3-1997

Robust Parameter Estimation for the Mixed Weibull (Seven Parameter) Including the Method of Minimum Likelihood and the Method of Minimum Distance

Donald A. Mumford

ROBUST PARAMETER ESTIMATION FOR THE MIXED WEIBULL (SEVEN PARAMETER) INCLUDING THE METHOD OF MAXIMUM LIKELIHOOD AND THE METHOD OF MINIMUM DISTANCE

THESIS

Donald A. Mumford, Captain, USAF

AFIT/GOR/ENY/97M-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENY/97M-1

ROBUST PARAMETER ESTIMATION FOR THE MIXED WEIBULL (SEVEN PARAMETER) INCLUDING THE METHOD OF MAXIMUM LIKELIHOOD AND THE METHOD OF MINIMUM DISTANCE

THESIS

Donald A. Mumford, Captain, USAF

AFIT/GOR/ENY/97M-1

AFIT/GOR/ENY/97M-1

ROBUST PARAMETER ESTIMATION FOR THE MIXED WEIBULL (SEVEN PARAMETER) INCLUDING THE METHOD OF MAXIMUM LIKELIHOOD AND THE METHOD OF MINIMUM DISTANCE

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Donald A. Mumford

Captain, USAF

March 1997

AFIT/GOR/ENY/97M-1

ROBUST PARAMETER ESTIMATION FOR THE MIXED WEIBULL (SEVEN PARAMETER) INCLUDING THE METHOD OF MAXIMUM LIKELIHOOD AND THE METHOD OF MINIMUM DISTANCE

Donald A. Mumford
Captain, USAF

Approved:

Advisor: Maj Edward A. Pohl, Ph.d.	10 March 97 date
Reader: Dr Albert C. Moore, Ph.D.	10 March 97 date
Reader: Maj W. Paul Murdock, Jr., Ph.D.	10 MARCH 97 date
Reader: Maj (Sel) John S. Crown, Ph.D.	10 March 97 date

Acknowledgments

First, I thank God that he has allowed me the opportunity to do something meaningful with my life. Second, I need to thank my wife (Yolanda) who received the least credit, but sacrificed the most.

I am greatly indebted to my advisors/readers, Major Edward A. Pohl, Dr Albert H. Moore and Major W. Paul Murdock, Jr. They inspired me to attain new levels of engineering, understanding and personal growth. Also, I thank them for believing in me. Finally, I would like to thank Dr Moore for allowing me to share in his research and insight of years of experience.

Don Mumford

Table of Contents

Appendices

Page

List of Figures

List of Tables

Abstract

Robust parameter estimation is successfully applied to the Mixed Weibull (seven parameter) using the Method of Minimum Distance and the Method of Maximum Likelihood. That is, parameters can now be estimated for a mixture of two Weibull distributions where the true populations are co-located, partially co-located or highly separated. Both techniques provided very robust estimates that were far superior to current parameter estimation techniques. Sample sizes as low as ten with mixing proportions down to 0.1 were investigated.

For the MLEs, innovative bounding techniques are presented to allow consistent and correct convergence using any reasonable point estimate. The likelihood function is solved numerically as a non-linear constrained optimization using a quasi-Newton method.

Minimum Distance Estimates (over three hundred scenarios investigated) are derived for some variation or combination of the mixing proportion and the location parameter(s), individually and simultaneously (the Anderson-Darling and Cramer-von Mises statistics were used). In fact, the MDE for the mixing proportion was so effective that future researchers should consider some permanent combination.

Primary measures of success were based on comparison of CDFs. Mean square error (MSE) and integrated absolute difference (IAF) between the estimated and true distributions were measured including confidence intervals.

ROBUST PARAMETER ESTIMATION FOR THE MIXED WEIBULL (SEVEN PARAMETER) INCLUDING THE METHOD OF MAXIMUM LIKELIHOOD AND THE METHOD OF MINIMUM DISTANCE

I. Introduction

One common thread in describing the life of a satellite, a billion dollar manufacturing process or environmental stress screening is the use of statistics and probability to mathematically model the multitude of characteristics and processes that cannot be modeled deterministically. The reality is that most characteristics or processes are not known with certainty, therefore, a good statistical and/or probability model provides the best mathematical representation. Of course, the ultimate goal may be either to cost effectively manage these processes or to preclude current system deficiencies in the next generation system design. Thus, the Department of Defense and the Air Force conducts numerous studies to understand military systems. One primary Air Force Agency which conducts such studies is the Air Force Operational Test and Evaluation Center (AFOTEC). While they are chartered to conduct operational test and evaluation, AFOTEC's agenda includes determining a weapon system's effectiveness and suitability.

When modeling a system, AFOTEC will collect information such as observed successes, observed failures and repair data. Unfortunately, statistical models require numerous observations to build confidence in their accuracy. Because additional observations usually equate to additional time and money, data is often very limited. Nonetheless, AFOTEC must draw conclusions based on the data they collect, no matter how small the sample size. Even in statistics, many functional forms have been developed in an attempt to provide better statistical and/or probability models. Historically, engineers have used the Weibull distribution because of its ability to accurately model an infinite number of distributional forms including the Exponential and Normal. Each functional form requires correct estimation of its parameters i.e. those parameters that best fit the limited data.

The key to success in this discipline is selection of a mathematically correct distribution and associated parameters based on the observed data. Stating the latter, formally, "the ability to accurately predict the parameters of a known or hypothesized distribution." In practice, this is accomplished by using estimators which have been shown to minimize the amount of error between the observed data and the true system in some meaningful and measurable way. Historically, the primary means of estimation has been the Method of Maximum Likelihood (thus, the common reference to Maximum Likelihood Estimation or specifically, Maximum Likelihood Estimates, MLEs).

Formal Problem Statement

Over the past decade, engineers expressed the need for robust bi-modal statistical models to represent a variety of real-world systems and processes. Bi-modal data can be modeled with a single distribution, but, often the quality of the model is poor. Hence, the introduction of a mixture of statistical models to represent multi-modality quickly gained popularity as a more mathematically correct representation. Multi-mode observations are inherent in many fields particularly logistics where the Weibull distribution is used extensively. Hence, the term Mixed Weibull to represent a mixture of two or more Weibull distributions (the term Mixed Weibull in this report indicating a mixture of two Weibulls). Additional applications of the Mixed Weibull include reliability engineering (bi-modal failure modes for electrical failures), Criminal Justice System (bi-modal rates of re-incarceration) and a variety of bi-modal medical data. Unfortunately, parameter estimation can deteriorate rapidly with mixtures of distributions because of the increase in the number of parameters.

The Method of Maximum Likelihood

The MLEs possess many desirable properties (Cox and Hinkley, 1974; and Wetherill, 1981). "Under certain general conditions the MLEs are consistent, and have the asymptotic properties of efficiency, normality and unbiasedness. Furthermore, the MLEs are functions of the sufficient statistics if they exist. And, the MLEs of the functions of unknown parameters are the functions of the MLEs of the parameters (Jaing, 1991)."

While MLEs enjoy many asymptotic properties, they have proven considerably less desirable under many common scenarios such as: 1) small or moderate sample sizes; and 2) distributions which have large number of parameters to estimate such as the Weibull, Gamma or a mixture of distributions (Dr Moore, 1996). "Suggested estimators for distributions include the 'average' (for the Normal, Poisson, and Exponential), the number of observed values (for the Uniform), and many other sample-based statistics (Mendenhall, 1990:370)." Most of these well-known sample statistics are Maximum Likelihood Estimates (MLEs). While computationally efficient and possessing many desirable asymptotic properties, MLEs make several problematic assumptions including: 1) the sample is an accurate representation of true population; 2) the user knows the correct family of distributions; 3) the desirable properties hold for small or moderate sample sizes; and 4) the sample contains no significant outliers.

The Method of Minimum Distance

A promising alternative to MLE, Minimum Distance Estimation (MDE) is less sensitive to these assumptions. Hence, its classification as a "robust estimation" technique. Robust estimation attempts to protect against minor deviations from underlying assumptions (Rey, 1983). Basically, the concept of MDE is that better estimates will be obtained by fitting a distribution to the sample data. While computationally more intensive, the theoretical quality is maximized since these functions are based on minimizing the distance between the cumulative distribution function (CDF) of the observed data (empirical distribution function, EDF) and the hypothesized (in this case, estimated) cumulative distribution function (CDF). As the number of observations grows larger, the EDF approaches the true population CDF.

In statistical linguistics, the observed data such as observed failures are called the 'sample.' Specifically, parameters are estimated iteratively until the 'error' (between the EDF and estimated CDF) is minimized. In this context, accuracy is defined as the ability to minimize 'error.' Primary measures of accuracy are a class of goodness-of-fit statistics which measure the distance between the estimated cumulative distribution function (CDF) and the EDF. Formally, the distance between the estimated CDF and the EDF are minimized using numerical analysis on the mathematical functions developed for goodness-of-fit tests. The type of statistic is determined by the user's focus or assumptions that need to be overcome. Over thirty years ago, MDE recorded as much as a one thousand percent improvement over MLE (Dr Moore, 1996).

Research Objectives

The objective of this research was to investigate the application of parameter estimation methods for the seven parameter Mixed Weibull. First, due to their desirable asymptotic properties, the Method of Maximum Likelihood was implemented. Second, the Method of Minimum Distance was applied since parameter estimation might be enhanced particularly for small sample sizes. This research extends previous work in two ways. Specifically, the Method of Minimum Distance and the Method of Maximum Likelihood were extended to the more useful seven parameter Mixed Weibull. Previously, MDE had not been applied to the Mixed Weibull. Also, MLEs have only been derived for at most the five parameter Mixed Weibull.

II. Literature Review

This research extends to several key areas including the following: 1) the history of MDE; 2) the application of MDE and MLE to the Weibull distribution; 3) the application of MLE to a mixture of distributions; 4) the application of MDE as a robust estimation technique; and 5) finally, the recent progress with the Mixed Weibull. A unique and important history associated with each key area is included after a general review of the larger scope, parameter estimation for mixtures of distributions. Since the Method of Minimum Distance and the Method of Maximum Likelihood were born out of non-Mixed Weibull environment, the latter part of this discussion is devoted to these topics.

Parameter Estimation for Mixtures of Distributions

Research on the mixed distribution began in 1894 with Karl Pearson who constructed moment estimators for the five parameters of a mixture of two normal distributions. Rao (1948) applied an iterative method to the maximum likelihood equations for the special case where common variance was assumed for two normal subpopulations of the mixture. Kao (1959) utilized Weibull probability paper and graphical techniques to obtain the parameter estimates for a failure model involving a mixture of Weibull populations. Hasselblad (1966) dealt with a more general case where the number of sub-populations was greater than or equal to three. He employed the method of steepest ascent and Newton's method to solve for the MLEs of normal distributions. Bhattacharya (1967) developed a method of resolution of a distribution into normal sub-populations when the sub-population distributions were well separated (contained no overlap). Tan and Chang (1972) derived the asymptotic covariance matrix of the moment estimators and the information for a mixture of two normal distributions assuming common variance. Dick and Bowden (1973) primarily dealt with the maximum likelihood equations for the case when independent sample information was available from one of the subpopulations. Peters and Walker (1977) developed an iterative scheme for obtaining the MLEs of the parameters of a mixture of two normal distributions. Various researchers have viewed the parameter estimation problem under a different setting. Hosmer (1973) researched Hasselblad's iterative MLEs for a mixture of two normal distributions and made observations that the estimates tended to have smaller variances when the component samples constituted even as little as ten percent of the total sample. In an effort to preclude the intensive computational effort, John (1970) proposed an alternative model based on the product of normal distributions, each one raised to the power of one or zero. Hill (1963) investigated the estimation of the mixing proportion (p). He derived a general power series expansion for the information and considered various approximations for the case of two normal distributions. They optimized the mixing proportion by maximizing the expected value of the function. Blischke (1964) attempted various estimation procedures for a mixture of binomial distributions. Rider (1961) investigated the method of moments for a mixture of two exponential distributions. He extended his results to other mixed populations including Poisson, binomial and Weibull mixtures. For the Weibull mixture, he assumed that the shape parameters were known. Cohen (1967) extended Rider's work by obtaining moment estimates for a mixture of two normal distributions. He assumed common variance for both sub-populations. Since then a considerable amount of work has been devoted to this area, but specifically devoted to a mixture of normal or exponential distributions, e.g. Hasselblad (1966), Day (1969), Wolfe (1970), McLachlan and Jones (1988), Ashour (1985) and Cheng, Fu and Sinha (1985).

Falls (1970) attempted to find the five parameters of a two-Weibull mixture by the method of moments. Unfortunately, he could not solve the resulting system of five equations. Later, Cran (1976) gave some theoretical bases to support Kao's procedure resulting in a well-known and commonly used graphical procedure called the 'Kao-Cran' graphical estimation method. Olsson (1979) directly searched the maximum of the log-likelihood function of the Mixed Weibull distribution through the Nelder-Mead Simplex Procedure. In a follow-on effort, Jensen and Petersen (1982) developed another graphical procedure for parameter estimation of a two-Weibull mixture when the two sub-populations are well separated. Cheng and Fu (1982) proposed a weighted least squares method for estimating the parameters of a mixture of two

Weibulls when the data are grouped postmortem. Sinha (1986) gave an iterative procedure to obtain the MLE of a two-Weibull mixture for postmortem data. The approach is extended from the approach of Mendenhall and Hader (1958) which developed the MLE of a Mixed Exponential distribution.

The major works closest to this study are those of Kaylan (1979), Kaylan and Harris (1981), Mandelbaum (1982) and Mendelbaum and Harris (1982). Similar to Hasselblad's (1969) scheme, Kaylan developed an iterative procedure for solving the likelihood equations of the likelihood function of a mixed Weibull distribution when all n times to failure are available. The procedure is a typical fixed point iteration procedure. While it has been proven that the direction of the two points generated from the two successive iterations is in the direction of increasing the log-likelihood function, there is no guarantee that there will be actual improvement. Thus, a secondary rule must be incorporated to check actual improvement. Kaylan also developed a second algorithm based on the second partial derivatives of the likelihood function with respect to the mixing weights. After Kaylan's work, Mandelbaum (1982) developed algorithms for the progressive censoring sample for non-postmortem and postmortem cases.

Since then significant research has been conducted for a variety of reasons including the appropriateness of Weibull distribution to the fields of reliability, environmental stress screening and other bi-modal data. In 1991, Jiang performed extensive research to support reliability engineering, "use of the Mixed Weibull as a statistical model for the lifetime of units with multiple modes of failure." Both graphical and numerical methods were developed. An algorithm is successfully applied to solve the MLE for Mixed Weibull distributions where the number of sub-populations is known. The algorithms for complete, censored, grouped and suspended samples with non-postmortem and post-mortem failures are developed accordingly. The next year (1992), Jiang and Kececioglu published a graphical approach for modeling failure data by a Mixed Weibull. A majority of his effort focused on graphical analysis of a mixture of two Weibull distributions including parameter estimation. He derived a variety of classes based on common properties including extensive graphical properties. He also investigated the applicability of the two existing methods of graphical parameter estimation (Kao-Cran and Jensen-Peterson methods). In 1994, Kececioglu extended their research providing a method to estimate parameters of the Mixed Weibull for burn-in data using a Bayesian estimator. Later, in an effort to clarify graphical parameter estimation, Jiang and Murphy (1995) published research on an improved graphical technique. In that same year, Jiang and Kececioglu also investigated and published a methodology for parameter estimation from censored data via the Method of Maximum Likelihood. The algorithm follows the principles set forth by Mandelbaum using his Expectation and Maximization algorithm, and it is derived for both the postmortem and non-postmortem data. Finally, Pohl (1995) demonstrated the utility of the Mixed Weibull in Environmental Stress Screening (ESS). That is, ESS was employed to reduce, if not eliminate, the occurrence of early field failures. Specifically, he developed stress screening strategies for multi-component systems with Weibull failure rates.

Minimum Distance Estimation (MDE)

Many experts in the field of statistics categorize MDE as a "robust estimation" technique. It is a non-classical approach attempting to improve traditional estimation procedures where the procedure attempts to protect against minor deviations from underlying assumptions (Rey, 1983). The following is a brief history tracing MDE as a robust technique quoted from Gallagher (1990):

> The term robustness was first proposed by Box in 1953. In 1970, six prominent statisticians spent a year developing and testing seventy robust estimators for the location parameter of symmetric distributions (Andrews, 1972). Of course, these early methods focused on limited problems such as symmetric PDFs or estimation of only the location parameter. Eventually, these methods were extended to estimation of the PDF shape and location parameters (Parr and Schucany, 1980:616)."

The following articles, quoted from Benton-Santo (1986), combine to provide a brief history and prove the validity of the Method of Minimum Distance:

> The origin of the Method of Minimum Distance starts with Wolfowitz who published two papers in the 1950's. He developed the theory and proved the consistency of the estimates. Later, Matusita (1959) proved the consistency of MDE with other distance measures. Sahler's (1970) paper

proves conditions for the existence and consistency of MDEs. Hobbs, Moore and James investigated MDE for the three parameter Gamma (1984). Varying only the location parameter, James (1980) demonstrated superiority of MDE over MLE for the Gamma. In the same year, Hobbs, Moore and Miller (1980) successfully applied the method to the three parameter Weibull. Miller found similar results for a limited class of the two parameter Weibull. Also, Parr and Schucany (1980) demonstrated MDE by estimating the location or mean of the normal distribution. Daniel found improved estimates for the t distribution (1980:12).

Minimum Distance Estimation continued on various distributions including the normal (Eslinger, 1990), Exponential and Weibull. Perhaps, the most pertinent and extensive research was conducted by Gallagher and Moore in 1990. Using MDE on a Weibull distribution, they evaluated several MDE methods compared to the MLEs. Gallagher's results indicated that MDEs were superior to the MLEs under various conditions particularly when the minimum distance estimates were derived for the location parameter only, and for all three parameters simultaneously. Gallagher's research represents one of few that did not reduce the three parameter Weibull to a two parameter Weibull and he may be the first to derive MDEs based on all three parameters simultaneously.

In early research on mixed distributions, Woodard, Schucany, Lindsey and Parr (1984) conducted a comparison of MDE and MLE for estimating the mixing proportion for a mixture of two normal distributions. Results indicate that MLE was superior to MDE when component distributions are actually normal, while MDE provides better estimation when there are symmetric departures from normality. When component distributions are not symmetric, however, it is seen that neither of these normal based techniques provide satisfactory results. For a mixture of two normal and two exponential, Benton-Santo (1986) developed and compared the Method of Moments and quasi-clustering techniques.

Maximum Likelihood Estimation (MLE)

Numerous texts document procedures for deriving MLEs for the Weibull distribution (Banks and Carson,1984:p 373). To date, most applications and research reduce the three parameter Weibull to a two-parameter Weibull by means of a simple translation of the coordinate axis by a distance approximately equal to the first order statistic so the location parameter is equated to zero. This translation, as Harter and Moore discovered in 1965, greatly simplifies calculations particularly for parameter estimation. With a single Weibull, the translation can be performed without loss of accuracy. However, for a Mixed Weibull, this procedure, when applied to both distributions simultaneously, can seriously undermine parameter estimation if the true location parameters are not the same, i.e. do not start at the same location on the x-axis. There are a few cases where the location parameters are the same or this assumption can be made without consequences, i.e. does not seriously degrade parameter estimation. When this assumption is made for a mixture of two Weibull distributions, the problem is reduced to a five parameter Mixed Weibull (from a seven parameter Mixed Weibull since the true mixture of two Weibulls consists of seven parameters by definition).

Nonetheless, Kaylan and Harris (1981) derived MLEs for the first Mixed Weibull and Mixed Exponential. To simplify the problem, they reduced the inherent seven parameter Mixed Weibull to a five parameter Mixed Weibull. Even so they noted, " the problem of obtaining the MLEs for the parameters in mixture models presents considerable difficulty due to the complexity of the Log-likelihood function..."

As they discovered, prior to their work, little research was conducted on a mixture of Weibull distributions:

> Some research was conducted by Hasselblad who developed an iterative scheme to obtain MLEs for a mixtures of exponential-family densities (1969). In 1971, Oppenheimer extended Hasselblad's research to various forms of censoring. For the Mixed Weibull, the only example for parameter estimation was Kao who developed a graphical technique for estimating parameters (1959)." This technique opened the door to many researchers that wanted to use a mixture of two Weibulls. While the technique proved useful, the method provides only crude estimates.

Since then, Mandelbaum (1982) extended Kaylan's work to progressively censored samples including postmortem and non-postmortem cases. However, since all research to date assumes that the distributions are co-located (five parameter mixed model), a large number of researchers have sought an improvement on the current method due to its inflexibility in dealing with mixed distributions that are moderately separated or well separated. That is, for the Mixed Weibull, a seven parameter distribution is desired.

III. Methodology / Research Approach

Specifically, this research compared several different parameter estimation methods for the seven parameter Mixed Weibull (ref Table 1). One of the parameter estimates was based strictly on the Method of Maximum Likelihood while a majority were based on the Method of Minimum Distance. Next, the necessary background is given including basic definitions and notation. The methodology used to derive the Minimum Distance Estimates (MDEs) is presented. Finally, the algorithm used to solve the Maximum Likelihood Estimates (MLEs) is given. This section begins with an executive overview of the methods evaluated for this research that tracks well with the organization of the results and many of the appendices.

Stochastic Nature of the Mixing Proportion

Unfortunately, in the real world of bimodal populations, the mixing proportion is not known with certainty. The mixing proportion is uniformly distributed from zero to one, $p \sim U(0,1)$. The uniformly generated mixing proportion dictates the number of points generated from each population. Specifically, a random saample is generated from a uniform (0,1) distribution. The uniform random samples that exceed (do not exceed) the true mixing proportion dictate the number of Weibull samples generated from each population. Therefore, the proportion of the actual number of Weibull sample values generated and the true mixing proportion were often not the same.

Background

Weibull probability density function (PDF) :

- $$f_j(x;\theta_j) = \left(\frac{\beta_j}{\eta_j}\right)\cdot\left(\frac{x-\delta_j}{\eta_j}\right)^{\beta_j-1}\cdot\exp\left[-\left(\frac{x-\delta_j}{\eta_j}\right)^{\beta_j}\right]$$ Equation (1)

$$(x > \delta_j;\ \beta_j, \delta_j, \eta_j > 0)$$

where θ_j a parameter vector i.e. $\theta_j = (\beta_j, \eta_j, \delta_j)$

(β_j is the 'shape' parameter, η_j is the 'scale' parameter, δ_j is the 'location' parameter)

For example, let $\eta = 1$ and $\delta = 0$, several common functional forms for values of β are:

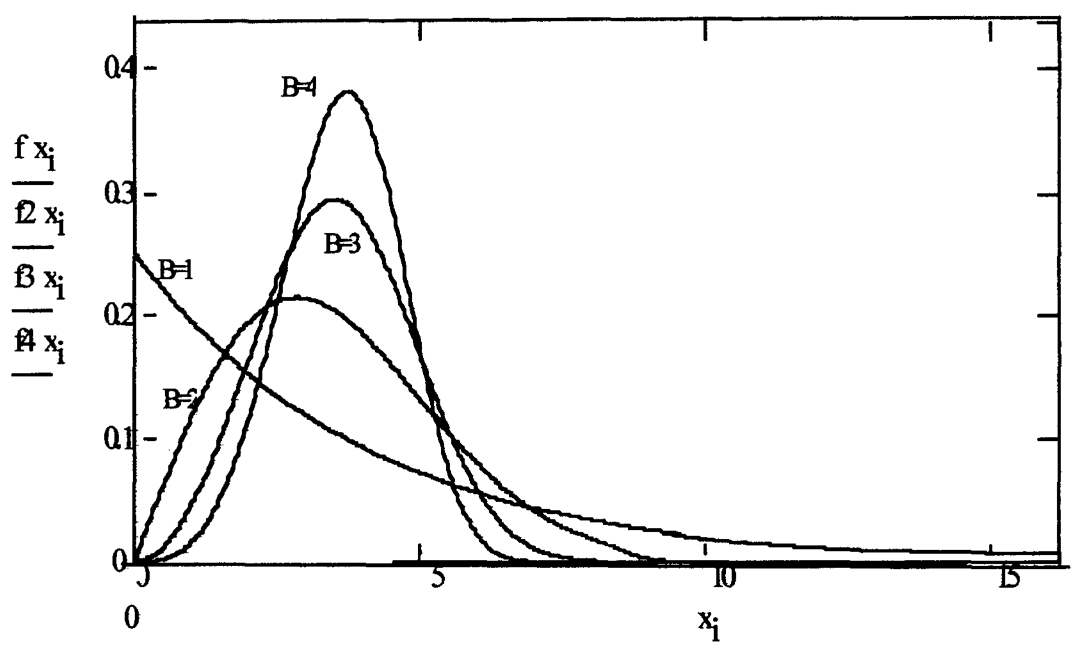


Figure 1. Preview of Weibull Probability Density Functions (PDFs)

Mixed Weibull PDF. Often data is not unimodal but bi-modal. In this case a mixture of two Weibulls (Mixed Weibull) can be used. The Mixed Weibull is formally expressed as a mixture where each PDF is weighted by the mixing proportion, p :

$$g(x;\alpha) = \sum_{j=1}^{2} p_j f_j(x;\theta_j) \qquad \text{Equation (2)}$$

$$\text{where } \sum_{j=1}^{2} p_j = 1 \text{ and } \alpha = (\theta_1, \theta_2, p)$$

For example, let **g(x)** = p*f1(x) + (1-p)*f2(x)

$p := 0.5 \qquad \beta_1 := 1 \quad \delta_1 := 0 \quad \eta_1 := 2 \quad f1(x) := \left[\frac{\beta_1}{\eta_1}\cdot\left[\left(\frac{x-\delta_1}{\eta_1}\right)^{\beta_1 - 1}\right]\cdot \exp\left[-\left(\frac{x-\delta_1}{\eta_1}\right)^{\beta_1}\right]\right]$

$\beta_2 := 4 \quad \delta_2 := 10 \quad \eta_2 := 2 \quad f2(x) := \begin{cases} 0 & \text{if } x \le 10 \\ \left[\frac{\beta_2}{\eta_2}\cdot\left[\left(\frac{x-\delta_2}{\eta_2}\right)^{\beta_2 - 1}\right]\cdot \exp\left[-\left(\frac{x-\delta_2}{\eta_2}\right)^{\beta_2}\right]\right] & \text{otherwise} \end{cases}$

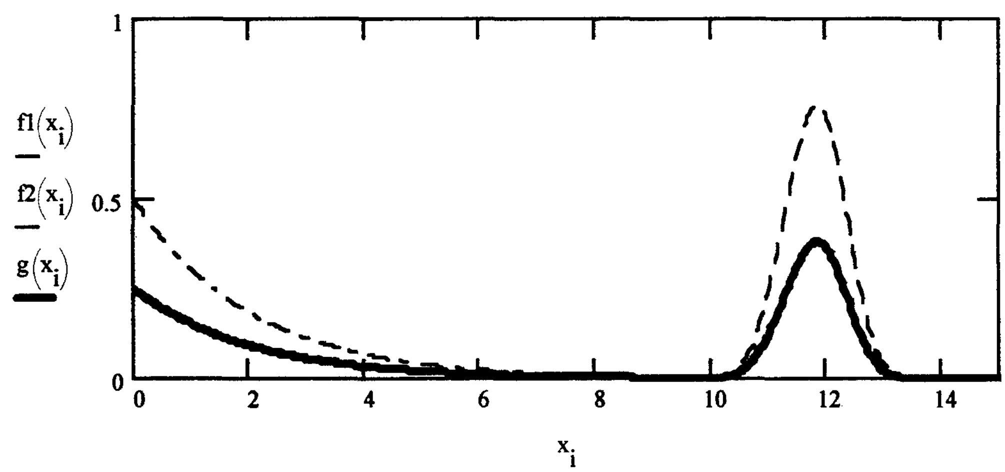


Figure 2. PDF for a mixture of two Weibulls

Cumulative Distribution Function. The cumulative density function (CDF) for the Weibull follows

$$\bullet F(x;\theta_j) = 1 - \exp(-((x-\delta)/\eta)^{\beta}) \qquad \text{Equation (3)}$$

Therefore, the CDF for a mixture of two Weibull distributions follows

$$\bullet G(x,\alpha) = \sum_{j=1}^{2} p_j F_j(x;\theta) \qquad \text{Equation (4)}$$

where p is known a the mixing proportion and where $\alpha = (\theta_1, \theta_2, p)$

For example, let

$p := 0.5$ $\qquad \delta1 := 0 \quad \eta1 := 2 \quad \beta1 := 1$ $\qquad F1(x) := 1 - \exp\left[-\left(\frac{x-\delta1}{\eta1}\right)^{\beta1}\right]$

$\delta2 := 10 \quad \eta2 := 2 \quad \beta2 := 4$ $\qquad F2(x) := \begin{cases} 0 & \text{if } x \le 10 \\ \left[1 - \exp\left[-\left(\frac{x-\delta2}{\eta2}\right)^{\beta2}\right]\right] & \text{otherwise} \end{cases}$

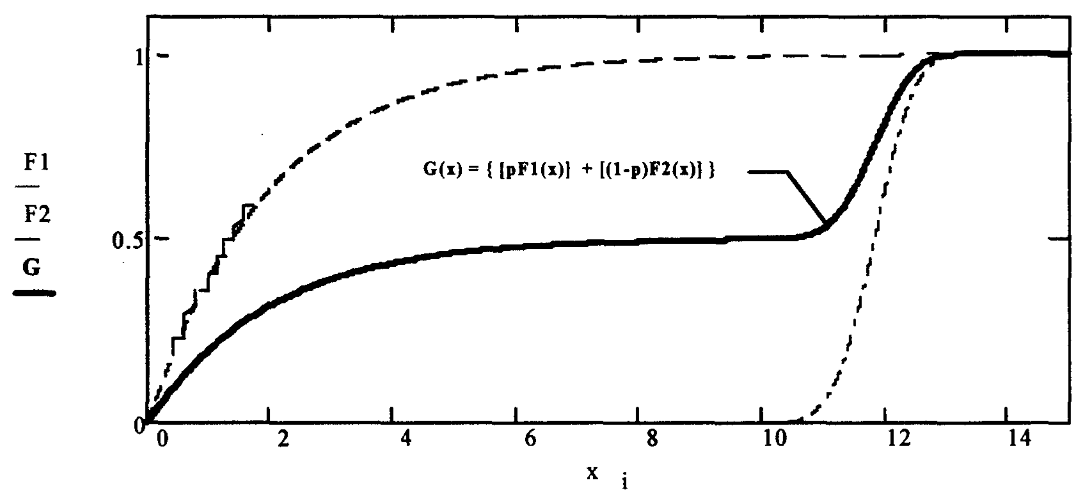


Figure 3. Mixed Weibull Cumulative Density Function

Minimum Distance Estimation (MDE). Minimum distance estimation was so named because the distribution parameters selected minimize the distance between the hypothesized distribution (in this case the estimated distribution) and the sample EDF. The measure of distance is determined by goodness-of-fit statistics which quantify the difference between the Empirical Distribution Function (EDF) and the estimated Cumulative Distribution Function (CDF). There exists a variety of goodness-of-fit statistics that weight the discrepancies between the EDF and CDF differently. For example, the Mixed Weibull EDF and CDF might look something like the following chart where the estimated CDFs are represented by straight lines. MDE would attempt to minimize the distance between the estimated CDFs and the sample data:

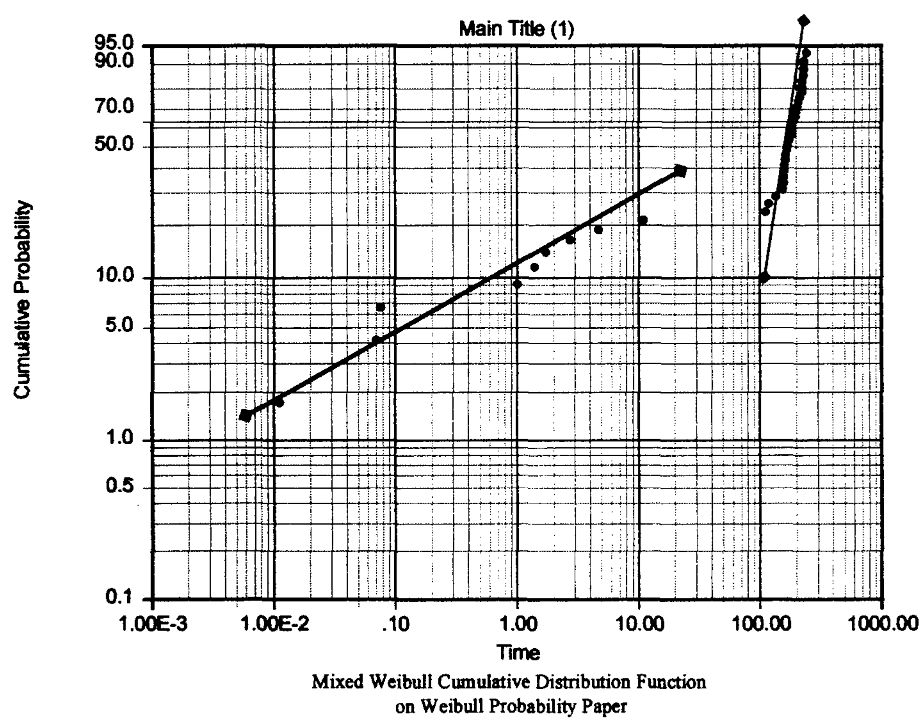


Figure 4. **Philosophy behind Minimum Distance Estimation**

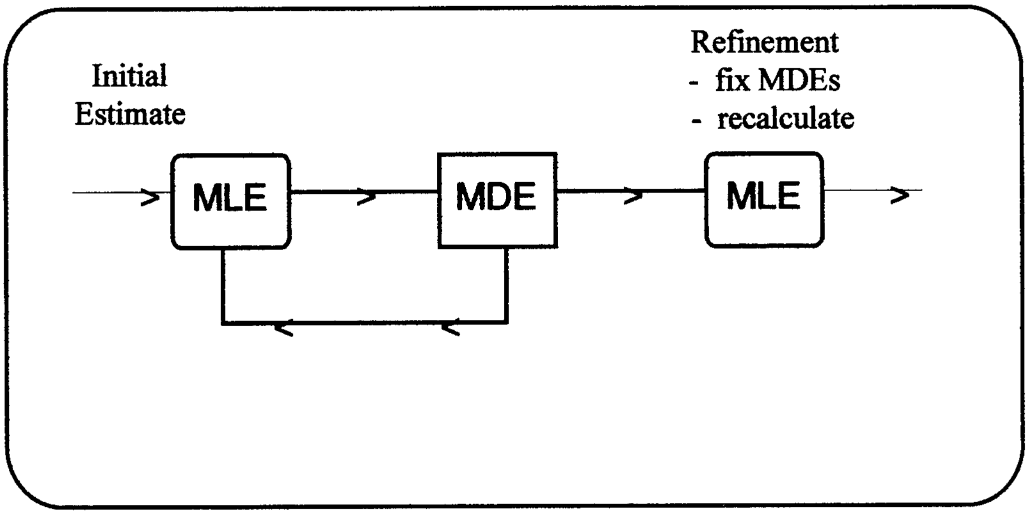
Method of Minimum Distance

Because parameters are selected to minimize the distance between a hypothesized distribution and the empirical distribution (distribution based on the sample observations or sample data), the name minimum distance estimation is given. The 'distance' is measured in several forms referred to as goodness-of-fit statistics. That is, the statistic quantifies the difference between the Empirical Distribution Function (EDF) and the hypothesized (estimated) Cumulative Distribution Function (CDF). The EDF is based on the sample, arranged in increasing order (ref Figure 4). The concept of minimum distance estimation is that better estimates will be obtained by fitting the distribution to the sample data. That is, the parameters of the estimated CDF are adjusted such that the smallest possible 'distance' remains between it and the data.

While applying MDE, one or more parameters can be estimated. This technique has also evolved to include simultaneous estimation of more than one parameter and sequential estimation of one or more parameters. Based on previous research on the Weibull distribution (Gallagher, 1990), the most promising parameters to estimate via minimum distance were the location parameter (δ_i) and the mixing proportion, p. In general, MDE assumes that an initial point estimate is available. The more accurate the point estimate, the better the MDE. Primarily due to their desirable asymptotic properties, this research assumed that the MLEs provided the best initial point estimate. Hence, MLEs were used as initial point estimates for the MDEs.

Given the MLE, a line search was conducted in the direction of decreasing "goodness of fit" statistic to find an interval that contained the minimum. Within this interval, the "golden search

(GS)" algorithm was applied for the same reasons as outlined by Gallagher (Gallagher, 1985: pp.25-28). The goodness-of-fit statistics were assumed to be unimodal with respect to a specific parameter. Finally, for MDE based on one parameter of a multi-parameter distribution such as the Mixed Weibull, the final MDE is calculated by re-estimating the MLE with the derived MDE as a fixed parameter (reference Figure 5). This is known as a refinement step.



• Figure 5. Basic Minimum Distance Estimation Process

For the seven parameters of the Mixed Weibull, some parameter estimation methods required several iterations of the Basic Minimum Distance Estimation Process. In general, the number of iterations depends on the method selected (Table 1). For example, since there were two location parameters (one for each PDF), MDL required two iterations whereas MDE on the mixing proportion required only one iteration. Based on the basic MDE process, there are many variations on the same theme which were investigated. The key, however, to successful parameter estimation is the correct selection of "goodness of fit" statistics.

Robustness and Goodness-of-Fit

In this context, the EDF is a step function, calculated from the sample, which estimates the population distribution function. EDF statistics are measures of the discrepancy between the EDF and a given distribution function, and are used for testing the fit of the sample to the distribution. In this research, the type of distribution was specified as Weibull, but the distribution contained parameters that were estimated from the sample.

