOK2
      CHARACTER DISTRI*18(6),NAMES*12(NSITES)
      DOUBLE PRECISION D(NSITES),DC1(14),DC2(18),H(3),PARA(5,6),
     *  PROB(NPROB),Q(MAXQ),RMOM(5),RPARA(4),SMAT(3,3),TMOM(4),T4FIT(5),
     *  VBAR(3),VOBS(3),VSD(3),WORK(MAXNS,3),X(MAXREC),XMOM(5,NSITES),
     *  Z(5)
      INTEGER LEN(NSITES)
      DATA BLANK/' '/,STAR/'*'/
      DATA ZERO/0D0/,ONE/1D0/,TWO/2D0/,THREE/3D0/
      DATA DISTRI/
     *  'GEN. LOGISTIC     ','GEN. EXTREME VALUE','GEN. NORMAL       ',
     *  'PEARSON TYPE III  ','GEN. PARETO       ','WAKEBY            '/
C
C         COEFFICIENTS OF POWER-SERIES APPROXIMATIONS OF TAU-4 IN TERMS
C         OF TAU-3, FOR THE FIRST 5 DISTRIBUTIONS IN ARRAY DISTRI
C
      DATA GLOC0,GLOC2/0.16667D0,0.83333D0/
      DATA GEVC0,GEVC1,GEVC2,GEVC3,GEVC4,GEVC5,GEVC6/
     *  0.10701D0, 0.11090D0, 0.84838D0,-0.06669D0,
     *  0.00567D0,-0.04208D0, 0.03763D0/
      DATA GNOC0,GNOC2,GNOC4,GNOC6,GNOC8/
     *  0.12282D0,0.77518D0,0.12279D0,-0.13638D0,0.11368D0/
      DATA PE3C0,PE3C2,PE3C4,PE3C6,PE3C8/
     *  0.12240D0,0.30115D0,0.95812D0,-0.57488D0,0.19383D0/
      DATA GPAC1,GPAC2,GPAC3,GPAC4/
     *  0.20196D0,0.95924D0,-0.20096D0,0.04061D0/
C
```

```
C         CRITICAL VALUES FOR D, H AND Z STATISTICS
C
      DATA DC1/4*3D0,1.3330D0,1.6481D0,1.9166D0,2.1401D0,2.3287D0,
     *               2.4906D0,2.6321D0,2.7573D0,2.8694D0,2.9709D0/
      DATA DC2/4*4D0,1.3333D0,1.6648D0,1.9821D0,2.2728D0,2.5337D0,
     *               2.7666D0,2.9748D0,3.1620D0,3.3310D0,3.4844D0,
     *               3.6246D0,3.7532D0,3.8718D0,3.9816D0/
      DATA HCRIT1,HCRIT2/1D0,2D0/
      DATA ZCRIT/1.645D0/
C
C         INITIALIZE ARRAYS
C
      NMAX=0
      SUMLEN=0
      DO 10 I=1,NSITES
      NREC=LEN(I)
      IF(NREC.GT.NMAX)NMAX=NREC
      SUMLEN=SUMLEN+NREC
   10 D(I)=ZERO
      DO 20 K=1,3
      VOBS(K)=ZERO
      VBAR(K)=ZERO
      VSD(K)=ZERO
      H(K)=ZERO
   20 CONTINUE
      DO 30 IDIST=1,5
   30 Z(IDIST)=ZERO
      DO 40 IPARA=1,5
      DO 40 IDIST=1,6
   40 PARA(IPARA,IDIST)=ZERO
      IF(NSITES.GT.MAXNS)GOTO 1000
C
C         CALCULATE THE WEIGHTED MEAN OF L-CV, L-SKEW, L-KURTOSIS
C
      DO 60 K=2,5
      RMOM(K)=ZERO
      DO 50 I=1,NSITES
   50 RMOM(K)=RMOM(K)+LEN(I)*XMOM(K,I)
   60 RMOM(K)=RMOM(K)/SUMLEN
      RMOM(1)=ONE
C
C         CALCULATE SUM OF SQUARES MATRIX
C
      IF(NSITES.LE.3)GOTO 135
      SUM2=ZERO
      SUM3=ZERO
      SUM4=ZERO
      DO 70 I=1,NSITES
      SUM2=SUM2+XMOM(2,I)
      SUM3=SUM3+XMOM(3,I)
      SUM4=SUM4+XMOM(4,I)
   70 CONTINUE
      SUM2=SUM2/NSITES
      SUM3=SUM3/NSITES
      SUM4=SUM4/NSITES
      DO 80 I=1,NSITES
      WORK(I,1)=XMOM(2,I)-SUM2
      WORK(I,2)=XMOM(3,I)-SUM3
      WORK(I,3)=XMOM(4,I)-SUM4
   80 CONTINUE
      DO 100 J=1,3
      DO 100 K=J,3
      SMAT(J,K)=ZERO
      DO 90 I=1,NSITES
   90 SMAT(J,K)=SMAT(J,K)+WORK(I,J)*WORK(I,K)
  100 CONTINUE
C
C         INVERT SUM OF SQUARES MATRIX
C
      DO 110 K=1,3
      IF(SMAT(1,1).LE.ZERO)GOTO 1030
```

```
      TEMP0=ONE/SMAT(1,1)
      TEMP1=-SMAT(1,2)*TEMP0
      TEMP2=-SMAT(1,3)*TEMP0
      IF(K.GT.2)TEMP1=-TEMP1
      IF(K.GT.1)TEMP2=-TEMP2
      SMAT(1,1)=SMAT(2,2)+TEMP1*SMAT(1,2)
      SMAT(1,2)=SMAT(2,3)+TEMP1*SMAT(1,3)
      SMAT(2,2)=SMAT(3,3)+TEMP2*SMAT(1,3)
      SMAT(1,3)=TEMP1
      SMAT(2,3)=TEMP2
      SMAT(3,3)=TEMP0
  110 CONTINUE
      SMAT(2,1)=SMAT(1,2)
      SMAT(3,1)=SMAT(1,3)
      SMAT(3,2)=SMAT(2,3)
C
C         CALCULATE DISCORDANCY MEASURES (D STATISTICS)
C
      FACTOR=NSITES/THREE
      DO 130 I=1,NSITES
      DO 120 J=1,3
      DO 120 K=1,3
  120 D(I)=D(I)+WORK(I,J)*WORK(I,K)*SMAT(J,K)
      D(I)=D(I)*FACTOR
      WORK(I,1)=D(I)
  130 CONTINUE
      CALL SORT(WORK(1,1),NSITES)
      GOTO 140
  135 DO 138 I=1,NSITES
  138 D(I)=ONE
C
C         PRINT DISCORDANCY MEASURES
C
  140 CONTINUE
      IF(KPRINT.LE.0)GOTO 160
      WRITE(KOUT,6000)
      DCRIT1=DC1(1)
      DCRIT2=DC2(1)
      IF(NSITES.LE.14)DCRIT1=DC1(NSITES)
      IF(NSITES.LE.18)DCRIT2=DC2(NSITES)
      KSTART=1
      DO 150 I=1,NSITES
      LOOK1=BLANK
      LOOK2=BLANK
      IF(D(I).GE.DCRIT1)LOOK1=STAR
      IF(D(I).GE.DCRIT2)LOOK2=STAR
      IF(D(I).LT.DCRIT1)KSTART=KSTART+1
      WRITE(KOUT,6010)I,LEN(I),NAMES(I),(XMOM(K,I),K=2,4),
     *  D(I),LOOK1,LOOK2
  150 CONTINUE
      WRITE(KOUT,6020)(RMOM(K),K=2,4)
      IF(KSTART.LT.NSITES)WRITE(KOUT,6030)(WORK(K,1),K=KSTART,NSITES)
  160 CONTINUE
C
      IF(NSIM.LE.0)RETURN
      IF(NPROB.GT.MAXQ)GOTO 1010
      IF(NSIM.EQ.1)GOTO 270
      IF(NMAX.GT.MAXREC)GOTO 1020
C
C         FIT KAPPA DISTRIBUTION TO REGIONAL L-MOMENTS
C
      CALL PELKAP(RMOM,RPARA,IFAIL)
      IF(IFAIL.EQ.0)GOTO 180
      CALL PELGLO(RMOM,RPARA)
      RPARA(4)=-ONE
  180 IF(KPRINT.GT.0)WRITE(KOUT,6040)(RPARA(K),K=1,4)
C
C         START THE NSIM REPETITIONS
C
      T4BAR=ZERO
      T4SD=ZERO
```

```
      DO 220 ISIM=1,NSIM
      SUM2=ZERO
      SUM3=ZERO
      SUM4=ZERO
C
C         START OF LOOP OVER SITES
C
      DO 200 I=1,NSITES
      NREC=LEN(I)
C
C         GET VECTOR OF UNIFORM RANDOM NUMBERS
C
      CALL DURAND(SEED,NREC,X)
C
C         TRANSFORM FROM UNIFORM TO KAPPA
C
      DO 190 J=1,NREC
      X(J)=QUAKAP(X(J),RPARA)
  190 CONTINUE
C
C         FIND L-MOMENTS OF SIMULATED DATA
C
      CALL SORT(X,NREC)
      CALL SAMLMR(X,NREC,TMOM,4,A,B)
      CV=TMOM(2)/TMOM(1)
      WORK(I,1)=CV
      WORK(I,2)=TMOM(3)
      WORK(I,3)=TMOM(4)
      SUM2=SUM2+NREC*CV
      SUM3=SUM3+NREC*TMOM(3)
      SUM4=SUM4+NREC*TMOM(4)
C
C         END OF LOOP OVER SITES
C
  200 CONTINUE
C
      SUM2=SUM2/SUMLEN
      SUM3=SUM3/SUMLEN
      SUM4=SUM4/SUMLEN
      T4BAR=T4BAR+SUM4
      T4SD=T4SD+SUM4**2
C
C         CALCULATE HETEROGENEITY V-STATISTICS FOR SIMULATED DATA
C
      IF(NSITES.EQ.1)GOTO 215
      V1=ZERO
      V2=ZERO
      V3=ZERO
      DO 210 I=1,NSITES
      NREC=LEN(I)
      TEMP2=(WORK(I,1)-SUM2)**2
      TEMP3=(WORK(I,2)-SUM3)**2
      TEMP4=(WORK(I,3)-SUM4)**2
      V1=V1+NREC*TEMP2
      V2=V2+NREC*DSQRT(TEMP2+TEMP3)
      V3=V3+NREC*DSQRT(TEMP3+TEMP4)
  210 CONTINUE
      V1=DSQRT(V1/SUMLEN)
      V2=V2/SUMLEN
      V3=V3/SUMLEN
      VBAR(1)=VBAR(1)+V1
      VBAR(2)=VBAR(2)+V2
      VBAR(3)=VBAR(3)+V3
      VSD(1)=VSD(1)+V1**2
      VSD(2)=VSD(2)+V2**2
      VSD(3)=VSD(3)+V3**2
  215 CONTINUE
C
C         END OF THE NSIM REPETITIONS
C
  220 CONTINUE
```