"Suppose a given random sample of size n is $x_1, x_2, \ldots, x_n$ and let $x_{(1)}, x_{(2)}, \ldots, x_{(n)}$ be the order statistics; suppose further that the cumulative distribution function (CDF) of x is F(x). For the present and in most of this chapter we assume this distribution to be continuous. The empirical distribution function (EDF) is $F_n(x)$ defined by

$$F_n(x) = (\text{number of observations} \leq x) / n \,; \quad -\infty < x < \infty$$

More precisely, the definition is

$$F_n(x) = 0, \qquad x < x_{(1)}$$

$$F_n(x) = i/n, \qquad x_{(i)} \leq x < x_{(i+1)}, \qquad i = 1, \ldots, n-1$$

$$F_n(x) = 1, \qquad x_{(n)} \leq x$$

Thus $F_n(x)$ is a step function, calculated from the data. As x increases, the EDF takes a step up of height 1/n as each sample observation is reached. For any x, $F_n(x)$ records the proportion of observation less than or equal to x. We can expect $F_n(x)$ to estimate F(x), and it is in fact a consistent estimator of F(x). As $n \rightarrow \infty$, $| F_n(x) - F(x) |$ decreases to zero with probability one." (D'Agostino and Stephens, 1986: p97-98)

The goodness-of-fit statistics measure the discrepancy between two distributions. Recall that for MDE, the distributions utilized were the estimated CDF and the EDF. The measured difference between them constitutes the 'statistic'. Two were selected for this research based on their previous success: the Cramer-Von Mises and the Anderson-Darling test statistics.

Cramer-von Mises. Let K and L represent two cumulative distributions and 'W' a weighting function, the theoretical CVM is equation (5) (Parr and Suchany, 1984:616):

$$\bullet\ CVM(K,L) = \int_{-\infty}^{\infty} [K(x) - L(x)]^2 * W(L(x))dL(x) \qquad \text{Equation (5)}$$

where the weighting function is a constant equal to one, W(x) = 1

When the weighting function is a constant equal to one, W(x) = 1, the CVM formula becomes the Cramer-von Mises statistic. When K(x) is an empirical distribution, the computational formula is equation (6) where $F(x_{(i)})$ is the estimated distribution (Stephens, 1980):

$$CVM = \sum_{i=1}^{n} [z_i - (2i - 1) / 2n]^2 + (1 / 12n) \qquad \text{Equation (6)}$$

where $z_i = F(x_{(i)})$ for i = 1,2, …, n

The Anderson-Darling statistic is derived by increasing the weights for the distribution tails as in equation (7) (Anderson and Darling, 1954: p 767):

$$W(x) = 1 / [F(X) * (1 - F(x)] \qquad \text{Equation (7)}$$

The computational formula is equation (8) (Stephens: p731)

$$AD = (-1 / n) \sum_{i=1}^{n} (2i - 1)[\ln(z_i) + \ln(1 - z_i)] - n \qquad \text{Equation (8)}$$

where $z_i = F(x_{(i)})$ for i = 1,2, …, n

Method of Maximum Likelihood

The method of maximum likelihood selects as distribution parameters those values that maximize the likelihood function of the observed sample. The likelihood function of the sample is described by the joint density function (Mendenhall, Wackerly, and Schaeffer:p362). Therefore, the probability of the observed sample is maximized by the choice of the distribution parameter values. That is, choose as estimates those values of the parameters that maximize the likelihood of the sample. The likelihood of the sample, L = $L(x_1,x_2,\ldots,x_n)$, is defined to be the joint density of $x_1,x_2,\ldots,x_n$. (Mendenhall, Wackerly, and Schaeffer, p402). Since the natural logarithm of L is a monotonically increasing function of L, both L and the natural logarithm of L are maximized by the same parameter values.

a. Background

Let $f = \{f_j(x;\theta_j), j = 1,2,\ldots K\}$ (where K is the number of Weibull distributions represented in a mixture) be a family of probability distribution functions (PDFs) where $\theta_j = (\beta_j, \delta_j, \eta_j)$ denotes the parameter vector. Recall the Weibull probability density function (PDF):

$$f_j(x;\theta_j) = \left(\frac{\beta_j}{\eta_j}\right)\cdot\left(\frac{x-\delta_j}{\eta_j}\right)^{\beta_j-1}\cdot\exp\left[-\left(\frac{x-\delta_j}{\eta_j}\right)^{\beta_j}\right] \qquad \text{repeat Eqn (1)}$$

$(x > \delta_j;\ \text{and}\ \beta_j,\delta_j,\eta_j > 0)$

where θ_j is a vector of parameters $(\beta_j, \eta_j, \delta_j)$

Also, recall the Mixed Weibull is formally expressed as mixture where each PDF is weighted by the mixing proportion, p:

$$g(x;\alpha) = \sum_{j=1}^{2} p_j f_j(x;\theta_j) \qquad \text{repeat Eqn (2)}$$

where $\alpha = (\theta_1, \theta_2, p)$ and where $\sum_{j=1}^{2} p_j = 1$

Also, recall the Cumulative Density Function (CDF):

$$F(x) = 1 - \exp(-((x - \delta)/\eta)^{\beta}) \qquad \text{repeat Eqn (3)}$$

Therefore, the mixed CDF is

$$G(x, \theta) = \sum_{j=1}^{2} p_j F_j(x;\theta) \qquad \text{repeat Eqn (4)}$$

Thus, $G(x; \theta)$ is obtained as a convex combination of the subpopulation CDFs $\{F(x; \theta_j)\}$ with mixing proportions given by the vector p. For the complete sample case, the log-Likelihood function is expressed as

$$LL(\alpha) = \sum_{i=1}^{N} \ln g(x_i; \alpha) \qquad \text{Equation (9)}$$

where $\alpha = (\theta_1, \theta_2, p)$, N denotes sample size, and x_i is the i^{th} observation.

b. Formal Statement of the Problem

We now wish to find the values of α that maximize LL. Consistent with classical optimization methods, the algorithm maximizes LL by finding the gradient and solving for the unknown parameters by setting the gradients equal to zero (i.e. finding the roots). Recall that the gradient is the first derivative that represents the slope of any function. Those parameter values that result from setting the functions slope equal to zero are the roots of the function. Geometrically, we refer to such a critical point as the maximum or minimum depending on the convexity or concavity of the function, respectively. Since the natural logarithm of L (ln L) is a monotonically increasing function of L, both ln L and L will be maximized (Mendenhall, 1990:p402). "In estimation problems related to mixtures, one has to take into account a set of constraints in addition to the objective function. That is, mixing proportions have to lie between 0 and 1, and there may exist other constraints related to the parameters of subpopulations. It is observed that the constraints are generally of a linear type, and hence the MLE problem can be formulated as a mathematical programming problem with non-linear objective function and linear constraints (Kaylan and Harris, 1982):"

$$\max LL(\alpha)$$

$$\alpha \in S$$

$$\text{where } S = \{\alpha | \sum_{j=1}^{K} p_j = 1, \alpha >= 0) \qquad \text{Equation (10)}$$

c. Solution Approach

As Kaylan and Harris (1982) noted for the five parameter Mixed Weibull, the problem of obtaining the MLEs for the parameters in mixture models presents considerable difficulty due to the complexity of the likelihood function (objective function, equation 10). Kaylan and Harris were successful using a common rule of substitution which could not be extended to the seven parameter objective function (likelihood function, Eqn 10). Thus, the MLE was solved as a non-linear constrained optimization problem using a FORTRAN 77 based IMSL subroutine.

The initial outstanding issue with this approach is the fact that there is no global maximum. Even for the five parameter Mixed Weibull, there exists multiple local maximum for the likelihood function (Kaylan and Harris, 1981). As Redner and Walker (1984) pointed out, there is currently no adequate, efficient and reliable way of systematically determining all local maximum. Fortunately, regardless of the number of local minima, most problems can be solved correctly if an adequate initial guess is provided. The greater the number of local maxima, the greater the demand for a more accurate initial guess. In theory, one desires to reduce the number of local maxima which allows a less accurate guess to converge to the correct solution. This is particularly true in the stochastic environment.

Obviously, transition to a seven parameter log-likelihood function makes the response surface more complex (Figure 6). However, based on observations from the response surface and contour plots, the response surface for the seven parameters did not change from the response surface for five parameters if and only if a conditional statement was employed as discussed below.

The conditional statement is consistent with traditional probability theory in that there is no such thing as a negative probability. For the Weibull distribution, this translates to data values less than the location parameter. Those data values less than the location parameter (usually near the first order statistic for a single distribution) are undefined and need to be excluded in the calculation of the likelihood function and the gradient particularly in the calculation of the MLEs. This is standard practice with a single distribution. In mathematical terms, this is known as defining the proper interval for the function of interest. What is unique to the objective function of the mixed distribution is the fact that the interval is undefined for different parts of the objective function (hence, related gradient) because the objective function is composed of two probability density functions. Thus, equation (9) is modified slightly to look like equation (11):

$$LL(\alpha) = \sum_{i=\delta_j}^{N} \ln [\, p\, f_1(x_i; \theta_1) + (1\text{-}p)\, f_2(x_i; \theta_2)] \qquad \text{Equation (11)}$$

where $\delta_j = \delta_1$ for PDF1 and $\delta_j = \delta_2$ for PDF2;

and δ_1 and δ_2 are the location parameters of PDF1 and PDF2, respectively.

After implementation of the conditional statement, the response surface was often unimodal with respect to each parameter except under extremely poor guesses (Figure 7). Hence, data values less than the location parameter are labeled as irrational values as opposed to rational values. This advantageous scheme may only be possible when the true gradient is used in the calculation of the MLE since only the true gradient equations allow exclusion of the values less than their respective location parameters.

Retaining irrational values results in a response surface that in many cases will not allow convergence or requires a highly accurate guess that typically is not available. Hence, a conditional statement was employed to exclude irrational values in the likelihood function to obtain a better response surface. Now, those previously established techniques developed to solve the MLE for the five parameter Mixed Weibull were applied successfully to the seven parameter Mixed Weibull. Specifically, this methodology assumed that a reasonable initial estimate was available. Mendelbaum (1982) recommended a graphical approach. By starting at this initial estimate, a quasi-Newton method was successfully applied.

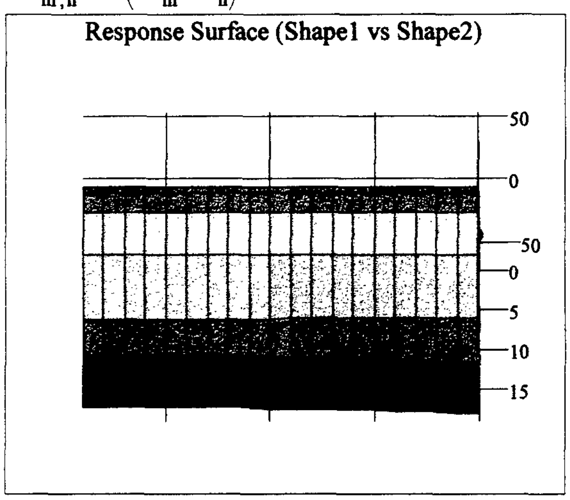
$m := 1..20 \quad n := 1..20 \quad B1_m := 2.0 + 0.1\cdot m \quad B2_n := 0.1 + 0.1\cdot n$

$p := .5 \quad D1 := 0 \quad E1 := 1 \quad N := 40 \quad x1 := \mathrm{rweibull}\left(\frac{N}{2}, 3\right)$

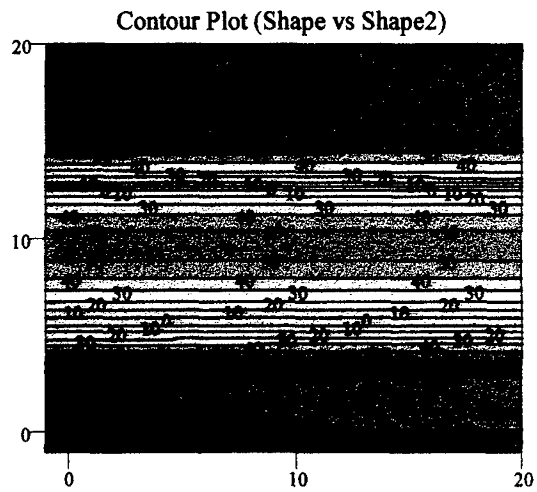
$D2 := 5.0 \quad E2 := 1 \quad x2 := \mathrm{rweibull}\left(\frac{N}{2}, 1\right) + 5.0$

$$L(B1,B2) := \sum_{i=1}^{N} \ln\left[\left[p\cdot\left[\frac{B1}{E1}\cdot\left[\left(\frac{x_i - D1}{E1}\right)^{B1-1}\right]\cdot\exp\left[-\left[\left(\frac{x_i - D1}{E1}\right)^{B1}\right]\right]\right]\right] \ldots \right.$$
$$\left. + \left[(1-p)\cdot\left[\frac{B2}{E2}\cdot\left[\left(\frac{x_i - D2}{E2}\right)^{B2-1}\right]\cdot\exp\left[-\left[\left(\frac{x_i - D2}{E2}\right)^{B2}\right]\right]\right]\right]\right]$$

$M2_{m,n} := L(B1_m, B2_n)$



M2



M2

Figure 6. Irrational Response Surface and Contour Plot

$N := 40 \qquad x1 := \text{rweibull}\left(\frac{N}{2}, 3\right) \qquad x2 := \text{rweibull}\left(\frac{N}{2}, 1\right) + 5.0 \quad x := \text{stack}(x1, x2)$

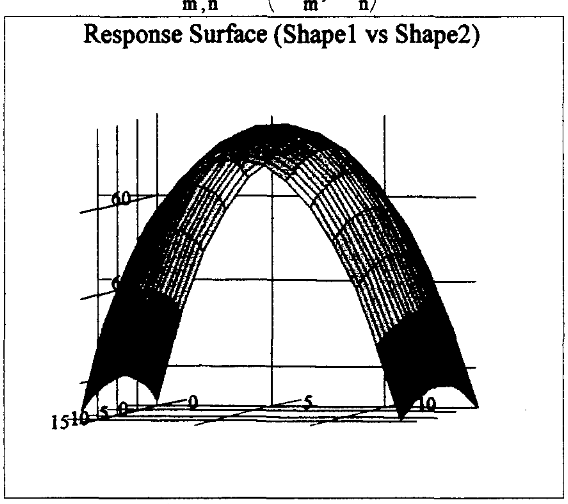
$m := 1..20 \qquad n := 1..15 \qquad B1_m := 2.0 + 0.1 \cdot m \qquad B2_n := 0.2 + 0.1 \cdot n$

$p := .5 \qquad D1 := 0 \qquad E1 := 1$

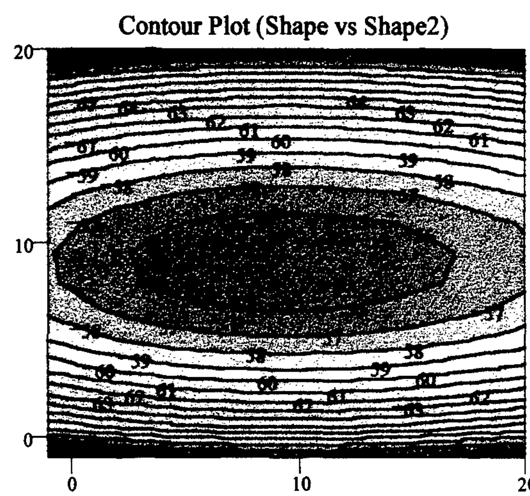
$\qquad\qquad D2 := 5.0 \qquad E2 := 1$

$$L(B1,B2) := \sum_{i=1}^{N} \ln\left[\left[p \cdot \left[\frac{B1}{E1} \cdot \left[\left(\frac{x_i - D1}{E1}\right)^{B1-1}\right] \cdot \exp\left[-\left[\left(\frac{x_i - D1}{E1}\right)^{B1}\right]\right]\right]\right] \ldots + \begin{cases} 0 & \text{if } x_i < D2 \\ \left[(1-p) \cdot \left[\frac{B2}{E2} \cdot \left[\left(\frac{x_i - D2}{E2}\right)^{B2-1}\right] \cdot \exp\left[-\left[\left(\frac{x_i - D2}{E2}\right)^{B2}\right]\right]\right]\right] & \text{otherwise} \end{cases}\right]$$

$M2_{m,n} := L\left(B1_m, B2_n\right)$



M2



M2

Figure 7. Rational Response Surface and Contour Plot

d. The Gradient of the Likelihood Function

We now discuss the development of an efficient algorithm to solve the maximization problem. For the sake of simplicity, we henceforth employ f_{ij}, g_i, and LL instead of $f_j(x_i; \beta_j, \eta_j, \delta_j)$, $g_j(x_i; \alpha)$ and LL(α), respectively. Taking the gradient of ln L, the following equations are obtained:

$$\partial LL/\partial\beta_j = \sum_{i=1}^{N} (1/g_i)(p_j)(\partial f_{ij}/\partial\beta_j) \qquad (j = 1, 2)$$

$$\partial LL/\partial\eta_j = \sum_{i=1}^{N} (1/g_i)(p_j)(\partial f_{ij}/\partial\eta_j) \qquad (j = 1, 2) \qquad \text{Equation (11)}$$

$$\partial LL/\partial\delta_j = \sum_{i=1}^{N} (1/g_i)(p_j)(\partial f_{ij}/\partial\delta_j) \qquad (j = 1, 2)$$

$$\partial LL/\partial p_j = \sum_{i=1}^{N} (1/g_j)(f_{ij} - f_{ik}) \qquad (j = 1, 2, \ldots K\text{-}1)$$

e. Description of the Quasi-Newton IMSL subroutine

The following is a synopsis of the algorithm (IMSL Manual, Math Library, Minimization with Simple Bounds, subroutine DBCONG, 1990): " The algorithm used a quasi-Newton method and an active set strategy to solve maximization subject to simple bounds on the variables. From a given starting point (x^c), an active set (IA), which contains the indices of the variables at their bounds is built. A variable is called a 'free variable' if it is not in the active set. The routine then computes the search direction for the free variables according to the formula

$$d = -B^{-1} g^{c}$$

where B is a positive definite approximation of the Hessian, and g^c is the gradient evaluated at x^c; both are computed with respect to the free variables. The search direction for the variables in IA is set to zero. A line search is used to find a new point x^n,

$$x^n = x^c + \lambda d, \quad \lambda \in (0,1)$$

such that

$$f(x^n) \leq f(x^c) + \alpha g^T d, \quad \alpha \in (0,0.5).$$

Finally, the optimality conditions are checked:

$$\| g(x_i) \| \leq \epsilon, \quad l_i < x_i < u_i$$

$$g(x_i) < 0, \quad x_i = u_i$$

$$g(x_i) > 0, \quad x_i = l_i$$

where ϵ is a gradient tolerance

When the optimality is not achieved, B is updated according to the following formula:

$$B \leftarrow B - [(Bss^T B) / (s^T Bs)] + [(yy^T)/(y^T s)]$$

where $s = x^n - x^c$ and $y = g^n - g^c$. Another search direction is then computed to begin the next iteration.

The active set is changed only when a free variable hits its bounds during an iteration, or the optimality condition is met for the free variables but not for all variables in IA, the active set. In the latter case, a variable which violates the optimality condition will be dropped out of IA. For more details on the quasi-Newton method, see Dennis and Schnabel (1983). For more detailed information on an active set strategy, see Gill and Murray (1976)."

Evaluation Criteria

To evaluate the nine parameter estimation methods, Monte Carlo simulations were conducted. The extreme cases (selected for this research) for the Mixed Weibull are the following: 1) the case where there exists a large amount of overlap between the two populations; and 2) the case where there is no overlap. In the former case (Figure 8. Non-separated (NS) Mixed Weibull), the maximum overlap occurs when the two populations share the same location parameter and a common shape or a common scale parameter. The non-separated (NS) case was evaluated where the populations shared a common location parameter ($\delta_1 = \delta_2 = 5$) and a common scale parameter ($\eta_1 = \eta_1 = 0.5$). Of course, then the two populations must have different shape parameters (e.g. $\beta_1 = 4$, $\beta_2 = 1$).

$p := 0.5 \qquad \beta_1 := 4 \quad \beta_2 := 1 \qquad \delta_1 := 5 \quad \delta_2 := 5 \qquad \eta_1 := 0.5 \quad \eta_2 := 0.5$

$$f1(x) := \left[\frac{\beta_1}{\eta_1}\cdot\left[\left(\frac{x-\delta_1}{\eta_1}\right)^{\beta_1-1}\right]\cdot\exp\left[-\left(\frac{x-\delta_1}{\eta_1}\right)^{\beta_1}\right]\right] \qquad f2(x) := \left[\frac{\beta_2}{\eta_2}\cdot\left[\left(\frac{x-\delta_2}{\eta_2}\right)^{\beta_2-1}\right]\cdot\exp\left[-\left(\frac{x-\delta_2}{\eta_2}\right)^{\beta_2}\right]\right]$$

$$g(x) := ((p)\cdot f1(x)) + ((1-p)\cdot f2(x))$$

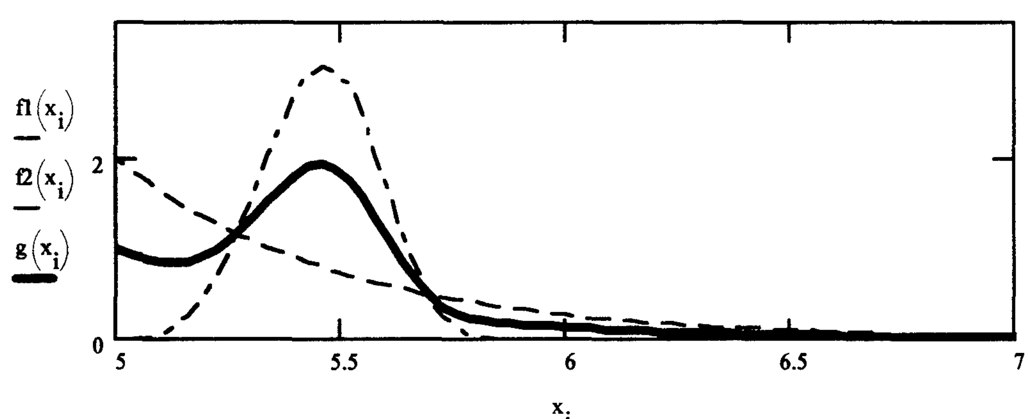


Figure 8. Non-separated (NS) Mixed Weibull

In the latter case (Figure 9. Well-separated, WS, for the Mixed Weibull), the Mixed Weibull can be thought of as a combination of two independent Weibull populations. For a single Weibull, the MLEs are scale and location invariant (Antle, p251). That is, the MLEs are equal variant with respect to the scale and location parameter. Therefore, only one set of location ($\delta_1 = 5, \delta_2 = 5$) and scale ($\eta_1 = \eta_2 = 0.5$) parameters were used for the well-separated cases. The shape for the well-separated (WS) populations were tested at two levels ($\beta_1 = \beta_2 = 3$ and $\beta_1 = \beta_2 = 0.9$). All variations were evaluated at four sample sizes (n = 10, 20, 40, 100).

$p := 0.5 \qquad \beta_1 := 3 \quad \beta_2 := 3 \qquad \delta_1 := 5 \quad \delta_2 := 10 \qquad \eta_1 := 0.5 \quad \eta_2 := 0.5$

$$f1(x1) := \left[\frac{\beta_1}{\eta_1} \cdot \left[\left(\frac{x1 - \delta_1}{\eta_1} \right)^{\beta_1 - 1} \right] \cdot \exp\left[-\left(\frac{x1 - \delta_1}{\eta_1} \right)^{\beta_1} \right] \right] \qquad f2(x2) := \left[\frac{\beta_2}{\eta_2} \cdot \left[\left(\frac{x2 - \delta_2}{\eta_2} \right)^{\beta_2 - 1} \right] \cdot \exp\left[-\left(\frac{x2 - \delta_2}{\eta_2} \right)^{\beta_2} \right] \right]$$

$$g(x) := ((p) \cdot f1(x1)) + ((1 - p) \cdot f2(x2))$$

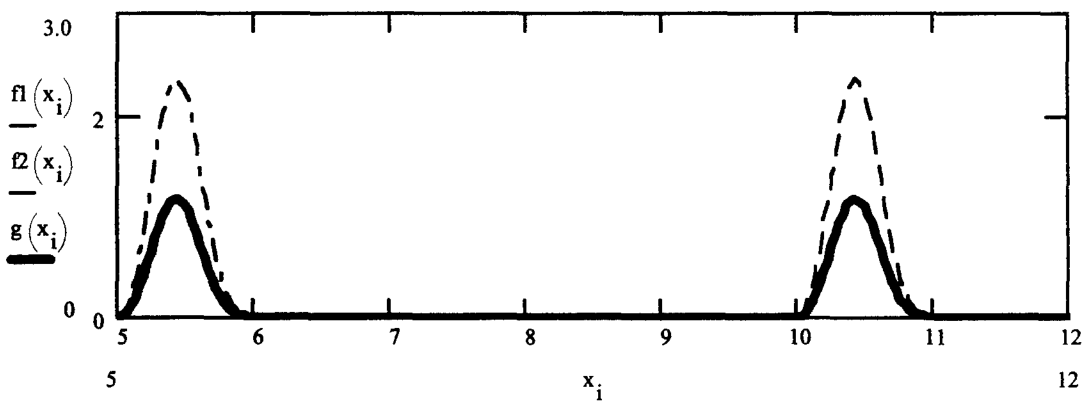


Figure 9. Well-Separated (WS) Mixed Weibull

The primary criteria falls in one general category, population distribution criteria. The population distribution criteria showed how well the estimated distribution matched the TRUE distribution usually measured in terms of the mean square error (MSE). Within the general category are two measures: 1) the integrated mean square error; and 2) the integrated absolute error. Also, for each of these criteria, the percentage of times each estimation technique was better than the initial MLE was calculated.

Reference to the TRUE parameters means the parameters used to generate the data sample. The TRUE parameters are the population parameters that statistics are attempting to make inferences about. EST_i represents a set of parameter estimates obtained by one of the six methods for the ith data set.

Integrated Absolute Difference Between CDFs. This measure shows the "area" between the estimated CDF and the TRUE CDF. The numerical integration algorithm used was Gauss-Legendre Quadrature. This algorithm worked in three steps: the integral was transformed to the range [-1, 1], evaluated at the roots of the Legendre polynomial, and then the weighted summation of evaluations was calculated. The transformation was Equation (12) (7 : 168). The transformation follows:

$$\int_a^b f(x)dx = (1/2)(b-a)\int_{-1}^{+1} f[(1/2)(b-a) * t + b + a)]dt \qquad \text{Equation (12)}$$

where a was the lower of the two location parameters and b was such that both CDF values exceed 0.999

A 48 degree Legendre polynomial was used with roots and weights as listed in Handbook of Mathematical Functions (1 : 917). The function [absolute values of [(F (TRUE, x) - F (EST, x))] was calculated at each of the roots. The numerical evaluation was completed by applying Gauss' Formula, Equation (13) (1: 887).

$$\int_{-1}^{+1} f(x)dx = \sum_{i=1}^{48} w_i f(x_i) \qquad \text{Equation (13)}$$

Since the objective of statistical estimation is to predict the population distribution from which the sample came, this criterion was a true measure of success.

Integrated Squared Difference Between CDFs. This test approximated the theoretical Cramer-von Mises statistic. The same Gauss-Legendre Quadrature numerical integration was used. Instead of "absolute value" between CDFs as used previously, the "squared" difference was used.

Percentage of Times Better

The final evaluation criteria were the percentages of times that the minimum distance estimates were "better" than the initial MLE, where "better" was determined by the previous criteria. This criteria ensured that a few extremely poor estimates could not skew the results against a generally good estimation technique.

Executive Overview

An overview of the methods is presented in Tables 1a,b,c. For several reasons (see justification), slightly different methods were used based on the nature of the data (for a description of non-separated (NS) versus a well-separated (WS) Mixed Weibull see Figures 8 and 9, respectively, Evaluation Criteria).

Table 1a. Basic Parameter Estimation Cycles

MLE	Maximum Likelihood Estimates
MDE	Minimum Distance Estimates
a. Minimum Distance via the Location Parameters, δ	
MDLA	The Minimum Distance for the location parameters using the Anderson-Darling "goodness-of-fit" statistic and maximum likelihood estimates for other parameters (sequentially, δ_1 then δ_2).
MDLC	The Minimum Distance for the location parameter using the Cramer -Von Mises "goodness-of-fit" statistic and maximum likelihood estimates for other parameters (sequentially, δ_1 then δ_2).
MDLSA	The Minimum Distance for the location parameters using the Anderson-Darling "goodness-of-fit" statistic and maximum likelihood estimates for other parameters (simultaneously).
MDLSC	The Minimum Distance for the location parameter using the Cramer -Von Mises "goodness-of-fit" statistic and maximum likelihood estimates for other parameters (simultaneously).
b. Minimum Distance via the Mixing Proportion, p	
MDPA	The Minimum Distance for the mixing proportion using the Anderson-Darling "goodness-of-fit" statistic and maximum likelihood estimates for other parameters.
MDPC	The Minimum Distance for the mixing proportion using the Cramer -Von Mises "goodness-of-fit" statistic and maximum likelihood estimates for other parameters.

* Table 1b. Non-Separated Mixed Weibull Parameter Estimation Methods

[1]MLE (1)	Maximum Likelihood Estimation
MDE	Minimum Distance Estimation
	a. Minimum Distance via the Mixing Proportion, p
MDPA (2)	Minimum Distance for the mixing parameter using the Anderson-Darling "goodness-of-fit" statistic and maximum likelihood estimates for other parameters.
MDPC (3)	The Minimum Distance for the mixing parameter using the Cramer - Von Mises "goodness-of-fit" statistic and maximum likelihood estimates for other parameters.
	b. Minimum Distance via Location & Mixing Proportion, p
MDPL1A (4)	Reverse order - Sequentially combined methods (MDPA, MDL1A)
MDPL1C (5)	Reverse order - Sequentially combined methods (MDPC, MDL1C)
MDPL2A (6)	Reverse order - Sequentially combined methods (MDPA, MDL1A, MDL2A)
MDPL2C (7)	Reverse order - Sequentially combined methods (MDPC, MDL1C, MDL2C)
MDPLSA (8)	Reverse order - Simultaneously combined methods (MDPA, MDLSA)
MDPLSC (9)	Reverse order - Simultaneously combined methods (MDPC, MDLSC)

* Note: Reference Appendix D for some Single-Run Samples

* Table 1c. Well-Separated Mixed Weibull Parameter Estimation Methods

MLE (1)	Maximum Likelihood Estimation
MDE	Minimum Distance Estimation
	a. Minimum Distance via the Mixing Proportion, p
MDPA (2)	Minimum Distance for the mixing proportion using the Anderson-Darling "goodness-of-fit" statistic and maximum likelihood estimates for other parameters.
MDPC (3)	The Minimum Distance for the mixing proportion using the Cramer -Von Mises "goodness-of-fit" statistic and maximum likelihood estimates for other parameters.
	b. Minimum Distance via Location
MDLSA (4)	The Minimum Distance for the location parameters using the Anderson-Darling "goodness-of-fit" statistic and maximum likelihood estimates for other parameters (simultaneously).
MDLSC (5)	The Minimum Distance for the location parameter using the Cramer -Von Mises "goodness-of-fit" statistic and maximum likelihood estimates for other parameters (simultaneously).
	c. Minimum Distance via Location & Mixing Proportion, p
MDLSPA (6)	Sequentially combined methods (MDLSA, MDPA)
MDLSPC (7)	Sequentially combined methods (MDLSC, MDPC)
MDPLSA (8)	Reverse order - Sequentially combined methods (MDPC, MDLSA)
MDPLSC (9)	Reverse order - Sequentially combined methods (MDPC, MDLSC)

* Note: Reference Appendix D for some Single-Run Samples

Justification for Selection of Methods

Based on Gallagher's research with the three parameter Weibull (1991), Minimum Distance Estimation was applied to several of the most promising parameters for the Mixed Weibull. Also, several alpha tests were conducted to determine the optimal methods. Not surprisingly, the results fell into two main groups, the non-separated (NS) and the well-separated (WS) Mixed Weibull. That is, the most promising methods for the well-separated distributions did not hold for the non-separated distributions. Hence, the divergence between Table 1b and Table 1c (for a complete discussion, see Minimum Distance Estimation (MDE) Conclusions). This is due in part to the nature of data and the method used to solve the MLE (reference MLE Methodology).

The alpha testing results indicated that for well-separated distributions, the mixing proportion was calculated with a significantly higher degree of accuracy. Obviously, the research needed to determine if calculating the mixing proportion via minimum distance (MDP) was beneficial. Also, the research needed to determine the best course of action for calculating the location parameter via minimum distance (MDL), before or after conducting MDP. In the well-separated cases, there were two reasons that led to a simultaneous estimation of the location parameters via minimum distance (MDLS). First, the distributions are completely independent. And, second performing them separately had a tendency to destroy an otherwise accurate mixing proportion.

In brief, the response surface for the likelihood function required a large sample size even for the well-separated distribution where the mixing proportion could be determined by a simple visual inspection. Hence, for the non-separated distributions, it was never more advantageous to calculate the location parameters (MDL) without first calculating the mixing proportion via minimum distance (MDP). However, there remained the goal of comparing the sequential and simultaneous calculation of the location parameters (MDL and MDLS, respectively).

Estimation Techniques

Regardless of the estimation method, the parameters were estimated using one of two techniques. Recall that the Mixed Weibull has seven parameters essentially representing the standard parameters for two probability density functions and one mixing proportion, p. For the MLE, a FORTRAN 77 based IMSL (International Mathematical and Statistical Library) subroutine was used (reference Methodology for MLE). Since minimum distance was applied to only one or two parameters at once (i.e. in conjunction with MLEs) a detailed breakdown of the estimation technique per parameter is presented for the core methods in Table 2.

• Table 2. Estimation Techniques per parameter

	δ_1	β_1	η_1	p	δ_2	β_2	η_2
	LOCATION	SHAPE	SCALE	MIX	LOCATION	SHAPE	SCALE
CORE							
MLE	Max Likelihood Estimation (MLE)						
MDL	MDE (GS)	MLE given δ			MDE (GS)	MLE given δ	
MDP	MLE given p			MDE (GS)	MLE given p		

IV. Mixed Weibull Results

To evaluate the performance of the parameter estimation techniques, Monte Carlo simulations were conducted. That is, for a true population with predetermined parameters as described below, 1000 random samples were generated. Confidence intervals for the smallest samples were calculated (reference Appendix B). For each sample, the estimates were calculated and the resulting parameters were evaluated with the criteria in Chapter 3. Since the Weibull MLEs are location and scale invariant, the well-separated distributions were tested at two different shape parameters. The following scenarios were tested:

- Three different mixing proportions (p = 0.5, 0.3, 0.1)

- Non-separated versus well-separated distributions

    -- Non-separated distributions (same location parameter, $\delta_1 = \delta_2 = 5$)

        --- one family of the shape parameter ($\beta_1 = 4$ and $\beta_2 = 1$)

        --- same scale parameter ($\eta_1 = \eta_2 = 0.5$)

    -- Well-separated distribution ($\delta_1 = 5$ and $\delta_2 = 10$)

        --- two families of the shape parameter ($\beta_1 = \beta_2 = 0.9$ and $\beta_1 = \beta_2 = 3$)

        --- same scale parameter ($\eta_1 = \eta_2 = 0.5$)

The above variations were each evaluated at four sample sizes (n = 10, 20, 40, 100).

Selected nominal-error samples are presented in Appendix A using the most successful method from this research (minimum distance via the mixing proportion using the Cramer-von Mises statistic, MDPC). Aggregated results are presented in Appendix B based strictly on one common measure of error (Integrated Absolute Difference) allowing the reader to readily compare the 288 scenarios (a scenario is defined as one method at one sample size under one set of true parameters). More detailed results for each scenario are presented in Appendix C including two measures of error including confidence intervals and several sub-totals on the number of times better than the MLE. Tables 3, 4 and 5 on the following page shows the best estimators by CDF comparison including the method that produced the best result (highest percentage) and the percentage of times better than the Maximum Liklihood Estimates (MLE).

Stochastic Generation of the Mixing Proportion

As discussed in the Methodology, the mixing proportion was generated stochastically during each monte carlo run. The mixing proportion (as dictated by reliability theory) is uniformly distributed from zero to one, $p \sim U(0,1)$. Specifically, a random sample was generated from a uniform (0,1) distribution. The number of uniform random samples that exceeded (did not exceed) the true mixing proportion dictated the number of Weibull samples generated from each population. Therefore, the proportion of the actual number of samples generated and the true mixing proportion were often not the same.

Minimum Distance Estimation (MD) Results

Consistent with the executive overview (including acronyms), MDE results are presented in Tables 3,4 and 5 based on Appendix C. Table 3 show results for non-separated (NS) populations. Tables 4 and 5 show results for the well-separated (WS) populations where the only difference is the true shape parameter. These results are a summarization in terms of the best estimation method for a given scenario and in terms of estimating a better set of parameters than the Method of Maximum Likelihood. Further, the results are grouped by mixing proportion and, finally, by sample size.