```
C
C         CALCULATE HETEROGENEITY V-STATISTICS FOR OBSERVED DATA
C
      IF(NSITES.EQ.1)GOTO 235
      V1=ZERO
      V2=ZERO
      V3=ZERO
      DO 225 I=1,NSITES
      NREC=LEN(I)
      TEMP2=(XMOM(2,I)-RMOM(2))**2
      TEMP3=(XMOM(3,I)-RMOM(3))**2
      TEMP4=(XMOM(4,I)-RMOM(4))**2
      V1=V1+NREC*TEMP2
      V2=V2+NREC*DSQRT(TEMP2+TEMP3)
      V3=V3+NREC*DSQRT(TEMP3+TEMP4)
  225 CONTINUE
      VOBS(1)=DSQRT(V1/SUMLEN)
      VOBS(2)=V2/SUMLEN
      VOBS(3)=V3/SUMLEN
C
C         CALCULATE AND PRINT HETEROGENEITY MEASURES (H STATISTICS)
C
      IF(KPRINT.GT.0)WRITE(KOUT,6050)NSIM
      DO 230 J=1,3
      VBAR(J)=VBAR(J)/NSIM
      VSD(J)=DSQRT((VSD(J)-NSIM*VBAR(J)**2)/(NSIM-ONE))
      H(J)=(VOBS(J)-VBAR(J))/VSD(J)
      IF(KPRINT.LE.0)GOTO 230
      LOOK1=BLANK
      LOOK2=BLANK
      IF(H(J).GE.HCRIT1)LOOK1=STAR
      IF(H(J).GE.HCRIT2)LOOK2=STAR
      IF(J.EQ.1)WRITE(KOUT,6060)VOBS(J),VBAR(J),VSD(J),H(J),LOOK1,LOOK2
      IF(J.EQ.2)WRITE(KOUT,6070)VOBS(J),VBAR(J),VSD(J),H(J),LOOK1,LOOK2
      IF(J.EQ.3)WRITE(KOUT,6080)VOBS(J),VBAR(J),VSD(J),H(J),LOOK1,LOOK2
  230 CONTINUE
  235 CONTINUE
C
C         FIND TAU-4 VALUES OF EACH CANDIDATE DISTRIBUTION
C
      S=RMOM(3)
      SS=S*S
      T4FIT(1)=GLOC0+SS*GLOC2
      T4FIT(2)=
     *  GEVC0+S*(GEVC1+S*(GEVC2+S*(GEVC3+S*(GEVC4+S*(GEVC5+S*GEVC6)))))
      T4FIT(3)=GNOC0+SS*(GNOC2+SS*(GNOC4+SS*(GNOC6+SS*GNOC8)))
      T4FIT(4)=PE3C0+SS*(PE3C2+SS*(PE3C4+SS*(PE3C6+SS*PE3C8)))
      T4FIT(5)=S*(GPAC1+S*(GPAC2+S*(GPAC3+S*GPAC4)))
C
C         CALCULATE GOODNESS-OF-FIT MEASURES (Z STATISTICS)
C
      T4BAR=T4BAR/NSIM
      T4SD=DSQRT((T4SD-NSIM*T4BAR**2)/(NSIM-ONE))
      DO 240 IDIST=1,5
      Z(IDIST)=(T4FIT(IDIST)+T4BAR-TWO*RMOM(4))/T4SD
  240 CONTINUE
C
C         PRINT Z STATISTICS
C
      IF(KPRINT.LE.0)GOTO 260
      WRITE(KOUT,6090)NSIM
      DO 250 IDIST=1,5
      LOOK1=BLANK
      IF(DABS(Z(IDIST)).LT.ZCRIT)LOOK1=STAR
  250 WRITE(KOUT,6100)DISTRI(IDIST),T4FIT(IDIST),Z(IDIST),LOOK1
  260 CONTINUE
C
C         FIT DISTRIBUTIONS
C
  270 CONTINUE
      CALL PELGLO(RMOM,PARA(1,1))
```

```
      CALL PELGEV(RMOM,PARA(1,2))
      CALL PELGNO(RMOM,PARA(1,3))
      CALL PELPE3(RMOM,PARA(1,4))
      CALL PELGPA(RMOM,PARA(1,5))
      CALL PELWAK(RMOM,PARA(1,6),IFAIL)
C
C         FOR SUCCESSFUL CANDIDATES AND WAKEBY, PRINT PARAMETERS ...
C
      IF(KPRINT.LE.0)GOTO 320
      IF(NSIM.EQ.1)WRITE(KOUT,6110)
      IF(NSIM.GT.1)WRITE(KOUT,6120)
      DO 280 IDIST=1,5
      IF(DABS(Z(IDIST)).LE.ZCRIT)
     *  WRITE(KOUT,6130)DISTRI(IDIST),(PARA(IPARA,IDIST),IPARA=1,3)
  280 CONTINUE
      WRITE(KOUT,6130)DISTRI(6),(PARA(IPARA,6),IPARA=1,5)
C
C         ... AND ESTIMATE AND PRINT QUANTILES
C
      IF(NPROB.EQ.0)GOTO 320
      WRITE(KOUT,6140)PROB
      DO 300 IDIST=1,5
      IF(DABS(Z(IDIST)).GT.ZCRIT)GOTO 300
      DO 290 IQ=1,NPROB
      IF(IDIST.EQ.1)Q(IQ)=QUAGLO(PROB(IQ),PARA(1,1))
      IF(IDIST.EQ.2)Q(IQ)=QUAGEV(PROB(IQ),PARA(1,2))
      IF(IDIST.EQ.3)Q(IQ)=QUAGNO(PROB(IQ),PARA(1,3))
      IF(IDIST.EQ.4)Q(IQ)=QUAPE3(PROB(IQ),PARA(1,4))
      IF(IDIST.EQ.5)Q(IQ)=QUAGPA(PROB(IQ),PARA(1,5))
  290 CONTINUE
      WRITE(KOUT,6150)DISTRI(IDIST),(Q(IQ),IQ=1,NPROB)
  300 CONTINUE
      DO 310 IQ=1,NPROB
  310 Q(IQ)=QUAWAK(PROB(IQ),PARA(1,6))
      WRITE(KOUT,6150)DISTRI(6),(Q(IQ),IQ=1,NPROB)
  320 CONTINUE
C
      RETURN
C
 1000 WRITE(KOUT,7000)'MAXNS'
      RETURN
 1010 WRITE(KOUT,7000)'MAXQ'
      RETURN
 1020 WRITE(KOUT,7000)'MAXREC'
      RETURN
 1030 WRITE(KOUT,7010)
      GOTO 140
C
 6000 FORMAT(/' SITE    N      NAME       L-CV   L-SKEW  L-KURT   D(I)')
 6010 FORMAT(2I5,2X,A12,3F8.4,F7.2,2X,2A1)
 6020 FORMAT(/5X,'WEIGHTED MEANS',5X,6F8.4)
 6030 FORMAT(/' FLAGGED TEST VALUES'/(15F5.1))
 6040 FORMAT(/' PARAMETERS OF REGIONAL KAPPA DISTRIBUTION ',4F8.4)
 6050 FORMAT(//' ***** HETEROGENEITY MEASURES *****'/
     *  ' (NUMBER OF SIMULATIONS  =',I6,')')
 6060 FORMAT(/' OBSERVED     S.D. OF GROUP L-CV          =',F8.4/
    *         ' SIM. MEAN OF S.D. OF GROUP L-CV          =',F8.4/
    *         ' SIM. S.D. OF S.D. OF GROUP L-CV          =',F8.4/
    *         ' STANDARDIZED TEST VALUE H(1)             =',F6.2,2X,2A1)
 6070 FORMAT(/' OBSERVED AVE.  OF L-CV / L-SKEW DISTANCE =',F8.4/
    *         ' SIM. MEAN OF AVE. L-CV / L-SKEW DISTANCE =',F8.4/
    *         ' SIM. S.D. OF AVE. L-CV / L-SKEW DISTANCE =',F8.4/
    *         ' STANDARDIZED TEST VALUE H(2)             =',F6.2,2X,2A1)
 6080 FORMAT(/' OBSERVED AVE.  OF L-SKEW/L-KURT DISTANCE =',F8.4/
    *         ' SIM. MEAN OF AVE. L-SKEW/L-KURT DISTANCE =',F8.4/
    *         ' SIM. S.D. OF AVE. L-SKEW/L-KURT DISTANCE =',F8.4/
    *         ' STANDARDIZED TEST VALUE H(3)             =',F6.2,2X,2A1)
 6090 FORMAT(//' ***** GOODNESS-OF-FIT MEASURES *****'/
    *  ' (NUMBER OF SIMULATIONS  =',I6,')'/)
 6100 FORMAT(1X,A18,2X,' L-KURTOSIS=',F6.3,2X,' Z VALUE=',F6.2,1X,A1)
 6110 FORMAT(//' PARAMETER ESTIMATES'/)
```

```
 6120 FORMAT(//' PARAMETER ESTIMATES FOR DISTRIBUTIONS ACCEPTED AT THE',
     *  ' 90% LEVEL'/)
 6130 FORMAT(1X,A18,1X,5F7.3)
 6140 FORMAT(/' QUANTILE ESTIMATES'/19X,(1X,14F7.3))
 6150 FORMAT(1X,A18,(1X,14F7.3))
C
 7000 FORMAT(' *** ERROR *** ROUTINE REGTST :',
     *  ' INSUFFICIENT WORKSPACE - RECOMPILE WITH LARGER VALUE OF ',A6)
 7010 FORMAT(' *** ERROR *** ROUTINE REGTST : UNABLE TO INVERT',
     *  ' SUM-OF-SQUARES MATRIX.'/31X,'D STATISTICS NOT CALCULATED.')
C
      END
```

```
      PROGRAM XCLUST
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  EXAMPLE PROGRAM FOR CLUSTER ANALYSIS.  THE PROGRAM READS IN
C  ATTRIBUTES FOR A NUMBER OF SITES, TRANSFORMS THE ATTRIBUTES, FORMS
C  CLUSTERS BY WARD'S METHOD, PRINTS INFORMATION ABOUT THE CLUSTERS,
C  AND REFINES THE CLUSTERS USING THE K-MEANS ALGORITHM.
C    THE ANALYSIS FOLLOWS HOSKING AND WALLIS ("REGIONAL FREQUENCY
C  ANALYSIS: AN APPROACH BASED ON L-MOMENTS", CAMBRIDGE UNIV. PRESS,
C  1997, SEC. 9.2).
C
C  PARAMETERS OF PROGRAM:
C  MAXNS  - MAXIMUM NUMBER OF SITES
C  NATMAX - MAXIMUM NUMBER OF ATTRIBUTES
C  INFILE - STREAM NUMBER TO WHICH INPUT FILE IS ATTACHED
C  KOUT   - STREAM NUMBER TO WHICH OUTPUT FILE IS ATTACHED
C
C  ROUTINES USED: CLUAGG,CLUINF,CLUKM,KMNS,OPTRA,QTRAN
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MAXNS=104,NATMAX=10,INFILE=7,KOUT=6)
C
      PARAMETER (NWORK=MAXNS*(MAXNS-1)/2,NRWORK=NWORK)
      CHARACTER*12 ID(MAXNS)
      INTEGER MERGE(2,MAXNS),IWORK(MAXNS),NREC(MAXNS),
     *  IASSGN(MAXNS),LIST(MAXNS),NUM(MAXNS)
      DOUBLE PRECISION X(MAXNS,NATMAX),WGSS(MAXNS),WORK(NWORK),
     *  WEIGHT(NATMAX),XMOM(5,MAXNS),Z(NATMAX),CENT(NATMAX,MAXNS)
      REAL RWORK(NRWORK)
      DATA ZERO/0D0/,ONE/1D0/,THREE/3D0/
C
      N=104
      READ(INFILE,*)
C
C         FOR EACH DATA POINT ...
C
      DO 10 I=1,N
C
C         ... READ THE DATA ...
C
      READ(INFILE,5000)ID(I),XLAT,XLONG,AREA,ELEV
C
C         ... AND COMPUTE THE TRANSFORMED ATTRIBUTES
C
      X(I,1)=DLOG(AREA)
      X(I,2)=DSQRT(ELEV)
      X(I,3)=XLAT
      X(I,4)=XLONG
C
C         END OF LOOP OVER DATA POINTS
C
   10 CONTINUE
C
C         SET WEIGHTS FOR EACH ATTRIBUTE
C
      NATT=4
      WEIGHT(1)=THREE
      WEIGHT(2)=ONE
```