Table 3 Best Estimators by CDF Comparison for Non-Separated Populations

NS	MIXING PROPORTION	N	INTEGRATED ABSOLUTE DIFFERENCE %		INTEGRATED SQUARED DIFFERENCE %	
	P = 0.5	10	MDPC MDPC	93.2 93.2	MDPC MDPC	93.2 93.2
		20	MDPA MDPA	98.1 98.1	MDPA MDPA	98.1 98.1
		40	MDPC MDPC	99.1 99.1	MDPC MDPC	99.1 99.1
		100	MDPC MDPC	99.0 99.0	MDPC MDPC	99.0 99.0
	P = 0.3	10	MDPC MDPLA	53.4 55.2	MDPC MDPLA	53.4 55.2
		20	MDPC MDPC	58.8 58.8	MDPA MDPA	55.6 55.6
		40	MDPC MDPA	54.1 57.7	MDPC MDPA	54.1 57.7
		100	MDPC MDPA	52.4 56.3	MDPA MDPA	56.3 56.3
	P = 0.1	10	MDPA MDPLA	46.4 53.5	MDPA MDPLA	46.4 53.5
		20	MDPC MDPLSA	44.3 50.9	MDPA MDPA	43.2 43.2
		40	MDPA MDPA	40.9 40.9	MDPA MDPLC	41.8 45.7
		100	MDPA MDPLSA	40.3 44.6	MDPA MDPA	42.8 42.8

Note: The first rows of techniques were the best by that columns criteria. The second rows were the techniques that were better than MLE most often. The percentage criteria was better, not equal.

Table 4. Best Estimators by CDF Comparison for Well-Separated Populations (Shape = 3)

WS $\beta_1 = 3$	MIXING PROPORTION	N	INTEGRATED ABSOLUTE DIFFERENCE %		INTEGRATED SQUARED DIFFERENCE %	
	P = 0.5	10	MDPC MDPC	82.8 82.8	MDPC MDPC	82.8 82.8
		20	MDPC MDPC	87.9 86.9	MDPC MDPC	87.9 86.9
		40	MDPC MDPC	89.0 89.0	MDPC MDPC	89.0 89.0
		100	MDPC MDPC	94.7 94.7	MDPC MDPC	94.7 94.7
	P = 0.3	10	MDPC MDPC	50.2 50.2	MDPA MDPA	41.6 41.6
		20	MDPC MDPLSC	29.0 37.8	MDPC MDPLSC	29.0 37.8
		40	MDPC MDLSPC	17.6 44.0	MDPC MDLSPC	17.6 44.0
		100	MDLA MDLA	39.6 39.6	MDLA MDLA	39.6 39.6
	P = 0.1	10	MDLSC MDLSPA	33.7 61.4	MDPA MDPA	60.9 60.9
		20	MDLSC MDLSPA	27.4 53.6	MDPC MDPC	44.2 44.2
		40	MDLSA MDPC	21.4 21.4	MDLSPA MDPLSA	21.4 34.2
		100	MDLSA MDLSA	28.4 28.4	MDLSA MDLSA	28.4 28.4

Note: The first rows of techniques were the best by that columns criteria. The second rows were the techniques that were better than MLE most often. The percentage criteria was better, not equal.

Table 5. Best Estimators by CDF Comparison for Well-Separated Populations (Shape = 0.9)

WS $\beta_I =$ 0.9	MIXING PROPORTION	N	INTEGRATED ABSOLUTE DIFFERENCE %		INTEGRATED SQUARED DIFFERENCE %	
	P = 0.5	10	MDPA	86.3	MDPA	86.3
			MDPA	86.3	MDPA	86.3
		20	MDPC	82.0	MDPC	82.0
			MDPC	82.0	MDPC	82.0
		40	MDPC	87.1	MDPC	87.1
			MDPC	87.1	MDPC	87.1
		100	MDPC	84.6	MDPC	84.6
			MDPC	84.6	MDPC	84.6
	P = 0.3	10	MDPA	41.2	MDPA	41.2
			MDPA	41.3	MDPA	41.3
		20	MDPC	45.9	MDPC	45.9
			MDPC	45.9	MDPC	45.9
		40	MDPC	43.1	MDPC	43.1
			MDPA	45.9	MDPA	45.9
		100	MDPA	51.9	MDPA	51.9
			MDPLSC	55.4	MDPLSC	55.4
	P = 0.1	10	MDPLSA	58.0	MDPLSA	58.0
			MDLSA	65.4	MDLSA	65.4
		20	MDLSC	61.1	MDLSC	61.1
			MDLSC	61.1	MDLSC	61.1
		40	MDPA	52.0	MDPC	46.3
			MDPA	52.0	MDPC	46.3
		100	MDPA	49.1	MDPA	49.1
			MDPA	49.1	MDPA	49.1

Note: The first rows of techniques were the best by that columns criteria. The second rows were the techniques that were better than MLE most often. The percentage criteria was better, not equal.

Maximum Liklihood Estimation (MLE) Results

Appendix B was specifically drafted to demonstrate the phenomenal success of estimating parameters for the seven parameter Mixed Weibull (See Conclusions for the MLE). Based strictly on our ability to estimate parameters quickly and accurately, these results set a precedent and make a strong argument for a permanent transition to non-linear techniques for all mixed distributions. Within this context, there is existing precedence for using gradient information as opposed to some approximation such as finite difference or least squares.

V. Conclusions

The net error after applying the recommended methods from Tables 3, 4 or 5 is very reasonable with an excellent confidence interval even for sample sizes as low as ten (reference MDPC, Appendix C) . One can expect the highest error to occur when the mixing proportion is equal to one half. Based on MDPC (the overall best method), nominal error for each sample size (n = 10,20,40,100) is plotted (estimated versus true Mixed Weibull PDF and CDF) for both well-separated and non-separated results in Appendix A.

One should not be surprised to find some counter-intuitive results. For example, MDPC often estimated a mixing proportion closer to the true proportion than the actual sample generated when the sample size was small. Also, there appears to be a balancing act occurring between the shape parameters and mixing proportion with the net result being a good fit. That is, a poor estimate of the mixing proportion can be accommodated by one small and one large shape parameter. This trend was easier to detect in the well-separated scenarios, but probably occurred any time the mixing proportion was in error. In the end, the only way to assess the performance of the methodology is with large sample sizes

Since we utilized the "extreme" cases for the Mixed Weibull in terms of the completely separated and the non-separated populations, the results can be viewed as worst case. That is, the researcher can expect to generate results that have the same amount of error or less in their parameter estimates. Also, the researcher needs to remember that one sample in this context is really estimating two probability density functions. Since the mixing proportion was also generated stochastically, the total error is often greater than the sum of the error from two stochastically generated probability distribution functions. Also, one cannot segregate the error by simple subtraction because the estimated mixing proportion was used in the calculation of the estimates for the parameters of the probability density functions.

Total error was predominantly a function of the mixing proportion for several reasons. Due to the fact that both estimation processes were affected by the mixing proportion, the problem will be discussed in this section. For now, suffice it to say that once the mixing proportion was estimated accurately (as in MDP), the error usually dropped to a fraction of the original MLE. While there is no way to prove it from these results, the remaining error appears to be strictly due to the stochastic nature of the sample. This fact can be verified by the universal success of minimum distance estimation of the mixing proportion (MDPC or MDPA) even for sample sizes of ten. The exception being the highly overlapping (non-separated) distributions where there is naturally some minor additional error induced by the overlap in the distributions.

Results demonstrated few universal properties with the exception of the mixing proportion where error always reached a maximum at mixing proportion equal to one half. Conclusions varied for several reasons but, primarily depending on whether the two true populations were well-separated or not. New properties were observed for both the well-separated and the non-separated results both in terms of both the MDEs and the MLEs. In some cases, the results followed traditional expectations. For example, MLEs became significantly more accurate with increased sample size. Recall, however, that the motivation for Minimum Distance Estimation is to enhance parameter estimation for the small to moderate sample sizes where the asymptotic properties might not hold.

The dominance of the mixing proportion may give the false perception that the MLE methodology(non-linear constrained optimization) did not perform adequately. As indicated by the results (after the mixing proportion was adequately estimated), the numerical methodology was as good as any parameter estimation to date. The accuracy of this estimation was further enhanced by utilization of the gradient information. And, this method can be applied to most mixed distributions (especially a mixture of two distributions) where traditional MLE algorithms cannot be extended from single to a mixed distribution.

The Mixing Proportion

There are several reasons that the mixing proportion became a dominant issue. First, estimation of the mixing proportion was poor. For the non-separated distributions, estimation remained poor even in large sample sizes. Mixing proportion estimation for the well-separated distributions started out fair and enhanced with increasing sample size. But, even for the well-separated results where one could perform a simple head count, a moderate sample size was required to estimate the mixing proportion. And, sample sizes of one hundred still did not result in estimation of the mixing proportion as good as that obtained by minimum distance of the same parameter (MDPC). Second, the mixing proportion dominates the objective function (the likelihood function) greater than any other parameter. Third, the mixing proportion was generated stochastically in addition to the sample. Fourth, due to the nature of the evaluation criteria, the mixing proportion dominates the calculation of error. Finally, the estimated mixing proportion was used in the calculation of the other parameter estimates. Hence, if the stochastically generated or calculated mixing proportion was in error even a small amount, the estimation of the other parameters was disturbed (not so much in location, but in magnitude affecting more the shape and scale parameters). In summary, the calculated error was sensitive to changes in the mixing proportion simply due to the nature of the evaluation criteria.

Minimum Distance Estimation (MDE) Conclusions

Application of traditional theories in a multi-modal distribution required careful consideration. For instance, the sample size per distribution is not a function of the mixing proportion. Rather, each distribution is weighted according to the mixing proportion at the expense of the other distributions. Thus, the better the estimate of the mixing proportion, the better the estimate for MDL. And, application of MDL after a good estimate of the mixing proportion was always beneficial.

In general, MDE via the mixing proportion proved very effective. Ironically, however, while there were some documented successes with MDE via the location parameters (MDL), the net benefit was overshadowed by domination of the mixing proportion to minor deviations. As discussed, small errors in the mixing proportion typically resulted in large errors in the parameter estimation for the individual distributions. MDE is performed using the estimated parameters from the MLE. Hence, the greater the error in the MLEs, the greater the error in the MDEs.

Another problem in applying MDL to the Mixed Weibull was observed. If MDL was applied sequentially, the mixing proportion (and hence all other parameters) was definitely affected by the application to only one location parameter. In an attempt to "balance" the impact, MDL was conducted simultaneously to both location parameters. The problem is that the MLE that followed was estimated with one less degree of freedom (5 df) than when the location parameters are estimated sequentially (6 df). Theoretically, accuracy in the estimation is lost with a reduced number of degrees of freedom. This is a trade-off that needs continued vigilance in the future for any mixture.

By further analysis of the results, one can reach a myriad of worthless conclusions about the effectiveness of the different methods under different circumstances. While these areas warranted investigation, the results (Tables 3, 4, and 5) suggest utilizing the most successful method for a given scenario. Most often, this translates to a simple choice of minimum distance estimation of the mixing proportion using the Cramer-von Mises test statistic (MDPC). Surprisingly, this finding held for both the well-separated and the non-separated populations. The only exception to this rule is in the case of a large sample size and well-separated populations where, not surprisingly, the MLEs dominate.

Maximum Likelihood Estimation (MLE) Conclusions

These results set a precedent and make a strong argument for a permanent transition to (MLE) parameter estimation as a non-linear constrained optimization problem particularly in the mixed distribution context. Most often, the error measured and observed was due to the need for large sample sizes to estimate the mixing proportion (specifically, the gradient of the likelihood function with respect to it) and the stochastic nature of the process. Precedence has already been established for utilizing the available gradient information as opposed to an approximation such as finite difference or least squares. Also, the gradient equations may be required to enable a reasonable response surface by excluding undefined data which is not constant in all parts of the likelihood function (reference the conditional statement to exclude data less than the location parameter).

Again, the results displayed a natural separation based first on whether or not the true populations were well-separated or not. In general, the seven parameter MLE for the Mixed Weibull followed traditional expectations for sample size. That is, as the sample size increased, the MLE became more accurate. Accessing the gradient information proved comparable to all previous parameter estimation techniques of the Weibull distribution. However, using the gradient, the mixing proportion was not accurately estimated except for large sample sizes on well separated populations. Hence, in general, MDE via a mixing proportion, p, typically improved the MLE. Also, consistent with previous research, error increases with values of the shape parameter less than three (shape, $\beta_i = 3$) assuming all other variables are held constant.

While the errors may seem quite reasonable, this researcher feels that the errors are deceptively high for several primary reasons. First , the error per distribution is heavily dependent upon the mixing proportion. If the mixing proportion was in error then the parameters of both distributions were estimated incorrectly. Also, the nature of the error calculation quickly displays small departures in the mixing proportion whereas large departures are needed for any of the other parameters (Appendix E). But, this may be appropriate since no other parameter has such an impact on the remaining estimated parameters. To make matters worse, the mix of samples from each population was generated stochastically (generation based on true mixing proportion) and the error is calculated by comparing the estimated mixing proportion to the theoretical, not the stochastically generated one. Hence, if a more accurate means were available for predicting the mixing proportion such as a simple count (well-separated) or Bayesian knowledge, most of the error could be eliminated.

Excluding the mixing proportion, MLE performed very well even for the non-separated populations. The accuracy of the MLE increased as expected with increasing sample size. For the well-separated distributions, better results would be obtained by simply counting the number of data points in each mode. Unfortunately, the MLE methodology never approximated the mixing proportion correctly for the non-separated cases even when using large sample sizes. Hence, the need for a bound on the mixing proportion and the superiority of MDPC. This is due in part to the sensitivity of the gradient of the mixing proportion and the fact that we placed no restriction on the location of the second location parameter.

Future Enhancements including Mitigation of Outliers

Until the mixing proportion can be determined with some greater accuracy, the method of minimum distance may not see consistent success outside of estimating the mixing proportion. While statistics were not generated specifically, the method of minimum distance on the mixing proportion did a good job of estimating the mixing proportions. At this time, no enhancements for minimum distance are recommended.

Since the MLE methodology worked so well, there are only a couple of enhancements suggested. The assumption is that one desires to estimate parameters for a mixed distribution continuing to use the Method of Maximum Likelihood. For the MLEs, one should search for better way to estimate the mixing proportion and use it as both a point estimate and a bound. Also, the error could be mitigated further by using the point estimates as reasonable bounds on the variables. In this research, the point estimates were not used to bound the variables except in the case of the mixing proportion. This action is equivalent to mitigating outliers.

In the future, one should generate statistics on the success of the mixing proportion such as mean square error for the mixing proportion. Also, alternative methods should be investigated for the mixing proportion.

Appendix A: Selected Nominal-Error Samples for Non-Separated Mixed Weibull

Part 1a: Well-Separated Mixed Weibull (Shape, $\beta_1 = \beta_2 = 3$)
Part 1b: Well-Separated Mixed Weibull (Shape, $\beta_1 = \beta_2 = 0.9$)
Part 2: Non-Separated Mixed Weibull

The following abbreviations were used in this appendix.

PDF	Probability Density Function
CDF	Cumulative Distribution Function
COUNT	Sample size
PDF SUBCOUNT	Sample size per PDF
TRUE	True Solution for seven parameter Mixed Weibull
(1-3)	PDF 1 Shape (β_1), Location (δ_1) and Scale (η_1) Parameters
(4-6)	PDF 2 Shape (β_2), Location (δ_2) and Scale (η_2) Parameters
(7)	Mixing Proportion (p)
MLE	Maximum Likelihood Estimate(s)
MDPC	Minimum Distance Estimate for the Mixing Proportion using the Cramer Von-Mises Statistic
p_{true}	True mixing proportion
f1t(x)	True PDF for Population #1
f2t(x)	True PDF for Population #2
g_{true}	True Mixed Weibull PDF
F1t(x)	True CDF for Population #1
F2t(x)	True CDF for Population #2
G_{true}	True Mixed Weibull CDF
p_{est}	Estimated mixing proportion
f1(x)	Estimated PDF for Population #1
f2(x)	Estimated PDF for Population #2
g_{est}	Estimated Mixed Weibull PDF
F1(x)	Estimated CDF for Population #1
F2(x)	Estimated CDF for Population #2
G_{est}	Estimated Mixed Weibull CDF

Mixed Weibull Probability Density Function (PDF) Equations

$$f1t(x) := \left[\frac{\beta t_1}{\eta t_1}\cdot\left(\frac{x-\delta t_1}{\eta t_1}\right)^{\beta t_1 - 1}\right]\cdot \exp\left[-\left(\frac{x-\delta t_1}{\eta t_1}\right)^{\beta t_1}\right]$$

$$f2t(x) := \left[\frac{\beta t_2}{\eta t_2}\cdot\left(\frac{x-\delta t_2}{\eta t_2}\right)^{\beta t_2 - 1}\right]\cdot \exp\left[-\left(\frac{x-\delta t_2}{\eta t_2}\right)^{\beta t_2}\right]$$

$$f1(z) := \left[\frac{\beta_1}{\eta_1}\cdot\left(\frac{z-\delta_1}{\eta_1}\right)^{\beta_1 - 1}\right]\cdot \exp\left[-\left(\frac{z-\delta_1}{\eta_1}\right)^{\beta_1}\right]$$

$$f2(z) := \left[\frac{\beta_2}{\eta_2}\cdot\left(\frac{z-\delta_2}{\eta_2}\right)^{\beta_2 - 1}\right]\cdot \exp\left[-\left(\frac{z-\delta_2}{\eta_2}\right)^{\beta_2}\right]$$

$$g_{est}(z) := \left[\left(p_{est}\right)\cdot f1(z)\right] + \left[\left(1-p_{est}\right)\cdot f2(z)\right]$$

$$g_{true}(x) := \left[\left(p_{true}\right)\cdot f1t(x)\right] + \left[\left(1-p_{true}\right)\cdot f2t(x)\right]$$

Mixed Weibull Cumulative Distribution Function (CDF) Equations

$$F1t(x) := \left[1-\exp\left[-\left(\frac{x-\delta t1}{\eta t1}\right)^{\beta t1}\right]\right]$$

$$F2t(x) := \left[1-\exp\left[-\left(\frac{x-\delta t2}{\eta t2}\right)^{\beta t2}\right]\right]$$

$$F1(z) := \left[1-\exp\left[-\left(\frac{z-\delta_1}{\eta_1}\right)^{\beta_1}\right]\right]$$

$$F2(z) := \left[1-\exp\left[-\left(\frac{z-\delta_2}{\eta_2}\right)^{\beta_2}\right]\right]$$

$$Gest(z) := (p\cdot F1(z)) + ((1.0-p)\cdot F2(z))$$

$$Gtrue(x) := (p\cdot F1t(x)) + ((1.0-p)\cdot F2t(x))$$

Part 1a of Appendix A: Selected Nominal-Error Well-Separated Mixed Weibull
(Shape, $\beta_1 = \beta_2 = 3$)

WELL-SEPARATED (WS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mixing proportion via AD
MDLSC	MDE of the location parameters simultaneously via CVM
MDLSA	MDE of the location parameters simultaneously via AD
MDLSPC	MDE of the location parameters simultaneously and then mixing proportion via CVM
MDLSPA	MDE of the location parameters simultaneously and then mixing proportion via AD
MDPLSC	MDE of the mixing proportion and then location parameters (simultaneously) via CVM
MDPLSA	MDE of the mixing proportion and then the location parameters (simultaneously) via AD

Sample MDPC for Well-Separated Mixed Weibull
(Shape = 3)

COUNT = 100
For this seed, PDF1 SUBCOUNT = 45
For this seed, PDF2 SUBCOUNT = 55
FOS = 5.1460951116265

LOS = 10.908389477633

TRUE(1-3)= 3.0 5.0 0.5
TRUE(4-7)= 3.0 10.0 0.5 0.5

INITIAL MLE Soln: 2.538 5.078 0.405
2.062 10.140 0.387 0.459

The function value = 26.624

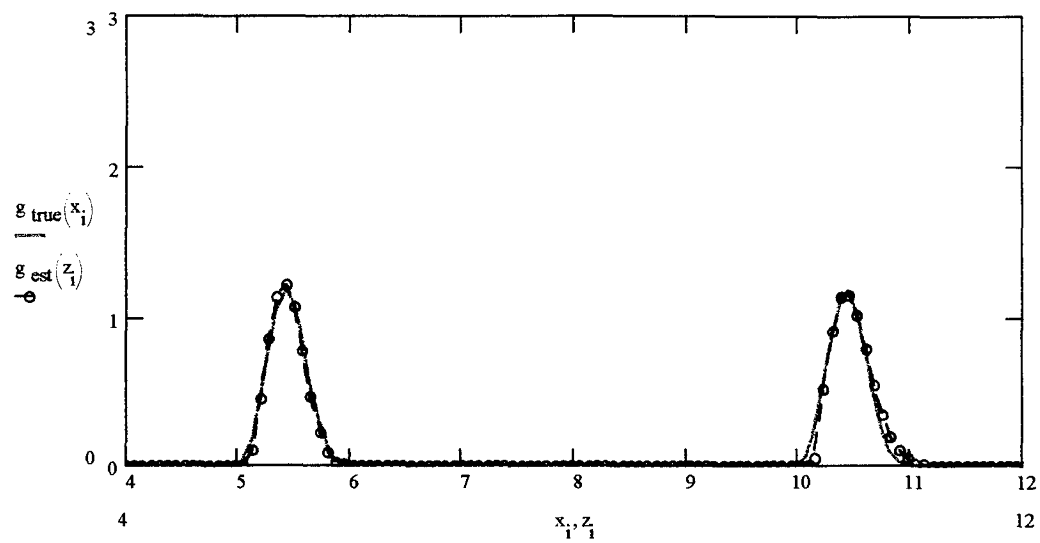
SUB-TOTALS FOR MLE phase 1
ph1TOTINTABS = 9.8619061424575D-02
ph1TOTINTMSE = 4.1667944346358D-03
INTABSPDF1 = 3.9025302296607D-02
INTMSEPDF1 = 1.4013473932253D-03
INTABSPDF2 = 5.9593759127967D-02
INTMSEPDF2 = 2.7654470414105D-03
MDPC = 0.48447875342725

Revised MLE Soln: 2.538 5.078 0.405
2.062 10.140 0.387 0.484

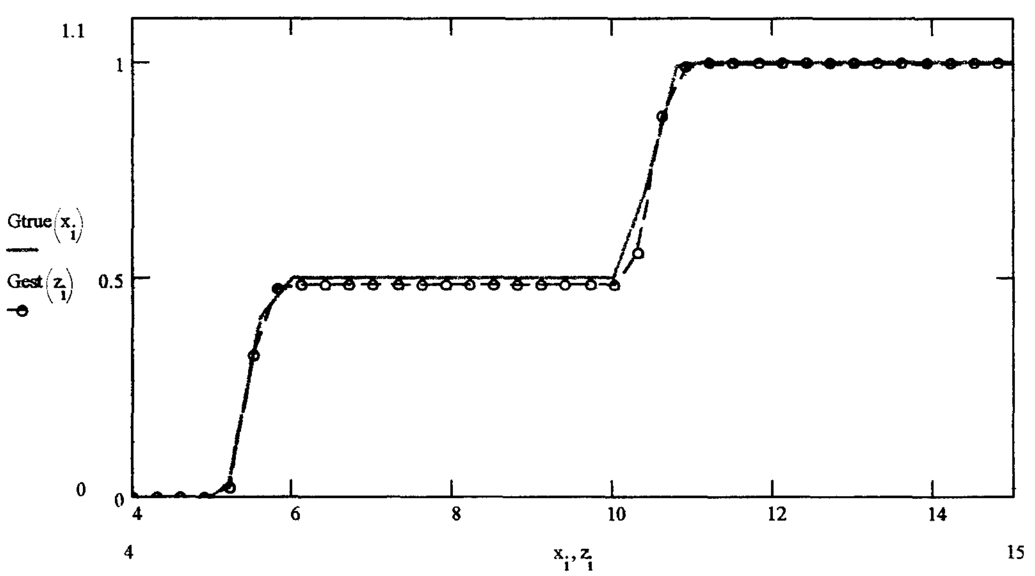
The function value = 26.749

SUB-TOTALS FOR MLE PHASE 4
ph4TOTINTABS = 4.9269505611561D-02
ph4TOTINTMSE = 1.1485930918393D-03
INTABSPDF1 = 1.5410369637805D-02
INTMSEPDF1 = 2.0221936225421D-04
INTABSPDF2 = 3.3859135973756D-02
INTMSEPDF2 = 9.4637372958514D-04

Probability Density Function (PDF)



Cumulative Distribution Function (CDF)



Sample MDPC for Well-Separated Mixed Weibull
(Shape = 3)

COUNT = 40
For this seed, PDF1 SUBCOUNT = 22
For this seed, PDF2 SUBCOUNT = 18
FOS = 5.2157392110824
LOS = 10.749425212536

TRUE(1-3)= 3.0 5.0 0.5
TRUE(4-7)= 3.0 10.0 0.5 0.5

INITIAL MLE Soln: 3.436 5.051 0.441
1.467 10.185 0.256 0.550

The function value = 3.818

SUB-TOTALS FOR MLE phase 1
ph1TOTINTABS = 0.13478094176074
ph1TOTINTMSE = 7.6607059335772D-03
INTABSPDF1 = 6.2280471076631D-02
INTMSEPDF1 = 3.2641820374831D-03
INTABSPDF2 = 7.2500470684105D-02
INTMSEPDF2 = 4.3965238960941D-03

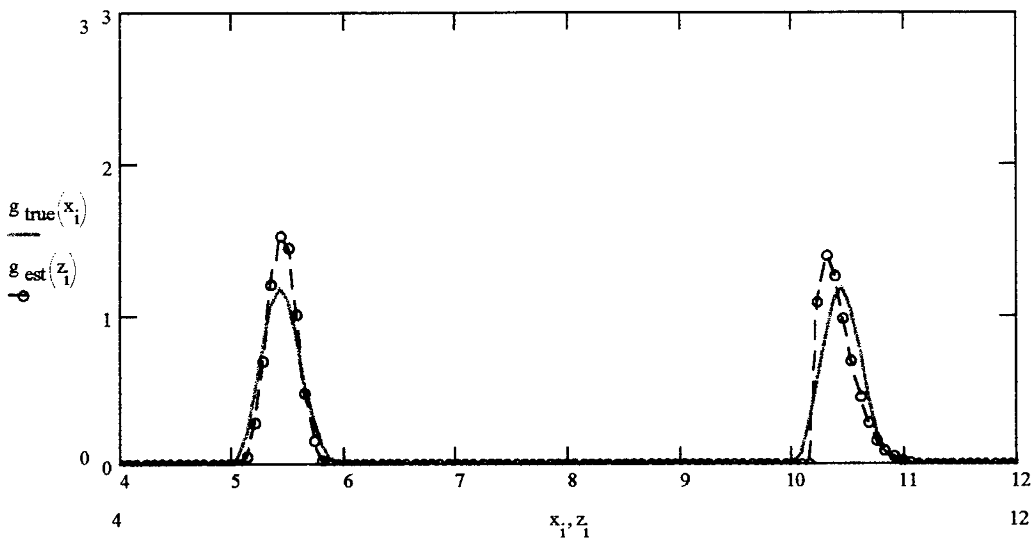
MDPC = 0.51684017596413

Revised Soln: 3.436 5.051 0.441
1.467 10.185 0.256 0.517

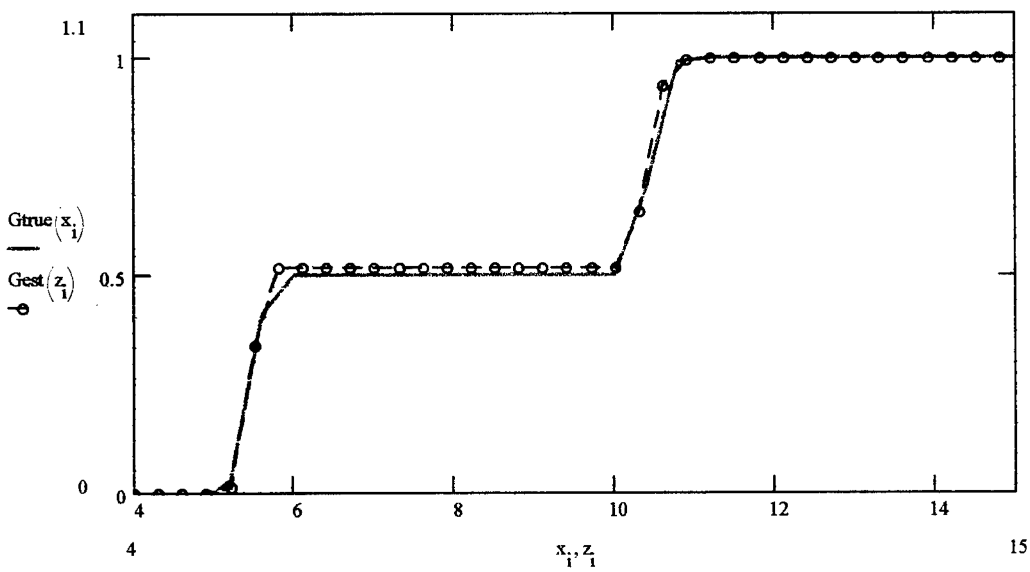
The function value = 3.906

SUB-TOTALS FOR MLE PHASE 4
ph4TOTINTABS = 6.6696381024139D-02
ph4TOTINTMSE = 2.3348880555180D-03
INTABSPDF1 = 2.9990211629377D-02
INTMSEPDF1 = 7.6629264896699D-04
INTABSPDF2 = 3.6706169394762D-02
INTMSEPDF2 = 1.5685954065510D-03

Probability Density Function (PDF)



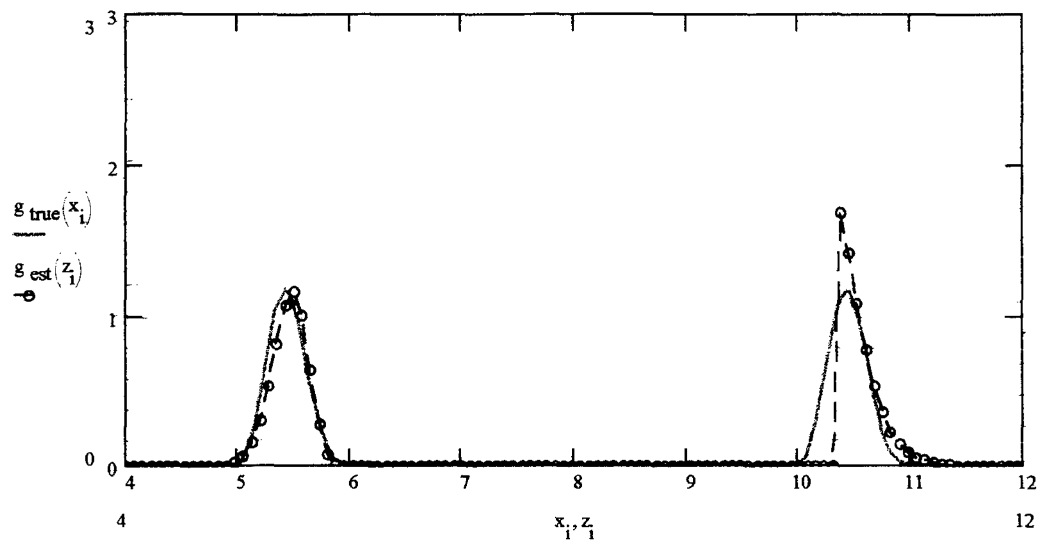
Cumulative Distribution Function (CDF)



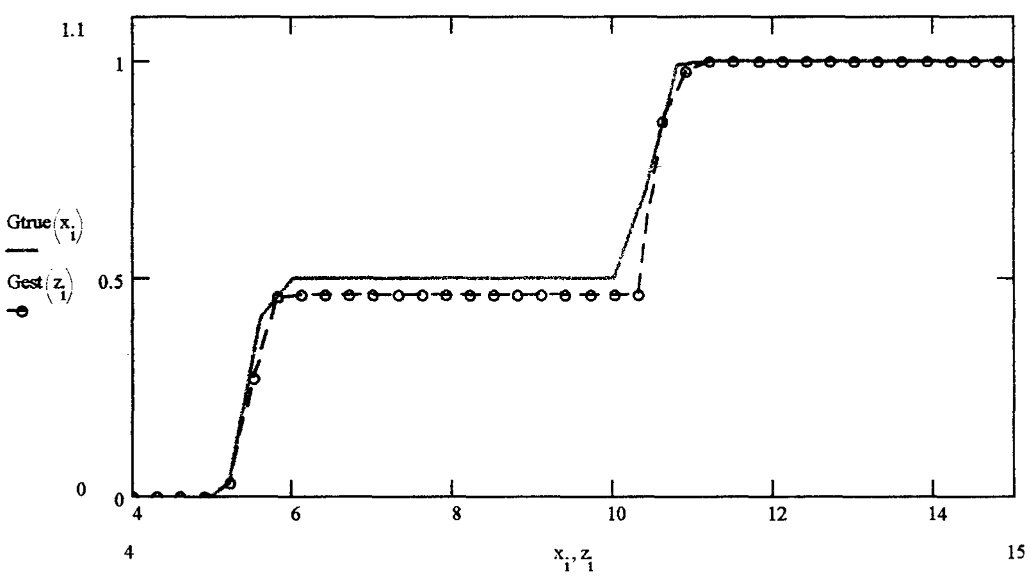
Sample MDPC for Well-Separated Mixed Weibull
(Shape = 3)

```
COUNT = 20
        For this seed, PDF1 SUBCOUNT =  8
        For this seed, PDF2 SUBCOUNT =  12
                FOS =    5.1835496657608
                LOS =    10.814596565814

 TRUE(1-3)=   3.0   5.0  0.5
 TRUE(4-7)=   3.0  10.0  0.5       0.5

INITIAL MLE Soln:        5.722  4.669  0.852
                         1.233  10.300  0.236          0.400

                         The function value =   3.739

                     SUB-TOTALS FOR MLE phase 1
                         ph1TOTINTABS =  0.24587558478945
                         ph1TOTINTMSE =   2.4245512009114D-02
                             INTABS1 =  0.11090467362401
                             INTMSE1 =   1.0126481581103D-02
                             INTABS2 =  0.13497091116544
                             INTMSE2 =   1.4119030428011D-02

MDPC =   0.46279508359642

        Revised MLE Soln:   5.722  4.669  0.852
                            1.233  10.300  0.236          0.463

                         The function value =   3.899

                     SUB-TOTALS FOR MLE PHASE 4
                         ph4TOTINTABS =  0.11942197373540
                         ph4TOTINTMSE =   5.9773539911757D-03
                             INTABSPDF1 =   4.5956752453868D-02
                             INTMSEPDF1 =   1.6671584645745D-03
                             INTABSPDF2 =   7.3465221281527D-02
                             INTMSEPDF2 =   4.3101955266013D-03
```

Probability Density Function (PDF)



Cumulative Distribution Function (CDF)



Sample MDPC for Well-Separated Mixed Weibull
(Shape = 3)

COUNT = 10
For this seed, PDF1 SUBCOUNT = 3
For this seed, PDF2 SUBCOUNT = 7
FOS = 5.1675338970734
LOS = 10.679696815739

TRUE(1-3)= 3.0 5.0 0.5
TRUE(4-7)= 3.0 10.0 0.5 0.5

INITIAL MLE Soln: 4.452 5.168 0.419
0.994 10.315 0.140 0.400

The function value = 997.388

SUB-TOTALS FOR MLE phase 1 and J =: 2
ph1TOTINTABS = 0.25563531333835
ph1TOTINTMSE = 2.7912134556282D-02
INTABSPDF1 = 0.14656380661449
INTMSEPDF1 = 1.7810429417008D-02
INTABSPDF2 = 0.10907150672386
INTMSEPDF2 = 1.0101705139274D-02

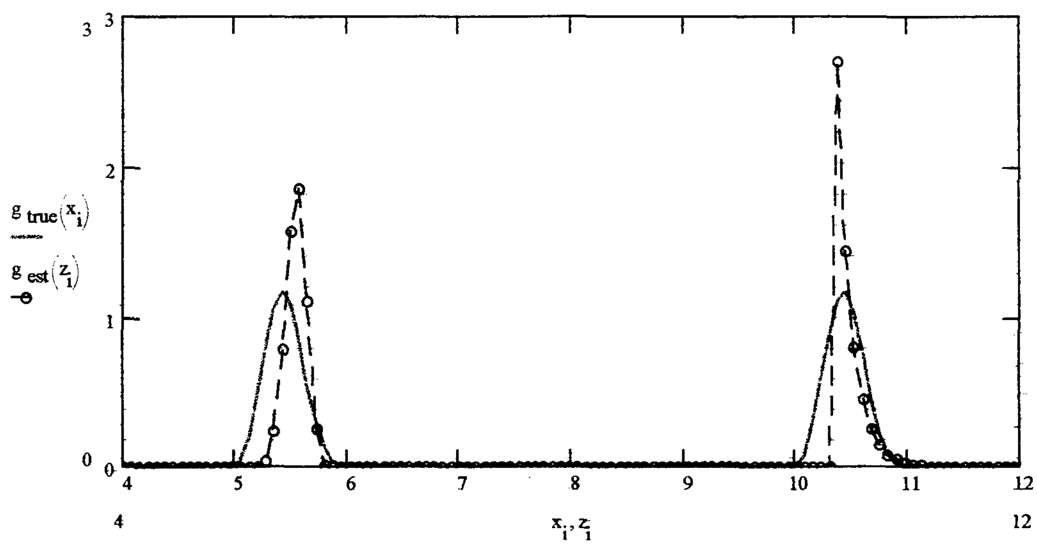
MDPC = 0.44118033899264

Revised MLE Soln: 4.694 5.168 0.412
0.956 10.315 0.124 0.441

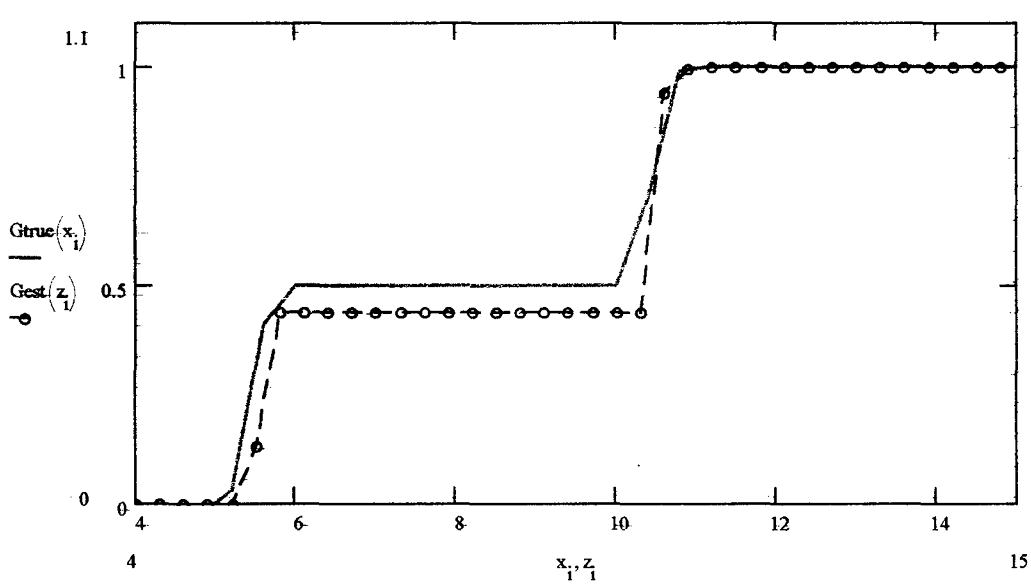
The function value = 1999.578

SUB-TOTALS FOR MLE PHASE 4
ph4TOTINTABS = 0.17333113705157
ph4TOTINTMSE = 1.4733297376409D-02
INTABSPDF1 = 0.10509606627328
INTMSEPDF1 = 1.0773771021652D-02
INTABSPDF2 = 6.8235070778296D-02
INTMSEPDF2 = 3.9595263547579D-03

Probability Density Function (PDF)



Cumulative Distribution Function (CDF)



Part 1b of Appendix A: Selected Nominal-Error Well-Separated Mixed Weibull
(Shape, $\beta_1 = \beta_2 = 0.9$)

WELL-SEPARATED (WS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mixing proportion via AD
MDLSC	MDE of the location parameters simultaneously via CVM
MDLSA	MDE of the location parameters simultaneously via AD
MDLSPC proportion	MDE of the location parameters simultaneously and then mixing via CVM
MDLSPA proportion	MDE of the location parameters simultaneously and then mixing via AD
MDPLSC (simultaneously)	MDE of the mixing proportion and then location parameters via CVM
MDPLSA	MDE of the mixing proportion and then the location parameters (simultaneously) via AD

Sample MDPC for Well-Separated Mixed Weibull
(Shape = 0.9)

COUNT = 100
For this seed, PDF1 SUBCOUNT = 48
For this seed, PDF2 SUBCOUNT = 52
FOS = 5.0014781821701
LOS = 16.926652000889

TRUE(1-3) = 0.9 5.0 0.5
TRUE(4-7) = 0.9 15.0 0.5 0.5

INITIAL MLE Soln: 0.500 5.001 0.546
1.373 14.862 0.738 0.400

The function value = 1120.529

SUB-TOTALS FOR MLE phase 1 and J =: 2
ph1TOTINTABS = 0.85355935088488
ph1TOTINTMSE = 9.8172954856908D-02
INTABS1 = 0.54001141924391
INTMSE1 = 6.9980344851740D-02
INTABS2 = 0.31354793164098
INTMSE2 = 2.8192610005168D-02

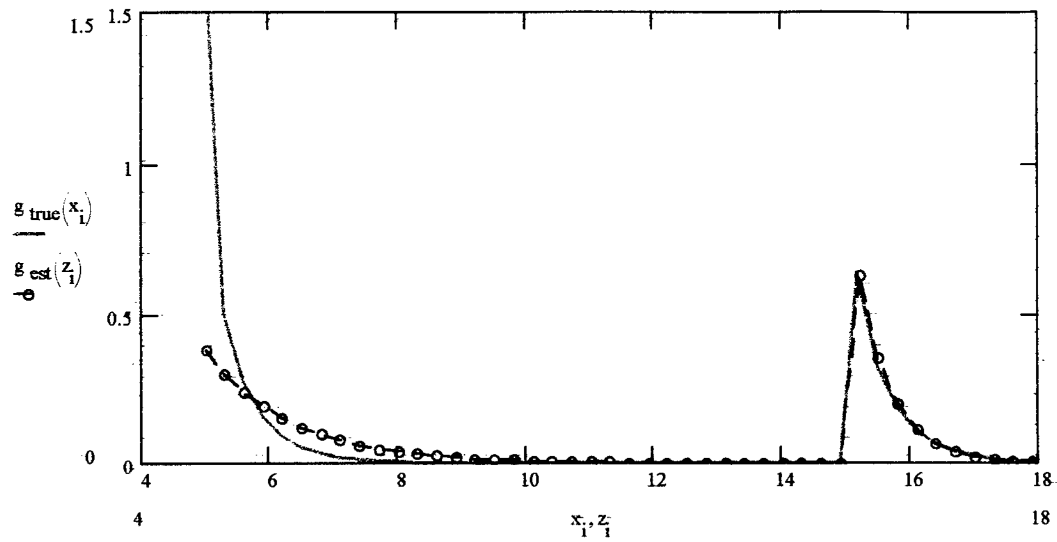
MDPC = 0.50404508267441

Revised MLE Soln: 0.977 5.001 1.336
0.984 15.000 0.535 0.504

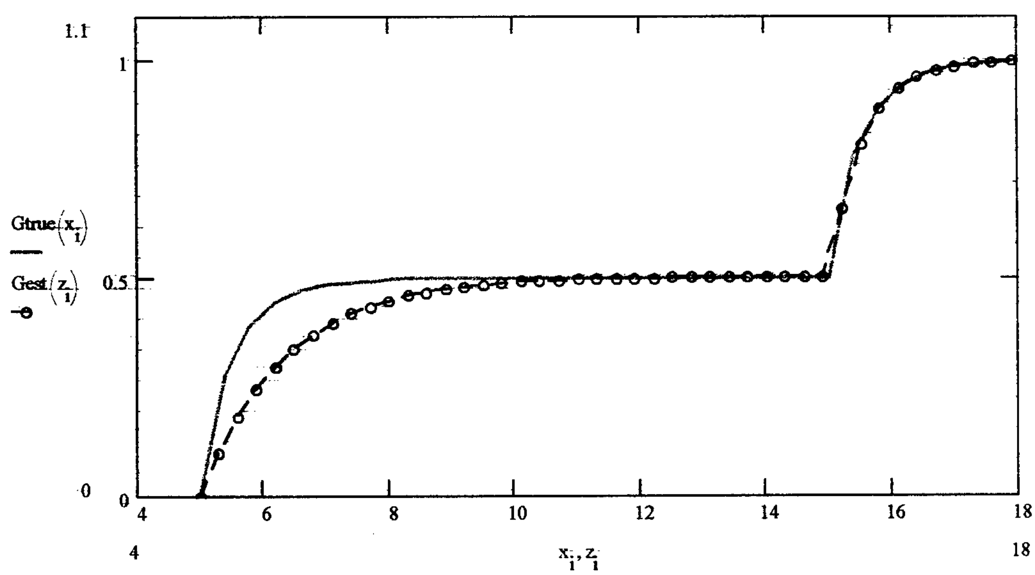
The function value = 1106.938

SUB-TOTALS FOR MLE PHASE 4
ph4TOTINTABS = 0.42255552312705
ph4TOTINTMSE = 4.2608063031293D-02
INTABS1 = 0.39606488312585
INTMSE1 = 4.2390146507697D-02
INTABS2 = 2.6490640001192D-02
INTMSE2 = 2.1791652359559D-04

Probability Density Function (PDF)



Cumulative Distribution Function (CDF)



Sample MDPC for Well-Separated Mixed Weibull
(Shape = 0.9)

COUNT = 40
For this seed, PDF1 SUBCOUNT = 19
For this seed, PDF2 SUBCOUNT = 21
FOS = 5.0025317918973
LOS = 16.581330169171

TRUE(1-3)= 0.9 5.0 0.5
TRUE(4-7)= 0.9 15.0 0.5 0.5

INITIAL MLE Soln: 0.891 5.003 0.651
0.999 15.022 0.512 0.400

The function value = 1044.893

SUB-TOTALS FOR MLE phase 1
ph1TOTINTABS = 0.86328490071668
ph1TOTINTMSE = 8.5379229014965D-02
INTABSPDF1 = 0.46232986482445
INTMSEPDF1 = 4.8812727121721D-02
INTABSPDF2 = 0.40095503589223
INTMSEPDF2 = 3.6566501893244D-02

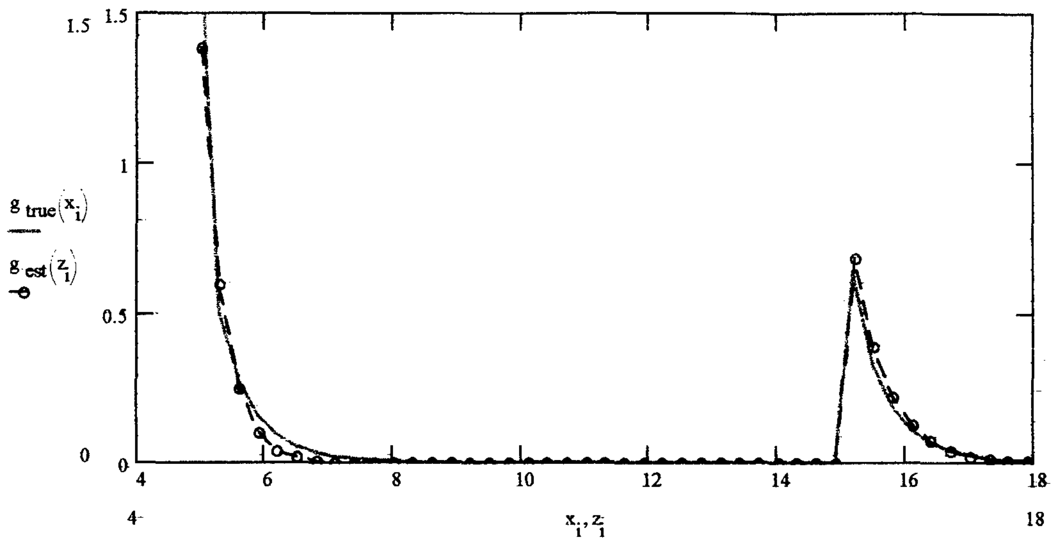
MDPC = 0.48904508301020

Revised Soln: 1.007 5.003 0.341
1.000 15.022 0.530 0.489

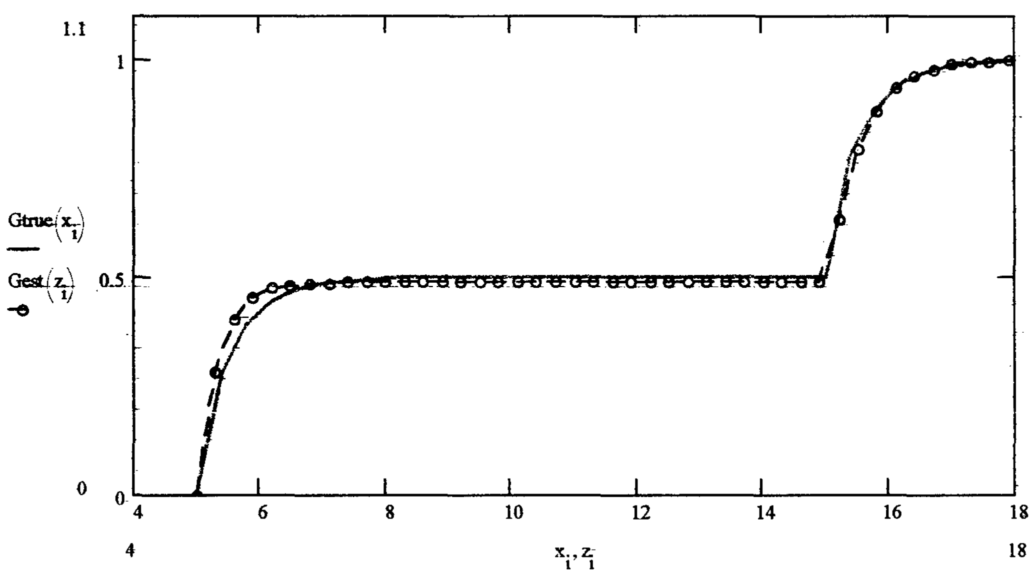
The function value = 1051.105

SUB-TOTALS FOR MLE PHASE 4
ph4TOTINTABS = 0.13909122466502
ph4TOTINTMSE = 3.9637805066710D-03
INTABSPDF1 = 8.2697747493030D-02
INTMSEPDF1 = 2.9601460137443D-03
INTABSPDF2 = 5.6393477171990D-02
INTMSEPDF2 = 1.0036344929266D-03

Probability Density Function (PDF)



Cumulative Distribution Function (CDF)



Sample MDPC for Well-Separated Mixed Weibull
(Shape = 0.9)

COUNT = 20
For this seed, PDF1 SUBCOUNT = 7
For this seed, PDF2 SUBCOUNT = 13
FOS = 5.0124423190355
LOS = 16.670192220847

TRUE(1-3)= 0.9 5.0 0.5
TRUE(4-7)= 0.9 15.0 0.5 0.5

INITIAL MLE Soln: 1.425 5.012 0.614
0.858 15.003 0.489 0.400

The function value = 1014.190

SUB-TOTALS FOR MLE phase 1
ph1TOTINTABS = 0.81516382283992
ph1TOTINTMSE = 7.6670938208509D-02
INTABSPDF1 = 0.41550929779042
INTMSEPDF1 = 3.9094579146734D-02
INTABSPDF2 = 0.39965452504951
INTMSEPDF2 = 3.7576359061775D-02

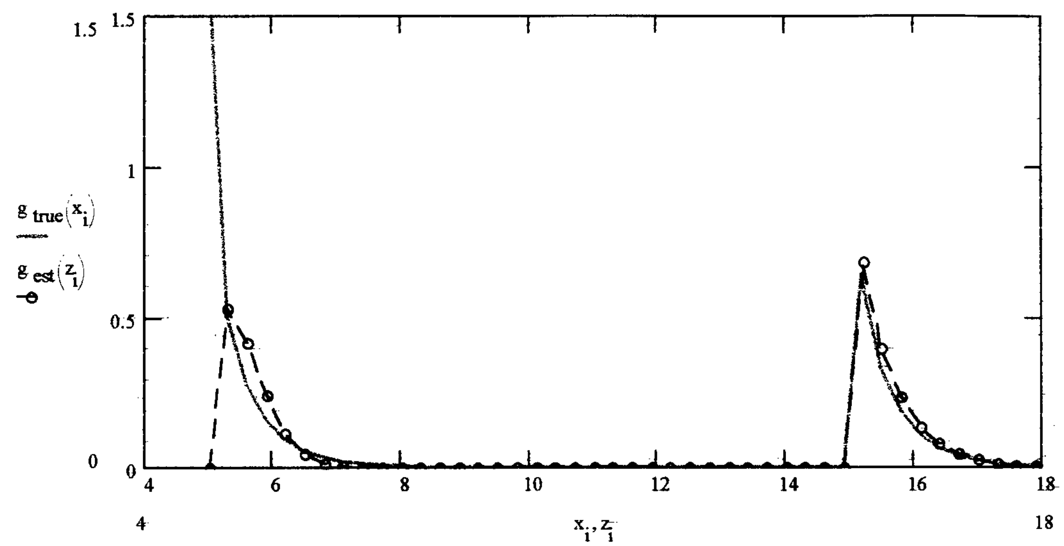
MDPC = 0.45243033874118

Revised MLE Soln: 1.491 5.012 0.638
0.995 15.003 0.555 0.452

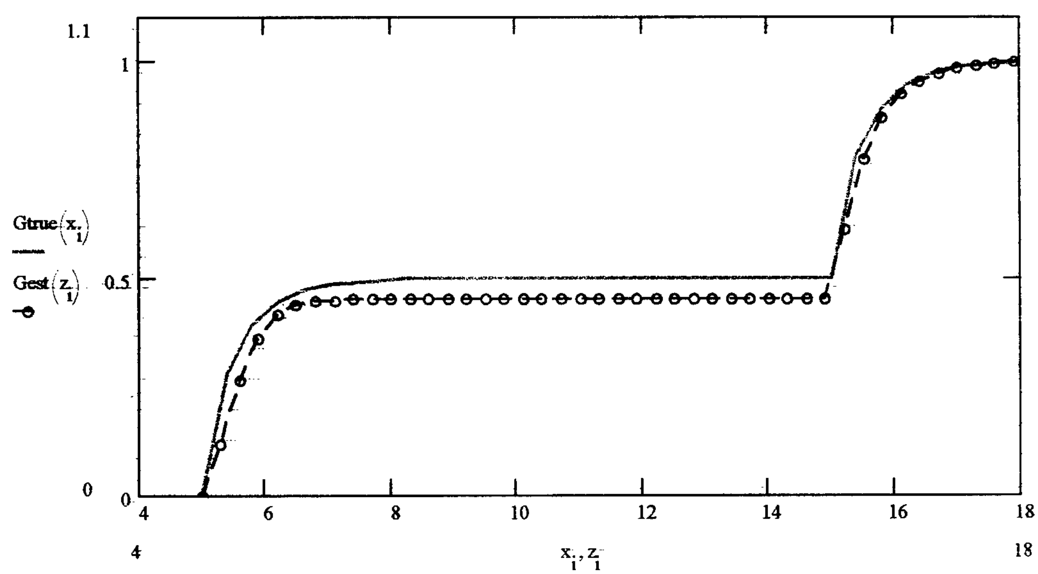
The function value = 29.685

SUB-TOTALS FOR MLE PHASE 4 and J =: 3
ph4TOTINTABS = 0.33358770050161
ph4TOTINTMSE = 1.7980879941690D-02
INTABSPDF1 = 0.12946907434405
INTMSEPDF1 = 8.6376951805304D-03
INTABSPDF2 = 0.20411862615756
INTMSEPDF2 = 9.3431847611597D-03

Probability Density Function (PDF)



Cumulative Distribution Function (CDF)



Sample MDPC for Well-Separated Mixed Weibull
(Shape = 0.9)

COUNT = 10
For this seed, PDF1 SUBCOUNT = 6
For this seed, PDF2 SUBCOUNT = 4
FOS = 5.0019494304593
LOS = 15.0019494304593

TRUE(1-3)= 0.9 5.0 0.5
TRUE(4-7)= 0.9 15.0 0.5 0.5

INITIAL MLE Soln: 1.079 5.002 0.945
0.995 15.013 0.155 0.600

The function value = 1021.092

SUB-TOTALS FOR MLE phase 1
ph1TOTINTABS = 0.91025476923598
ph1TOTINTMSE = 0.11910581216087
INTABSPDF1 = 0.29812235118837
INTMSEPDF1 = 2.3293678020428D-02
INTABSPDF2 = 0.61213241804760
INTMSEPDF2 = 9.5812134140443D-02

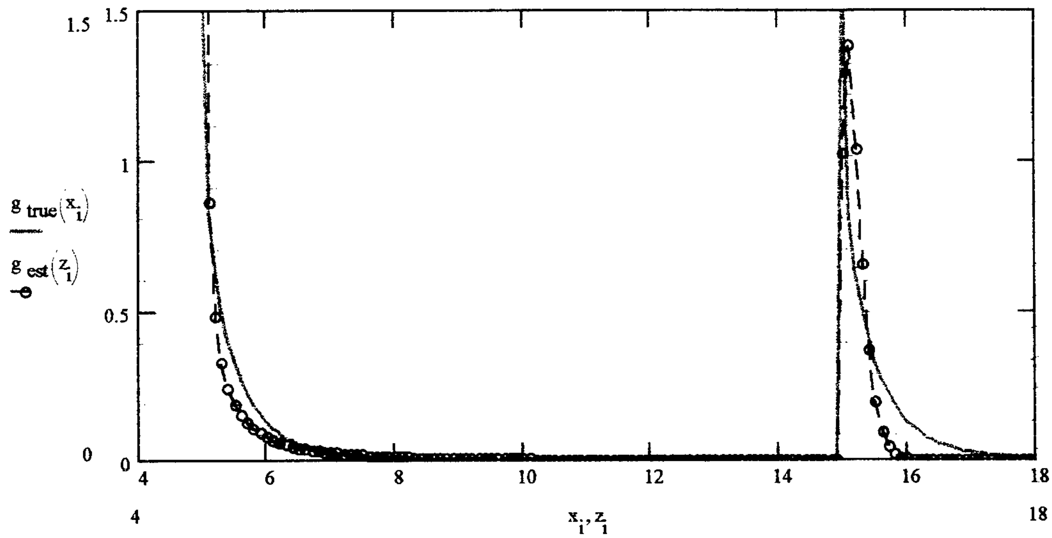
MDPC = 0.54220491629182

Revised MLE Soln: 0.500 5.002 0.290
1.352 14.987 0.217 0.542

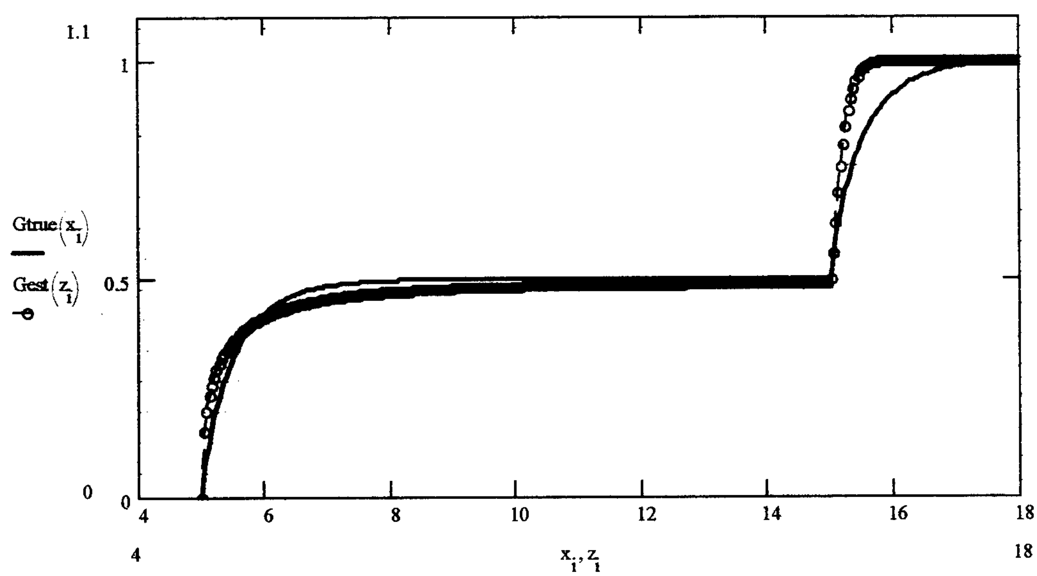
The function value = 1009.938

SUB-TOTALS FOR MLE PHASE 4
ph4TOTINTABS = 0.33225026498452
ph4TOTINTMSE = 3.8054716916735D-02
INTABSPDF1 = 0.16766219830378
INTMSEPDF1 = 1.0013769351279D-02
INTABSPDF2 = 0.16458806668074
INTMSEPDF2 = 2.8040947565456D-02

Probability Density Function (PDF)



Cumulative Distribution Function (CDF)



Part 2 of Appendix A: Selected Nominal-Error Non-Separated Mixed Weibull

Non-Separated (NS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mixing proportion via AD
MDPL(1-2)C	MDE of the location parameters (sequentially) via CVM
MDPL(1-2)A	MDE of the location parameters (sequentially) via AD
MDPLSC	MDE of the location parameters (simultaneously) via CVM
MDPLSA	MDE of the location parameters (simultaneously) via AD

Selected MDPC for Non-Separated Mixed Weibull

COUNT = 100
For this seed, PDF1 SUBCOUNT = 53
For this seed, PDF2 SUBCOUNT = 47
FOS = 5.0157295668406
LOS = 7.5789691820566

TRUE(1-3) = 4.0 5.0 0.5
TRUE(4-7) = 1.0 5.0 0.5 0.5

INITIAL MLE Soln: 4.954 5.016 0.444
0.978 5.016 0.561 0.400

The function value = -0.419

SUB-TOTALS FOR MLE phase 1
ph1TOTINTABS = 0.52615183541213
ph1TOTINTMSE = 5.1507373965766D-02
INTABSPDF1 = 9.2604750982949D-02
INTMSEPDF1 = 8.5478564915129D-03
INTABSPDF2 = 0.43354708442918
INTMSEPDF2 = 4.2959517474253D-02

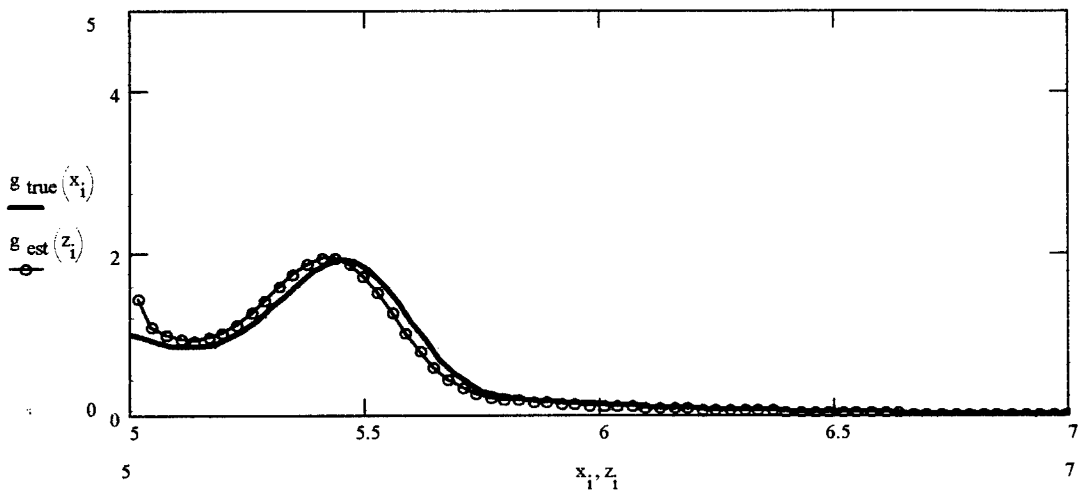
MDPC = 0.49654508284273

Revised MLE Soln: 3.697 5.016 0.455
0.940 5.016 0.472 0.497

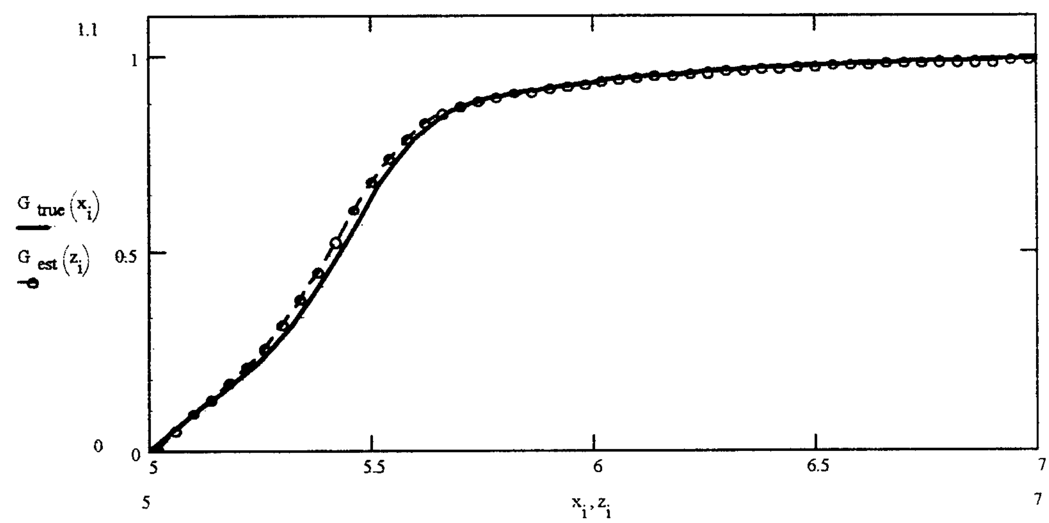
The function value = -4.202

SUB-TOTALS FOR MLE PHASE 4
ph4TOTINTABS = 2.9507926348969D-02
ph4TOTINTMSE = 4.4208574000542D-04
INTABSPDF1 = 1.4693500680389D-02
INTMSEPDF1 = 3.7708911956011D-04
INTABSPDF2 = 1.4814425668580D-02
INTMSEPDF2 = 6.4996620445307D-05

Mixed Weibull Probability Density Function (PDF)



Mixed Weibull Cumulative Distribution Function (CDF)



Sample MDPC for Non-Separated Mixed Weibull

COUNT = 40

For this seed, PDF1 SUBCOUNT = 21
For this seed, PDF2 SUBCOUNT = 19
FOS = 5.0002596553541
LOS = 7.4472265311814

TRUE(1-3) = 4.0 5.0 0.5
TRUE(4-7) = 1.0 5.0 0.5 0.5

INITIAL MLE Soln: 10.000 5.000 0.540
0.959 5.000 0.395 0.400

The function value = 1001.329

SUB-TOTALS FOR MLE phase 1
ph1TOTINTABS = 0.35733566871914
ph1TOTINTMSE = 3.3714864541222D-02
INTABSPDF1 = 0.12905860516313
INTMSEPDF1 = 1.4033346410764D-02
INTABSPDF2 = 0.22827706355601
INTMSEPDF2 = 1.9681518130458D-02

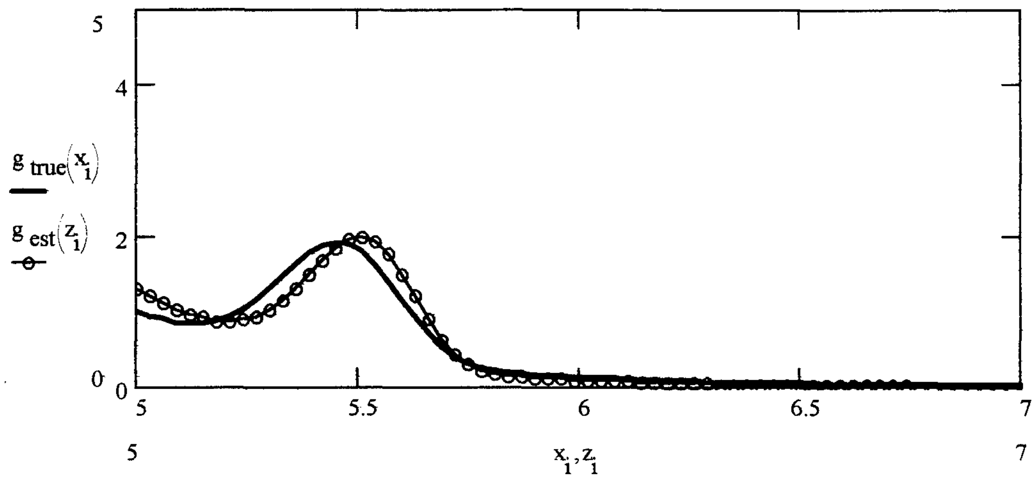
MDPC = 0.48279508314939

Revised MLE Soln: 4.894 5.000 0.583
1.000 5.000 0.458 0.483

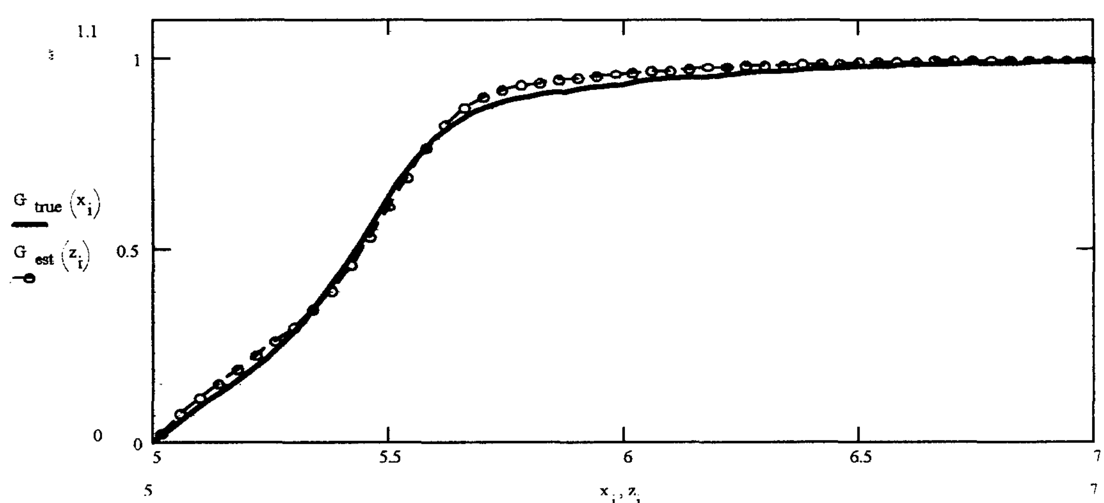
The function value = 1000.218

SUB-TOTALS FOR MLE PHASE 4
ph4TOTINTABS = 9.5097848118644D-02
ph4TOTINTMSE = 5.0643449134269D-03
INTABSPDF1 = 5.7237378113412D-02
INTMSEPDF1 = 4.5472748103043D-03
INTABSPDF2 = 3.7860470005232D-02
INTMSEPDF2 = 5.1707010312255D-04

Mixed Weibull Probability Density Function (PDF)



Mixed Weibull Cumulative Distribution Function (CDF)



Sample MDPC for Non-Separated Mixed Weibull

COUNT = 20

For this seed, PDF1 SUBCOUNT = 7

For this seed, PDF2 SUBCOUNT = 13

FOS = 5.0361368118333

LOS = 6.8861048364758

TRUE(1-3) = 4.0 5.0 0.5

TRUE(4-7) = 1.0 5.0 0.5 0,5

INITIAL MLE Soln: 4.991 5.036 0.521

1.000 5.036 1.006 0.506

The function value = 1002.037

SUB-TOTALS FOR MLE phase 1

ph1TOTINTABS = 0.27470402893859

ph1TOTINTMSE = 2.4855348161004D-02

INTABSPDF1 = 3.3748617355952D-02

INTMSEPDF1 = 2.2260368119689D-03

INTABSPDF2 = 0.24095541158264

INTMSEPDF2 = 2.2629311349035D-02

MDPC = 0.52356894263326

Revised MLE Soln: 5.388 5.036 0.485

0.748 5.034 0.468 0.524

The function value = 0.009

SUB-TOTALS FOR MLE PHASE 4

ph4TOTINTABS = 0.13939827220376

ph4TOTINTMSE = 3.3378852878963D-03

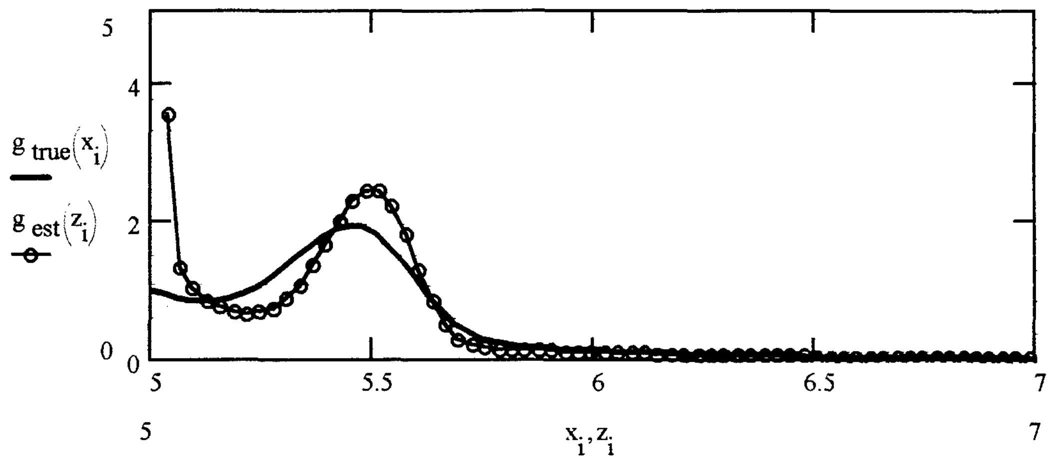
INTABSPDF1 = 3.8758370994966D-02

INTMSEPDF1 = 1.3671301147967D-03

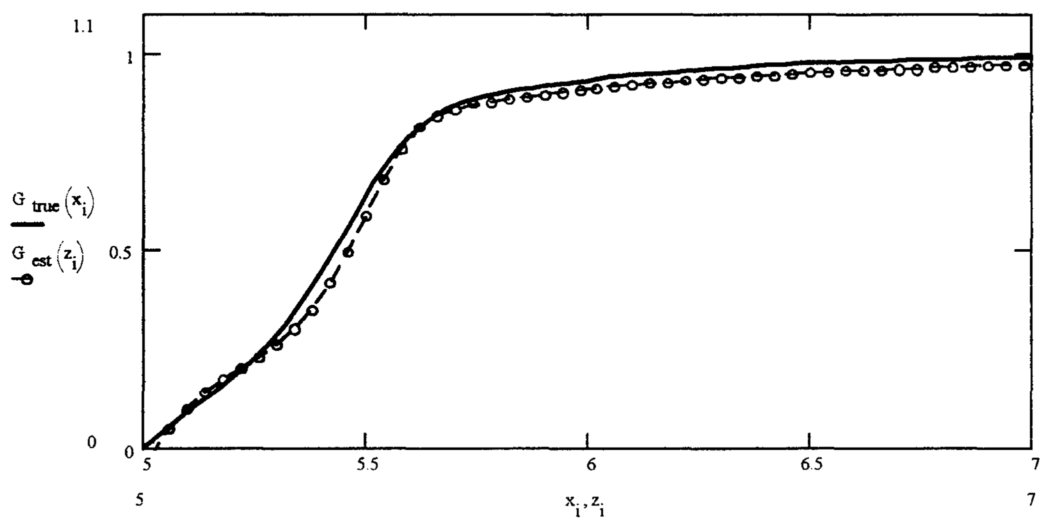
INTABSPDF2 = 1.0063990120879D-01

INTMSEPDF2 = 1.9707551730996D-03

Mixed Weibull Probability Density Function (PDF)