```
      WEIGHT(3)=ONE
      WEIGHT(4)=ONE
C
C         FOR EACH ATTRIBUTE ...
C
      DO 40 J=1,NATT
C
C         ... CALCULATE ITS STANDARD DEVIATION ACROSS THE DATA POINTS ...
C
      SUM1=ZERO
      SUM2=ZERO
      DO 20 I=1,N
      SUM1=SUM1+X(I,J)
      SUM2=SUM2+X(I,J)**2
   20 CONTINUE
      SD=DSQRT((SUM2-SUM1*SUM1/N)/(N-ONE))
C
C         ... DIVIDE THE WEIGHT BY THIS STANDARD DEVIATION ...
C
      WEIGHT(J)=WEIGHT(J)/SD
C
C         ... AND APPLY THE WEIGHT TO EACH DATA POINT
C
      DO 30 I=1,N
   30 X(I,J)=X(I,J)*WEIGHT(J)
C
C         END OF LOOP OVER ATTRIBUTES
C
   40 CONTINUE
C
C         WARD'S ALGORITHM
C
      NX=MAXNS
      NW=NWORK
      CALL CLUAGG(3,X,NX,N,NATT,MERGE,WGSS,IWORK,WORK,NW)
      WRITE(KOUT,6000)
      DO 50 I=1,N-1
      WRITE(KOUT,6010)I,N-I,MERGE(1,I),MERGE(2,I),WGSS(I)
   50 CONTINUE
C
C         PRINT INFORMATION ABOUT THE 7-CLUSTER GROUPING
C
      NCLUST=7
      CALL CLUINF(NCLUST,N,MERGE,IASSGN,LIST,NUM)
      WRITE(KOUT,6020)
      WRITE(KOUT,6030)(IASSGN(I),I=1,N)
      WRITE(KOUT,6040)
      IORIG=0
      DO 60 ICL=1,NCLUST
      NN=NUM(ICL)
      WRITE(KOUT,6050)ICL,NN
      WRITE(KOUT,6060)(IABS(LIST(I)),I=IORIG+1,IORIG+NN)
      IORIG=IORIG+NN
   60 CONTINUE
C
C         ADJUST CLUSTERS BY K-MEANS ALGORITHM
C
      MAXIT=10
      CALL CLUKM(X,NX,N,NATT,NCLUST,IASSGN,LIST,NUM,SS,MAXIT,IWORK,
     *  RWORK,NRWORK)
C
C         PRINT INFORMATION ABOUT ADJUSTED CLUSTERS
C
      WRITE(KOUT,6070)SS
      WRITE(KOUT,6020)
      WRITE(KOUT,6030)(IASSGN(I),I=1,N)
      WRITE(KOUT,6040)
      IORIG=0
      DO 70 ICL=1,NCLUST
      NN=NUM(ICL)
      WRITE(KOUT,6050)ICL,NN
```

```
      WRITE(KOUT,6060)(IABS(LIST(I)),I=IORIG+1,IORIG+NN)
      IORIG=IORIG+NN
   70 CONTINUE
C
C         FIND CLUSTER CENTERS, IN SPACE OF TRANSFORMED ATTRIBUTES
C
      WRITE(KOUT,6080)
      DO 80 ICL=1,NCLUST
      DO 80 IATT=1,NATT
   80 CENT(IATT,ICL)=ZERO
      ICL=1
      DO 100 I=1,N
      L=LIST(I)
      ISITE=IABS(L)
      DO 90 IATT=1,NATT
      CENT(IATT,ICL)=CENT(IATT,ICL)+X(ISITE,IATT)
   90 CONTINUE
      IF(L.LT.0)ICL=ICL+1
  100 CONTINUE
      DO 110 ICL=1,NCLUST
      NN=NUM(ICL)
      DO 110 IATT=1,NATT
  110 CENT(IATT,ICL)=CENT(IATT,ICL)/NN
C
C         TRANSFORM BACK TO ORIGINAL ATTRIBUTES
C
      DO 120 ICL=1,NCLUST
      Z(1)=DEXP(CENT(1,ICL)/WEIGHT(1))
      Z(2)=(CENT(2,ICL)/WEIGHT(2))**2
      Z(3)=CENT(3,ICL)/WEIGHT(3)
      Z(4)=CENT(4,ICL)/WEIGHT(4)
      WRITE(KOUT,6090)ICL,(Z(J),J=1,NATT)
  120 CONTINUE
C
      STOP
C
 5000 FORMAT(A8,4F8.0)
 6000 FORMAT(' MERGING SEQUENCE FROM WARD''S ALGORITHM'//
     * 8X,'NUMBER OF    MERGED       SUM OF'/
     * ' STAGE  CLUSTERS    CLUSTERS      SQUARES'/)
 6010 FORMAT(1X,I3,I9,I10,I5,F12.2)
 6020 FORMAT(/' ASSIGNMENT OF SITES TO CLUSTERS')
 6030 FORMAT(1X,10I4)
 6040 FORMAT(//' CLUSTER MEMBERSHIP')
 6050 FORMAT(/' CLUSTER',I4,'  HAS',I4,' MEMBERS:')
 6060 FORMAT(1X,10I4)
 6070 FORMAT(///' ADJUSTED CLUSTERS FROM K-MEANS ALGORITHM'/
     *  ' (SUM OF SQUARES =',F12.2,')')
 6080 FORMAT(/' TYPICAL MEMBERS OF EACH CLUSTER'/
     *  '          AREA      ELEV       LAT      LONG')
 6090 FORMAT(1X,I3,6F10.2)
C
      END
```

```
      PROGRAM XFIT
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  EXAMPLE PROGRAM FOR REGIONAL FREQUENCY ANALYSIS USING THE METHOD OF
C  L-MOMENTS. THE PROGRAM FITS A DISTRIBUTION TO REGIONAL DATA AND USES
C  IT TO ESTIMATE QUANTILES AT EACH SITE.
C
C  THIS EXAMPLE FITS A WAKEBY DISTRIBUTION, USING A VARIANT (PLOTTING
C  POSITION ESTIMATORS INSTEAD OF UNBIASED) OF THE REGIONAL L-MOMENT
C  ALGORITHM DESCRIBED BY HOSKING AND WALLIS ("REGIONAL FREQUENCY
C  ANALYSIS: AN APPROACH BASED ON L-MOMENTS", CAMBRIDGE UNIV. PRESS,
C  1997).  TO FIT A DIFFERENT DISTRIBUTION, REPLACE THE CALLS TO
C  SUBROUTINES PELWAK AND QUAWAK BY THE APPROPRIATE PEL... AND QUA...
C  ROUTINES, CHANGE THE 'WAKEBY' IN FORMAT STATEMENT 6030 AND CHANGE
C  THE VALUE OF PARAMETER NPAR.
C
C  PARAMETERS OF PROGRAM:
C  MAXNS  - SHOULD BE AT LEAST AS LARGE AS THE NUMBER OF SITES IN THE
C           REGION
C  MAXN   - SHOULD BE AT LEAST AS LARGE AS THE LARGEST RECORD LENGTH
C           AT ANY SITE IN THE REGION
C  NPAR   - NUMBER OF PARAMETERS IN THE DISTRIBUTION TO BE FITTED
C           (5 FOR WAKEBY, OF COURSE)
C  NPROB  - NUMBER OF FLOOD QUANTILES TO BE ESTIMATED AT EACH SITE
C  INFILE - STREAM NUMBER TO WHICH INPUT FILE IS ATTACHED
C
C  ARRAYS TO BE INITIALIZED IN DATA STATEMENTS:
C  PROB(NPROB) - PROBABILITIES FOR WHICH QUANTILES ARE TO BE ESTIMATED
C
C  VARIABLES TO BE INITIALIZED IN DATA STATEMENTS:
C  A      - ) PARAMETERS OF
C  B      - ) PLOTTING POSITION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MAXNS=100,MAXN=200,NPAR=5,NPROB=10,INFILE=7)
      CHARACTER*32 SITEID
      DOUBLE PRECISION PROB(NPROB),QUANT(NPROB),PARA(5),RMOM(5),
     *  RQUANT(NPROB),WEIGHT(MAXNS),X(MAXN),XMOM(5,MAXNS)
C
      DATA PROB/0.1D0,0.2D0,0.5D0,0.8D0,0.9D0,0.95D0,0.98D0,0.99D0,
     *  0.999D0,0.9999D0/
      DATA A,B/-0.35D0,0D0/
C
      IF(A.EQ.0D0.AND.B.EQ.0D0)WRITE(6,6000)
      IF(A.NE.0D0.OR.B.NE.0D0)WRITE(6,6010)A,B
C
C         READ THE DATA AND CALCULATE AT-SITE L-MOMENTS.
C         ASSUMED STRUCTURE OF DATA FILE IS AS FOLLOWS.
C         1. ONE RECORD CONTAINING THE NUMBER OF SITES IN THE REGION.
C         2. FOR EACH SITE:
C            A  ONE RECORD CONTAINING AN IDENTIFYING LABEL FOR THE SITE;
C            B. ONE RECORD CONTAINING THE RECORD LENGTH AT THE SITE;
C            C. THE DATA VALUES, IN FREE FORMAT.
C
      READ(INFILE,*)NSITE
      DO 10 ISITE=1,NSITE
      READ(INFILE,'(A32)')SITEID
      READ(INFILE,*)N
```