Mixed Weibull Cumulative Distribution Function (CDF)



Sample MDPC for Non-Separated Mixed Weibull

COUNT = 10

For this seed, PDF1 SUBCOUNT = 4
For this seed, PDF2 SUBCOUNT = 6
FOS = 5.1361625093515
LOS = 6.9105651798572

TRUE(1-3)= 4.0 5.0 0.5
TRUE(4-7)= 1.0 5.0 0.5 0.5

INITIAL MLE Soln: 5.007 5.136 0.300
0.964 5.136 0.473 0.400
The function value = 1.013

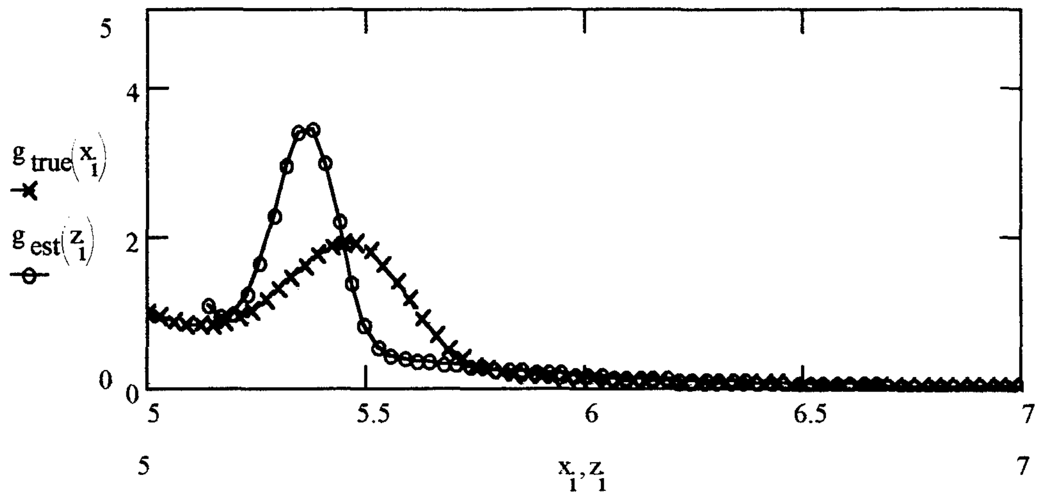
SUB-TOTALS FOR MLE phase 1
ph1TOTINTABS = 0.48890929755808
ph1TOTINTMSE = 4.7920076166326D-02
INTABSPDF1 = 9.9601118688938D-02
INTMSEPDF1 = 8.9053825702508D-03
INTABSPDF2 = 0.38930817886914
INTMSEPDF2 = 3.9014693596075D-02

MDPC = 0.49095491296700

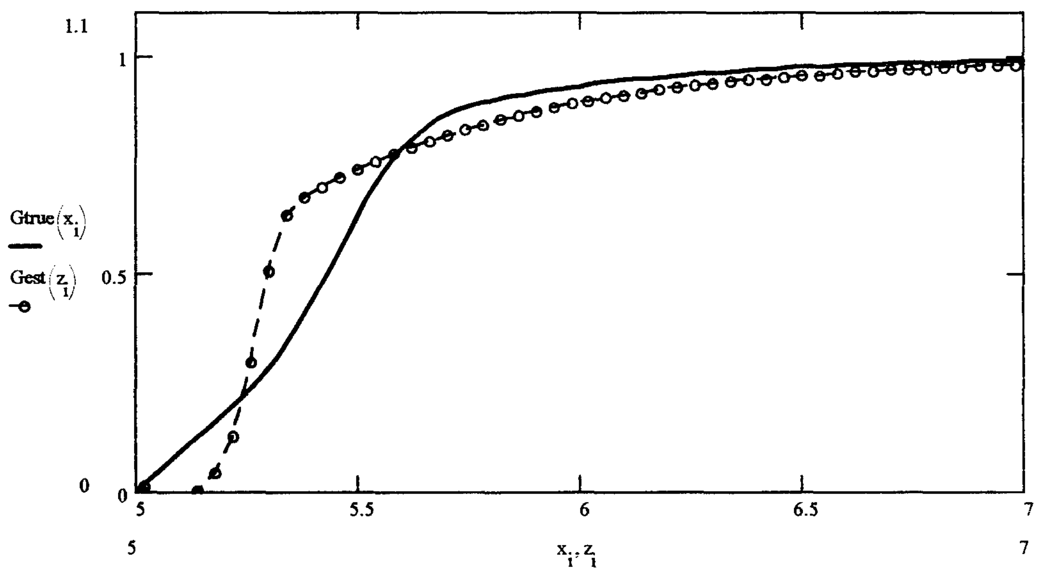
Revised MLE Soln: 3.818 5.136 0.150
0.973 5.136 0.539 0.491
The function value = 1005.080

SUB-TOTALS FOR MLE PHASE 4
ph4TOTINTABS = 0.21857220897944
ph4TOTINTMSE = 2.9663715348184D-02
INTABS1 = 9.3901002185331D-02
INTMSE1 = 2.3095166235675D-02
INTABS2 = 0.12467120679411
INTMSE2 = 6.5685491125098D-03

Mixed Weibull Probability Density Function (PDF)



Mixed Weibull Cumulative Distribution Function (CDF)



Appendix B. By Integrated Absolute Error, Results for Mixed Weibull

Part 1a. Well-Separated Mixed Weibull (Shape = $\beta_1 = \beta_2 = 3$)
Part 1b. Well-Separated Mixed Weibull (Shape = $\beta_1 = \beta_2 = 0.9$)
Part 2. Non-Separated Mixed Weibull

The following abbreviations were used in this appendix (ref Chap 3. Methodology).

GENERAL

P	Mixing Proportion
L	Shorthand notation
L(1-2)	Sequential estimation of the location parameters
LS	Simultaneous estimation of the location parameters
N	Sample Size
I-ABS	Error reported via Integrated Absolute Difference
I-MSE	Error reported via Mean Square Error
MDE	Minimum Distance Estimate
MLE	Maximum Likelihood Estimate
CVM	Cramer Von-Mises Test Statistic
AD	Anderson-Darling Test Statistic
SCORE	Number of times better than MLE out of 1000
CI	Confidence Interval at the alpha equal to ten percent level

Part 1a of Appendix B. By Method Results for Well-Separated Mixed Weibull
(Shape = $\beta_1 = \beta_2 = 3$)

WELL-SEPARATED (WS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mixing proportion via AD
MDLSC	MDE of the location parameters simultaneously via CVM
MDLSA	MDE of the location parameters simultaneously via AD
MDLSPC proportion	MDE of the location parameters simultaneously and then mixing via CVM
MDLSPA proportion	MDE of the location parameters simultaneously and then mixing via AD
MDPLSC (simultaneously)	MDE of the mixing proportion and then location parameters via CVM
MDPLSA	MDE of the mixing proportion and then the location parameters (simultaneously) via AD

Table 7. By Integrated Absolute Error, Aggregated Results for Well-Separated Mixed Weibull

B = 3									
	10	20	40	100		10	20	40	100
	{------------------- CVM			-------------------}		{------------------- AD			-------------------}
P = 0.5	***MDLSPC***					***MDLSPA***			
MLE	0.2552	0.2069	0.1431	0.0947		0.2902	0.1954	0.1422	0.0841
MDL	0.2859	0.2395	0.1765	0.0592		0.3097	0.227	0.16	0.0956
MDP	0.2156	0.1623	0.0918	0.1384		0.2485	0.1733	0.1192	0.1115
	MDPLSC					***MDPLSA***			
MLE	0.2701	0.2065	0.1411	0.0971		0.2827	0.2021	0.1489	0.0837
MDP	0.1987	0.129	0.0823	0.0495		0.2048	0.1494	0.1131	0.1032
MDL	0.2256	0.1426	0.0929	0.0569		0.2431	0.1491	0.1211	0.1133
P = 0.3	***MDLSPC***					***MDLSPA***			
MLE	0.1684	0.1187	0.0928	0.0725		0.155	0.1176	0.0936	0.0712
MDL	0.1947	0.1518	0.1335	0.1128		0.1718	0.1270	0.0956	0.0809
MDP	0.1881	0.1459	0.1401	0.1215		0.1759	0.1374	0.1209	0.1121
	MDPLSC					***MDPLSA***			
MLE	0.1562	0.1106	0.091	0.0744		0.1488	0.1125	0.0923	0.0719
MDP	0.1533	0.1247	0.1194	0.1098		0.1534	0.1266	0.1198	0.1097
MDL	0.1622	0.1355	0.1223	0.1113		0.1849	0.1408	0.1232	0.1121
P = 0.1	***MDLSPC***					***MDLSPA***			
MLE	0.1319	0.1064	0.0844	0.0583		0.1316	0.1136	0.0793	0.0564
MDL	0.1291	0.1182	0..1092	0.1073		0.1361	0.1183	0.0942	0.0792
MDP	0.1343	0.1224	0.1123	0.1059		0.1364	0.1241	0.1077	0.1031
	MDPLSC					***MDPLSA***			
MLE	0.1373	0.1110	0.0901	0.0581		0.1356	0.1123	0.0810	0.0592
MDP	0.1560	0.1179	0.1121	0.1056		0.1352	0.1205	0.1095	0.1054
MDL	0.1371	0.1184	0.1139	0.1055		0.1352	0.1207	0.1093	0.1035

Part 1b of Appendix B. By Method Results for Well-Separated Mixed Weibull
(Shape = $\beta_1 = \beta_2 = 0.9$)

WELL-SEPARATED (WS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mixing proportion via AD
MDLSC	MDE of the location parameters simultaneously via CVM
MDLSA	MDE of the location parameters simultaneously via AD
MDLSPC proportion	MDE of the location parameters simultaneously and then mixing via CVM
MDLSPA proportion	MDE of the location parameters simultaneously and then mixing via AD
MDPLSC (simultaneously)	MDE of the mixing proportion and then location parameters via CVM
MDPLSA	MDE of the mixing proportion and then the location parameters (simultaneously) via AD

Table 8. By Integrated Absolute Error, Aggregated Results for Well-Separated Mixed Weibull

B = 0.9									
	10	20	40	100		10	20	40	100
	{------------------- CVM -------------------}					{------------------- AD -------------------}			
P = 0.5	***MDLSPC***					***MDLSPA***			
MLE	0.7068	0.6238	0.644	0.7		0.7463	0.7119	0.6789	0.7144
MDL	0.7313	0.711	0.8323	0.9817		0.7244	0.8168	0.8383	1.0128
MDP	0.5166	0.4774	0.4759	0.549		0.5779	0.5228	0.4833	0.6286
	MDPLSC					***MDPLSA***			
MLE	0.7415	0.6872	0.7172	0.7081		0.7114	0.7117	0.6489	0.7333
MDP	0.4538	0.3819	0.3235	0.3617		0.4331	0.4146	0.3563	0.4661
MDL	0.5661	0.5108	0.5012	0.5289		0.5582	0.5527	0.5027	0.5419
P = 0.3	***MDLSPC***					***MDLSPA***			
MLE	0.4267	0.3847	0.3724	0.3958		0.4329	0.3914	0.3735	0.4048
MDL	0.4703	0.4601	0.479	0.6546		0.4754	0.4847	0.4903	0.6074
MDP	0.4684	0.4478	0.4434	0.5297		0.4908	0.4618	0.4218	0.4549
	MDPLSC					***MDPLSA***			
MLE	0.4147	0.377	0.3784	0.3968		0.4192	0.3734	0.3558	0.4001
MDP	0.4261	0.3707	0.3736	0.3911		0.4469	0.3747	0.3731	0.3885
MDL	0.4608	0.4242	0.4526	0.4911		0.434	0.409	0.3941	0.3978
P = 0.1									
	MDLSPC					***MDLSPA***			
MLE	0.4999	0.3462	0.3291	0.3427		0.5181	0.3856	0.3181	0.334
MDL	0.4605	0.3442	0.3807	0.5323		0.4578	0.3954	0.4085	0.61
MDP	0.476	0.3832	0.3713	0.522		0.4474	0.3703	0.3826	0.5144
	MDPLSC					*MDPLSA*			
MLE	0.5429	0.377	0.3146	0.3292		0.4582	0.3734	0.3558	0.3516
MDP	0.5474	0.3707	0.3303	0.3584		0.4939	0.3747	0.3731	0.3416
MDL	0.508	0.4242	0.413	0.5982		0.4252	0.409	0.3941	0.5798

Part 2 of Appendix B. By Method Results for Non-Separated Mixed Weibull

Non-Separated (NS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mixing proportion via AD
MDPL(1-2)C	MDE of the location parameters (sequentially) via CVM
MDPL(1-2)A	MDE of the location parameters (sequentially) via AD
MDPLSC	MDE of the location parameters (simultaneously) via CVM
MDPLSA	MDE of the location parameters (simultaneously) via AD

Table 6. By Integrated Absolute Error, Aggregated Results for Non-Separated Mixed Weibull

	10	20	40	100		10	20	40	100
	{------------------- CVM -------------------}					{------------------------ AD ------------------------}			
P = 0.5	MDPLC					MDPLA			
MLE	0.4375	0.3982	0.3991	0.3698		0.4395	0.4189	0.3905	0.3539
MDP	0.1873	0.1352	0.0957	0.0641		0.2487	0.2088	0.1854	0.1464
MDL 1	0.1923	0.1372	0.0959	0.0669		0.2718	0.2105	0.1951	0.1542
MDL 2	0.2441	0.2228	0.2086	0.2083		0.2574	0.2314	0.1812	0.1675
	MDPLSC					MDPLSA			
MLE	0.4281	0.4065	0.3709	0.3734		0.4315	0.4124	0.3957	0.3795
MDP	0.1799	0.1248	0.0966	0.0667		0.2454	0.2145	0.1889	0.1452
	na	na	na	na		na	na	na	na
MDLS	0.2356	0.2293	0.2026	0.2049		0.2696	0.2256	0.1997	0.1641
P = 0.3	MDPLC					MDPLA			
MLE	0.2652	0.2509	0.2376	0.2378		0.267	0.2562	0.2435	0.2364
MDP	0.2615	0.2344	0.2272	0.2252		0.2531	0.2348	0.2282	0.2258
MDL 1	0.2694	0.2388	0.2343	0.2252		0.2505	0.2411	0.2346	0.2273
MDL 2	0.2452	0.2553	0.2474	0.2371		0.2531	0.2414	0.2357	0.2251
	MDPLSC					MDPLSA			
MLE	0.2551	0.2607	0.2447	0.2437		0.2696	0.2498	0.2493	0.238
MDP	0.2463	0.2329	0.2295	0.224		0.2499	0.2335	0.2301	0.2268
	na	na	na	na		na	na	NA	NA
MDLS	0.257	0.2515	0.2521	0.2312		0.2597	0.2392	0.2307	0.2397
P = 0.1	MDPLC					MDPLA			
MLE	0.226	0.2185	0.2121	0.2089		0.2208	0.2218	0.2204	0.1997
MDP	0.2157	0.2113	0.2198	0.218		0.2091	0.2122	0.2184	0.2165
MDL 1	0.2243	0.2231	0.2242	0.2281		0.2193	0.2179	0.227	0.2273
MDL 2	0.2138	0.2154	0.2181	0.2251		0.2122	0.2138	0.2189	0.2217
	MDPLSC					MDPLSA			
MLE	0.2183	0.2263	0.2102	0.2034		0.2302	0.2298	0.2199	0.2032
MDP	0.2146	0.2169	0.2188	0.2165		0.2141	0.2144	0.2186	0.2163
	na	na	na	na		na	na	NA	na
MDLS	0.2191	0.2201	0.2187	0.2234		0.2162	0.2137	0.2178	0.2189

Appendix C. By Method Results including Confidence Intervals

Part 1a. Well-Separated Mixed Weibull (Shape = $\beta_1 = \beta_2 = 3$)
Part 1b. Well-Separated Mixed Weibull (Shape = $\beta_1 = \beta_2 = 0.9$)
Part 2. Non-Separated Mixed Weibull

The following abbreviations were used in this appendix (ref Chap 3. Methodology).

GENERAL

P	Mixing Proportion
L	Shorthand notation
L(1-2)	Sequential estimation of the location parameters
LS	Simultaneous estimation of the location paramters
N	Sample Size
I-ABS	Error reported via Integrated Absolute Difference
I-MSE	Error reported via Mean Square Error
MDE	Minimum Distance Estimate
MLE	Maximum Likelihood Estamate
CVM	Cramer Von_Mises Test Statistic
AD	Anderson-Darling Test Statistic
SCORE	Number of times better than MLE out of 1000
CI	Confidence Interval at the alpha equal to ten percent level

Part 1a of Appendix C. By Method Results for Well-Separated Mixed Weibull
(Shape = $\beta_1 = \beta_2 = 3$)

WELL-SEPARATED (WS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mxing proportion via AD
MDLSC	MDE of the location parameters simultaneously via CVM
MDLSA	MDE of the location parameters simultaneously via AD
MDLSPC	MDE of the location parameters simultaneously and then mixing proportion via CVM
MDLSPA	MDE of the location parameters simultaneously and then mixing proportion via AD
MDPLSC	MDE of the mixing proportion and then location parameters (simultaneously) via CVM
MDPLSA	MDE of the mixing proportion and then the location parameters (simultaneously) via AD

Table 10. By Method Results for Well Separated Mixed Weibull (Shape = 3)

WS (B = 3)

		MLE	MDPC	MDPA	MDLSC	MDLSA	MDLSPC	MDLSPA	MDPLSA	MDPLSC
P = 0.5	I-ABS	0.2701	0.1987	0.2048	0.2859	0.3097	0.2156	0.2485	0.2431	0.2256
N = 10	CI +/-	2.02E-03	2.47E-03	2.47E-03	2.24E-03	1.63E-03	2.47E-03	1.19E-02	1.63E-03	2.59E-03
	SCORE		828	455	270	206	637	408	387	766
	I-MSE	0.0457	0.0358	0.0399	0.0484	0.0653	0.0378	0.0544	0.0492	0.0425
	CI +/-	6.33E-04	1.37E-03	1.37E-03	1.26E-03		1.37E-03	1.10E-02	8.69E-04	1.41E-03
	SCORE		828	455	270	206	637	408	387	828
N = 20	I-ABS	0.2069	0.129	0.1494	0.2396	0.2269	0.1623	0.1733	0.1491	0.1426
	SCORE		879	733	405	408	815	721	734	813
	I-MSE	0.0245	0.0165	0.0173	0.0416	0.0318	0.0313	0.0229	0.0177	0.0214
	SCORE		880	732	405	408	815	721	734	813
N = 40	I-ABS	0.1431	0.0823	0.1131	0.1765	0.1599	0.0918	0.1192	0.1211	0.0929
	SCORE		890	590	395	348	848	579	554	854
	I-MSE	0.0113	0.0044	0.0066	0.0172	0.0141	0.0066	0.0077	0.0076	0.0068
	SCORE		890	579	395	348	848	579	554	854
N = 100	I-ABS	0.0947	0.0496	0.1032	0.1384	0.0956	0.0592	0.1115	0.1133	0.0569
	SCORE		947	290	218	130	497	148	73	880
	I-MSE	0.0053	0.0015	0.0048	0.0108	0.0052	0.0026	0.0054	0.0058	0.0023
	SCORE		949	290	218	130	497	148	73	880
P=0.3		**MLE**	MDPC	MDPA	MDLSC	MDLSA	MDLSPC	MDLSPA	MDPLSA	MDPLSC
N = 10	I-ABS	0.1684	0.1533	0.1534	0.1948	0.1718	0.1881	0.1759	0.1849	0.1622
	SCORE		502	416	76	468	118	466	481	443
	I-MSE	,0180	0.0146	0.0143	0.0253	0.0219	0.0238	0.0221	0.0255	0.0163
	SCORE		502	416	76	468	118	466	481	443
N = 20	I-ABS	0.1187	0.1247	0.1266	0.1518	0.127	0.1459	0.1374	0.1355	0.1408
	SCORE		290	284	220	447	234	337	295	378
	I-MSE	0.0071	0.0071	0.0074	0.0139	0.0104	0.0128	0.0112	0.0095	0.0127
	SCORE		290	284	220	447	234	337	295	378
N = 40	I-ABS	0.0928	0.1194	0.1198	0.1335	0.0956	0.1401	0.1209	0.1232	0.1223
	SCORE		176	161	225	440	120	276	255	211
	I-MSE	0.0043	0.061	0.0062	0.0115	0.0046	0.0111	0.0064	0.0069	0.0069
	SCORE		175	161	225	440	120	276	255	211
N = 100	I-ABS	0.0725	0.1098	0.1097	0.1129	0.0809	0.1215	0.1121	0.1121	0.1113
	SCORE		98	105	252	396	78	128	150	105
	I-MSE	0.0028	0.0052	0.0052	0.0076	0.0036	0.0075	0.0055	0.0055	0.0055
	SCORE		98	100	252	396	78	128	150	105
P = 0.1		**MLE**	MDPC	MDPA	MDLSC	MDLSA	MDLSPC	MDLSPA	MDPLSA	MDPLSC
N = 10	I-ABS	0.1319	0.136	0.1352	0.1291	0.1361	0.1343	0.1364	0.1352	0.1371
	SCORE		567	514	337	296	609	614	668	597
	I-MSE	0.0089	0.0889	0.0088	0.0088	0.0095	0.0087	0.0089	0.0088	0.0091
	SCORE		560	515	337	296	609	614	668	597
N = 20	I-ABS	0.1064	0.1179	0.1205	0.1182	0.1183	0.1224	0.1241	0.1207	0.1184
	SCORE		442	447	274	241	478	536	517	475
	I-MSE	0.0062	0.0065	0.0069	0.0071	0.0073	0.0071	0.0073	0.0069	0.0065
	SCORE		440	441	274	241	478	536	517	475
N = 40	I-ABS	0.0844	0.1121	0.105	0.1092	0.0942	0.1123	0.1077	0.1093	0.1139
	SCORE		272	239	138	214	332	333	342	284
	I-MSE	0.0039	0.0057	0.0053	0.0058	0.0046	0.0058	0.0053	0.0054	0.0059
	SCORE		270	239	138	214	332	333	342	284
N = 100	I-ABS	0.0583	0.1056	0.1054	0.1073	0.0792	0.1059	0.1031	0.1035	0.1055
	SCORE		90	99	124	284	140	143	149	103

Part 1b of Appendix C. By Method, Results for Well-Separated Mixed Weibull
(Shape = $\beta_1 = \beta_2 = 0.9$)

WELL-SEPARATED (WS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mxing proportion via AD
MDLSC	MDE of the location parameters simultaneously via CVM
MDLSA	MDE of the location parameters simultaneously via AD
MDLSPC	MDE of the location parameters simultaneously and then mixing proportion via CVM
MDLSPA	MDE of the location parameters simultaneously and then mixing proportion via AD
MDPLSC	MDE of the mixing proportion and then location parameters (simultaneously) via CVM
MDPLSA	MDE of the mixing proportion and then the location parameters (simultaneously) via AD

Table 11. By Method Results for the Well-Separated Mixed Weibull (Shape = 0.9)

		MLE	MDPC	MDPA	MDLSC	MDLSA	MDLSPC	MDLSPA	MDPLSC	MDPLSA
P = 0.5	I-ABS	0.7068	0.4538	0.4331	0.7312	0.7244	0.5166	0.5779	0.5661	0.5582
N = 10	CI +/-	2.02E-03	2.42E-03	2.21E-02	2.78E-03	2.31E-02	2.02E-03	3.89E-02	3.57E-03	2.71E-02
	SCORE		830	863	455	474	727	719	714	738
	I-MSE	0.0907	0.0573	0.0557	0.1129	0.1189	0.0737	0.096	0.1002	0.0842
	CI +/-	6.33E-04	7.17E-04	2.46E-03	7.83E-04	5.34E-03	6.33E-04	4.77E-03	1.37E-04	3.60E-03
	SCORE		830	863	455	474	727	719	714	738
N = 20	I-ABS	0.6238	0.3819	0.4146	0.711	0.8168	0.4774	0.5228	0.5108	0.5527
	SCORE		820	869	357	380	684	692	671	698
	I-MSE	0.0658	0.0311	0.0379	0.099	0.1106	0.0642	0.0659	0.0638	0.0674
	SCORE		820	869	357	380	684	692	671	698
N = 40	I-ABS	0.644	0.3235	0.3563	0.8323	0.8383	0.4759	0.4833	0.5012	0.5027
	SCORE		871	824	335	330	699	692	688	637
	I-MSE	0.0639	0.0246	0.0272	0.1145	0.1179	0.0562	0.0611	0.0618	0.0503
	SCORE		871	824	335	330	699	692	688	637
N = 10C	I-ABS	0.7004	0.3617	0.4661	0.9817	1.013	0.5489	0.6286	0.5289	0.5419
	SCORE		846	715	266	139	648	295	195	254
	I-MSE	0.0729	0.0312	0.0418	0.1508	0.1861	0.0818	0.1275	0.0655	0.0608
	SCORE		846	715	266	139	648	295	195	254
P = 0.3		**MLE**	**MDPC**	**MDPA**	**MDLSC**	**MDLSA**	**MDLSPC**	**MDLSPA**	**MDPLSC**	**MDPLSA**
N = 10	I-ABS	0.4267	0.4261	0.4469	0.4703	0.4754	0.4684	0.4908	0.4609	0.434
	SCORE		412	391	471	231	430	188	413	402
	I-MSE	0.0414	0.0406	0.0478	0.0595	0.0584	0.0566	0.06	0.0561	0.0446
	SCORE		412	391	471	231	430	188	413	402
N = 20	I-ABS	0.3847	0.3707	0.3747	0.4601	0.4847	0.4478	0.4618	0.4242	0.409
	SCORE		459	288	379	363	320	376	359	260
	I-MSE	0.0247	0.023	0.0239	0.0486	0.0453	467	0.0403	0.0411	0.0279
	SCORE		459	288	379	363	320	376	359	260
N = 40	I-ABS	0.3724	0.3736	0.3731	0.479	0.4903	0.4434	0.4218	0.4526	0.3941
	SCORE		431	459	268	299	313	372	321	473
	I-MSE	0.0208	0.0206	0.0212	0.0511	0.039	0.0446	0.0301	0.0437	0.0242
	SCORE		431	459	268	299	313	372	321	473
N = 10C	I-ABS	0.3958	0.3912	0.3885	0.6546	0.6074	0.5297	0.4549	0.4911	0.3978
	SCORE		504	519	159	226	299	408	304	554
	I-MSE	0.0217	0.0209	0.0207	0.0786	0.0557	0.0628	0.0376	0.0502	0.0261
	SCORE		500	519	159	226	299	408	304	554
P = 0.1		**MLE**	**MDPC**	**MDPA**	**MDLSC**	**MDLSA**	**MDLSPC**	**MDLSPA**	**MDPLSC**	**MDPLSA**
N = 10	I-ABS	0.4999	0.5474	0.4939	0.4605	0.4578	0.4759	0.4474	0.508	0.4254
	SCORE		519	517	654	570	643	612	643	580
	I-MSE	0.0348	0.0396	0.0351	0.0338	0.0319	0.0347	0.0311	0.0371	0.0284
	SCORE		519	517	654	570	654	612	643	580
N = 20	I-ABS	0.3462	0.388	0.3627	0.3442	0.3954	0.3832	0.3703	0.3704	0.3743
	SCORE		496	519	611	501	607	506	590	514
	I-MSE	0.0196	0.0236	0.0214	0.0213	0.0237	0.0246	0.0214	0.234	0.0219
	SCORE		496	519	611	501	607	506	590	514
N = 40	I-ABS	0.3291	0.3303	0.3285	0.3807	0.4085	0.3713	0.3826	0.4129	0.4302
	SCORE		463	520	526	362	519	419	473	383
	I-MSE	0.0167	0.0161	0.0161	0.0237	0.0235	0.0231	0.0212	0.0265	0.0263
	SCORE		463	520	526	362	519	419	473	383
N = 10C	I-ABS	0.3427	0.3584	0.3416	0.5323	0.61	0.5219	0.5144	0.5982	0.5798
	SCORE		444	450	400	178	51	363	283	56
	I-MSE	0.0159	0.0167	0.0159	0.0389	0.0424	0.0378	0.0347	0.0453	0.0411
	SCORE		444	450	400	178	51	363	283	56

Part 2 of Appendix C. By Method Results for Non-Separated Mixed Weibull

Non-Separated (NS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mxing proportion via AD
MDPL(1-2)C	MDE of the location parameters (sequentially) via CVM
MDPL(1-2)A	MDE of the location parameters (sequentially) via AD
MDPLSC	MDE of the location parameters (simultaneously) via CVM
MDPLSA	MDE of the location parameters (simultaneously) via AD

Table 9. By Method Results for the Non-Separated Mixed Weibull

P = 0.5

		MLE	MDPC	MDPA	MDPL1C	MDPL2C	MDPL1A	MDPL2A	MDPLSC	MDPLSA
N = 10	I-ABS	0.4375	0.1873	0.2487	0.1923	0.2441	0.2574	0.2718	0.2356	0.2696
	CI	1.79E-03	6.46E-04	9.85E-04	4.76E-04	6.17E-04	1.04E-01	9.02E-04	6.92E-04	7.38E-04
		score:	932	791	912	842	790	763	827	723
	I-MSE	0.0513	0.0194	0.0289	0.0206	0.0349	0.0332	0.0382	0.337	0.0289
	CI	2.60E-04	2.10E-04	2.00E-04	9.12E-04	1.39E-04	2.31E-04	2.00E-04	1.55E-04	1.56E-04
		score:	932	791	912	842	790	763	827	723
N = 20	I-ABS	0.3982	0.1352	0.2088	0.1372	0.2228	0.2106	0.2314	0.2293	0.2226
		score:	979	840	967	892	852	815	804	894
	I-MSE	0.0418	0.0108	0.0175	0.0113	0.0301	0.0177	0.0235	0.0234	0.0301
		score:	979	840	967	892	852	815	804	894
N = 40	I-ABS	0.3991	0.0957	0.1812	0.0959	0.2086	0.1854	0.1951	0.2026	0.1997
		score:	991	883	980	874	867	858	884	839
	I-MSE	0.0385	0.005	0.0111	0.0051	0.0259	0.0116	0.0154	0.0251	0.0161
		score:	991	883	980	874	867	858	884	839
N = 10(	I-ABS	0.3698	0.0641	0.1464	0.0669	0.2083	0.1543	0.1675	0.2049	0.1641
		score:	990	896	983	853	887	877	864	888
	I-MSE	0.0333	0.0021	0.0065	0.0028	0.0255	0.0077	0.0114	0.025	0.0108
		score:	990	896	983	853	887	877	864	888
P = 0.3										
N=10	I-ABS	0.2652	0.2452	0.2477	0.2615	0.2694	0.2531	0.2505	0.257	0.2599
		score:	534	516	517	487	542	552	513	548
	I-MSE	0.0193	0.0178	0.0181	0.0212	0.0297	0.0202	0.0223	0.027	0.025
		score:	534	516	517	487	542	552	513	548
N = 20	I-ABS	0.2509	0.2344	0.2348	0.2388	0.2553	0.2411	0.2414	0.2515	0.2392
		score:	519	579	573	466	620	549	479	562
	I-MSE	0.0156	0.0152	0.0149	0.0169	0.0263	0.0177	0.0192	0.0266	0.0184
		score:	519	579	573	466	620	549	479	562
N = 40	I-ABS	0.2376	0.2272	0.2282	0.2343	0.2474	0.2346	0.2357	0.2526	0.2307
		score:	541	577	532	430	545	549	448	577
	I-MSE	0.0128	0.0127	0.0127	0.0146	0.026	0.014	0.0155	0.0261	0.0146
		score:	541	577	532	430	545	549	448	577
N = 10(	I-ABS	0.2378	0.2252	0.2251	0.2252	0.2371	0.2258	0.2273	0.2312	0.2397
		score:	564	563	571	457	551	557	502	379
	I-MSE	0.0137	0.0111	0.0111	0.0117	0.0233	0.0116	0.0129	0.0133	0.0234
		score:	564	563	571	457	551	557	502	379
P = 0.1										
N=10	I-ABS	0.226	0.2157	0.2091	0.2243	0.2138	0.2193	0.2122	0.2191	0.2162
		score:	500	464	459	509	476	535	506	514
	I-MSE	0.0111	0.011	0.0108	0.0125	0.0128	0.0121	0.0119	0.013	0.0121
		score:	500	464	459	509	476	535	506	514
N = 20	I-ABS	0.2185	0.2113	0.2122	0.2231	0.2154	0.2179	0.2138	0.2201	0.2137
		score:	443	432	444	495	465	476	482	509
	I-MSE	0.0105	0.0107	0.0106	0.0121	0.0127	0.0117	0.0117	0.0108	0.0117
		score:	443	432	444	495	465	476	482	509
N = 40	I-ABS	0.2121	0.2198	0.2184	0.2242	0.2181	0.227	0.2189	0.2187	0.2178
		score:	409	409	405	457	378	445	412	452
	I-MSE	0.01	0.0109	0.0109	0.0121	0.0131	0.0121	0.0119	0.0129	0.0116
			409	409	405	457	378	445	412	452
N = 10(	I-ABS	0.2089	0.218	0.2165	0.2281	0.2251	0.2273	0.2217	0.2234	0.2189
			344	305	326	322	278	309	369	180

Appendix D. Single Sample Runs for Each Method

Part 1a. Well-Separated Mixed Weibull (Shape = $\beta_1 = \beta_2 = 3$)
Part 1b. Well-Separated Mixed Weibull (Shape = $\beta_1 = \beta_2 = 0.9$)
Part 2. Non-Separated Mixed Weibull

The following abbreviations were used in this appendix (ref Chap 3. Methodology).

GENERAL

P	Mixing Proportion
L	Shorthand notation
L(1-2)	Sequential estimation of the location parameters
LS	Simultaneous estimation of the location paramters
N	Sample Size
I-ABS	Error reported via Integrated Absolute Difference
I-MSE	Error reported via Mean Square Error
MDE	Minimum Distance Estimate
MLE	Maximum Likelihood Estamate
CVM	Cramer Von_Mises Test Statistic
AD	Anderson-Darling Test Statistic
SCORE	Number of times better than MLE out of 1000
CI	Confidence Interval at the alpha equal to ten percent level

Part 1a of Appendix D. Single Sample Run for Well-Separated Mixed Weibull (Shape = $\beta_1 = \beta_2 = 3$)

WELL-SEPARATED (WS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mxing proportion via AD
MDLSC	MDE of the location parameters simultaneously via CVM
MDLSA	MDE of the location parameters simultaneously via AD
MDLSPC proportion	MDE of the location parameters simultaneously and then mixing via CVM
MDLSPA proportion	MDE of the location parameters simultaneously and then mixing via AD
MDPLSC (simultaneously)	MDE of the mixing proportion and then location parameters via CVM
MDPLSA	MDE of the mixing proportion and then the location parameters (simultaneously) via AD

Sample for MDLSPC

COUNT = 10

For this seed, PDF1 SUBCOUNT = 5
For this seed, PDF2 SUBCOUNT = 5

FOS = 5.2528127194075
LOS = 10.3968802790539

TRUE(1-3)=	3.0	5.0	0.5	
TRUE(4-7)=	3.0	15.0	0.5	0.5

INITIAL MLE ...

4.195	5.253	0.379	
2.393	10.249	0.100	0.444

The function value = 993.076

MDL1C = 5.3616323769752

MDL2C = 10.2017474577584

Revised MLE:

2.786	5.362	0.265	
3.719	10.202	0.148	0.444

The function value = 993.196

MDPC = 0.50635426732366

Revised MLE:

2.786	5.362	0.265	
3.719	10.202	0.148	0.506

The function value = 993.265

Sample for MDLSPA

COUNT = 10

For this seed, PDF1 SUBCOUNT = 6
For this seed, PDF2 SUBCOUNT = 4

FOS = 5.1946464500596
LOS = 10.3603981497592

TRUE(1-3)=	3.0	5.0	0.5	
TRUE(4-7)=	3.0	15.0	0.5	0.5

INITIAL MLE ...

1.500	5.195	0.270	
3.384	10.226	0.100	0.556

The function value = 994.869

MDL1A = 5.2185976658729

MDL2A = 10.0683577962409

Revised MLE:	1.301	5.219	0.237	
	8.747	10.068	0.259	0.556

The function value = 995.046

MDPA = 0.5040108024391

Revised MLE:	1.301	5.219	0.237	
	8.747	10.068	0.259	0.504

The function value = 995.094

Sample for MDPC & MDPLSC

COUNT = 10

For this seed, CVM1 = 4
For this seed, CVM2 = 6
FOS = 5.2167419148015
LOS = 10.4808556003185

TRUE(1-3)= 3.0 5.0 0.5
TRUE(4-7)= 3.0 10.0 0.5 0.5

INITIAL MLE ...
2.612 5.217 0.206
2.688 10.033 0.318 0.400

The function value = 997.726

MDPC = 0.47368033826620

Revised MLE: 2.612 5.217 0.206
2.688 10.033 0.318 0.474

The function value = 998.005

MDL1C = 5.2321927644561

MDL2C = 10.0480235732543

Revised MLE: 2.363 5.232 0.189
2.513 10.048 0.301 0.474

The function value = 997.983

Sample for MDPA & MDPLSA

COUNT = 10

For this seed, CVM1 = 3
For this seed, CVM2 = 7

FOS = 5.1845715038107
LOS = 10.688001991511

TRUE(1-3)= 3.0 5.0 0.5
TRUE(4-7)= 3.0 10.0 0.5 0.5

INITIAL MLE ...

2.240 5.185 0.212
1.728 10.135 0.275 0.400

The function value = 999.233

MDPA = 0.41868033949555

Revised MLE: 2.240 5.185 0.212
1.728 10.135 0.275 0.419

The function value = 999.363

MDL1A = 5.23942244183989

MDL2A = 10.0314915981867

Revised MLE: 1.419 5.239 0.150
2.555 10.031 0.394 0.419

The function value = 999.493

Part 1b of Appendix D. Single Sample Runs for Well-Separated Mixed Weibull
(Shape = $\beta_1 = \beta_2 =$ 0.9)

WELL-SEPARATED (WS)

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mxing proportion via AD
MDLSC	MDE of the location parameters simultaneously via CVM
MDLSA	MDE of the location parameters simultaneously via AD
MDLSPC proportion	MDE of the location parameters simultaneously and then mixing via CVM
MDLSPA proportion	MDE of the location parameters simultaneously and then mixing via AD
MDPLSC (simultaneously)	MDE of the mixing proportion and then location parameters via CVM
MDPLSA	MDE of the mixing proportion and then the location parameters (simultaneously) via AD

Sample for Well-Separated Mixed Weibull

COUNT = 10

For this seed, PDF1 SUBCOUNT = 6
For this seed, PDF2 SUBCOUNT = 4

FOS = 5.0368507578617
LOS = 15.878272472209

TRUE(1-3)=	0.9	5.0	0.5	
TRUE(4-7)=	0.9	15.0	0.5	0.5

INITIAL MLE ...

1.471	5.037	0.800	
1.876	14.995	0.560	0.556

The function value = 1009.239

MDL1C = 5.0924738156716

MDL2C = 14.601669949573

Revised MLE: 1.366	5.092	0.729		
	3.721	14.602	0.990	0.556

The function value = 1009.082

MDPC = 0.56214434883697

Revised MLE:	1.366	5.092	0.729	
	3.723	14.602	0.990	0.562

The function value = 1009.083

Sample for MDLSP

COUNT = 10

For this seed, PDF1 SUBCOUNT = 6
For this seed, PDF2 SUBCOUNT = 4

FOS = 5.4557230964974
LOS = 16.149584079853

TRUE(1-3)= 0.9 5.0 0.5
TRUE(4-7)= 0.9 15.0 0.5 0.5

INITIAL MLE ...
0.967 5.456 0.580
0.946 15.103 0.440 0.600

The function value = 1007.572

MDL1A = 5.4988539184841

MDL2A = 15.102514094312

Revised MLE: 0.500 5.499 1.019
0.500 15.103 0.400 0.600

The function value = 1016.478

MDPA = 0.55720491595654

Revised MLE: 0.500 5.499 1.119
0.500 15.103 0.349 0.557

The function value = 1016.625

Sample for MDPC & MDPLSC

COUNT = 10

For this seed, PDF1 SUBCOUNT = 5
For this seed, PDF2 SUBCOUNT = 5

FOS = 5.3964532271532
LOS = 15.670319494084

TRUE(1-3)=	0.9	5.0	0.5	
TRUE(4-7)=	0.9	15.0	0.5	0.5

INITIAL MLE ...

1.082	5.396	0.475	
1.000	15.044	0.312	0.400

The function value = 1006.323

MDPC = 0.50743033751215

Revised MLE:	1.111	5.396	0.515	
	0.997	15.044	0.285	0.507

The function value = 1006.27

MDL1C = 5.4447156052831

MDL2C = 15.044130141483

Revised MLE:	0.500	5.445	1.601	
	0.500	15.044	0.366	0.507

The function value = 1015.236

Sample for MDPA & MDPLSA

COUNT = 10

For this seed, PDF1 SUBCOUNT = 6
For this seed, PDF2 SUBCOUNT = 4

FOS = 5.0019494304593
LOS = 15.365247463217

TRUE(1-3)= 0.9 5.0 0.5
TRUE(4-7)= 0.9 15.0 0.5 0.5

INITIAL MLE ...

1.079 5.002 0.945
0.995 15.013 0.155 0.600

The function value = 1021.092

MDPA = 0.54220491629182

Revised MLE: 0.500 5.002 0.290
1.352 14.987 0.217 0.542

The function value = 1009.938

MDL1A = 5.104479945233

MDL2A = 13.494479945233

Revised MLE: 1.076 5.104 0.894
10.000 13.494 1.731 0.542

The function value = 1008.670

Part 2 of Appendix D. Single Sample Runs for **Non-Separated** Mixed Weibull

Non-Separated (NS) Methods

MLE	Maximum Likelihood Estimation
MDPC	MDE of the mixing proportion via CVM
MDPA	MDE of the mxing proportion via AD
MDPL(1-2)C	MDE of the location parameters (sequentially) via CVM
MDPL(1-2)A	MDE of the location parameters (sequentially) via AD
MDPLSC	MDE of the location parameters (simultaneously) via CVM
MDPLSA	MDE of the location parameters (simultaneously) via AD

Sample Run for MDPC & MDPLC

COUNT = 10

For this seed, PDF1 SUBCOUNT = 4
For this seed, PDF2 SUBCOUNT = 6

FOS = 5.0186625015626
LOS = 6.2767729694646

TRUE(1-3)=	4.0	5.0	0.5	
TRUE(4-7)=	1.0	5.0	0.5	0.5

INITIAL MLE ...

5.292	5.019	0.510	
0.715	5.019	0.296	0.400

The function value = -6.577

MDPC = 0.48315982820764

Revised MLE:

5.073	5.019	0.481	
0.999	5.019	0.444	0.483

The function value = -0.642

MDL1C = 4.9674008566246

Revised MLE:

5.190	4.967	0.531	
1.000	5.019	0.406	0.483

The function value = -0.598

MDL2C = 5.0044015888227

Revised MLE:

10.000	4.967	0.560	
0.500	5.004	0.133	0.483

The function value = -1.500

Sample Run for MDPA & MDPLA

COUNT = 10

For this seed, ADT1 = 5
For this seed, ADT2 = 5

FOS = 5.2319329793737
LOS = 6.5415151342765

TRUE(1-3)=	4.0	5.0	0.5	
TRUE(4-7)=	1.0	5.0	0.5	0.5

INITIAL MLE ...

6.223	5.232	0.255	
0.996	5.232	0.695	0.400

The function value = -4.577

MDPA = 0.47065982860115

Revised MLE:	1.651	5.232	0.150	
	0.906	5.232	0.733	0.471

The function value = 1002.156

MDL1A = 5.2450449387151

Revised MLE:	2.824	5.245	0.150	
	0.896	5.232	0.483	0.471

The function value = 0.066

MDL2A = 5.2142278811673

Revised MLE:	7.250	5.245	0.246	
	0.500	5.214	0.217	0.471

The function value = -4.164

Sample Run for MDPC & MDPLSC

COUNT = 10
PDF1 SUBCOUNT = 6
PDF2 SUBCOUNT = 4
FOS = 5.0235825959959
LOS = 6.5095421369077

TRUE(1-3)= 4.0 5.0 0.5
TRUE(4-7)= 1.0 5.0 0.5 0.5

INITIAL MLE: 6.259 5.024 0.418
0.929 5.024 0.673 0.600

The function value = 998.351

MDPC = 0.49131966251611

Revised MLE: 5.728 5.024 0.440
0.873 5.024 0.588 0.491

The function value = -3.811

MDL1C = 5.0464192499178

MDL2C = 5.0708774965770

Revised MLE: 10.000 5.046 0.422
1.251 5.071 0.698 0.491

The function value = 995.873

Sample Run for MDPA & MDPLSA

COUNT = 10

PDF1 SUBCOUNT = 4
PDF2 SUBCOUNT = 6

FOS = 5.0537649570432
LOS = 6.1571406477402

TRUE(1-3)=	4.0	5.0	0.5	
TRUE(4-7)=	1.0	5.0	0.5	0.5

INITIAL MLE ...

4.523	5.054	0.392	
0.888	5.054	0.399	0.400

The function value = -1.053

MDPA = 0.51868033726038

Revised MLE:	4.176	5.054	0.392	
	1.000	5.054	0.482	0.519

The function value = 0.885

MDL1 = 5.0191057105040

MDL2 = 5.0394743882462

Revised MLE:	3.458	5.019	0.396	
	0.987	5.039	0.521	0.519

The function value = 0.819

Appendix E: Sample FORTRAN for Non-Separated Mixed Weibull

The following abbreviations were used in this appendix.