```
      READ(INFILE,*)(X(I),I=1,N)
      WEIGHT(ISITE)=N
      CALL SORT(X,N)
      CALL SAMLMR(X,N,XMOM(1,ISITE),NPAR,A,B)
      WRITE(6,6020)ISITE,SITEID,N,(XMOM(I,ISITE),I=1,NPAR)
   10 CONTINUE
C
C         CALCULATE REGIONAL AVERAGE L-MOMENTS
C
      CALL REGLMR(NSITE,NPAR,5,XMOM,WEIGHT,RMOM)
      WRITE(6,6030)(RMOM(I),I=1,NPAR)
C
C         FIT REGIONAL FREQUENCY DISTRIBUTION
C
      CALL PELWAK(RMOM,PARA,IFAIL)
      IF(IFAIL.NE.0)WRITE(6,6040)IFAIL
      WRITE(6,6050)(PARA(I),I=1,NPAR)
C
C         CALCULATE QUANTILES OF REGIONAL FREQUENCY DISTRIBUTION
C
      WRITE(6,6060)(PROB(IQ),IQ=1,NPROB)
      DO 20 IQ=1,NPROB
   20 RQUANT(IQ)=QUAWAK(PROB(IQ),PARA)
      WRITE(6,6070)(RQUANT(IQ),IQ=1,NPROB)
C
C         CALCULATE QUANTILE ESTIMATES FOR EACH SITE
C
      DO 40 ISITE=1,NSITE
      DO 30 IQ=1,NPROB
   30 QUANT(IQ)=XMOM(1,ISITE)*RQUANT(IQ)
      WRITE(6,6080)ISITE,(QUANT(IQ),IQ=1,NPROB)
   40 CONTINUE
C
      STOP
C
 6000 FORMAT(' REGIONAL ANALYSIS, UNBIASED L-MOMENTS'/)
 6010 FORMAT(' REGIONAL ANALYSIS,',
     *  ' L-MOMENT PLOTTING POSITION PARAMETERS ',2F8.4/)
 6020 FORMAT(' SITE',I3,1X,A32,'N=',I3,'   L-MOMENT RATIOS', F9.2,4F9.4)
 6030 FORMAT(//' REGIONAL AVERAGE L-MOMENT RATIOS',5F9.4)
 6040 FORMAT(/' PARAMETER ESTIMATION: FAIL FLAG',I2)
 6050 FORMAT(/' REGIONAL WAKEBY PARAMETERS',5F12.4)
 6060 FORMAT(///'  SITE',25X,'QUANTILES'/' NUMBER',10F10.4/1X,106('-'))
 6070 FORMAT(' REGION',10F10.2/)
 6080 FORMAT(1X,I4,2X,10F10.2)
      END
```

```
      PROGRAM XTEST
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  EXAMPLE PROGRAM TO ILLUSTRATE THE USE OF ROUTINE REGTST
C
C  THE ROUTINE READS IN THE SAMPLE L-MOMENTS FOR DIFFERENT SITES IN A
C  REGION AND CALLS ROUTINE REGTST TO COMPUTE DISCORDANCY, HETEROGENEITY
C  AND GOODNESS-OF-FIT STATISTICS.
C
C  PARAMETERS OF PROGRAM:
C  MAXNS  - MAXIMUM NUMBER OF SITES
C  SSEED  - SEED FOR RANDOM-NUMBER GENERATOR
C  NSIM   - NSIM PARAMETER OF ROUTINE REGTST
C  KPRINT - OUTPUT FLAG, KPRINT PARAMETER OF ROUTINE REGTST
C  INFILE - STREAM NUMBER TO WHICH INPUT FILE IS ATTACHED
C  KOUT   - STREAM NUMBER TO WHICH OUTPUT FILE IS ATTACHED
C  NPROB  - NUMBER OF QUANTILES TO BE FOUND
C  A      - ) PARAMETERS OF
C  B      - ) PLOTTING POSITION
C
C  ARRAYS TO BE INITIALIZED IN DATA STATEMENTS:
C  PROB   - PROBABILITIES FOR WHICH QUANTILES ARE TO BE FOUND
C
C  ROUTINES USED: REGTST AND ROUTINES CALLED BY REGTST
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (SSEED=619145091D0,NSIM=500,MAXNS=200)
      PARAMETER (KPRINT=1,INFILE=7,KOUT=6,NPROB=10)
C
      CHARACTER NAMES*12(MAXNS),REGNAM*60
      DOUBLE PRECISION D(MAXNS),H(3),PARA(5,6),PROB(NPROB),RMOM(5),
     *  VBAR(3),VOBS(3),VSD(3),XMOM(5,MAXNS),Z(5)
      INTEGER LEN(MAXNS)
C
      DATA A/0D0/,B/0D0/
      DATA PROB/0.01D0,0.02D0,0.05D0,0.1D0,0.2D0,
     *  0.5D0,0.90D0,0.95D0,0.99D0,0.999D0/
C
C         READ IN THE AT-SITE L-MOMENTS.
C         DATA FILE MAY CONTAIN ANY NUMBER OF REGIONAL DATA STRUCTURES.
C         A 'REGIONAL DATA STRUCTURE' CONSISTS OF:
C         1. ONE RECORD CONTAINING:
C            (COLUMNS  1- 4) NUMBER OF SITES IN REGION;
C            (COLUMNS  5-56) IDENTIFYING LABEL FOR THE REGION.
C         2. FOR EACH SITE, ONE RECORD CONTAINING:
C            (COLUMNS  1-12) AN IDENTIFYING LABEL FOR THE SITE;
C            (COLUMNS 13-16) THE RECORD LENGTH AT THE SITE;
C            (COLUMNS 17-56) SAMPLE L-MOMENTS L-1, L-CV, T-3, T-4, T-5.
C
    1 READ(INFILE,5000,END=900)NSITES,REGNAM
      WRITE(KOUT,6000)REGNAM,NSITES
      DO 10 ISITE=1,NSITES
      READ(INFILE,5010)NAMES(ISITE),LEN(ISITE),(XMOM(I,ISITE),I=1,5)
   10 CONTINUE
C
C         CALCULATE TEST STATISTICS
C
      SEED=SSEED
```

```
      CALL REGTST(NSITES,NAMES,LEN,XMOM,A,B,SEED,NSIM,NPROB,PROB,
     *  KPRINT,KOUT,RMOM,D,VOBS,VBAR,VSD,H,Z,PARA)
      GOTO 1
C
  900 CONTINUE
      WRITE(KOUT,6010)
      STOP
C
 5000 FORMAT(I4,A52)
 5010 FORMAT(A12,I4,5F8.0)
 6000 FORMAT(///1X,A52,I8,' SITES')
 6010 FORMAT(///' ALL DATA PROCESSED')
      END
```

```
      PROGRAM XSIM
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  Regional frequency analysis - world specified by its L-moments.
C  Includes calculation of heterogeneity measures.
C
C  Parameters of program:
C  SEED   - Seed for random-number generator.
C  NREP   - Number of simulated regions.
C  NSIM   - Number of simulations used by routine REGTST.  Set it to
C           zero if simulation of heterogeneity and goodness-of-fit
C           measures is not required.  N.B. Total number of simulated
C           regions is NREP*(NSIM+1): large values of NREP and NSIM will
C           need a lot of computing time!
C  NSITE  - Number of sites in region.
C  NMAX   - Maximum record length at any site.
C  RMED   - Correlation between each pair of sites.  At least 0, less
C           than 1.
C  NQ     - Number of quantiles to be estimated (their values are held
C           in array FVAL).
C  NQQ    - Number of empirical quantiles of the distribution of
C           quantile estimates to be found (their values are held in
C           array QUANT).
C  NGROUP - Number of groups in the histogram used to accumulate
C           empirical distribution of quantile estimates.  Include two
C           extra groups for points falling outside the range of the
C           histogram.
C  START  - Lower endpoint for the above histogram.
C  GRINT  - Group interval for the above histogram.
C  KPRINT - Flag for printing simulation results:
C            0 - no printing;
C            1 - print only regional averages;
C            2 - print results for all sites.
C  KOUT   - I/O stream number for printed output.  N.B. If NQ is larger
C           than 16, the output width will be greater than 133 columns;
C           the output stream must be able to accept these long records.
C
C  Arrays whose elements are set in DATA statements:
C  NREC   - Integer array of length NSITE.  Record lengths at each site.
C           Record lengths must be in descending order, and none may
C           exceed NMAX.
C  CV     - Array of length NSITE.  L-CVs at each site.
C  SKEW   - Array of length NSITE.  L-skewnesses at each site.
C  FVAL   - Array of length NQ.  Quantiles to be estimated.
C  QUANT  - Array of length NQQ.  Empirical quantiles to be estimated.
C           They should be in ascending order.
C
C  Arrays used to store the simulation results:
C  BIAS   - Array of dimension (NQ,NS,2).  BIAS(I,J,1) contains the
C           relative bias of the Ith quantile, and BIAS(I,J,2) the
C           relative bias of the Ith 'growth curve component'
C           (quantile divided by the mean), for site J.
C  RMSE   - Array of dimension (NQ,NS,2).  RMSE(I,J,1) contains the
C           relative RMSE of the Ith quantile, and RMSE(I,J,2) the
C           relative RMSE of the Ith growth curve component, for site J.
C  QEMP   - Array of dimension (NQQ,NQ,NS,2).  QEMP(L,I,J,1) contains
C           the Lth empirical quantile of the distribution of
C           (estimated quantile)/(true quantile) for the Ith quantile
```

```
C           at site J; QEMP(L,I,J,2) contains the corresponding quantity
C           for (estimated growth curve component)/(true growth curve
C           component).
C             The empirical quantile may lie outside the range of the
C           histogram used to accumulate the values -- this range being
C           from START to START+GRINT*(NGROUP-2).  In this case the
C           corresponding element of QEMP is unchanged from its initial
C           value, BIG as defined in a DATA statement below.
C              In arrays BIAS, RMSE, and QEMP, the elements for site NS
C           (where NS=NSITE+1) contain averages over all sites of the
C           corresponding array elements.  E.g. BIAS(I,NS,1) is the
C           average of BIAS(I,1,1), BIAS(I,2,1), ..., BIAS(1,NSITE,1).
C  HAVE   - Array of dimension 3.  Contains the average, over all
C           simulated regions, of the three heterogeneity statistics
C           calculated by routine REGTST.
C  ACCEPT - Array of dimension 4.  Contains the proportion of
C           simulations in which each of four distributions
C           (GLO = generalized logistic, GEV = generalized
C           extreme-value, LN3 = lognormal, PE3 = Pearson type III)
C           gave an acceptable fit to the simulated data, according to
C           the goodness-of-fit statistic calculated by routine REGTST.
C  CHOOSE - Array of dimension 4.  Contains the proportion of
C           simulations in which each of the above four distributions
C           gave the best fit to the simulated data.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      PARAMETER (SEED=619145091D0,NREP=10000,NSIM=0)
      PARAMETER (NSITE=19,NMAX=100,RMED=0.64D0)
      PARAMETER (NQ=6,NQQ=2,NGROUP=302,START=0.5D0,GRINT=0.005D0)
      PARAMETER (KPRINT=2,KOUT=6)
C
      PARAMETER (NS=NSITE+1)
      CHARACTER*10 HAAB,HBIAS,HRMSE
      CHARACTER*12 NAMES(NSITE)
      INTEGER IHIST(NGROUP,NQ,NS,2),NREC(NSITE)
      DOUBLE PRECISION
     *  AAB(NQ,2),ACCEPT(4),BIAS(NQ,NS,2),CHOOSE(4),COR(NSITE,NSITE),
     *  CV(NSITE),DD(NSITE),FVAL(NQ),H(3),HAVE(3),PARA(5,NSITE),
     *  PPARA(5,6),PROB(1),QEMP(NQQ,NQ,NS,2),QUANT(NQQ),RMOM(5),
     *  RMSE(NQ,NS,2),RPARA(5),RRMOM(5),SKEW(NSITE),SMEAN(NSITE),
     *  SMOM(5,NSITE),TRUEAV(NQ),TRUEGC(NQ),TRUEP(3,NSITE),
     *  TRUEQ(NQ,NSITE),VBAR(3),VOBS(3),VSD(3),X(NMAX,NSITE),XX(NSITE),
     *  YY(NSITE),ZZZ(5)
      DATA HAAB/'  ABS.BIAS'/,HBIAS/'      BIAS'/,HRMSE/'      RMSE'/
      DATA ZERO/0D0/,ONE/1D0/
C
C         ZCRIT - Critical value for Z statistic
C         SMALL - A small number, used to avoid having to evaluate a
C                 quantile function at a value too close to 0 or 1.
C         BIG   - A large number, used to initialize array QEMP and
C                 variable ZMIN.
C
      DATA ZCRIT/1.645D0/,SMALL/1D-10/,BIG/1D10/
C
      DATA FVAL/0.01D0,0.1D0,0.5D0,0.9D0,0.99D0,0.999D0/
      DATA QUANT/0.05D0,0.95D0/
      DATA (NREC(ISITE),CV(ISITE),SKEW(ISITE),ISITE=1,NSITE)/
     *  98,  0.0978D0, 0.0279D0,
     *  59,  0.0992D0, 0.0279D0,
     *  90,  0.1006D0, 0.0279D0,
     *  61,  0.1020D0, 0.0279D0,
     *  65,  0.1034D0, 0.0279D0,
     *  86,  0.1047D0, 0.0279D0,
     *  78,  0.1061D0, 0.0279D0,
     *  72,  0.1075D0, 0.0279D0,
     *  67,  0.1089D0, 0.0279D0,
     *  99,  0.1103D0, 0.0279D0,
     *  49,  0.1117D0, 0.0279D0,
     *  61,  0.1131D0, 0.0279D0,
     *  69,  0.1145D0, 0.0279D0,
```