PDF	Probability Density Function
CDF	Cumulative Distribution Function
COUNT	Sample size
PDF SUBCOUNT	Sample size per PDF
TRUE	True Solution for seven parameter Mixed Weibull
(1-3)	PDF 1 Shape (β_1), Location (δ_1) and Scale (η_1) Parameters
(4-6)	PDF 2 Shape (β_2), Location (δ_2) and Scale (η_2) Parameters
(7)	Mixing Proportion (p)
MLE	Maximum Likelihood Estimate(s)
MDPC	Minimum Distance Estimate for the Mixing Proportion using the Cramer Von-Mises Statistic
p_{true}	True mixing proportion
f1t(x)	True PDF for Population #1
f2t(x)	True PDF for Population #2
g_{true}	True Mixed Weibull PDF
F1t(x)	True CDF for Population #1
F2t(x)	True CDF for Population #2
G_{true}	True Mixed Weibull CDF
p_{est}	Estimated mixing proportion
f1(x)	Estimated PDF for Population #1
f2(x)	Estimated PDF for Population #2
g_{est}	Estimated Mixed Weibull PDF
F1(x)	Estimated CDF for Population #1
F2(x)	Estimated CDF for Population #2
G_{est}	Estimated Mixed Weibull CDF

```
      PROGRAM NSPLC

      INTEGER COUNT,I,J,K,SCabs21,SCmse21,
     6     SCabs31,SCmse31,SCabs32,SCmse32,
     6     SCabs41,SCmse41,RUN,SEED,SEED1,DIV

      REAL*8 MLE(1:7),RAW(5000),TRUE(1:7),
     6 MDLCVM1(0:3),MDLCVM2(0:3),MDPCVM1(0:3),MDPCVM2(0:3),
     6 TRUEPDF1(0:3), TRUEPDF2(0:3),
     6 INTABS,INTMSE,INTABS1,INTMSE1,INTABS2,INTMSE2,
     6 ph3TOTINTABS,ph3TOTINTMSE,ph4TOTINTABS,ph4TOTINTMSE,
     6 ph2totintabs,ph2totintmse,ph1totintabs,ph1totintmse,
     6 SUM1INTABS, SUM1INTMSE,sum2intabs,sum2intmse,
     6 sum3intabs,sum3intmse,sum4intabs,sum4intmse,
     6 MEAN4INTABS,MEAN4INTMSE,MEAN1INTABS,MEAN1INTMSE,
     6 MEAN2INTABS,MEAN2INTMSE,MEAN3INTABS,MEAN3INTMSE,
     6    XGUESS(7), X(7),XLB(7),XUB(7), Di, P

      EXTERNAL RSORT
      COMMON / GLOBALDATA / COUNT, raw


      DATA TRUE/4.0,5.0,0.5,  1.0,5.0,0.5,   0.5/
      DATA SEED / 242234567.0 /
      DATA COUNT / 40 /

C *** ROEs
c     1) D1 CANNOT be .GT. FOS
c     2) D2 cannot be .GT. LOS
C     -NOTE: to prevent underflow, do not allow any to equal zero

      DATA   XLB/0.5E0,5.0E0, 0.15E0, 0.5E0,5.0E0,0.1E0,0.4E0/,
     6     XUB/10.0E0,5.01E0,3.1E0,10.0E0,6.0E0,3.1E0,0.6E0/
      DATA XGUESS/5.0E0,5.0E0,0.5E0,  1.5E0,5.0E0,0.5E0,0.5E0/



** MAIN

              RUN = 0

      CALL READ ( RUN, SEED, SCABS21, SCABS31,
     6     SCABS31,SCabs32,SCABS41,
     6 MEAN4INTABS,MEAN4INTMSE,MEAN1INTABS,MEAN1INTMSE,
     6 MEAN2INTABS,MEAN2INTMSE,MEAN3INTABS,MEAN3INTMSE )

              RUN = RUN + 1

              SEED1 = SEED
              sum1intabs = 0.0
              sum1intmse = 0.0
```

```
              sum2intabs = 0.0
              sum2intmse = 0.0
              sum3intabs = 0.0
              sum3intmse = 0.0


        DO 17  J = RUN, 5000
              DIV = J - (RUN-1)


C45           Generate samples

              CALL MONTE (SEED1,TRUE)


C       SORT an 'CVMjustable' (M&S, p 453) array of of observations subsequently referred to as 'raw'
C             - an real array of length equal to the count


              CALL RSORT
        PRINT *, ' main successfully exited the CALL RSORT '


C       After sorting but before calling DBCONG, reset XUB-D1 = FOS
C             RESET ALTERED INITIAL CONDITIONS

              XLB(7) = 0.4
              XUB(7) = 0.6
              XGUESS(7) = (1.0 - 0.5)

              XLB(2) = 0.5
              XUB(2)= RAW(1)
              XGUESS(2) = RAW(1)

              XLB(5) = 0.5
              XUB(5)= RAW(COUNT-4)
              XGUESS(5) = 5.0

              XUB(3) = RAW(COUNT-3)
              XUB(6) = RAW(COUNT-3)


c             PRINT*,'UPDATED XUB(2) =',XUB(2)


C       INITIAL MLE (MLE PH1, VARY ALL SEVEN PARS)
C       PASS OUT MLE FOR PDF1 ONLY REQURIED FOR MDE PH1
```

```
      PRINT*,' ENTERING INITIAL MLE ...'

            CALL SMLE (X,XGUESS,XLB,XUB)

      PRINT*,' SUCCESSFULLY EXITED INITIAL MLE ...'


                  MLE(1) = X(1)
                  MLE(2) = X(2)
                  MLE(3) = X(3)


C     Now, calculate the error for phase 1
C
            INTABS = 0.0
            INTMSE = 0.0
            INTABS1 = 0.0
            INTMSE2 = 0.0
            INTABS1 = 0.0
            INTMSE2 = 0.0
            ph1totINTABS = 0.0
            ph1totINTMSE = 0.0

      DO 26  I = 1,3
            MDLCVM1 (0) = X(7)
            MDLCVM2 (0) = X(7)
            TRUEPDF1(0) = TRUE(7)
            TRUEPDF2(0) = TRUE(7)
            MDLCVM1 (I) = X(I)
            MDLCVM2 (I) = X(I+3)
            TRUEPDF1(I) = TRUE(I)
            TRUEPDF2(I) = TRUE(I+3)
26    CONTINUE


C           * calc subtotal error for PDF 1 *

            CALL INTEGRATE (TRUEPDF1,MDLCVM1,INTABS1,INTMSE1)

C           * calc subtotal error for PDF 1 *

            CALL INTEGRATE (TRUEPDF2,MDLCVM2,INTABS2,INTMSE2)

            ph1TOTINTABS = INTABS1 + INTABS2
            ph1TOTINTMSE = INTMSE1 + INTMSE2

            PRINT*,'SUB-TOTALS FOR MLE phase 1 and J =:  ',J

            PRINT*,'  ph1TOTINTABS =',ph1TOTINTABS
```

```
            PRINT*,' ph1TOTINTMSE =',ph1TOTINTMSE
            PRINT*,'   INTABS1 =',INTABS1
            PRINT*,'   INTMSE1 =',INTMSE1
            PRINT*,'   INTABS2 =',INTABS2
            PRINT*,'   INTMSE2 =',INTMSE2

            SUM1INTABS = SUM1INTABS + ph1TOTINTABS
            SUM1INTMSE = SUM1INTMSE + ph1TOTINTMSE
            MEAN1INTABS = ((SUM1INTABS) + (MEAN1INTABS*(J-DIV)))/J
            MEAN1INTMSE = ((SUM1INTMSE) + (MEAN1INTMSE*(J-DIV)))/J


      PRINT*,' MEAN-ph1-INT-ABS =',MEAN1INTABS
      PRINT*,' MEAN-ph1-INT-MSE =',MEAN1INTMSE



C     *** INSERT MIN DISTANCE PROPORTION ***

C      MDE ON P ( FIX PREV Six PARS, VARY P )
C
            DO 33  K = 1,7
                  MLE(K) = X(K)
33          CONTINUE


            CALL PSMDE (MLE,P)

            IF (P .LT. XLB(7)) THEN
                  P = XLB(7)
            END IF
            IF (P .GT. XUB(7)) THEN
                  P = XUB(7)
            END IF


C     2nd MLE (MLE Ph2, fix D1, vary six pars)
C     NOW, set rerun MLE with MDPCVM fixed:


            IF (P .LT. 1.0E-6) THEN
                  P = 1.0E-6
            END IF

            PRINT*,'P = ',P
            XLB(7) = (P - 1.0E-7)
            XUB(7) = (P + 1.0E-7)
            XGUESS(7) = P

            CALL SMLE(X,XGUESS,XLB,XUB)
```

```
C       Now, calculate the error for phase 2
C
              INTABS = 0.0
              INTMSE = 0.0
              INTABS1 = 0.0
              INTMSE1 = 0.0
              INTABS2 = 0.0
              INTMSE2 = 0.0
              ph4totINTABS = 0.0
              ph4totINTMSE = 0.0

        DO 29  I = 1,3
              MDPCVM1 (0) = X(7)
              MDPCVM2 (0) = X(7)
              TRUEPDF1(0) = TRUE(7)
              TRUEPDF2(0) = TRUE(7)
              MDPCVM1 (I) = X(I)
              MDPCVM2 (I) = X(I+3)
              TRUEPDF1(I) = TRUE(I)
              TRUEPDF2(I) = TRUE(I+3)
29      CONTINUE


              CALL INTEGRATE (TRUEPDF1,MDPCVM1,INTABS1,INTMSE1)

              CALL INTEGRATE (TRUEPDF2,MDPCVM2,INTABS2,INTMSE2)

              ph4TOTINTABS = INTABS1 + INTABS2
              ph4TOTINTMSE = INTMSE1 + INTMSE2

              PRINT*,'SUB-TOTALS FOR MLE PHASE 4 and J =:  ',J
              PRINT*,'  ph4TOTINTABS =',ph4TOTINTABS
              PRINT*,'  ph4TOTINTMSE =',ph4TOTINTMSE
              PRINT*,'    INTABS1 =',INTABS1
              PRINT*,'    INTMSE1 =',INTMSE1
              PRINT*,'    INTABS2 =',INTABS2
              PRINT*,'    INTMSE2 =',INTMSE2

              SUM4INTABS = SUM4INTABS + ph4TOTINTABS
              SUM4INTMSE = SUM4INTMSE + ph4TOTINTMSE
              MEAN4INTABS = ((SUM4INTABS) + (MEAN4INTABS*(J-DIV)))/J
              MEAN4INTMSE = ((SUM4INTMSE) + (MEAN4INTMSE*(J-DIV)))/J

        PRINT*,'  MEAN-ph4-ABS =',MEAN4INTABS
        PRINT*,'  MEAN-ph4-MSE =',MEAN4INTMSE
```

```
C       INITIAL MDE (MDE PH1, FIX D1, VARY Six PARS)
C

                CALL SMDE (MLE,Di)

C       2nd MLE (MLE Ph2, fix D1, vary six pars)
C       NOW, set rerun MLE with MDLCVM fixed:



                PRINT*,'D1 = ',Di
                XLB(2) = (Di - 1.0E-7)
                XUB(2) = (Di + 1.0E-7)
                XGUESS(2) = Di



                CALL SMLE(X,XGUESS,XLB,XUB)

C       Now, calculate the error for phase 2
C
                INTABS = 0.0
                INTMSE = 0.0
                INTABS1 = 0.0
                INTMSE2 = 0.0
                INTABS1 = 0.0
                INTMSE2 = 0.0
                ph2totINTABS = 0.0
                ph2totINTMSE = 0.0

** now, calculate error

        DO 27  I = 1,3
                MDLCVM1 (0) = X(7)
                MDLCVM2 (0) = X(7)
                TRUEPDF1(0) = TRUE(7)
                TRUEPDF2(0) = TRUE(7)
                MDLCVM1 (I) = X(I)
                MDLCVM2 (I) = X(I+3)
                TRUEPDF1(I) = TRUE(I)
                TRUEPDF2(I) = TRUE(I+3)
27      CONTINUE


                CALL INTEGRATE (TRUEPDF1,MDLCVM1,INTABS1,INTMSE1)
```

```
          CALL INTEGRATE (TRUEPDF2,MDLCVM2,INTABS2,INTMSE2)

          ph2TOTINTABS = INTABS1 + INTABS2
          ph2TOTINTMSE = INTMSE1 + INTMSE2

          PRINT*,'SUB-TOTALS FOR MLE phase 2 and J =:  ',J

          PRINT*,' ph2TOTINTABS =',ph2TOTINTABS
          PRINT*,' ph2TOTINTMSE =',ph2TOTINTMSE
          PRINT*,'    INTABS1 =',INTABS1
          PRINT*,'    INTMSE1 =',INTMSE1
          PRINT*,'    INTABS2 =',INTABS2
          PRINT*,'    INTMSE2 =',INTMSE2

          SUM2INTABS = SUM2INTABS + ph2TOTINTABS
          SUM2INTMSE = SUM2INTMSE + ph2TOTINTMSE
          MEAN2INTABS = ((SUM2INTABS) + (MEAN2INTABS*(J-DIV)))/J
          MEAN2INTMSE = ((SUM2INTMSE) + (MEAN2INTMSE*(J-DIV)))/J


      PRINT*,' MEAN-INT-ph2-ABS =',MEAN2INTABS
      PRINT*,' MEAN-INT-ph2-MSE =',MEAN2INTMSE


C     2nd MDE (calculate Min Distance for D2)

          MLE(1) = X(4)
          MLE(2) = X(5)
          MLE(3) = X(6)


          CALL SMDE (MLE,Di)


C     3Rd MLE (MLE Ph2, fix D1, vary six pars)
C     NOW, set rerun MLE with MDLCVM fixed:

c         IF (Di .LT. 1.0) THEN
c             Di = 1.0
c         END IF

          PRINT*,'D2 = ',Di
          XLB(5) = (Di - 1.0E-7)
          XUB(5) = (Di + 1.0E-7)
          XGUESS(5) = Di


          CALL SMLE(X,XGUESS,XLB,XUB)

C
C     Now, calculate the error for phase 3
```

```
C
          INTABS = 0.0
          INTMSE = 0.0
          INTABS1 = 0.0
          INTMSE2 = 0.0
          INTABS1 = 0.0
          INTMSE2 = 0.0
          ph3totINTABS = 0.0
          ph3totINTMSE = 0.0

      DO 28  I = 1,3
          MDLCVM1 (0) = X(7)
          MDLCVM2 (0) = X(7)
          TRUEPDF1(0) = TRUE(7)
          TRUEPDF2(0) = TRUE(7)
          MDLCVM1 (I) = X(I)
          MDLCVM2 (I) = X(I+3)
          TRUEPDF1(I) = TRUE(I)
          TRUEPDF2(I) = TRUE(I+3)
28    CONTINUE


          CALL INTEGRATE (TRUEPDF1,MDLCVM1,INTABS1,INTMSE1)

          CALL INTEGRATE (TRUEPDF2,MDLCVM2,INTABS2,INTMSE2)

          ph3TOTINTABS = INTABS1 + INTABS2
          ph3TOTINTMSE = INTMSE1 + INTMSE2

          PRINT*,'SUB-TOTALS FOR MLE phase 3 and J =:  ',J
          PRINT*,'  ph3TOTINTABS =',ph3TOTINTABS
          PRINT*,'  ph3TOTINTMSE =',ph3TOTINTMSE
          PRINT*,'     INTABS1 =',INTABS1
          PRINT*,'     INTMSE1 =',INTMSE1
          PRINT*,'     INTABS2 =',INTABS2
          PRINT*,'     INTMSE2 =',INTMSE2

          SUM3INTABS = SUM3INTABS + ph3TOTINTABS
          SUM3INTMSE = SUM3INTMSE + ph3TOTINTMSE
          MEAN3INTABS = ((SUM3INTABS) + (MEAN3INTABS*(J-DIV)))/J
          MEAN3INTMSE = ((SUM3INTMSE) + (MEAN3INTMSE*(J-DIV)))/J


      PRINT*,' MEAN-ph3-INT-ABS =',MEAN3INTABS
      PRINT*,' MEAN-ph3-INT-MSE =',MEAN3INTMSE



          IF (PH4TOTINTABS .LT. PH1TOTINTABS) THEN
                SCABS41 = SCABS41 + 1
          END IF
```

```
            IF (PH4TOTINTMSE .LT. PH1TOTINTMSE) THEN
                  SCMSE41 = SCMSE41 + 1
            END IF
ccc

ccc
            IF (PH2TOTINTABS .LT. PH1TOTINTABS) THEN
                  SCABS21 = SCABS21 + 1
            END IF
            IF (PH2TOTINTMSE .LT. PH1TOTINTMSE) THEN
                  SCMSE21 = SCMSE21 + 1
            END IF
ccc
            IF (PH3TOTINTABS .LT. PH1TOTINTABS) THEN
                  SCABS31 = SCABS31 + 1
            END IF
            IF (PH3TOTINTMSE .LT. PH1TOTINTMSE) THEN
                  SCMSE31 = SCMSE31 + 1
            END IF
ccc
            IF (PH3TOTINTABS .LT. PH2TOTINTABS) THEN
                  SCABS32 = SCABS32 + 1
            END IF
            IF (PH3TOTINTMSE .LT. PH2TOTINTMSE) THEN
                  SCMSE32 = SCMSE32 + 1
            END IF
C
C     In case fail to finish, print summary statistics
C

      PRINT*,' ... SUMMARY STATISTICS for seed =',SEED,' J =',J
      PRINT*,'-------------------------------------------------'
      PRINT*,'  SCabs21=',SCabs21
      PRINT*,'  SCmse21=',SCmse21
      PRINT*,'  SCabs31=',SCabs31
      PRINT*,'  SCmse31=',SCmse31
      PRINT*,'  SCabs41=',SCabs41
      PRINT*,'  SCmse41=',SCmse41
      PRINT*,'  SCabs32=',SCabs32
      PRINT*,'  SCmse32=',SCmse32

      PRINT*,'  MEAN-ph1-INT-ABS =',MEAN1INTABS
      PRINT*,'  MEAN-ph1-INT-MSE =',MEAN1INTMSE
      PRINT*,'  MEAN-INT-ph2-ABS =',MEAN2INTABS
      PRINT*,'  MEAN-INT-ph2-MSE =',MEAN2INTMSE
      PRINT*,'  MEAN-ph3-INT-ABS =',MEAN3INTABS
      PRINT*,'  MEAN-ph3-INT-MSE =',MEAN3INTMSE
      PRINT*,'  MEAN-ph4-INT-ABS =',MEAN4INTABS
      PRINT*,'  MEAN-ph4-INT-MSE =',MEAN4INTMSE
      PRINT*,'-------------------------------------------------'
```

```
      CALL TIMER (J,SCABS21,SCABS31,
   6     SCABS31,SCabs32,SCABS41,
   6 MEAN4INTABS,MEAN4INTMSE,MEAN1INTABS,MEAN1INTMSE,
   6 MEAN2INTABS,MEAN2INTMSE,MEAN3INTABS,MEAN3INTMSE)


17    CONTINUE


      PRINT*,' ... SUMMARY STATISTICS for seed =',SEED1,' J =',J
      PRINT*,'-----------------------------------------------'
      PRINT*,'  SCabs21=',SCabs21
      PRINT*,'  SCmse21=',SCmse21
      PRINT*,'  SCabs31=',SCabs31
      PRINT*,'  SCmse31=',SCmse31
      PRINT*,'  SCabs41=',SCabs41
      PRINT*,'  SCmse41=',SCmse41
      PRINT*,'  SCabs32=',SCabs32
      PRINT*,'  SCmse32=',SCmse32

      PRINT*,'  MEAN-ph1-INT-ABS =',MEAN1INTABS
      PRINT*,'  MEAN-ph1-INT-MSE =',MEAN1INTMSE
      PRINT*,'  MEAN-INT-ph2-ABS =',MEAN2INTABS
      PRINT*,'  MEAN-INT-ph2-MSE =',MEAN2INTMSE
      PRINT*,'  MEAN-ph3-INT-ABS =',MEAN3INTABS
      PRINT*,'  MEAN-ph3-INT-MSE =',MEAN3INTMSE
      PRINT*,'  MEAN-ph4-INT-ABS =',MEAN4INTABS
      PRINT*,'  MEAN-ph4-INT-MSE =',MEAN4INTMSE

      END


C*****************************************************************************
C* SUBROUTINE SUMMARY OUTPUT - READS AND COUNTS
C*      ALSO declares the real array 'raw' as an adjustable array
C*      returns a one-dimensional real array raw called raw   P455          *
C*      into ascending order by the selection-sort algorithm               *
C*****************************************************************************
C490


      SUBROUTINE SUMMARY(SEED,J,DIV,SCabs21,SCmse21,
   6     SCabs31,SCmse31,SCabs32,SCmse32,
   6     SCabs41,SCmse41,
   6 MEAN4INTABS,MEAN4INTMSE,MEAN1INTABS,MEAN1INTMSE,
   6 MEAN2INTABS,MEAN2INTMSE,MEAN3INTABS,MEAN3INTMSE)

* Constant:
*   INTEGER: The maximum number of data items that can be stored
```

```
      INTEGER LIMIT
      PARAMETER  ( LIMIT = 5000 )

      INTEGER COUNT,J,DIV,SCabs21,SCmse21,
     6   SCabs31,SCmse31,SCabs32,SCmse32,
     6   SCabs41,SCmse41

      REAL*8 TRUE(1:7),SEED,
     6 MEAN4INTABS,MEAN4INTMSE,MEAN1INTABS,MEAN1INTMSE,
     6 MEAN2INTABS,MEAN2INTMSE,MEAN3INTABS,MEAN3INTMSE
* Variables:
*   COUNT:The total number of raw observations read from frm
*   raw:Array of real observations

      INTEGER I, count, ERRCOD
      LOGICAL ENDFIL
      REAL*8 raw(5000), X(7)

      COMMON / GLOBALDATA / count, raw


C-OPEN THE INPUT FILE
      INTEGER INP, IOUT, J
      CHARACTER CPD*30
      IOUT=4
      INP=3
c     WRITE(6,*) ' '
c140  WRITE(6,FMT='($,A)') ' INPUT FILE NAME = '
c     READ(5,150) CPD
c150  FORMAT(A30)


      cpd = 'rnslcpc.RES'

      OPEN(UNIT=INP,FILE=CPD,ACCESS='APPEND')


c     WRITE (INP,*) '----------------------'



C     WRITE (NOUT,99999) X, FL, (IPARAM(L), L=3,5)

C

      WRITE (INP,99) J,DIV,
     6   SCABS21,SCmse21,SCabs31,
     6   SCmse31,SCabs41,SCmse41,
     6   SCabs32,SCmse32,MEAN1INTABS,
     6   MEAN1INTMSE,MEAN2INTABS,MEAN2INTMSE,
     6   MEAN3INTABS,MEAN3INTMSE,MEAN4INTABS,
```

```
   6    MEAN4INTMSE

99      FORMAT(9(1X,I3),3X,8(1X,F10.7))

c       WRITE (INP,*) '------------------------'

c       CLOSE (INP)


        IF ( ERRCOD .LT. 0) THEN
                ENDFIL = .TRUE.
        END IF
        IF (ENDFIL) GO TO 20


20      PRINT *, ' END OF FILE REACHED WITHIN RSORT SUBROUTINE '
c       PRINT *,'WITHIN READ, COUNT =', COUNT

        END



C******************************************************************************
C* SUBROUTINE READ OUTPUT - READS AND COUNTS
C*      ALSO declares the real array 'raw' as an adjustable array
C*      returns a one-dimensional real array raw called raw    P455          *
C*      into ascending order by the selection-sort algorithm                 *
C******************************************************************************
C490
        SUBROUTINE READ (RUN,SEED,SCABS21,SCABS31,
   6    SCABS31,SCabs32,SCABS41,
   6 MEAN4INTABS,MEAN4INTMSE,MEAN1INTABS,MEAN1INTMSE,
   6 MEAN2INTABS,MEAN2INTMSE,MEAN3INTABS,MEAN3INTMSE)

        INTEGER LIMIT
        PARAMETER ( LIMIT = 5000 )


        INTEGER count,ERRCOD,RUN
        INTEGER LIMIT, SCABS21, SCABS31,
   6    SCABS31,SCabs32,SCABS41
        REAL*8
   6 MEAN4INTABS,MEAN4INTMSE,MEAN1INTABS,MEAN1INTMSE,
   6 MEAN2INTABS,MEAN2INTMSE,MEAN3INTABS,MEAN3INTMSE

        LOGICAL ENDFIL
        REAL*8 raw(5000), SEED
```

```
        COMMON / GLOBALDATA / count, raw

C-      OPEN THE INPUT FILE

        INTEGER INP, IOUT, J
        CHARACTER CPD*30
        IOUT=4
        INP=3
c       WRITE(6,*) ' '
c140    WRITE(6,FMT='($,A)') ' INPUT FILE NAME = '
c       READ(5,150) CPD
c150    FORMAT(A30)

        cpd = 'nslcpc.in'

        OPEN(UNIT=INP,FILE=CPD)

        CONTINUE
        REWIND(INP)

        READ (INP,*)  RUN
        READ (INP,*)  seed
        READ (INP,*)  SCABS21
        READ (INP,*)  SCABS31
        READ (INP,*)  SCABS41
        READ (INP,*) MEAN1INTABS
        READ (INP,*) MEAN1INTMSE
        READ (INP,*) MEAN2INTABS
        READ (INP,*) MEAN2INTMSE


        READ (INP,*) MEAN3INTABS
        READ (INP,*) MEAN3INTMSE
        READ (INP,*) MEAN4INTABS
        READ (INP,*) MEAN4INTMSE


        END


C*****************************************************************************
C* SUBROUTINE TIMER OUTPUT - READS AND COUNTS
C*      ALSO declares the real array 'raw' as an adjustable array
C*      returns a  one-dimensional real array raw called raw    P455          *
C*      into ascending order by the selection-sort algorithm                 *
C*****************************************************************************
C490
        SUBROUTINE TIMER (J,SCABS21,SCABS31,
   6    SCABS31,SCabs32,SCABS41,
   6 MEAN4INTABS,MEAN4INTMSE,MEAN1INTABS,MEAN1INTMSE,
```

```
 6 MEAN2INTABS,MEAN2INTMSE,MEAN3INTABS,MEAN3INTMSE)

      INTEGER LIMIT, SCABS21, SCABS31,
 6    SCABS31,SCabs32,SCABS41
      REAL*8
 6 MEAN4INTABS,MEAN4INTMSE,MEAN1INTABS,MEAN1INTMSE,
 6 MEAN2INTABS,MEAN2INTMSE,MEAN3INTABS,MEAN3INTMSE

      PARAMETER  ( LIMIT = 5000 )
      INTEGER I, count, ERRCOD
      LOGICAL ENDFIL
      REAL*8 raw(5000), X(7)

      COMMON / GLOBALDATA / count, raw


C-    OPEN THE INPUT FILE

      INTEGER INP, IOUT, J
      CHARACTER CPD*30
      IOUT=4
      INP=3

      cpd = 'nslcpc.out'

      OPEN(UNIT=INP,FILE=CPD)

      REWIND(INP)


      PRINT*,'WITHIN SUBR TIMER'
      PRINT*,'RUN     =',RUN
      PRINT*,'SCABS21    =',SCABS21
      PRINT*,'SCABS31    =',SCABS31


      WRITE (INP,*)  J
      WRITE (INP,*) SCABS21
      WRITE (INP,*) SCABS31
      WRITE (INP,*) SCABS41

      WRITE (INP,*) MEAN1INTABS
      WRITE (INP,*) MEAN1INTMSE
      WRITE (INP,*) MEAN2INTABS
      WRITE (INP,*) MEAN2INTMSE


      WRITE (INP,*) MEAN3INTABS
      WRITE (INP,*) MEAN3INTMSE
      WRITE (INP,*) MEAN4INTABS
      WRITE (INP,*) MEAN4INTMSE
```

```
      END

******************************************************************
* THIS FUNCTION  RETURNS A MONTE CARLO ESTIMATE
*
******************************************************************
C166

      REAL*8 FUNCTION MONTE (SEED1,TRUE)
c             ( GENERATES A UNIFORM RANODM COUNTBER )

      CHARACTER CPD*30
      INTEGER  COUNT, I, CVM1, CVM2
      REAL*8 RAW(5000),TRUE(1:7),SEED1,SEED2,SEED3,
     6  U1,U2,U3,RG1,RG2,RG3
      INTRINSIC DEXP, DLOG
      EXTERNAL RSORT,RG1,RG2,RG3
      COMMON /GLOBALDATA/ COUNT, raw

            CVM1 = 0
            CVM2 = 0

            SEED2 = SEED1 + 10.0
            SEED3 = SEED1 - 10.0

      DO 55  I = 1, COUNT

                  U1 = RG1 (SEED1)
                  U2 = RG2 (SEED2)
                  U3 = RG3 (SEED3)

            IF (U1 .LT. TRUE(7)) THEN
                  CVM1 = CVM1 + 1
c                 IF (TRUE(2) .LT. 1.0E-7) THEN
c                       RAW(I)=TRUE(3)
c    6            *(((-1.0*DLOG(1.0-U2))**(1.0/TRUE(1))))
c                 ELSE
                        RAW(I)=TRUE(3)
     6            *(((-1.0*DLOG(1.0-U2))**(1.0/TRUE(1))))
     6            +TRUE(2)
c                 END IF
            ELSE
                  CVM2 = CVM2 + 1
c                 IF (TRUE(5) .LT. 1.0E-7) THEN
c                       PRINT*,'CAUTION, TRUE(5)=',TRUE(5)
c                       RAW(I)=TRUE(6)
c    6      *(((-1.0*DLOG(1.0-U3))**(1.0/TRUE(4))))
c                 ELSE
                        RAW(I)=TRUE(6)
     6            *(((-1.0*DLOG(1.0-U3))**(1.0/TRUE(4))))
```

```
   6                    +TRUE(5)
c                     END IF
               END IF

c       PRINT*,' SEED1 =',SEED1
c       PrINT*,'   U1 =',  U1
c       PRINT*,' SEED2 =',SEED2
c       PRINT*,'   U2 =',  U2
c       PRINT*,' SEED3 =',SEED3
c       PRINT*,'   U3 =',  U3

C       PRINT*,' FOR I = ',I,'  RAW(I) = ',RAW(I)


55      CONTINUE


               PRINT*,'For this seed, CVM1 =',CVM1
               PRINT*,'For this seed, CVM2 =',CVM2

        END

*********************************************************************
* THIS FUNCTION  RETURNS A UNIFORM RANDOM COUNTBER
*        *
*********************************************************************


        REAL*8 FUNCTION RG1 (SEED1)
c              ( GENERATES A UNIFORM RANODM COUNTER )

        REAL*8 PROD, SEMI, SEED1
        INTRINSIC DMOD

               PROD = 16807.D0*SEED1

               SEMI = DMOD(PROD,2147483647.D0)

               RG1 = SEMI*0.4656613E-9
C250
               SEED1 = SEMI

        RETURN

        END



*********************************************************************
```

```
* THIS FUNCTION  RETURNS A UNIFORM RANDOM COUNTBER
*        *
******************************************************************

      REAL*8 FUNCTION RG2 (SEED2)
c            ( GENERATES A UNIFORM RANDOM COUNTER )

      REAL*8 PROD, SEMI, SEED2
      INTRINSIC DMOD

            PROD = 16807.D0*SEED2

            SEMI = DMOD(PROD,2147483647.D0)

            RG2 = SEMI*0.4656613E-9

            SEED2 = SEMI

      RETURN

      END

******************************************************************
* THIS FUNCTION  RETURNS A UNIFORM RANDOM COUNTBER
*        *
******************************************************************

      REAL*8 FUNCTION RG3 (SEED3)
c            ( GENERATES A UNIFORM RANDOM COUNTER )

      REAL*8 PROD, SEMI, SEED3
      INTRINSIC DMOD

            PROD = 16807.D0*SEED3

            SEMI = DMOD(PROD,2147483647.D0)

            RG3 = SEMI*0.4656613E-9
C250
            SEED3 = SEMI

      RETURN

      END
c256
```

```
*******************************************************************************
* SUBROUTINE RSORT
* Sorts count = COUNT values in a one-dimensional real array 'raw     P409
*
*        into ascending order by the selection-sort algorithm
*******************************************************************************
* This is not a big deal for our two population Mixed Weibull
* Lets assume initially that the mixing proportion  (P) is equal to 0.5
C
      SUBROUTINE RSORT
      INTEGER LIMIT
      PARAMETER (LIMIT = 5000)
      INTEGER count, I, J
      REAL*8 LOW, raw(5000)
      COMMON / GLOBALDATA / COUNT, raw
      EXTERNAL RSWAP


c     PRINT*,' WITHIN RSORT COUNT = ',COUNT

      DO 20  I = 1, count-1
            LOW = I
            DO 10  J = (I+1), count
                  IF (raw(J) .LT. RAW(LOW) ) THEN
                     LOW = J
                  END IF
10          CONTINUE
                  CALL RSWAP ( raw(I), RAW(LOW) )
20    CONTINUE
      PRINT *, 'WITHIN RSORT, FOS = ',RAW(1)
      PRINT *, 'WITHIN RSORT, LOS = ',RAW(I)


c           DO 30 J = 1,COUNT
c                 PRINT*,RAW(J)
c30         CONTINUE

c     PRINT ' (30(1X,F8.3))',(raw(J),J=1,count)
C200
      RETURN
      END
C
C     ****************************************************************************
C     * SUBROUTINE RSWAP
C     *  swaps two real values
C225  ****************************************************************************

      SUBROUTINE RSWAP (r1,r2)
```

```
      REAL*8 r1, r2, temp
      EXTERNAL RSORT
            temp = r1
            r1 = r2
            r2 = temp
      RETURN
      END


**********************************************************************************
* MLE         SUBROUTINE MLE
**********************************************************************************

*** original data set was file:  "raw40"
*     a well-separated (GT 5) data set

      SUBROUTINE SMLE(X,XGUESS,XLB,XUB)

      INTEGER N
      PARAMETER (N=7)

      INTEGER IPARAM(7), ITP, L, NOUT, I, COUNT
      REAL*8 FL,FLSCALE,GRCVM,FLOG,RPARAM(7),
   &    X(7),XGUESS(7),XLB(7),XSCALE(7),XUB(7),
   &    raw(5000), PDF, MPDF, TOL1, TOL2
      EXTERNAL DBCONG, FLOG, GRCVM, RECVM, RSORT, UMACH,
   &        PDF, MPDF, PDPDF, DU4INFC
      INTRINSIC DEXP, DLOG
      COMMON / GLOBALDATA / COUNT, raw


      DATA XSCALE/   7*1.0E-1/,    FLSCALE/1.0E0/

c     CRITICAL INITIALIZATION

            DO 14  I = 1,7
                  X(I) = 0.0
14          CONTINUE



C                 All the bounds are provided
      ITP = 0

C                 Default parameters are not used

C     *****************************************************
C     * DOUBLE PRECISION BLOCK ...
C     *****************************************************
```

```
C-TOL1 = SQRT(sum of (XSCALE(I)*XGUESS(I))**2) for I= 1,...,N
C-TOL2 = 2 NORM OF X-SCALE

      TOL1 = 0.2
      TOL2 = 0.2

      CALL DU4INF(IPARAM, RPARAM)
      IPARAM(1) = (1)
c     IPARAM(2) = (15)
      IPARAM(3) = (1000)
      IPARAM(4) = (2000)
      IPARAM(5) = (2000)
c     RPARAM(1) = ((eps)**(2/3))
c     RPARAM(2) = ((eps)**(2/3))
c     RPARAM(3) = MAX(1.0E-20,((eps)**(2/3)))
c     RPARAM(4) = MAX(1.0E-20,((eps)**(2/3)))
c     RPARAM(5) = (100*((eps)**(2/3)))
c     RPARAM(6) = (1000*MAX(TOL1, TOL2))
c     RPARAM(7) = 10.0

      CALL DBCONG(FLOG,GRCVM,N,XGUESS,ITP,XLB,XUB,XSCALE,
   &                 FLSCALE, IPARAM, RPARAM, X, FL)

c                          Print results

      CALL UMACH (2, NOUT)

      WRITE (NOUT,99999) X, FL, (IPARAM(L), L=3,5)

C
99999 FORMAT('Soln is ',6X,7F8.3,//,'The function ',
   &    'value = ',F8.3,//,'The number of iterations is',
   &    10X,I3,/,'The number of function evaluations is',
   &    I3, /,' The number of grCVMient evaluations is ',I3)

      END



*     BASIC FUNCTIONS:
*******************************************************************************
* f - WEIBULL PDF DENOTED BY f where J = 1 or 2  where
*********************************************************************************

      REAL*8 FUNCTION PDF(raw, count, X, I, J)

      INTEGER count, I, J

      REAL*8 raw(count), X(7)
      INTRINSIC DLOG, DEXP
```

```
* critical initialization

        PDF = 0.0

* start execution

        IF (J .EQ. 1)  THEN

*  recall that if this value is GT 10.0, all pfs will be = 0.0
*   i.e if ((raw - D1) / E1 ) GT 3, DEXP will generate an underflowc
**  prevent underflow

                IF ((raw(I) .LT. X(2))
     6          .OR.(ABS(raw(I)-X(2)).LT.1E-10)
     6       .OR.(ABS(((raw(I)-X(2))/X(3))**X(1)).GT.100.0)
     6          .OR.(ABS(raw(I)-X(2)).LT.1E-10)) THEN
                        PDF = 0.0
c122
                ELSE IF (ABS(X(1)-1.0) .LE. 1.0E-10) THEN
                        PDF = (X(1)/X(3))
     6                  *1.0
     6          *(DEXP(-1.0*(((raw(I)-X(2))/X(3))**X(1))))
                ELSE
                        PDF = (X(1)/X(3))
     6                  *(((raw(I)-X(2))/X(3))**(X(1)-1.0))
     6          *(DEXP(-1.0*(((raw(I)-X(2))/X(3))**X(1))))
                END IF
        ELSE
c  if (raw(I) .LT. D2) THEN does not apply to second pdf
c140
                IF ((raw(I) .LT. X(5))
     6          .OR. (ABS(((raw(I)-X(5))/X(6))**X(4)).GT.100.0)
     6          .OR. (ABS(raw(I)-X(5)).LT.1E-10)) THEN
                        PDF = 0.0
                ELSE IF (ABS(X(4)-1.0) .LT. 1E-10) THEN
                                PDF = (X(4)/X(6))
     6                          *1.0
     6  *(DEXP(-1.0*(((raw(I)-X(5))/X(6))**X(4))))
                ELSE
                                PDF = (X(4)/X(6))
     6          *(((raw(I)-X(5))/X(6))**(X(4)-1.0))
     6  *(DEXP(-1.0*(((raw(I)-X(5))/X(6))**X(4))))
                END IF
        END IF
c       PRINT *, ' PDFij = ', PDF
        RETURN
        END
*********************************************************************************
*  g  -  MIXED WEIBULL PDF DENOTED BY g(RAW; ALPHA) = g(I)
```

```
********************************************************************************

        REAL*8 FUNCTION MPDF(raw, count, X, I)
        INTEGER count, I
        REAL*8 raw(5000), X(7), PDF, pdf1, pdf2
        INTRINSIC DEXP, DLOG
        EXTERNAL PDF
* initialization

        pdf2 = 0.0
        pdf1 = 0.0
        MPDF = 0.0
** start
        pdf1 = PDF(raw,count,X,I,1)
        pdf2 = PDF(raw,count,X,I,2)
C       PRINT*, 'pdf1 = ', pdf1
C       PRINT*, 'pdf2 = ', pdf2
c       PRINT*, ' ... within MPDF, X = '
c       PRINT*, ' X(1-3) = ',X(1),X(2),X(3)
c       PRINT*,           ' X(4-6)   ',X(4),X(5),X(6),X(7)


        IF (ABS(pdf1) .LT. 1.0E-10) THEN
c               PRINT*, 'pdf1 is ZERO for I = ', I
                        MPDF = ((1.0-X(7))*pdf2)
        ELSE IF (ABS(pdf2) .LT. 1.0E-10) THEN
C               PRINT*, 'pdf2 is ZERO! for I = ', I
                        MPDF = (X(7)*pdf1)
        ELSE IF((ABS(pdf1).LT.1.0E-10).AND.(ABS(pdf2).LT.1.0E-10)) THEN
                PRINT*,'WARNING, BOTH PDFS=ZERO (NOT POSSIBLE)'
                PRINT*,'  pdf1 = ', pdf1
                PRINT*,'  pdf2 = ', pdf2
        ELSE
                MPDF = (X(7)*pdf1)+((1.0-X(7))*pdf2)
        END IF
C       PRINT*, 'mpdf = ', MPDF
        RETURN
        END
********************************************************************************
* EQN#5  - calc of  GRCVMIENT vector  stored in GR(I)
********************************************************************************

        SUBROUTINE GRCVM(N, X, GR )
        INTEGER N, I, count, J, K, L
        REAL*8 GR(N), X(7), pf(7),
    &           raw(5000), temp(7),
    &           PDF, MPDF, mt
        INTRINSIC DEXP, DLOG
        COMMON / GLOBALDATA / COUNT, raw
        EXTERNAL PDPDF, PDF, MPDF

c       PRINT*, ' did GRCVM receive? COUNT = ',COUNT
```

```
c        PRINT*, ' X(1-3) = ',X(1),X(2),X(3)
c        PRINT*, ' X(4-6) ',X(4),X(5), X(6),X(7)
C    6   ' raw = ',raw(I)

**       CRITICAL INITIALIZATION **

         DO 33  K = 1,7
                GR(K) = 0.0
33       continue


**** CALCULATIONS



c- check null


         DO 20  I = 1,count


C               ** initialization **

                mt=0.0

                DO 44  L = 1,7
                       temp(L) = 0.0
44              continue


C        ** MAIN **


         CALL PDPDF(pf, raw, count, X, I)

         mt = MPDF(raw,count,X,I)


c               PRINT*, ' COUNT = ', I
c               PRINT*, '  PF1-3=', PF(1),PF(2),PF(3)
c               PRINT*, '  PF4-6=', PF(4),PF(5),PF(6)

c        PRINT*, 'mpdf = ', mt




         DO 77  J = 1,3

                IF (mt .LT. 1E-10) THEN
```

```
                temp(J) = 0.0

            ELSE IF(ABS(pf(J)).LT.1.0E-50) THEN
                        temp(J) = 0.0
            ELSE
                        temp(J) = (pf(J)/mt)*X(7)
            END IF

            GR(J) = GR(J) - temp(J)

77          continue


      DO 88  J = 4,6

            IF (mt .LT. 1E-10) THEN

                temp(J) = 0.0

            ELSE IF(ABS(pf(J)).LT.1.0E-50)THEN
                        temp(J) = 0.0
            ELSE
                        temp(J) = (pf(J)/mt)*(1.0-X(7))
            END IF

            GR(J) = GR(J) - temp(J)

88          continue

            IF (mt .LT. 1E-10) THEN

                temp(J) = 0.0

            ELSE IF(ABS((PDF(raw,count,X,I,1))
     6          -(PDF(raw,count,X,I,2)))
     6          .LT.  1.0E-10) THEN

                temp(7) = 0.0
            ELSE
                temp(7)=(((PDF(raw,count,X,I,1))
     6          -(PDF(raw,count,X,I,2))))/mt
            END IF

            GR(7) = GR(7) - temp(7)

20    CONTINUE

c     PRINT*,'WITHIN GRCVM, THE CALCULATED GRCVM IS = '
c     PRINT*,'  GR(1-3) = ',GR(1),GR(2),GR(3)
c     PRINT*,'  GR(4-6) = ',GR(4),GR(5),GR(6),GR(7)
      RETURN
      END
```

```
c         ************************************************************************
c         * EQN#5a  -  calc of partial derivatives wrt pdf                              *
c         ************************************************************************
c320
          SUBROUTINE PDPDF (pf, raw, count, X, I)
          INTEGER count, I, K
          REAL*8  raw(5000), X(7), pf(7),
   &              B1, D1, E1, B2, D2, E2
          INTRINSIC DEXP, DLOG

C         PRINT*,'WITHIN PDPDF, COUNT = ', COUNT

C         ****  CALCULATIONS WITHIN FOR THE SUMMATIONS  ****
                  B1 = X(1)
                  D1 = X(2)
                  E1 = X(3)
                  B2 = X(4)
                  D2 = X(5)
                  E2 = X(6)
C         PRINT*, ' did PDPDF receive? = ', count,X(1),E1
          IF (ABS(E1) .LT. 1.0E-2) THEN
                  PRINT*, ' WARN, E1 = 0, PARTIALS DIV ZERO !?'
          END IF
          IF (ABS(B1) .LE. 1.0E-1) THEN
                  PRINT*, ' WARNING, B1 = ZERO !?'
          END IF
c339
          IF (ABS(E2) .LT. 1.0E-2) THEN
                  PRINT*, ' WARNING, E2 = 0, PARTIALS DIV ZERO !?'
          END IF
          IF (ABS(B2) .LE. 1.0E-1) THEN
                  PRINT*, ' WARN, B2 = 0, ERROR,  ZERO !?'
          END IF

** initialization

          DO 33  K = 1,7
                  pf(K) = 0.0
33        continue

* recall that if this value is GT 10.0, all pfs will be = 0.0
*  i.e if ((raw - D1) / E1 ) GT 3, DEXP will generate an underflow

          IF ( (RAW(I) .LT. D1)
   6        .OR.(ABS(((raw(I)-D1)/E1)**B1) .GT. 100.00)
   6              .OR.(ABS(raw(I)-D1).LT.1E-10)) THEN
                  pf(1) = 0.0
                  pf(2) = 0.0
                  pf(3) = 0.0
          ELSE IF ((B1-1.0) .LT. 1E-10) THEN
                  pf(1) =(((1.0/E1)*1.0
   6              *(DEXP(-((raw(I)-D1)/E1)**B1)))
```

```
 6               +((B1/E1)*1.0
 6               *DLOG(((raw(I)-D1)/E1))
 6               *(DEXP(-((raw(I)-D1)/E1)**B1)))
 6                     -((B1/E1)*1.0
 6               *(((raw(I)-D1)/E1)**B1)
 6                     *DLOG(((raw(I)-D1)/E1))
 6                     *(DEXP(-((raw(I)-D1)/E1)**B1))))
                 pf(2) = ((((B1**2)/E1)*1.0
 6               *((((raw(I)-D1)/E1)**B1)/(raw(I)-D1))
 6                     *(DEXP(-((raw(I)-D1)/E1)**B1))))
                 pf(3)=(((-B1/(E1**2))*1.0
 6               *(DEXP(-((raw(I)-D1)/E1)**B1)))
 6                     +(((B1**2)/(E1**2))*1.0
 6               *(((raw(I)-D1)/E1)**B1)
 6               *(DEXP(-((raw(I)-D1)/E1)**B1))))
           ELSE
                 pf(1) =(((1.0/E1)*(((raw(I)-D1)/E1)**(B1-1.0))
 6               *(DEXP(-((raw(I)-D1)/E1)**B1)))
 6               +((B1/E1)*(((raw(I)-D1)/E1)**(B1-1.0))
 6               *DLOG(((raw(I)-D1)/E1))
 6               *(DEXP(-((raw(I)-D1)/E1)**B1)))
 6                     -((B1/E1)*(((raw(I)-D1)/E1)**(B1-1.0))
 6               *(((raw(I)-D1)/E1)**B1)
 6                     *DLOG(((raw(I)-D1)/E1))
 6                     *(DEXP(-((raw(I)-D1)/E1)**B1))))
                 pf(2) = (((-B1/E1)
 6                     *(((raw(I)-D1)/E1)**(B1-1.0))
 6                     *((B1-1.0)/(raw(I)-D1))
 6                     *(DEXP(-((raw(I)-D1)/E1)**B1)))
 6               +(((B1**2)/E1)*(((raw(I)-D1)/E1)**(B1-1.0))
 6               *((((raw(I)-D1)/E1)**B1)/(raw(I)-D1))
 6                     *(DEXP(-((raw(I)-D1)/E1)**B1))))
                 pf(3)=(((-B1/(E1**2))
 6                     *(((raw(I)-D1)/E1)**(B1-1.0))
 6               *(DEXP(-((raw(I)-D1)/E1)**B1)))
 6               -((B1/(E1**2))*(((raw(I)-D1)/E1)**(B1-1.0))
 6               *(B1-1.0)*(DEXP(-((raw(I)-D1)/E1)**B1)))
 6               +(((B1**2)/(E1**2))*(((raw(I)-D1)/E1)**(B1-1.0))
 6               *(((raw(I)-D1)/E1)**B1)
 6               *(DEXP(-((raw(I)-D1)/E1)**B1))))
c405
           END IF

** CHECK VALIDITY OF OBSERVATION FOR SECOND DISTRIBUTION

           IF ((raw(I).LT. D2)
 6         .OR. (ABS(raw(I)-D2).LT.1.0E-10)
 6         .OR. (ABS(((raw(I)-D2)/E2)**B2).GT.100.00))THEN
                 pf(4) = 0.0
                 pf(5) = 0.0
                 pf(6) = 0.0
           ELSE IF ((B2-1.0) .LT. 1.0E-10) THEN
```

```
                pf(4) =(((1.0/E2)*1.0
     6          *(DEXP(-((raw(I)-D2)/E2)**B2)))
     6          +((B2/E2)*1.0
     6          *DLOG(((raw(I)-D2)/E2))
     6          *(DEXP(-((raw(I)-D2)/E2)**B2)))
     6                -((B2/E2)*1.0
     6          *(((raw(I)-D2)/E2)**B2)
     6                *DLOG(((raw(I)-D2)/E2))
     6                *(DEXP(-((raw(I)-D2)/E2)**B2))))
                pf(5) = ((((B2**2)/E2)*1.0
     6          *((((raw(I)-D2)/E2)**B2)/(raw(I)-D2))
     6               *(DEXP(-((raw(I)-D2)/E2)**B2))))
                pf(6)=(((-B2/(E2**2))*1.0
     6          *(DEXP(-((raw(I)-D2)/E2)**B2)))
     6                +(((B2**2)/(E2**2))*1.0
     6          *(((raw(I)-D2)/E2)**B2)
     6          *(DEXP(-((raw(I)-D2)/E2)**B2))))
          ELSE
                pf(4) =(((1.0/E2)*(((raw(I)-D2)/E2)**(B2-1.0))
     6          *(DEXP(-((raw(I)-D2)/E2)**B2)))
     6          +((B2/E2)*(((raw(I)-D2)/E2)**(B2-1.0))
     6          *DLOG(((raw(I)-D2)/E2))
     6          *(DEXP(-((raw(I)-D2)/E2)**B2)))
     6                -((B2/E2)*(((raw(I)-D2)/E2)**(B2-1.0))
     6          *(((raw(I)-D2)/E2)**B2)
     6                *DLOG(((raw(I)-D2)/E2))
     6                *(DEXP(-((raw(I)-D2)/E2)**B2))))
                pf(5) = (((-B2/E2)*(((raw(I)-D2)/E2)**(B2-1.0))
     6                *((B2-1.0)/(raw(I)-D2))
     6                *(DEXP(-((raw(I)-D2)/E2)**B2)))
     6                +(((B2**2)/E2)*(((raw(I)-D2)/E2)**(B2-1.0))
     6          *((((raw(I)-D2)/E2)**B2)/(raw(I)-D2))
     6                *(DEXP(-((raw(I)-D2)/E2)**B2))))
                pf(6)=(((-B2/(E2**2))
     6                *(((raw(I)-D2)/E2)**(B2-1.0))
     6          *(DEXP(-((raw(I)-D2)/E2)**B2)))
     6          -((B2/(E2**2))*(((raw(I)-D2)/E2)**(B2-1.0))
     6          *(B2-1.0)*(DEXP(-((raw(I)-D2)/E2)**B2)))
     6          +(((B2**2)/(E2**2))*(((raw(I)-D2)/E2)**(B2-1.0))
     6          *(((raw(I)-D2)/E2)**B2)
     6          *(DEXP(-((raw(I)-D2)/E2)**B2))))
          END IF

          RETURN
          END

C********************************************************************************
C FL  -  CALC THE NATURAL DLOG- LIKLIHOOD SUBROUTINE                  *
C        (subroutine was necessary because function do not handle summations)
C        MULTIPLY BY NEGATIVE ONE FOR imsl TO CONVERT TO MAX PROBLEM
*
```

```
C******************************************************************************
C       for the complete sample case, count:
C
        SUBROUTINE FLOG ( N, X, FL )

        INTEGER N, I, count
        REAL*8 raw(5000), FL, X(7),PDF, MPDF, MTI, Q
        intrinsic DLOG
        COMMON / GLOBALDATA / COUNT, raw

        EXTERNAL PDF, MPDF
**** CALCULATIONS WITHIN FOR THE SUMMATIONS ****

        FL = 0.0

        DO 10  I = 1, count
                MTI = 0.0
                Q = 0.0
C       PRINT*,' for count =',I,' RAW(I)=',RAW(I)
                MTI = MPDF(raw,count,X,I )
C       PRINT*,' MPDFi=',MTI
                IF (MTI .LE. 1.0E-10) THEN
                        Q = -1000.0
                ELSE
                        Q = DLOG(MTI)
                END IF
c5000    PRINT*,' NATURAL DLOG OF MPDFi =', Q
                FL = FL - Q
10      CONTINUE
c       PRINT*,'VALUE OF DLOG-LIKLIHOOD( = -FL ): FL = ', FL
        RETURN
        END




****************************************************************************
*
* SMDE          SUBR SMDE
*
****************************************************************************

        SUBROUTINE SMDE (MLE, Di)

        INTEGER I, COUNT
        CHARACTER*3  WHICH
        REAL*8 err, tol, reps, Di
        PARAMETER( err  = 1.0E-6)
c                                       (* error and tolerance are limits*)
        PARAMETER( tol  = 1.0E-6)
c                                       (*  used in the numerical routines *)
```

```
      PARAMETER( reps = 1000)
c                              (* the number of DATA generated *)
c        - DECLARE FUNCTIONS
c-       - DECLARE  ARRAYS -

         REAL*8          RAW(5000),
   6             MLE(1:3),
   6             MDLCVM(1:3)
c                                        (* WEIBULL random  variables      *)
c                          (* position  0 is  the number of  RVs.*)

c                                        (* evaluation  values  for  different  paras        *)
         COMMON / GLOBALDATA / COUNT,RAW

C        PRINT*,'WITHIN SUBR SMDE ALL CALCS BASED ON ORIGINAL MLES:'
C        PRINT*,'  COUNT', COUNT
C        PRINT*,'   MLE(1) ',MLE(1)
C        PRINT*,'   MLE(2) ',MLE(2)
C        PRINT*,'   MLE(3) ',MLE(3)

C        CRITICAL INITIALIZATION

         Di = 0.0


         DO 99  I = 1,3
                 MDLCVM (I) = MLE (I)
99       CONTINUE


         WHICH = 'CVM'
         CALL GSEARCH ( WHICH, MDLCVM )
C                PRINT*,'THE MINIMUM DISTANCE LOCATION VIA CVM :'
C                PRINT*,'MDLCVM(1) ',MDLCVM(1)
C                PRINT*,'MDLCVM(2) ',MDLCVM(2)
C                PRINT*,'MDLCVM(3) ',MDLCVM(3)

         Di = MDLCVM(2)


         END

*************************************************************************
*        INPUTS: DESIRED STAT: WHICH,  HOW : PARAS
*  STARTING AT "A" SEARCHES IN "DIRECTION" UNTIL THE FUNCTION STOPS
*  DECREASING.  THEN BEGINS A GOLDEN SEARCH ON THE LAST TWO
*  INTERVALS JUST PRIOR TO THE FUNCTION INCREASING.
*  THE LOCATION SHOULD HAVE BEEN BOUNDED BELOW THE FIRST ORDER STATIS
*************************************************************************


         SUBROUTINE GSEARCH ( WHICH,PARAM )
```

```
      INTEGER REPS,K,NUM
      REAL*8 err, tol

      PARAMETER( err  = 1.0E-6)
c                                    (* error and tolerance are limits*)
      PARAMETER( tol  = 1.0E-6)
c                                    (*  used in the numerical routines *)
    PARAMETER( reps = 1000)
c
      CHARACTER*3  WHICH
      INTEGER  COUNT, REPCOUNT
      REAL*8  GOF, A, B,
C                                    (* CURRENT RIGHT AND LEFT ENDPOINTS *)
  6            AB,
C                                (*  MIDPOINT BETWEEN A AND B  *)
  6     LEFT, RIGHT,
C                                    (*  GOLDEN  SEARCH  MIDPOINTS *)
  6            FA,  FAB,  FB,
  6            FLEFT,  FRIGHT ,
C                                    (*  FUNCTION  VALUE  AT  CURRENT  POINTS *)
  6            STEP ,
C                                    (*  LINE  SEARCH  INTERVAL  LENGHT *)
  6            R ,
C                                          (*  SETS  GOLDEN  SEARCH  INTERVAL  WIDTH
*)
  6       BOUND,
C200                                    (*  GOLDEN  SEARCH  ITERATION  ERROR  BOUND
*)
C                             (*  BOUND  COULD  BE  A  STOPPING  RULE  BUT  *)
C                             (*  I DON'T  THINK  I  USED  IT  IN  THE  END *)


C       DECLARE ALL VARS IN COMMON

  6            PARAM(1:3), RAW(5000), CUM(5000)

C       EXTERNAL DECLARE ANY EXTERNAL FUNCTIONS USED

        EXTERNAL  GOF
        COMMON / GLOBALDATA / COUNT,RAW


        PRINT*,'WITHIN SUBR GSEARCH:'
        PRINT*,'  COUNT', COUNT

C       INITIALIZATION

               REPCOUNT = 0
               A = 0.0
               B = 0.0
               AB = 0.0
```

```
              LEFT = 0.0
              RIGHT = 0.0
              FA = 0.0
              FAB = 0.0
              FB = 0.0
                 FLEFT = 0.0
                 FRIGHT = 0.0
                 STEP = 0.0
                 R = 0.0
                 BOUND = 0.0
                 SUM  = 0.0
                 TEMP = 0.0
                 NUM = 0

                 DO 22  K = 1,COUNT
                        CUM(K) = 0.0
22               continue
c         PRINT*,'WHICH = ','CVM'
c         PRINT*,'  fos =',RAW(1)
C         PRINT*,'WITHIN GS PARAMETERS =',
C    6     PARAM(1),PARAM(2),PARAM(3)




C-        BEGIN

          STEP  =  1.0 / 50.0
c250                                   (* LINE  INTERVAL  STEP  SIZE *)
      R  =  0.618034
c                                 (* GOLDEN  SEARCH  MULTIPLIER *)
          A = PARAM(2)

C         PRINT*,' A = ',A

      FA  =  GOF (WHICH,PARAM )
c                              (*  CURRENT  OBJECTIVE  VALUE  *)
      FB  =  FA  +  1.0
c                                      (*  INITIATE  LOOP *)
C         PRINT*,' FA = ',FA
C         PRINT*,' FB = ',FB

C         WHILE  ( FB  -  FA )  >  ERROR  DO
C       (*  LOOP  DETERMINES  DIRECTION  TO  *)

10        IF ((FB - FA) .GT. 1.0E-6) THEN
C                                      (*  DECREASE  THE  FUNCTION  OR  IF  *)
          B  =  A + STEP
C                PRINT*,'  B =',B
C                                      (*  CURRENT  POINT  IS  THE  MINIMUM *)
            PARAM(2) = B
```

```
C       PRINT*,' PARAM(2) =',PARAM(2)
         FB =  GOF( WHICH, PARAM )

         IF  ( FB  .GT.  FA ) THEN
C                                                     (* TRY THE OTHER DIRECTION *)
C278
              STEP  = -1.0 *  STEP
              B  =  A  +  STEP
              PARAM ( 2 )  =  B
              FB  = GOF (WHICH,PARAM)
         END IF
         STEP =   STEP / 4.0
C                                               (* REDUCES STEP - IF THE CURRENT POINT  *)
        GO TO 10
        END IF
C                                                     (* IS THE MIN, STEP WILL REDUCE SO A =
B *)

        IF  ( FB  .GT.  FA)  THEN
C                                                     (* THE ORIGINAL POINT WAS THE
MINIMUM   *)
          PARAM ( 2 )  =  A
        ELSE
C                           (* LINE SEARCH TO FIND INTERVAL WITH MINIMUM*)
         AB  =  A
C                                                     (* INITIALIZES SEARCH  *)
         FAB  =  FA

C               REPEAT UNTIL                          (* LINE SEARCH CHECKS EVERY STEP TO
FIND  *)

20      A  =  AB
C                                                     (* WHERE THE FUNCTION STARTS TO
INCREASE  *)
              FA  =  FAB
              AB  =  B
              FAB =  FB
              B   =  B  +  STEP
              PARAM ( 2 )  = B
              FB  =  GOF ( WHICH,PARAM )
              IF ( FB .LT. FAB ) GO TO 20

C                                                           *** GOLDEN SEARCH BEGINS  ***
              LEFT          = B - R * ( B - A )
              RIGHT         = A + R * ( B - A )
              BOUND         = 2 * ABS (STEP)

              PARAM (2) =  LEFT
              FLEFT  =  GOF ( WHICH,PARAM )
              PARAM (2)  =  RIGHT
              FRIGHT =  GOF ( WHICH,PARAM )
```

```
C             WHILE  ABS  (FB - FA)  >  ERROR  DO
30       IF ((ABS(FB - FA) .GT. 1.0E-6)
  6                    .AND. (REPCOUNT .LT. REPS)) THEN

                       REPCOUNT = REPCOUNT + 1

                       IF  (FLEFT  .LT.  FRIGHT)  THEN
C                                              (* DELETE  RIGHT  INTERVAL  *)
                   B  =  RIGHT
                  FB  =  FRIGHT
                       RIGHT  =  LEFT
                       FRIGHT  = FLEFT
                       LEFT  =  B - R*(B-A)
                       PARAM (2)  =  LEFT
              FLEFT  =  GOF (WHICH,PARAM)
                       END IF

              IF  (FRIGHT .LE.  FLEFT)  THEN
C                                                  (* DELETE  LEFT  INTERVAL  *)
              A  =  LEFT
              FA  =  FLEFT
              LEFT  =  RIGHT
              FLEFT  =  FRIGHT
                            RIGHT  =  A + R * (B-A)
                            PARAM ( 2 )  = RIGHT
                       FRIGHT  =  GOF(WHICH,PARAM)
                   END IF

                   BOUND  =  R * BOUND

              GO TO 30
              END IF
C        (*  END OF  WHILE  *)              (*  END  GOLDEN  SEARCH  ROUTINE  *)

C                                    (*  PICKS  MIN  POINT  AS  THE  SOLUTION  *)

              IF (FLEFT  .LT. FRIGHT) THEN

               IF  (FA  .LT.  FLEFT)  THEN
                            PARAM ( 2 )  =  A
              ELSE
                         PARAM ( 2 )  =  LEFT
                       END IF

              ELSE

                       IF  (FB  .LT.  FRIGHT)  THEN
                       PARAM  ( 2 )  =  B
                   ELSE
                       PARAM ( 2 )  = RIGHT
```

```
            END IF

            END IF
C                     (* OF ELSE  ( FROM LONG TIME AGO )  *)


C                     (* OF PROCEDURE GOLDENSEARCH  *)
      END IF

      END


C NOW, RE-ESTIMATE MLEs USING THIS MIN DIST ESTIMATE OF LOCATION





************************************************************
*CVM  THIS FUNCTION RETURNS THE CRAMER VON-MISES GOODNESS OF FIT   *)
*  STATISTIC FOR THE THREE PARAMETER WEIBULL.                      *)
*  FORMULAS PUBLISHED IN WOODRUFF, MOORE, AND DUNNE (1983)         *)
*  DATA MUST BE ORDERED !                                 *)
************************************************************

      REAL*8 FUNCTION CVMGOF (param)

      INTEGER I,J,K,COUNT,SUBCOUNT,NLB,NUM
      REAL*8 RAW(5000),CUM(5000), SUM, TEMP,
     6   PARAM(1:3)
      INTRINSIC DLOG, DEXP
      COMMON / GLOBALDATA / COUNT, RAW

C     PRINT*,'From CVM PARAMETERS =',
C    6  PARAM(1),PARAM(2),PARAM(3)


C     INITIALIZATION

            SUM  = 0.0
            TEMP = 0.0
            NUM = 0
            NLB = 1
            subcount = 0
            CVMGOF = 0.0

      DO 12  K = 1,COUNT
            CUM(K) = 0.0
12    continue
```

```
C       BEGIN

        NUM = COUNT
C       PRINT*,'WITHIN SUBR CVM,  COUNT =',NUM


        DO 27  I = 1, NUM

             IF (RAW (I) .LE. PARAM (2)) THEN
c                            (* OBS LT LOC : UNDEFINED *)
                  CUM(I) = 1.0E-10
             ELSE
                  TEMP = 0.0

                  TEMP =  -1.0
     6  *DEXP(PARAM(1)*DLOG((RAW(I)-PARAM(2))/PARAM(3)))

             CUM(I)  = (1.0 - DEXP (TEMP))
             END IF

27      CONTINUE


        DO 13  J  =  1, NUM

             TEMP = 0.0

        TEMP = CUM(J)
     6    -((2.0*J-1.0)/(2.0*NUM))

C       PRINT*,'----------------------------------------'
C       PRINT*,'FOR J = ',J,' RAW(J) = ',RAW(J),'CUM(J)=',CUM(J)
C       PRINT*,'  K = ',K,' TEMP = ',TEMP
C       PRINT*,'----------------------------------------'

          SUM = SUM + TEMP*TEMP

13      CONTINUE

        CVMGOF = (1.0/12.0*NUM)+SUM



c       PRINT*,'---------------------------------------------'
c       PRINT*,'CVMGOF FUNCTION COMPLETED'

c       PRINT*,'    NUM = ',NUM
c       PRINT*,'    COUNT =',COUNT
c       PRINT*,'    SUM =',SUM
```

```
c       PRINT*,'   TEMP =',TEMP
c       PRINT*,'   CUM(count) = ', CUM(count)



        END

******************************************************************
* FUNCTION WHICH GOF
*       *
******************************************************************

        REAL*8 FUNCTION GOF (WHICH,PARAM)


        CHARACTER*3 WHICH
        REAL*8 PARAM(1:3), CVMGOF

        EXTERNAL CVMGOF

c       PRINT*,'WITHIN SUBR GOF:'
C       PRINT*,'From GOF PARAMETERS =',
C    6    PARAM(1),PARAM(2),PARAM(3)

                GOF = CVMGOF(PARAM)


        END




********************************************************
*       subroutine INTEGRATE
*               calculates the diff between CDFs
*               1) ICVM
*               2) imse
********************************************************

        SUBROUTINE INTEGRATE (DISTA, DISTB, INTABS,INTMSE)

        REAL*8 TEMP1,TEMP2,DISTA(0:3),DISTB(0:3),INTABS,INTMSE,
     6  UPLIMA,UPLIMB,UPLIM,LOWLIM, CUMWEIBULL
        EXTERNAL INT, CUMWEIBULL

C       PRINT*,'WITHIN SUBR INTEGRATE: '
C       PRINT*,'MDLCVM(0-3)=',DISTB(0),DISTB(1),DISTB(2),DISTB(3)
C       PRINT*,'TRUE (0-3) =',DISTA(0),DISTA(1),DISTA(2),DISTA(3)
C       PRINT*,'INTABS =',INTABS
```

```
C       PRINT*,'INTMSE =',INTMSE

        IF (DISTA(2) .EQ. 0.0) THEN
                DISTA(2) = 0.00001
        END IF

        UPLIMA = DISTA(2) + 3.0*DISTA(3)
        TEMP1 = CUMWEIBULL (DISTA, UPLIMA)
c       WHILE TEMP1 LT 0.999
10      IF (TEMP1 .LT. 0.999) THEN
                UPLIMA = UPLIMA + DISTA(3)
                TEMP1 = CUMWEIBULL(DISTA,UPLIMA)
        GO TO 10
        END IF


        UPLIMB = DISTB(2) + 3.0*DISTB(3)
        TEMP2 = CUMWEIBULL (DISTB, UPLIMB)
c       WHILE TEMP2 LT  0.999 DO
20      IF (TEMP2 .LT. 0.999) THEN
                UPLIMB = UPLIMB + DISTB(3)
                TEMP2 = CUMWEIBULL(DISTB,UPLIMB)
        GO TO 20
        END IF


        IF (TEMP1 .LT. TEMP2) THEN
                UPLIM = UPLIMB
        ELSE
                UPLIM = UPLIMA
        END IF

        IF (DISTA(2) .LT. DISTB(2)) THEN
                LOWLIM = DISTA(2)
        ELSE
                LOWLIM = DISTB(2)
        END IF

        CALL INT(DISTA,DISTB,LOWLIM,UPLIM,INTABS,INTMSE)

        END


*********************************************************

        SUBROUTINE INT(DISTA,DISTB,LOWERLIM,UPPERLIM,INTABS,INTMSE)
C1200
        INTEGER I
        REAL*8 TEMP1,TEMP2,TEMP3,VAL1,VAL2,SUM2,SUM1,
   6    W(1:24),X(1:24),CUMWEIBULL,DISTA(0:3),DISTB(0:3),
   6    LOWERLIM, UPPERLIM, INTABS, INTMSE
        EXTERNAL CUMWEIBULL,FX
```

```
C       PRINT*,'WITHIN SUBR INT: '
C       PRINT*,' LOWERLIM =',LOWERLIM
C       PRINT*,' UPPERLIM =',UPPERLIM
C       PRINT*,'INTABS =',INTABS
C       PRINT*,'INTMSE =',INTMSE
C       PRINT*,' MDLCVM(1-3)=',DISTB(1),DISTB(2),DISTB(3)
C       PRINT*,' TRUE (1-3)', DISTA(1),DISTA(2),DISTA(3)

        TEMP1 = 0.0
        TEMP2 = 0.0
        TEMP3 = 0.0
        SUM1 = 0.0
        SUM2 = 0.0

        X(1) = 0.03238017096286
        X(2)= 0.09700469920946
        X(3)= 0.16122235606889
        X(4)= 0.22476379039469
        X(5)= 0.28736248735546
        X(6)= 0.34875588629216
        X(7)= 0.40868648199072
        X(8)= 0.46690290475096
        X(9)= 0.52316097472223
        X(10)= 0.57722472608397
        X(11)= 0.62887396776514
        X(12)= 0.67787237963266
        X(13)= 0.72403413092381
        X(14)= 0.76715903251574
        X(15)= 0.80706620402944
        X(16)= 0.84258826162439
        X(17)= 0.87657202027424
        X(18)= 0.90587913671557
        X(19)= 0.93138669070655
        X(20)= 0.95298770316043
        X(21)= 0.97059159254625
        X(22)= 0.98412458372283
        X(23)= 0.99353017226635
        X(24)= 0.99877100725243
C1230
        W(1) = 0.06473769681
        W(2)= 0.06446616444
        W(3)= 0.06392423858
        W(4)= 0.06311419229
        W(5)= 0.06203942316
        W(6)= 0.06070443917
        W(7)= 0.05911483969
        W(8)= 0.05727729210
        W(9)= 0.05519950370
        W(10)= 0.05289018949
        W(11)= 0.05035903555
        W(12)= 0.04761665849
```

```
      W(13)= 0.04467456085
      W(14)= 0.04154508294
      W(15)= 0.03824135107
      W(16)= 0.03477722256
      W(17)= 0.03116722783
      W(18)= 0.02742650971
      W(19)= 0.02357076084
      W(20)= 0.01961616046
      W(21)= 0.01557931572
      W(22)= 0.01147723458
      W(23)= 0.00732755390
      W(24)= 0.00315334605


      TEMP1 = (UPPERLIM-LOWERLIM)/2
      TEMP2 = (UPPERLIM+LOWERLIM)/2
C1260
      SUM = 0.0

      DO 17  I = 1,24

            VAL1 = 0.0
            VAL2 = 0.0
            TEMP3 = 0.0

            TEMP3 = TEMP1*X(I)+TEMP2
            CALL FX(DISTA,DISTB,TEMP3, VAL1, VAL2)
            SUM1 = SUM1 + VAL1*W(I)
            SUM2 = SUM2 + VAL2*W(I)

17    CONTINUE


      DO 33  I = 1,24

            TEMP3 = 0.0

            TEMP3 = TEMP1*-1.0*X(I)+TEMP2
            CALL FX(DISTA,DISTB,TEMP3, VAL1, VAL2)
            SUM1 = SUM1 + VAL1*W(I)
            SUM2 = SUM2 + VAL2*W(I)
C1280
33    CONTINUE

      INTABS = TEMP1 * SUM1
      INTMSE = TEMP1 * SUM2

      END
```