```
     *  73,  0.1159D0, 0.0279D0,
     *  70,  0.1172D0, 0.0279D0,
     *  66,  0.1186D0, 0.0279D0,
     *  59,  0.1200D0, 0.0279D0,
     *  74,  0.1214D0, 0.0279D0,
     *  82,  0.1228D0, 0.0279D0/
C
C         CALCULATE POPULATION PARAMETERS FOR EACH SITE
C
      NNMAX=0
      DO 10 ISITE=1,NSITE
      RMOM(1)=ONE
      RMOM(2)=CV(ISITE)
      RMOM(3)=SKEW(ISITE)
      CALL PELGNO(RMOM,TRUEP(1,ISITE))
      IF(NREC(ISITE).GT.NNMAX)NNMAX=NREC(ISITE)
   10 CONTINUE
      IF(NNMAX.GT.NMAX)GOTO 1000
C
C         CALCULATE POPULATION QUANTILES FOR EACH SITE
C
      DO 30 IQ=1,NQ
      SUMAV=ZERO
      SUMGC=ZERO
      DO 20 ISITE=1,NSITE
      Q=QUAGNO(FVAL(IQ),TRUEP(1,ISITE))
      TRUEQ(IQ,ISITE)=Q
      SUMAV=SUMAV+Q
      SUMGC=SUMGC+ONE/Q
   20 CONTINUE
      TRUEAV(IQ)=SUMAV/NSITE
      TRUEGC(IQ)=NSITE/SUMGC
   30 CONTINUE
C
C         PRINT DESCRIPTION OF WORLD
C
      IF(KPRINT.EQ.0)GOTO 80
      IF(NQ.GE.13)GOTO 50
      WRITE(KOUT,6000)NREP,SEED,(FVAL(IQ),IQ=1,NQ)
      DO 40 ISITE=1,NSITE
      WRITE(KOUT,6010)ISITE,(TRUEP(I,ISITE),I=1,3),
     *  CV(ISITE),SKEW(ISITE),NREC(ISITE),(TRUEQ(IQ,ISITE),IQ=1,NQ)
   40 CONTINUE
      WRITE(KOUT,6020)(TRUEAV(IQ),IQ=1,NQ)
      WRITE(KOUT,6030)(TRUEGC(IQ),IQ=1,NQ)
      GOTO 80
   50 CONTINUE
      WRITE(KOUT,6000)NREP,SEED
      DO 60 ISITE=1,NSITE
      WRITE(KOUT,6010)ISITE,(TRUEP(I,ISITE),I=1,3),
     *  CV(ISITE),SKEW(ISITE),NREC(ISITE)
   60 CONTINUE
      WRITE(KOUT,6040)(FVAL(IQ),IQ=1,NQ)
      DO 70 ISITE=1,NSITE
      WRITE(KOUT,6050)ISITE,(TRUEQ(IQ,ISITE),IQ=1,NQ)
   70 CONTINUE
      WRITE(KOUT,6060)(TRUEAV(IQ),IQ=1,NQ)
      WRITE(KOUT,6070)(TRUEGC(IQ),IQ=1,NQ)
   80 CONTINUE
C
C         INITIALIZE CORRELATION MATRIX
C
      IF(KPRINT.GT.0)WRITE(KOUT,6080)RMED
      IF(RMED.EQ.ZERO.OR.NSITE.EQ.1)GOTO 140
      COR(1,1)=ONE
      DO 100 I=2,NSITE
      DO 90 J=1,I-1
   90 COR(I,J)=RMED
  100 COR(I,I)=ONE
C
C         CHOLESKY DECOMPOSITION OF CORRELATION MATRIX
```

```
C         FIRST COLUMN IS ALREADY CORRECT, BECAUSE COR(1,1)=1,
C         SO TRANSFORM ONLY THE SECOND AND SUBSEQUENT COLUMNS
C
      DO 130 I=2,NSITE
C
C         - DIAGONAL ELEMENTS
C
      SUM=ZERO
      DO 110 K=1,I-1
  110 SUM=SUM+COR(I,K)**2
      SUM=COR(I,I)-SUM
      COR(I,I)=DSQRT(SUM)
      IF(I.EQ.NSITE)GOTO 140
C
C         - OFF-DIAGONAL ELEMENTS
C
      DO 130 J=I+1,NSITE
      SUM=ZERO
      DO 120 K=1,I-1
  120 SUM=SUM+COR(I,K)*COR(J,K)
      COR(J,I)=(COR(J,I)-SUM)/COR(I,I)
  130 CONTINUE
C
  140 CONTINUE
C
C         INITIALIZE ARRAYS
C
      DO 150 IH=1,3
  150 HAVE(IH)=ZERO
      DO 160 IZ=1,4
      ACCEPT(IZ)=ZERO
  160 CHOOSE(IZ)=ZERO
      DO 190 IQ=1,NQ
      DO 190 IS=1,NS
      BIAS(IQ,IS,1)=ZERO
      BIAS(IQ,IS,2)=ZERO
      RMSE(IQ,IS,1)=ZERO
      RMSE(IQ,IS,2)=ZERO
      DO 170 IG=1,NGROUP
      IHIST(IG,IQ,IS,1)=0
  170 IHIST(IG,IQ,IS,2)=0
      DO 180 IQQ=1,NQQ
      QEMP(IQQ,IQ,IS,1)=BIG
  180 QEMP(IQQ,IQ,IS,2)=BIG
  190 CONTINUE
      NFAIL=0
      NWARN=0
      SSEED=SEED
C
C         START OF SIMULATION LOOP
C
      DO 400 IREP=1,NREP
C
      XSEED=SSEED
C
C         GENERATE INDEPENDENT UNIFORM RANDOM VARIATES
C
      DO 200 ISITE=1,NSITE
      CALL DURAND(SSEED,NNMAX,X(1,ISITE))
  200 CONTINUE
C
C         FOR CORRELATED DATA: TRANSFORM TO NORMAL, FORM CORRELATED
C         NORMAL VARIATES, TRANSFORM BACK TO UNIFORM
C
      IF(RMED.EQ.ZERO.OR.NSITE.EQ.1)GOTO 260
      DO 250 IX=1,NNMAX
      DO 210 ISITE=1,NSITE
      X(IX,ISITE)=QUASTN(X(IX,ISITE))
  210 CONTINUE
      DO 240 I=NSITE,1,-1
      SUM=COR(I,I)*X(IX,I)
```

```
      IF(I.EQ.1)GOTO 230
      DO 220 J=1,I-1
  220 SUM=SUM+COR(I,J)*X(IX,J)
  230 CONTINUE
      TEMP=CDFSTN(SUM)
      IF(TEMP.EQ.ONE)TEMP=ONE-SMALL
      IF(TEMP.EQ.ZERO)TEMP=SMALL
      X(IX,I)=TEMP
  240 CONTINUE
  250 CONTINUE
  260 CONTINUE
C
C         TRANSFORM TO CORRECT MARGINAL DISTRIBUTION,
C         DIVIDE BY SAMPLE MEAN AND ORDER THE DATA
C
      DO 290 ISITE=1,NSITE
      N=NREC(ISITE)
      SUM=ZERO
      DO 270 I=1,N
      X(I,ISITE)=QUAGNO(X(I,ISITE),TRUEP(1,ISITE))
  270 SUM=SUM+X(I,ISITE)
      SMEAN(ISITE)=SUM/N
      DO 280 I=1,N
  280 X(I,ISITE)=X(I,ISITE)/SMEAN(ISITE)
      CALL SORT(X(1,ISITE),N)
  290 CONTINUE
C
C         CALCULATE L-MOMENT RATIOS
C
      DO 300 I=1,5
  300 RMOM(I)=ZERO
      NSY=0
      DO 310 ISITE=1,NSITE
      N=NREC(ISITE)
      NSY=NSY+N
      CALL SAMLMR(X(1,ISITE),N,SMOM(1,ISITE),5,ZERO,ZERO)
      DO 310 I=1,5
  310 RMOM(I)=RMOM(I)+N*SMOM(I,ISITE)
      DO 320 I=1,5
  320 RMOM(I)=RMOM(I)/NSY
C
C         CALCULATE HETEROGENEITY STATISTICS
C
      IF(NSIM.EQ.0)GOTO 350
      RSEED=XSEED
      NPROB=0
      JPRINT=0
      CALL REGTST(NSITE,NAMES,NREC,SMOM,ZERO,ZERO,RSEED,NSIM,NPROB,
     *   PROB,JPRINT,KOUT,RRMOM,DD,VOBS,VBAR,VSD,H,ZZZ,PPARA)
      DO 330 IH=1,3
  330 HAVE(IH)=HAVE(IH)+H(IH)
      ZMIN=BIG
      DO 340 IZ=1,4
      ZABS=DABS(ZZZ(IZ))
      IF(ZABS.LE.ZCRIT)ACCEPT(IZ)=ACCEPT(IZ)+ONE
      IF(ZABS.GE.ZMIN)GOTO 340
      ICHOOS=IZ
      ZMIN=ZABS
  340 CONTINUE
      CHOOSE(ICHOOS)=CHOOSE(ICHOOS)+1
  350 CONTINUE
C
C         ESTIMATE PARAMETERS
C
      IFAIL=0
      CALL PELGNO(RMOM,RPARA)
      IF(RPARA(2).LT.ZERO)IFAIL=2
      IF(IFAIL.EQ.0)GOTO 360
      NFAIL=NFAIL+1
      GOTO 390
  360 CONTINUE
```

```
C
C         COMPUTE QUANTILES ...
C
      DO 380 IQ=1,NQ
      GC=QUAGNO(FVAL(IQ),RPARA)
C
C         ... AND ACCUMULATE HISTOGRAMS, OF EMPIRICAL DISTRIBUTIONS
C             OF QUANTILE ESTIMATES ...
C
      QSUM=ZERO
      DO 370 ISITE=1,NSITE
      Q=GC*SMEAN(ISITE)/TRUEQ(IQ,ISITE)
      QSUM=QSUM+Q
      BIAS(IQ,ISITE,1)=BIAS(IQ,ISITE,1)+(Q-ONE)
      RMSE(IQ,ISITE,1)=RMSE(IQ,ISITE,1)+(Q-ONE)**2
      IG=(Q-START)/GRINT+2
      IF(IG.LT.1)IG=1
      IF(IG.GT.NGROUP)IG=NGROUP
      IHIST(IG,IQ,ISITE,1)=IHIST(IG,IQ,ISITE,1)+1
      Q=GC/TRUEQ(IQ,ISITE)
      BIAS(IQ,ISITE,2)=BIAS(IQ,ISITE,2)+(Q-ONE)
      RMSE(IQ,ISITE,2)=RMSE(IQ,ISITE,2)+(Q-ONE)**2
      IG=(Q-START)/GRINT+2
      IF(IG.LT.1)IG=1
      IF(IG.GT.NGROUP)IG=NGROUP
      IHIST(IG,IQ,ISITE,2)=IHIST(IG,IQ,ISITE,2)+1
  370 CONTINUE
C
C         ... OF THE 'AVERAGE FOR ALL SITES'
C
      Q=QSUM/NSITE
      IG=(Q-START)/GRINT+2
      IF(IG.LT.1)IG=1
      IF(IG.GT.NGROUP)IG=NGROUP
      IHIST(IG,IQ,NS,1)=IHIST(IG,IQ,NS,1)+1
C
C         ... AND OF THE 'GROWTH CURVE COMPONENT'
C
      Q=GC/TRUEGC(IQ)
      IG=(Q-START)/GRINT+2
      IF(IG.LT.1)IG=1
      IF(IG.GT.NGROUP)IG=NGROUP
      IHIST(IG,IQ,NS,2)=IHIST(IG,IQ,NS,2)+1
  380 CONTINUE
  390 CONTINUE
C
C         END OF SIMULATION LOOP
C
  400 CONTINUE
C
C         CALCULATE FINAL RESULTS FOR HET AND GOF STATISTICS
C
      IF(NSIM.EQ.0)GOTO 430
      DO 410 IH=1,3
  410 HAVE(IH)=HAVE(IH)/NREP
      IF(KPRINT.GT.0)WRITE(KOUT,6090)HAVE,NSIM
      DO 420 IZ=1,4
      ACCEPT(IZ)=ACCEPT(IZ)/NREP
  420 CHOOSE(IZ)=CHOOSE(IZ)/NREP
      IF(KPRINT.GT.0)WRITE(KOUT,6100)(ACCEPT(IZ),IZ=1,4)
      IF(KPRINT.GT.0)WRITE(KOUT,6110)(CHOOSE(IZ),IZ=1,4)
  430 CONTINUE
C
C         CALCULATE FINAL RESULTS FOR ESTIMATION
C
      IF(NREP.LE.1)RETURN
      NOK=NREP-NFAIL
      DO 510 IQ=1,NQ
      DO 510 ITYPE=1,2
C
C         -  BIAS AND RMSE, SITE BY SITE ...
```

```
C
      DO 440 ISITE=1,NSITE
      BIAS(IQ,ISITE,ITYPE)=BIAS(IQ,ISITE,ITYPE)/NOK
      RMSE(IQ,ISITE,ITYPE)=DSQRT(RMSE(IQ,ISITE,ITYPE)/NOK)
  440 CONTINUE
C
C         - ... AND AVERAGED OVER ALL SITES
C
      BIAS(IQ,NS,ITYPE)=ZERO
      RMSE(IQ,NS,ITYPE)=ZERO
      AAB(IQ,ITYPE)=ZERO
      DO 450 ISITE=1,NSITE
      BIAS(IQ,NS,ITYPE)=BIAS(IQ,NS,ITYPE)+BIAS(IQ,ISITE,ITYPE)
      RMSE(IQ,NS,ITYPE)=RMSE(IQ,NS,ITYPE)+RMSE(IQ,ISITE,ITYPE)
      AAB(IQ,ITYPE)=AAB(IQ,ITYPE)+DABS(BIAS(IQ,ISITE,ITYPE))
  450 CONTINUE
      BIAS(IQ,NS,ITYPE)=BIAS(IQ,NS,ITYPE)/NSITE
      RMSE(IQ,NS,ITYPE)=RMSE(IQ,NS,ITYPE)/NSITE
      AAB(IQ,ITYPE)=AAB(IQ,ITYPE)/NSITE
C
C        - QUANTILES OF EMPIRICAL DISTRIBUTIONS OF ESTIMATORS
C
      DO 500 ISITE=1,NS
      IG=0
      SUM=ZERO
      DO 480 IQQ=1,NQQ
      CRIT=QUANT(IQQ)*NOK
      DO 460 IGG=1,NGROUP
      IF(SUM.GE.CRIT)GOTO 470
      IG=IG+1
      HH=IHIST(IG,IQ,ISITE,ITYPE)
      SUM=SUM+HH
  460 CONTINUE
  470 IF(IG.LE.1)GOTO 480
      IF(IG.GE.NGROUP)GOTO 490
      QEMP(IQQ,IQ,ISITE,ITYPE)=START+GRINT*(IG-ONE-(SUM-CRIT)/HH)
  480 CONTINUE
  490 CONTINUE
  500 CONTINUE
C
  510 CONTINUE
C
C         PRINT RESULTS
C
      IF(KPRINT.LE.0)GOTO 570
C
C         - HEADER LINES
C
      WRITE(KOUT,6120)NFAIL,NWARN
      WRITE(KOUT,6130)(FVAL(IQ),IQ=1,NQ)
C
C         - SITE BY SITE
C
      IF(KPRINT.EQ.1)GOTO 540
      DO 530 ISITE=1,NSITE
      WRITE(KOUT,6140)ISITE
      WRITE(KOUT,6170)HBIAS,(BIAS(IQ,ISITE,1),IQ=1,NQ)
      WRITE(KOUT,6170)HRMSE,(RMSE(IQ,ISITE,1),IQ=1,NQ)
      DO 520 IQQ=1,NQQ
      WRITE(KOUT,6180)QUANT(IQQ),(QEMP(IQQ,IQ,ISITE,1),IQ=1,NQ)
  520 CONTINUE
  530 CONTINUE
  540 CONTINUE
C
C         - 'AVERAGE FOR ALL SITES' AND 'GROWTH CURVE COMPONENT'
C
      WRITE(KOUT,6150)
      WRITE(KOUT,6170)HAAB,(AAB(IQ,1),IQ=1,NQ)
      WRITE(KOUT,6170)HBIAS,(BIAS(IQ,NS,1),IQ=1,NQ)
      WRITE(KOUT,6170)HRMSE,(RMSE(IQ,NS,1),IQ=1,NQ)
      DO 550 IQQ=1,NQQ
```

```
  550 WRITE(KOUT,6180)QUANT(IQQ),(QEMP(IQQ,IQ,NS,1),IQ=1,NQ)
      WRITE(KOUT,6160)
      WRITE(KOUT,6170)HAAB,(AAB(IQ,2),IQ=1,NQ)
      WRITE(KOUT,6170)HBIAS,(BIAS(IQ,NS,2),IQ=1,NQ)
      WRITE(KOUT,6170)HRMSE,(RMSE(IQ,NS,2),IQ=1,NQ)
      DO 560 IQQ=1,NQQ
  560 WRITE(KOUT,6180)QUANT(IQQ),(QEMP(IQQ,IQ,NS,2),IQ=1,NQ)
  570 CONTINUE
C----------------------------------------------------------------------
C  NOTE -- The section of the program between the dashed lines is
C  included for interest only, and is not an integral part of the
C  simulation.  It shows how Table 6.2 of Hosking and Wallis (1997) is
C  derived from the simulation results for the 'growth curve component'.
C
C         SET REGIONAL L-MOMENTS FOR 'NORTH CASCADES' REGION
C         AND ESTIMATE PARAMETERS OF REGIONAL LOGNORMAL DISTRIBUTION
C
      RMOM(1)=ONE
      RMOM(2)=0.1103D0
      RMOM(3)=0.0279D0
      CALL PELGNO(RMOM,RPARA)
C
C         COMPUTE AND PRINT THE ENTRIES FOR THE TABLE
C
      WRITE(KOUT,9010)
      DO 900 IQ=1,NQ
      Q=QUAGNO(FVAL(IQ),RPARA)
      RMS=RMSE(IQ,NS,2)
      ERRLOW=Q/QEMP(2,IQ,NS,2)
      ERRHI =Q/QEMP(1,IQ,NS,2)
      WRITE(KOUT,9020)FVAL(IQ),Q,RMS,ERRLOW,ERRHI
  900 CONTINUE
 9010 FORMAT(//' Hosking and Wallis (1997), Table 6.2'//
     *  '         F      qhat(F)     RMSE      Error bounds')
 9020 FORMAT(1X,5F10.3)
C----------------------------------------------------------------------
C
C         THE END
C
      RETURN
C
 1000 WRITE(6,7000)
      RETURN
C
 6000 FORMAT(' REGIONAL LOGNORMAL SIMULATIONS    NREP=',I6,
     *  '  SEED=',F12.0////' SITE      XI   ALPHA       K',
     *  '    L-CV  L-SKEW   N',:,'    QUANTILES .....'/49X,12F8.4)
 6010 FORMAT(1X,I3,5F8.4,I4,12F8.3)
 6020 FORMAT(9X,'AVERAGE FOR ALL SITES (ARITHMETIC MEAN)',12F8.3)
 6030 FORMAT(9X,'REGIONAL GROWTH CURVE (HARMONIC MEAN)  ',12F8.3)
 6040 FORMAT(/' SITE   QUANTILES .....'/4X,16F8.4)
 6050 FORMAT(1X,I3,16F8.3)
 6060 FORMAT(' AVERAGE FOR ALL SITES (ARITHMETIC MEAN)'/4X,16F8.3)
 6070 FORMAT(' REGIONAL GROWTH CURVE (HARMONIC MEAN)  '/4X,16F8.3)
 6080 FORMAT(//' INTER-SITE CORRELATION=',F6.2)
 6090 FORMAT(//' AVERAGE HETEROGENEITY MEASURES',3F8.2,
     *  '   (BASED ON',I6,' SIMULATIONS)')
 6100 FORMAT(' DISTRIBUTIONS ACCEPTED:',
     *  ' GLO',F7.4,', GEV',F7.4,', LN3',F7.4,', PE3',F7.4)
 6110 FORMAT(' DISTRIBUTIONS CHOSEN  :',
     *  ' GLO',F7.4,', GEV',F7.4,', LN3',F7.4,', PE3',F7.4)
 6120 FORMAT(//' L-MOMENT ESTIMATION OF LOGNORMAL DISTRIBUTION ',10X,
     *  'FAILURES:',I6,8X,'WARNINGS:',I6)
 6130 FORMAT(/9X,'F',5X,16F7.4)
 6140 FORMAT(/22X,'SITE',I3,' QUANTILES')
 6150 FORMAT(/20X,'AVERAGE FOR ALL SITES')
 6160 FORMAT(/20X,'GROWTH CURVE COMPONENT')
 6170 FORMAT(A10,5X,16F7.3)
 6180 FORMAT(1X,F5.3,' PT.',5X,16F7.3)
 7000 FORMAT('*** ERROR *** PROGRAM XSIM   : NMAX PARAMETER TOO SMALL')
      END
```

```
      DOUBLE PRECISION FUNCTION DERF(X)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  ERROR FUNCTION
C
C  BASED ON ALGORITHM 5666, J.F.HART ET AL. (1968) 'COMPUTER
C  APPROXIMATIONS'
C
C  ACCURATE TO 15 DECIMAL PLACES
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DATA ZERO/0D0/,ONE/1D0/,TWO/2D0/,THREE/3D0/,FOUR/4D0/,P65/0.65D0/
C
C         COEFFICIENTS OF RATIONAL-FUNCTION APPROXIMATION
C
      DATA P0,P1,P2,P3,P4,P5,P6/
     *  0.22020 68679 12376 1D3,    0.22121 35961 69931 1D3,
     *  0.11207 92914 97870 9D3,    0.33912 86607 83830 0D2,
     *  0.63739 62203 53165 0D1,    0.70038 30644 43688 1D0,
     *  0.35262 49659 98910 9D-1/
      DATA Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7/
     *  0.44041 37358 24752 2D3,   0.79382 65125 19948 4D3,
     *  0.63733 36333 78831 1D3,   0.29656 42487 79673 7D3,
     *  0.86780 73220 29460 8D2,   0.16064 17757 92069 5D2,
     *  0.17556 67163 18264 2D1,   0.88388 34764 83184 4D-1/
C
C         C1 IS SQRT(2), C2 IS SQRT(2/PI)
C         BIG IS THE POINT AT WHICH DERF=1 TO MACHINE PRECISION
C
      DATA C1/1.4142 13562 37309 5D0/
      DATA C2/7.9788 45608 02865 4D-1/
      DATA BIG/6.25D0/,CRIT/5D0/
C
      DERF=ZERO
      IF(X.EQ.ZERO)RETURN
      XX=DABS(X)
      IF(XX.GT.BIG)GOTO 20
      EXPNTL=DEXP(-X*X)
      ZZ=DABS(X*C1)
      IF(XX.GT.CRIT)GOTO 10
      DERF=EXPNTL*((((((P6*ZZ+P5)*ZZ+P4)*ZZ+P3)*ZZ+P2)*ZZ+P1)*ZZ+P0)/
     *  (((((((Q7*ZZ+Q6)*ZZ+Q5)*ZZ+Q4)*ZZ+Q3)*ZZ+Q2)*ZZ+Q1)*ZZ+Q0)
      IF(X.GT.ZERO)DERF=ONE-TWO*DERF
      IF(X.LT.ZERO)DERF=TWO*DERF-ONE
      RETURN
C
   10 DERF=EXPNTL*C2/(ZZ+ONE/(ZZ+TWO/(ZZ+THREE/(ZZ+FOUR/(ZZ+P65)))))
      IF(X.GT.ZERO)DERF=ONE-DERF
      IF(X.LT.ZERO)DERF=DERF-ONE
      RETURN
C
   20 DERF=ONE
      IF(X.LT.ZERO)DERF=-ONE
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION DIGAMD(X)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  DIGAMMA FUNCTION (EULER'S PSI FUNCTION) - THE FIRST DERIVATIVE OF
C  LOG(GAMMA(X))
C
C  BASED ON ALGORITHM AS103, APPL. STATIST. (1976) VOL.25 NO.3
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/
      DATA SMALL/1D-9/,CRIT/13D0/
C
C         C1...C7 ARE THE COEFFTS OF THE ASYMPTOTIC EXPANSION OF DIGAMD
C         D1 IS  -(EULER'S CONSTANT)
C
      DATA C1,C2,C3,C4,C5,C6,C7,D1/
     *  0.83333 33333 33333 333D-1,  -0.83333 33333 33333 333D-2,
     *  0.39682 53968 25396 825D-2,  -0.41666 66666 66666 666D-2,
     *  0.75757 57575 75757 575D-2,  -0.21092 79609 27960 928D-1,
     *  0.83333 33333 33333 333D-1,  -0.57721 56649 01532 861D 0/
      DIGAMD=ZERO
      IF(X.LE.ZERO)GOTO 1000
C
C         USE SMALL-X APPROXIMATION IF X.LE.SMALL
C
      IF(X.GT.SMALL)GOTO 10
      DIGAMD=D1-ONE/X
      RETURN
C
C         REDUCE TO DIGAMD(X+N) WHERE X+N.GE.CRIT
C
   10 Y=X
   20 IF(Y.GE.CRIT)GOTO 30
      DIGAMD=DIGAMD-ONE/Y
      Y=Y+ONE
      GOTO 20
C
C         USE ASYMPTOTIC EXPANSION IF Y.GE.CRIT
C
   30 DIGAMD=DIGAMD+DLOG(Y)-HALF/Y
      Y=ONE/(Y*Y)
      SUM=((((((C7*Y+C6)*Y+C5)*Y+C4)*Y+C3)*Y+C2)*Y+C1)*Y
      DIGAMD=DIGAMD-SUM
      RETURN
C
 1000 WRITE(6,7000)X
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE DIGAMD :',
     *  ' ARGUMENT OUT OF RANGE :',D24.16)
      END
```

```
      DOUBLE PRECISION FUNCTION DLGAMA(X)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  LOGARITHM OF GAMMA FUNCTION
C
C  BASED ON ALGORITHM ACM291, COMMUN. ASSOC. COMPUT. MACH. (1966)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA SMALL,CRIT,BIG,TOOBIG/1D-7,13D0,1D9,2D36/
C
C         C0 IS 0.5*LOG(2*PI)
C         C1...C7 ARE THE COEFFTS OF THE ASYMPTOTIC EXPANSION OF DLGAMA
C
      DATA C0,C1,C2,C3,C4,C5,C6,C7/
     *   0.91893 85332 04672 742D 0,  0.83333 33333 33333 333D-1,
     *  -0.27777 77777 77777 778D-2,  0.79365 07936 50793 651D-3,
     *  -0.59523 80952 38095 238D-3,  0.84175 08417 50841 751D-3,
     *  -0.19175 26917 52691 753D-2,  0.64102 56410 25641 026D-2/
C
C         S1 IS -(EULER'S CONSTANT), S2 IS PI**2/12
C
      DATA S1/-0.57721 56649 01532 861D 0/
      DATA S2/ 0.82246 70334 24113 218D 0/
C
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/,TWO/2D0/
      DLGAMA=ZERO
      IF(X.LE.ZERO)GOTO 1000
      IF(X.GT.TOOBIG)GOTO 1000
C
C         USE SMALL-X APPROXIMATION IF X IS NEAR 0, 1 OR 2
C
      IF(DABS(X-TWO).GT.SMALL)GOTO 10
      DLGAMA=DLOG(X-ONE)
      XX=X-TWO
      GOTO 20
   10 IF(DABS(X-ONE).GT.SMALL)GOTO 30
      XX=X-ONE
   20 DLGAMA=DLGAMA+XX*(S1+XX*S2)
      RETURN
   30 IF(X.GT.SMALL)GOTO 40
      DLGAMA=-DLOG(X)+S1*X
      RETURN
C
C         REDUCE TO DLGAMA(X+N) WHERE X+N.GE.CRIT
C
   40 SUM1=ZERO
      Y=X
      IF(Y.GE.CRIT)GOTO 60
      Z=ONE
   50 Z=Z*Y
      Y=Y+ONE
      IF(Y.LT.CRIT)GOTO 50
      SUM1=SUM1-DLOG(Z)
C
C         USE ASYMPTOTIC EXPANSION IF Y.GE.CRIT
C
   60 SUM1=SUM1+(Y-HALF)*DLOG(Y)-Y+C0
      SUM2=ZERO
```

```
      IF(Y.GE.BIG)GOTO 70
      Z=ONE/(Y*Y)
      SUM2=((((((C7*Z+C6)*Z+C5)*Z+C4)*Z+C3)*Z+C2)*Z+C1)/Y
   70 DLGAMA=SUM1+SUM2
      RETURN
C
 1000 WRITE(6,7000)X
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE DLGAMA :',
     *  ' ARGUMENT OUT OF RANGE :',D24.16)
      END
```

```
      SUBROUTINE DURAND(SEED,N,X)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  PSEUDO RANDOM NUMBER GENERATOR
C
C  PARAMETERS OF ROUTINE:
C  SEED   *IN/OUT* SEED FOR RANDOM NUMBER GENERATOR. SHOULD BE A WHOLE
C                  NUMBER IN THE RANGE 2D0 TO 2147483647D0.
C  N      * INPUT* NUMBER OF NUMBERS TO BE GENERATED
C  X      *OUTPUT* ARRAY OF LENGTH N. ON EXIT, CONTAINS RANDOM NUMBERS.
C
C  METHOD USED: MULTIPLICATIVE CONGRUENTIAL GENERATOR WITH BASE 2**31-1
C  AND MULTIPLIER 7**5 (P.A.W. LEWIS ET AL., 1969, IBM SYSTEMS JOURNAL)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION X(N)
      DATA AMULT/16807D0/
      DATA BASE,RBASE/2147483647D0,4.65661287524579692D-10/
      DO 10 I=1,N
      SEED=DMOD(SEED*AMULT,BASE)
      X(I)=SEED*RBASE
   10 CONTINUE
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION GAMIND(X,ALPHA,G)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  THE INCOMPLETE GAMMA INTEGRAL
C
C  BASED ON ALGORITHM AS239, APPL. STATIST. (1988) VOL.37 NO.3
C
C  PARAMETERS OF ROUTINE:
C  X      * INPUT* ARGUMENT OF FUNCTION (UPPER LIMIT OF INTEGRATION)
C  ALPHA  * INPUT* SHAPE PARAMETER
C  G      * INPUT* LOG(GAMMA(ALPHA)). MUST BE SUPPLIED BY THE PROGRAM,
C                  E.G. AS DLGAMA(ALPHA).
C
C  OTHER ROUTINES USED: DERF
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/,TWO/2D0/,THREE/3D0/,X13/13D0/,
     *  X36/36D0/,X42/42D0/,X119/119D0/,X1620/1620D0/,X38880/38880D0/,
     *  RTHALF/0.70710 67811 86547 524D0/
C
C         EPS,MAXIT CONTROL THE TEST FOR CONVERGENCE OF THE SERIES AND
C           CONTINUED-FRACTION EXPANSIONS.
C         OFL IS A LARGE NUMBER, USED TO RESCALE THE CONTINUED FRACTION.
C         UFL IS SUCH THAT EXP(UFL) IS JUST .GT. ZERO.
C         AHILL CONTROLS THE SWITCH TO HILL'S APPROXIMATION.
C
      DATA EPS/1D-12/,MAXIT/100000/,OFL/1D30/,UFL/-180D0/,AHILL/1D4/
      GAMIND=ZERO
      IF(ALPHA.LE.ZERO)GOTO 1000
      IF(X.LT.ZERO)GOTO 1010
      IF(X.EQ.ZERO)RETURN
C
      IF(ALPHA.GT.AHILL)GOTO 100
      IF(X.GT.ONE.AND.X.GE.ALPHA)GOTO 50
C
C         SERIES EXPANSION
C
      SUM=ONE
      TERM=ONE
      A=ALPHA
      DO 10 IT=1,MAXIT
      A=A+ONE
      TERM=TERM*X/A
      SUM=SUM+TERM
      IF(TERM.LE.EPS)GOTO 20
   10 CONTINUE
      WRITE(6,7020)
   20 ARG=ALPHA*DLOG(X)-X-G+DLOG(SUM/ALPHA)
      GAMIND=ZERO
      IF(ARG.GE.UFL)GAMIND=DEXP(ARG)
      RETURN
C
C         CONTINUED-FRACTION EXPANSION
C
   50 CONTINUE
      A=ONE-ALPHA
      B=A+X+ONE
      TERM=ZERO
```

```
      PN1=ONE
      PN2=X
      PN3=X+ONE
      PN4=X*B
      RATIO=PN3/PN4
      DO 70 IT=1,MAXIT
      A=A+ONE
      B=B+TWO
      TERM=TERM+ONE
      AN=A*TERM
      PN5=B*PN3-AN*PN1
      PN6=B*PN4-AN*PN2
      IF(PN6.EQ.ZERO)GOTO 60
      RN=PN5/PN6
      DIFF=DABS(RATIO-RN)
      IF(DIFF.LE.EPS.AND.DIFF.LE.EPS*RN)GOTO 80
      RATIO=RN
   60 PN1=PN3
      PN2=PN4
      PN3=PN5
      PN4=PN6
      IF(DABS(PN5).LT.OFL)GOTO 70
      PN1=PN1/OFL
      PN2=PN2/OFL
      PN3=PN3/OFL
      PN4=PN4/OFL
   70 CONTINUE
      WRITE(6,7020)
   80 ARG=ALPHA*DLOG(X)-X-G+DLOG(RATIO)
      GAMIND=ONE
      IF(ARG.GE.UFL)GAMIND=ONE-DEXP(ARG)
      RETURN
C
C         ALPHA IS LARGE: USE HILL'S APPROXIMATION (N.L. JOHNSON AND
C         S. KOTZ, 1970, 'CONTINUOUS UNIVARIATE DISTRIBUTIONS 1', P.180)
C
C         THE 'DO 110' LOOP CALCULATES 2*(X-ALPHA-ALPHA*DLOG(X/ALPHA)),
C         USING POWER-SERIES EXPANSION TO AVOID ROUNDING ERROR
C
  100 CONTINUE
      R=ONE/DSQRT(ALPHA)
      Z=(X-ALPHA)*R
      TERM=Z*Z
      SUM=HALF*TERM
      DO 110 I=1,12
      TERM=-TERM*Z*R
      SUM=SUM+TERM/(I+TWO)
      IF(DABS(TERM).LT.EPS)GOTO 120
  110 CONTINUE
  120 WW=TWO*SUM
      W=DSQRT(WW)
      IF(X.LT.ALPHA)W=-W
      H1=ONE/THREE
      H2=-W/X36
      H3=(-WW+X13)/X1620
      H4=(X42*WW+X119)*W/X38880
      Z=(((H4*R+H3)*R+H2)*R+H1)*R+W
      GAMIND=HALF+HALF*DERF(Z*RTHALF)
      RETURN
C
 1000 WRITE(6,7000)ALPHA
      RETURN
 1010 WRITE(6,7010)X
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE GAMIND :',
     *  ' SHAPE PARAMETER OUT OF RANGE :',D16.8)
 7010 FORMAT(' *** ERROR *** ROUTINE GAMIND :',
     *  ' ARGUMENT OF FUNCTION OUT OF RANGE :',D16.8)
 7020 FORMAT(' ** WARNING ** ROUTINE GAMIND :',
     *  ' ITERATION HAS NOT CONVERGED. RESULT MAY BE UNRELIABLE.')
```

```
END
```

```
      DOUBLE PRECISION FUNCTION QUASTN(F)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  QUANTILE FUNCTION OF THE STANDARD NORMAL DISTRIBUTION
C
C  BASED ON ALGORITHM AS241, APPL. STATIST. (1988) VOL.37 NO.3
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA ZERO/0D0/,HALF/0.5D0/,ONE/1D0/
      DATA SPLIT1/0.425D0/,SPLIT2/5D0/,CONST1/0.180625D0/,CONST2/1.6D0/
C
C         COEFFICIENTS OF RATIONAL-FUNCTION APPROXIMATIONS
C
      DATA A0,A1,A2,A3,A4,A5,A6,A7,B1,B2,B3,B4,B5,B6,B7/
     *                                 0.33871 32872 79636 661D  1,
     *  0.13314 16678 91784 377D  3,  0.19715 90950 30655 144D  4,
     *  0.13731 69376 55094 611D  5,  0.45921 95393 15498 715D  5,
     *  0.67265 77092 70087 009D  5,  0.33430 57558 35881 281D  5,
     *  0.25090 80928 73012 267D  4,  0.42313 33070 16009 113D  2,
     *  0.68718 70074 92057 908D  3,  0.53941 96021 42475 111D  4,
     *  0.21213 79430 15865 959D  5,  0.39307 89580 00927 106D  5,
     *  0.28729 08573 57219 427D  5,  0.52264 95278 85285 456D  4/
      DATA C0,C1,C2,C3,C4,C5,C6,C7,D1,D2,D3,D4,D5,D6,D7/
     *                                 0.14234 37110 74968 358D  1,
     *  0.46303 37846 15654 530D  1,  0.57694 97221 46069 141D  1,
     *  0.36478 48324 76320 461D  1,  0.12704 58252 45236 838D  1,
     *  0.24178 07251 77450 612D  0,  0.22723 84498 92691 846D -1,
     *  0.77454 50142 78341 408D -3,  0.20531 91626 63775 882D  1,
     *  0.16763 84830 18380 385D  1,  0.68976 73349 85100 005D  0,
     *  0.14810 39764 27480 075D  0,  0.15198 66656 36164 572D -1,
     *  0.54759 38084 99534 495D -3,  0.10507 50071 64441 684D -8/
      DATA E0,E1,E2,E3,E4,E5,E6,E7,F1,F2,F3,F4,F5,F6,F7/
     *                                 0.66579 04643 50110 378D  1,
     *  0.54637 84911 16411 437D  1,  0.17848 26539 91729 133D  1,
     *  0.29656 05718 28504 891D  0,  0.26532 18952 65761 230D -1,
     *  0.12426 60947 38807 844D -2,  0.27115 55568 74348 758D -4,
     *  0.20103 34399 29228 813D -6,  0.59983 22065 55887 938D  0,
     *  0.13692 98809 22735 805D  0,  0.14875 36129 08506 149D -1,
     *  0.78686 91311 45613 259D -3,  0.18463 18317 51005 468D -4,
     *  0.14215 11758 31644 589D -6,  0.20442 63103 38993 979D-14/
C
      Q=F-HALF
      IF(DABS(Q).GT.SPLIT1)GOTO 10
      R=CONST1-Q*Q
      QUASTN=Q*(((((((A7*R+A6)*R+A5)*R+A4)*R+A3)*R+A2)*R+A1)*R+A0)
     *        /(((((((B7*R+B6)*R+B5)*R+B4)*R+B3)*R+B2)*R+B1)*R+ONE)
      RETURN
   10 R=F
      IF(Q.GE.ZERO)R=ONE-F
      IF(R.LE.ZERO)GOTO 1000
      R=DSQRT(-DLOG(R))
      IF(R.GT.SPLIT2)GOTO 20
      R=R-CONST2
      QUASTN=(((((((C7*R+C6)*R+C5)*R+C4)*R+C3)*R+C2)*R+C1)*R+C0)
     *      /(((((((D7*R+D6)*R+D5)*R+D4)*R+D3)*R+D2)*R+D1)*R+ONE)
      GOTO 30
   20 R=R-SPLIT2
      QUASTN=(((((((E7*R+E6)*R+E5)*R+E4)*R+E3)*R+E2)*R+E1)*R+E0)
```

```
     *      /(((((((F7*R+F6)*R+F5)*R+F4)*R+F3)*R+F2)*R+F1)*R+ONE)
   30 IF(Q.LT.ZERO)QUASTN=-QUASTN
      RETURN
C
 1000 WRITE(6,7000)F
      QUASTN=ZERO
      RETURN
C
 7000 FORMAT(' *** ERROR *** ROUTINE QUASTN :',
     *  ' ARGUMENT OF FUNCTION INVALID')
      END
```

```
      SUBROUTINE SORT(X,N)
C***********************************************************************
C*                                                                     *
C*  FORTRAN CODE WRITTEN FOR INCLUSION IN IBM RESEARCH REPORT RC20525, *
C*  'FORTRAN ROUTINES FOR USE WITH THE METHOD OF L-MOMENTS, VERSION 3' *
C*                                                                     *
C*  J. R. M. HOSKING                                                   *
C*  IBM RESEARCH DIVISION                                              *
C*  T. J. WATSON RESEARCH CENTER                                       *
C*  YORKTOWN HEIGHTS                                                   *
C*  NEW YORK 10598, U.S.A.                                             *
C*                                                                     *
C*  VERSION 3     AUGUST 1996                                          *
C*                                                                     *
C***********************************************************************
C
C  SORTS THE ARRAY X INTO ASCENDING ORDER
C
C  PARAMETERS OF ROUTINE:
C  X      *IN/OUT* ARRAY OF LENGTH N. CONTAINS THE NUMBERS TO BE SORTED.
C                  ON EXIT, CONTAINS THE SORTED NUMBERS.
C  N      * INPUT* NUMBER OF ELEMENTS TO BE SORTED
C
C  METHOD USED IS SHELL SORT WITH SEQUENCE OF INCREMENTS AS IN
C  D.F.KNUTH (1969) 'THE ART OF COMPUTER PROGRAMMING', VOL.3, P.95
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION X(N)
      IF(N.LE.1)RETURN
      J=4
      DO 10 I=1,100
      J=3*J+1
      IF(J.GE.N)GOTO 20
   10 CONTINUE
   20 CONTINUE
      M=(J/3)
      DO 60 MM=1,100
      M=M/3
      IF(M.EQ.0)RETURN
      DO 50 I=M+1,N
      TEST=X(I)
      J=I
      DO 30 JJ=1,100
      J=J-M
      IF(J.LE.0)GOTO 40
      IF(TEST.GE.X(J))GOTO 40
   30 X(J+M)=X(J)
   40 CONTINUE
   50 X(J+M)=TEST
   60 CONTINUE
      END
```