****

```
      SUBROUTINE FX(DISTA,DISTB,PT,FX1,FX2)

      REAL*8 DISTA(0:3),DISTB(0:3),TEMP1,TEMP2,PT
      REAL*8 CUMWEIBULL, FX1, FX2
      EXTERNAL CUMWEIBULL

C1300
C     PRINT*,'WITHIN SUBR FX: '
C     PRINT*,' X =',PT
C     PRINT*,'MDLCVM(0-3)=',DISTB(0),DISTB(1),DISTB(2),DISTB(3)
C     PRINT*,'TRUE (0-3) =',DISTA(0),DISTA(1),DISTA(2),DISTA(3)

      TEMP1 = 0.0
      TEMP2 = 0.0
      FX1 = 0.0
      FX2 = 0.0

      TEMP1 = CUMWEIBULL(DISTA,PT) * DISTA(0)
      TEMP2 = CUMWEIBULL(DISTB,PT) * DISTB(0)
      FX1 = ABS(TEMP1-TEMP2)
      FX2 = (TEMP1-TEMP2)*(TEMP1-TEMP2)

C     PRINT*,' FX1 =',FX1
C     PRINT*,' FX2 =',FX2

      END

******************************************************************
* REAL FUNCTION CUMWEIBULL
*     returns the cumulative weibull value for point x
*     dist contains the weibull shape, scale and location
*
******************************************************************
C1314

      REAL*8 FUNCTION CUMWEIBULL(DIST, PT)

      REAL*8  TEMP,DIST(0:3),PT
      INTRINSIC DEXP,DLOG

C     PRINT*,'WITHIN FUNC CUMWEIBULL: '
C     PRINT*,' X =',PT
C     PRINT*,' DIST(1-3)=',DIST(1),DIST(2),DIST(3)

      TEMP = 0.0

      IF (PT  .LE.  DIST(2)) THEN
            CUMWEIBULL = 0.0
      ELSE
            TEMP = DEXP(DIST(1)*DLOG((PT-DIST(2))/DIST(3) ))
            IF (TEMP .GT. 20) THEN
                  CUMWEIBULL = 1.0
```

```
          ELSE
               CUMWEIBULL = 1.0 - DEXP(-1.0*TEMP)
          END IF

     END IF

     END
```

```
************************************************************************
*
* PSMDE                    SUBR SMDE ON P
*
************************************************************************

      SUBROUTINE PSMDE (MLE, P)

      INTEGER I, COUNT
      CHARACTER*3  WHICH
      REAL*8 err, tol, reps,P
      PARAMETER( err  = 1.0E-6)
c                                        (* error and tolerance are limits*)
      PARAMETER( tol  = 1.0E-6)
c                                        (*  used in the numerical routines *)
    PARAMETER( reps = 1000)
c                                   (* the number of DATA generated *)
c     - DECLARE FUNCTIONS
c-    - DECLARE  ARRAYS -

      REAL*8        RAW(5000),
     6          MLE(1:7),
     6          MDPCVM(1:7)
c                                        (* WEIBULL random  variables       *)
c                              (* position  0 is  the number of  RVs.*)

c                                        (* evaluation  values for  different  paras        *)
      COMMON / GLOBALDATA / COUNT,RAW

C     PRINT*,'WITHIN SUBR SMDE ALL CALCS BASED ON ORIGINAL MLES:'
C     PRINT*,'  COUNT', COUNT
C     PRINT*,'   MLE(1) ',MLE(1)
C     PRINT*,'   MLE(2) ',MLE(2)
C     PRINT*,'   MLE(3) ',MLE(3)

C     CRITICAL INITIALIZATION
```

```
      P = 0.0

      DO 99  I = 1,7
            MDPCVM (I) = MLE (I)
99    CONTINUE


      WHICH = 'CVM'
      CALL PGSEARCH ( WHICH, MDPCVM)

      P = MDPCVM(7)

      END

****************************************************************************
*     INPUTS: DESIRED STAT: WHICH,  HOW : PARAS
* STARTING AT "A" SEARCHES IN "DIRECTION" UNTIL THE FUNCTION STOPS
* DECREASING. THEN BEGINS A GOLDEN SEARCH ON THE LAST TWO
* INTERVALS JUST PRIOR TO THE FUNCTION INCREASING.
* THE LOCATION SHOULD HAVE BEEN BOUNDED BELOW THE FIRST ORDER STATIS
****************************************************************************


      SUBROUTINE PGSEARCH ( WHICH,PARAM )

      INTEGER REPS,K
      REAL*8 err, tol

      PARAMETER( err  = 1.0E-6)
c                                  (* error and tolerance are limits*)
      PARAMETER( tol  = 1.0E-6)
c                                  (*  used in the numerical routines *)
    PARAMETER( reps = 1000)
c
      CHARACTER*3  WHICH
      INTEGER  COUNT, REPCOUNT
      REAL*8  PGOF, A, B,
C                                  (*  CURRENT RIGHT AND LEFT ENDPOINTS *)
  6           AB,
C                              (*  MIDPOINT BETWEEN A AND B  *)
  6   LEFT, RIGHT,
C                                  (*  GOLDEN  SEARCH  MIDPOINTS *)
  6           FA,  FAB,  FB,
  6           FLEFT,  FRIGHT ,
C                                  (*  FUNCTION  VALUE  AT  CURRENT  POINTS *)
  6           STEP ,
C                                  (*  LINE  SEARCH  INTERVAL  LENGHT *)
  6           R ,
C                                        (*  SETS  GOLDEN  SEARCH  INTERVAL  WIDTH
*)
  6     BOUND,
```

```
C200                              (* GOLDEN SEARCH ITERATION ERROR BOUND
*)
C                      (* BOUND COULD BE A STOPPING RULE BUT *)
C                      (* I DON'T THINK I USED IT IN THE END *)


C      DECLARE ALL VARS IN COMMON

   6        PARAM(1:7), RAW(5000), CUM(5000)

C      EXTERNAL DECLARE ANY EXTERNAL FUNCTIONS USED

       EXTERNAL PGOF
       COMMON / GLOBALDATA / COUNT, RAW



C      PRINT*,'WITHIN SUBR GSEARCH:'
C      PRINT*,' COUNT', COUNT

C      INITIALIZATION

            REPCOUNT = 0
            A = 0.0
            B = 0.0
            AB = 0.0
          LEFT = 0.0
          RIGHT = 0.0
          FA = 0.0
          FAB = 0.0
          FB = 0.0
            FLEFT = 0.0
            FRIGHT = 0.0
            STEP = 0.0
            R = 0.0
            BOUND = 0.0
            SUM  = 0.0
            TEMP = 0.0
            NUM = 0.0


            DO 22  K = 1,COUNT
                 CUM(K) = 0.0
22          continue
c      PRINT*,'WHICH = ','CVM'
c      PRINT*,'COUNT =',RAW(0)
c      PRINT*,'  fos =',RAW(1)
C      PRINT*,'WITHIN GS PARAMETERS =',
C   6   PARAM(1),PARAM(2),PARAM(3)
```

```
C-      BEGIN

        STEP = 1.0 / 200.0
c250                              (* LINE INTERVAL STEP SIZE *)
   R = 0.618034
c                           (* GOLDEN SEARCH MULTIPLIER *)
        A = PARAM(7)

C       PRINT*,' A = ',A

   FA = PGOF ( WHICH,PARAM )
c                        (* CURRENT OBJECTIVE VALUE *)
        FB = FA + 1.0
c                                 (* INITIATE LOOP *)
C       PRINT*,' FA = ',FA
C       PRINT*,' FB = ',FB

C   WHILE ( FB - FA ) > ERROR DO
C    (* LOOP DETERMINES DIRECTION TO *)

10      IF ((FB - FA) .GT. 1.0E-6) THEN
C                                 (* DECREASE THE FUNCTION OR IF *)
       B = A + STEP
C              PRINT*,' B =',B
C                                 (* CURRENT POINT IS THE MINIMUM *)
       PARAM(7) = B
C              PRINT*,' PARAM(7) =',PARAM(7)
         FB = PGOF( WHICH, PARAM )

         IF ( FB .GT. FA ) THEN
C                                         (* TRY THE OTHER DIRECTION *)
C278
               STEP = -1.0 * STEP
             B = A + STEP
             PARAM (7) = B
             FB = PGOF ( WHICH,PARAM )
         END IF
         STEP = STEP / 4.0
C                                 (* REDUCES STEP - IF THE CURRENT POINT  *)
        GO TO 10
        END IF
C                                         (* IS THE MIN, STEP WILL REDUCE SO A =
B *)

        IF ( FB .GT. FA) THEN
C                                         (* THE ORIGINAL POINT WAS THE
MINIMUM  *)
           PARAM (7) = A
        ELSE
C                            (* LINE SEARCH TO FIND INTERVAL WITH MINIMUM*)
```

```
        AB = A
C                                          (* INITIALIZES SEARCH  *)
        FAB = FA

C            REPEAT UNTIL                  (* LINE SEARCH CHECKS EVERY STEP TO
FIND *)

20      A  = AB
C                                          (* WHERE THE FUNCTION STARTS TO
INCREASE *)
             FA = FAB
             AB = B
             FAB = FB
             B  = B + STEP
             PARAM (7) = B
             FB = PGOF ( WHICH,PARAM )
             IF ( FB .LT. FAB ) GO TO 20

C                                                 *** GOLDEN SEARCH BEGINS ***
             LEFT        = B - R * ( B - A )
             RIGHT       = A + R * ( B - A )
             BOUND              = 2 * ABS (STEP)

             PARAM (7) = LEFT
             FLEFT = PGOF ( WHICH,PARAM )
             PARAM (7) = RIGHT
             FRIGHT = PGOF ( WHICH,PARAM )


C            WHILE ABS (FB - FA) > ERROR DO
30      IF ((ABS(FB - FA) .GT. 1.0E-6)
   6                 .AND. (REPCOUNT .LT. REPS)) THEN

                     REPCOUNT = REPCOUNT + 1

                     IF (FLEFT .LT. FRIGHT) THEN
C                                            (* DELETE RIGHT INTERVAL  *)
                     B = RIGHT
                FB = FRIGHT
                RIGHT = LEFT
                   FRIGHT = FLEFT
                LEFT = B - R*(B-A)
                PARAM (7) = LEFT
            FLEFT = PGOF ( WHICH,PARAM )
                     END IF

             IF (FRIGHT .LE. FLEFT) THEN
C                                                (* DELETE LEFT INTERVAL *)
            A  = LEFT
            FA = FLEFT
        LEFT = RIGHT
        FLEFT = FRIGHT
```

```
               RIGHT  =  A + R * (B-A)
               PARAM (7)  = RIGHT
               FRIGHT  =  PGOF( WHICH,PARAM )
                END IF

                BOUND  =  R * BOUND

            GO TO 30
            END IF
C         (* END OF  WHILE  *)                  (* END  GOLDEN  SEARCH  ROUTINE  *)


C                                       (* PICKS  MIN  POINT  AS  THE  SOLUTION  *)

            IF (FLEFT  .LT. FRIGHT) THEN

            IF  (FA  .LT.  FLEFT)  THEN
                         PARAM (7)  =  A
                  ELSE
                         PARAM (7)  =  LEFT
                         END IF

            ELSE

                  IF  (FB  .LT.  FRIGHT)  THEN
                  PARAM (7)  =  B
            ELSE
                  PARAM (7)  = RIGHT
            END IF

            END IF
C                 (*  OF  ELSE   ( FROM  LONG  TIME  AGO )    *)



C                 (*  OF  PROCEDURE  GOLDENSEARCH   *)
      END IF

      END

C NOW, RE-ESTIMATE MLEs USING THIS MIN DIST ESTIMATE OF LOCATION

*********************************************************************
* FUNCTION WHICH PGOF
*       *
*********************************************************************

      REAL*8 FUNCTION PGOF ( WHICH,PARAM )


      CHARACTER*3 WHICH
      REAL*8 PARAM(1:7)
```

```
      EXTERNAL PCVMGOF

C     PRINT*,'WITHIN SUBR GOF:'
C     PRINT*,' COUNT', COUNT
C     PRINT*,'From GOF PARAMETERS =',
C   6  PARAM(1),PARAM(2),PARAM(3)

              PGOF = PCVMGOF( PARAM )

      END





*****************************************************************
*CVM  THIS FUNCTION RETURNS THE CRAMER VON-MISES GOODNESS OF FIT      *)
*  STATISTIC FOR THE THREE PARAMETER WEIBULL.                     *)
*  FORMULAS PUBLISHED IN WOODRUFF, MOORE, AND DUNNE (1983)        *)
*  DATA MUST BE ORDERED !                                     *)
*****************************************************************

      REAL*8 FUNCTION PCVMGOF ( param )

      INTEGER I,J,K,COUNT,SUBCOUNT,NLB,NUM
      REAL*8 RAW(5000),CUM(5000), SUM, TEMP,
    6   PARAM(1:7),CVMGOF1,CVMGOF2
      INTRINSIC DLOG, DEXP
      COMMON / GLOBALDATA / COUNT, RAW

C     PRINT*,'From CVM PARAMETERS =',
C   6 PARAM(1),PARAM(2),PARAM(3),
C   6 PARAM(4),PARAM(5),PARAM(6),PARAM(7)


C     INITIALIZATION

          SUM  = 0.0
          TEMP = 0.0
          NUM = 0
          NLB = 1
          subcount = 0
          CVMGOF1 = 0.0

      DO 12  K = 1,COUNT
          CUM(K) = 0.0
12    continue
```

```
C       BEGIN

        NUM = COUNT

c       PRINT*,'  WITHIN SCVMGOF, NUM =',NUM


        DO 13  I = 1, NUM

                IF (RAW (I) .LE. PARAM (2)) THEN
c                                       (* OBS LT LOC : UNDEFINED *)
                        CUM(I) = 1.0E-10
                ELSE
                        TEMP = 0.0

                        TEMP =  -1.0
   6    *DEXP(PARAM(1)*DLOG((RAW(I)-PARAM(2))/PARAM(3)))

            CUM(I) = (1.0 - DEXP (TEMP))*PARAM(7)
                END IF

13      CONTINUE


        DO 14  J  =  1, NUM

                TEMP = 0.0

        TEMP = CUM(J)
   6      -((2.0*J-1.0)/(2.0*NUM))

C       PRINT*,'----------------------------------------'
C       PRINT*,'FOR J = ',J,' RAW(J) = ',RAW(J),'CUM(J)=',CUM(J)
C       PRINT*,'  K = ',K,' TEMP = ',TEMP
C       PRINT*,'----------------------------------------'

          SUM = SUM + TEMP*TEMP

14      CONTINUE

        CVMGOF1 = (1.0/12.0*NUM)+SUM



c       PRINT*,'-------------------------------------------'
c       PRINT*,'CVMGOF FUNCTION COMPLETED'
c       PRINT*,'  CVMGOF = ',CVMGOF
c       PRINT*,'    NUM = ',NUM
c       PRINT*,'    COUNT =',COUNT
c       PRINT*,'    SUM =',SUM
```

```
c       PRINT*,'   TEMP =',TEMP
c       PRINT*,'   CUM(count) = ', CUM(count)


C       INITIALIZATION

                SUM  = 0.0
                TEMP = 0.0
                NUM = 0
                NLB = 1
                subcount = 0
                CVMGOF2 = 0.0
                CVMGOF = 0.0

        DO 22  K = 1,COUNT
                CUM(K) = 0.0
22      continue



C       BEGIN

        NUM = COUNT
c       PRINT*,'  COUNT =',NUM


        DO 23  I = 1, NUM

                IF (RAW (I) .LE. PARAM (5)) THEN
c                                       (* OBS LT LOC : UNDEFINED *)
                        CUM(I) = 1.0E-10
                ELSE
                        TEMP = 0.0

                        TEMP =  -1.0
     6  *DEXP(PARAM(4)*DLOG((RAW(I)-PARAM(5))/PARAM(6)))

            CUM(I)=(1.0-PARAM(7))*(1.0 - DEXP (TEMP))
                END IF

23      CONTINUE


        DO 24  J   =  1, NUM

                TEMP = 0.0

        TEMP = CUM(J)
     6    -((2.0*J-1.0)/(2.0*NUM))

C       PRINT*,'--------------------------------------'
C       PRINT*,'FOR J = ',J,' RAW(J) = ',RAW(J),'CUM(J)=',CUM(J)
```

```
C       PRINT*,' K = ',K,' TEMP = ',TEMP
C       PRINT*,'--------------------------------------'

          SUM = SUM + TEMP*TEMP

24      CONTINUE

                CVMGOF2 = (1.0/12.0*NUM)+SUM


        PCVMGOF = CVMGOF1 + CVMGOF2

C       PRINT*,' CVMGOF1 = ',CVMGOF1
C       PRINT*,' CVMGOF2 = ',CVMGOF2
C       PRINT*,' PCVMGOF = ',PCVMGOF

        END
```

Bibliography

Abramowitz, Milton and Irene A. Stegun, Editors. Handbook of Mathematical Functions with Formulas,Graphs, and Mathematical Tables. U. S. Dept. of Commerce, National Bureau of Standards, Applied Mathematics Series, (March 1965).

Adatia, A. and L. K. Chan. "Robust Estimators of the Three-Parameter Weibull Distribution." IEEE Transactions on Reliability, R-34, No. 5: 347-351, (October 1985).

Ahmad & Abd-Elrahm. "Updating a non-linear discriminant function estimated from a mixture of two Weibulls." Mathematical & Computer Modeling, (1994).

Anderson, T. W. and D. A. Darling. "A Test of Goodness of Fit." Journal of the American Statistical Association, 765- 769, (December 1954).

Andrews, D. F., et al., Robust Estimates of Location, Princeton, New Jersey: Princeton University Press, (1972).

Antle, Charles E. and Lee J. Bain. " A Property of Maximum Likelihood Estimators of Location and Scale Parameters." SIAM Review, 11, No. 2: 251-253, (April 1969).

Ashour, S.K., "Asymptotic Distribution of Bayes Estimators in Censored Type-I Samples from A Mixed Exponential Population," IEEE Transactions on Reliability, Vol. R-34, pp. 489-490, (1985).

August, R; Zaretsky. " Incorporating Finite Element Analysis into Component Life and Reliability." (1991).

Banks, Jerry and John S. Carson, II. Discrete-Event System Simulation, Engelwood Cliffs, New Jersey: Prentice-Hall, Inc., (1984).

Behboodian, J., "On the Modes of a Mixture of Two Normal Distributions. "Technometrics 12, 131-139 (1970).

Benton-Santo. "Minimum Distance Estimation of Mixing Proportions." M. S. Thesis, Scool of Engineering, Air Force Instititute of Technology (1986).

Bergman, B. "Estimation of Weibull Parameters using a weight function." Journal of Material Science Letters, (1986).

Bhattacharya, C.G., "A Simple Method of Resolution of a Distribution into Gaussian Components." Biometrics, Vol. 23, pp. 115-137, (1967).

Blischke, W.R., "Estimating the Parameters of Mixtures of Binomial Distributions." Journal of the American Statistical Asociation, Vol. 59, pp. 510-528, (1964).

Burden, Richard L., J. Douglas Faires, and Albert C. Reynolds. Numerical Analysis, Boston: Prindle, Weber & Schmidt, (1981).

Cacciari, M. " Use of MW for the identification of PD phenomena." IEEE Transactions on Dielectrics and Electrical Insulation , (1995).

Cacciari, M. "Estimating the CDF of failure data points to be plotted on WPP." IEEE Transactions on Electrical Insulation, (1991).

Cao. " Minimum Distance Density-Based Estimation." Computational Statistics & Data Analysis, (1995).

Chanute Technical Training Center (ATC). System Reliability/ Maintainability Textbook. No. 30ZR2895G-102 and 40ST289G-1- 102. Chanute Air Force Base, IL.

Cheng S.W. and J.C. Fu, "Estimation of Mixed Weibull Parameters in Life Testing." IEEE Transactions on Reliability, Vol. R-31, pp. 377-381, (1982).

Cheng S.W., J.C. Fu and S.K. Sinha, "An Empirical Procedure for Estimating the Parameters of a Mixed Exponential Life Testing Model," IEEE Transactions on Reliability, Vol. R-34, pp. 60-64, (1985).

Cohen, A.C., "Estimation in Mixtures of Two Normal Distributions," Technometrics, Vol. 9, pp. 15-28, (1967).

Contin, Cacciari. " Estimation of Weibull distribution parameters for partial discharge inference." IEEE Ann Rep 94, (1994).

Cordero Brana. "Minimum Hellinger Distance Estimation for Finite Mixture Models." Thesis-Utah State, (1994).

Cran, G.W., "Graphical Estimation Methods for Weibull Distributions." Microelectronics and Reliability, Vol. 15, pp. 47-52, (1976).

Crowder, H. P., R. S. Dembo, and J. Mulvey. Guidelines for reporting computational experiments in mathematical programming. Mathematical Programming 15, 316-329 (1978).

D'Agostino, R. B. and Stephens, M.A. (edited by) Goodness-of-Fit Techniques, Marcell Decker, Inc., (1986).

Daniels, Tony G. Robust Estimation of the Generalized Distribution Using Minimum Distance Estimation. MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, (December 1980).

Day, N.E., "Estimating the Components of a Mixture of Normal Distributions." Biometrika, Vol. 56, pp. 463-474, (1969).

Dick, N.P. and D.C. Bowden, "Maximum Likelihood Estimation for Mixtures of Two Normal Distributions." Biometrics, Vol. 29, pp. 781-790, (1973).

Eslinger, Paul W. "Minimum Hellinger Distance Estimation for Normal Models." Journal of Statistical Computation and Simulation (1995).

Everitt, B. S. and J. D. Hand, Finite Mixture Distributions; Chapman and Hall Publishing (1981).

Falls, L. W.. "Estimation of Parameters in Compound Weibull Distributions," Technometrics, 12:399-407, (1970).

Gallagher, Mark A. "Robust Minimum Distance Estimation using the three parameter Weibull." M. S. Thesis, Scool of Engineering, Air Force Instititute of Technology (1986).

Gallagher, Mark A. "Robust Minimum Distance Estimation using the three parameter Weibull." IEEE Trans on Reliability, Vol. 39: 575-580, (1991).

Gera, Amos E. "Unimodality and Bi-modality of mixed Weibull distributions." Quality and Reliability Engineering International, (1995).

Hager, Harold. W., Lee J. Bain, and Charles E. Antle. "Reliability Estimation for the Generalized Gamma Distribution and Robustness of the Weibull Model." Technometrics, Vol 13, No. 3: 547-557, (August 1971).

Harris C. M., Kaylan A. R. and M. D. Maltz. Refinaments on the statistics of recidivism measurement; Mathematical Frontiers of Criminology (Ed. J. Fox) Academic Press New York (1980).

Harter, H. Leon and Albert H. Moore. "Maximum Likelihood Estimation of the Parameters of Gamma and Weibull Populations from Complete and from Censored Samples." Technometrics, Vol. 4, No. 7: 639-643, (November 1965).

Hasselblad, V., "Estimation of Parameters for a Mixture of Normal Distributions." Technometrics Vol. 8: 431-444, (1966).

Hasselblad V., "Estimation of finite mixtures of distributions from the exponential family." Journal of the American Statistical Association, Vol. 64: 1459-1471, (1969).

Hill, B. M., "Information for Estimating the Proportions in Mixtures of Exponential and Normal Distributions." Journal of the American Statistical Association, Vol 58, pp. 918-932, (1963).

Himmelbau, David M. Applied Nonlinear Programming, St. Louis, Missouri: McGraw-Hill Inc, (1972).

Hines, William W. and Douglas C. Montogomery. Probability and Statistics in Engineering and Managment Science, New York: John Wiley & Sons, (1980).

Hobbs, Jon R., Albert H. Moore, and William James. "Minimum Distance Estimation of the Three Parameters of the Gamma Distribution." IEEE Transactions on Reliability, R-33, No. 3: 237-240, (August 1984).

Hobbs, Jon R., Albert H. Moore, and Robert M. Miller. "Minimum-Distance Estimation of the Parameters of the Three Parameter Weibull Distribution." IEEE Transactions on Reliability, R-34, No. 5: 495-496, (December 1985).

Hogg, Robert V. "Adaptive Robust Procedures: A Partial Review and Some Suggestions for Future Applications and Theory." Journal of the American Statistical Association, 69, No. 348: 909-927, (December 1974).

Hooke, Robert and T. A. Jeeves. " 'Direct Search' Solution of Numerical and Statistical Problems." Journal of the Association of Computer Machines, Vol. 8: 212-229, (1962).

Hosmer, D. W., Jr. "A Comparison of Iterative Maximum Likelihood Estimates of the Parameters of a Mixture of Two Normal Distributions Under Three Different Types of Sample." Biometrics, Vol. 29, 761-770, (1973).

Huber, Peter J. Robust Statistics. New York: John Wiley & Sons, (1981).

James, W. L., "Robust Minimum Distance Estimation Based on a Family of Three-Parameter Gamma distributions." M.S. Thesis. School of Engineering , Air Force Institute of Technology (AU), Wright-Patterson AFB OH (1980).

Jensen, F. and N.E. Petersen, "Burn-In, An Engineering Aproach to the Design and Analysis of Burn-In Procedures," John Wiley & Sons, (1982).

Jiang and D. Kececioglu, "Maximum Likelihood Estimation (MLE) from censored data for the Mixed-Weibull Distributions." IEEE Transactions on Reliability, Vol. 41: 248-255 (1992).

Jiang, S. and D. Kececioglu, "Graphical representation of two Mixed-Weibull distributions", IEEE Transactions on Reliability, Vol. 41: 241-247, (1992).

Jiang, S. and Murthy. "Modeling Failure-data by Mixture of Two Weibull Distributions: A Graphical Approach." IEEE Transactions, (1995).

Jiang, S., "Mixed Weibull Distributions in Reliability Engineering: Statistical Models for the Lifetime of Units with Multiple Modes of Failure." Ph.D. Dissertation, University of Arizona (1991).

John S., "On Identifying the Population of Origin of Each Observation in a Mixture of Observations from Two Normal Populations." Technometrics, Vol. 12, pp. 553-564, (1970a).

John S., "On Identifying the Population of Origin of Each Observation in a Mixture of Observations from Two Gamma Distributions." Technometrics, Vol. 12, pp. 565-568, (1970b).

Kao J. H. K., " A Graphical Estimation of Mixed Weibull parameters in life testing of electron tubes." Technometrics, Vol. 1: 389-407, (1959).

Kapur, K. C. and L. R. Lamberson. Reliability in Engineering Design. New York: John Wiley & Sons, (1977).

Kaylan, Ali R. and Carl M. Harris. "Efficient Algorithms to derive MLE for Finite Exponential and Weibull Mixtures." Computers and Operations Research Vol. 8, pp99-104 (1981).

Kaylan, Ali R., Statistical Analysis of Finite Mixtures of Exponential and Weibull Distributions. Ph.D. Dissertation, University of North Carolina, (1978).

Kaylan, A.R., "Statistical Analysis of Finite Mixtures of Exponential and Weibull Distributions." Doctoral Dissertation, Syracuse University, (1979).

Kececioglu, D. "Mixed Weibull Parameter Estimation for Burn-in Data Using the Bayesian Approach." Microelectronics and Reliability, (1994).

Koenker, R. Machad. "Moments Expansions and the Robustness of Minimum Distance Estimation." Economic Theory, (1994).

Laud, P. W. and R. Saunders, "Estimation of Stock Model Parameters by the EM Algorithm." Technometrics, (1979).

Launer, Robert L. and Graham N. Wilkinson. Editors. Robustness inn Statistics. Orlando: Academic Press, Inc., (1979).

Lindsay, H., Parr, W. C., William, R., and Woodward, W. A., " A Comparison of Minimum Distance and Maximum Liklihood Estimation of a Mixture Proportion." Journal of the American Statistical Association, 79:590-598, (1984).

Luenberger, David G. Introduction to linear and Nonlinear Programming. Reading, Massachusetts: Addison-Wesley Pub. Co. (1973).

MacDonald P.D.M. Estimation of Finite Distribution Mixtures. In Applied Statistics: Proc. Conf. Dalhhousie University, Halifax, May 2-4, 1974 (Ed R.P. Gupta) North-Holland, Amsterdam, 231-245 (1975).

Maguluri, G. "Estimation in a Semi-parametric Model by the Method of Minimum Distance." Journal of Statistical Planning and Inference, 30: 23-32 (1992).

Mandelbaum, J., "Numerical issues in inference for finite mixtures of Weibulls under Progressive Censoring." Ph.D. Dissertation (1982).

Mandelbaum, J. and C. M. Harris, "Parameter estimation under under progressive censoring condtions for a finite mixture of Weibull distributions." TIMS/Studies in Management Sciences, Vol. 19: 239-260, (1982).

Matusita, Kameo. "On the Estimation by the Minimum Distance Method." Institute of Statistical Mathematics. 5: 59-66 (1959).

McLachlan, G.J. and P.N. Jones, "Finite Mixture Models to Grouped and Truncated Data via the EM Algorithm." Biometrics, Vol.44, pp. 571-578, (1988).

Mendenhall, W. and R.J. Hader, "Estimation of Parameters of Mixed Exponentially Distributed Failure Time Distributions from Censored Life Test Data," Biometrika, Vol. 45, pp. 504-520, (1958).

Mendenhall, W., Scheaffer, R. L., and Wackerly, R.L. Mathematical Statistics with Applications, Boston: Duxbury Press, (1981).

Miller, Robert M. Robust Minmum Distance Estimation of the Three Parameter Weibull Distribution. MS thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December, (1980).

Moore, Dr Albert C., Interview with Dr Albert H. Moore, 28 Sep 1996.

Mukherjee, R. "Minimum Distance Estimation in Linear Models with Long-range Dependent Errors." Statistics and Probability Letters, Vol. 21: 347-355, (1994).

Olsson, D.M., "Estimation for Mixtures of Distributions by Direct Maximization of the Likelihood Function." Journal of Quality Technology, Vol. 11: 153-159, (1979).

Oppenheimer L. Estimation of a Mixture of Exponentials for Complete and Censored Samples. Ph.D. Dissertation, University of North Carolina (1971).

Pak, R. J. "Minimum Hellinger Distance Estimation in Simple Linear Regression Models." Statistics & Probability Letters, (1996).

Pandley, M. and S. K. Upadhyay. "Selection of Weibull Shape Parameter, Based on Adaptive Estimation." IEEE Transactions on Reliability, R-32, No. 2, (June 1983).

Parr, William C. Minimum Distance and Robust Estimation. Ph.D. Dissertation. The School of Humanities and Sciences, Southern Methodist University, (1978).

Parr, William C. and William R. Schucany. "Minimum Distance and Robust Estimation." Journal of the American Statistical Association, 75, No. 3: 616-624, (1980).

Payne, James A. Introduction to Simulation: Programming Techniques and Methods of Analysis. New York: McGraw-Hill Book Co., (1982).

Pearson, Karl, "Contributions to the Mathematical Theory of Evolution." Philosophical Transactions of the Royal Society, Vol. 185A, 77-110, (1894).

Peters, B. C. and H. F. Walker, "An Iterative Procedure for Obtaining Maximum-Likelihood Estimates of the Parameters for a Mixture of Normal Distributions." University of Houston, (1977).

Pohl, E., "Environmental Stress Screening Strategies for multi-component systems with Weibull failure times." IEEE ARMS, (1995).

Rao, C. R., "The Utilization of Multiple Measurements in Problems of Biological Classification." Journal of the Royal Statistical Society, Vol. 10B, 159-203, (1948).

Redner, R.A., "Maximum Liklihood Estimation for Mixture Models" JSC-1632, (October 1980).

Redner, R. A. and Walker, H.F., "Mixture Densities, Maximum Likelihood and the EM Algorithm." SIAM Review, Vol. 26: 195-239, (1984).

Rey, William J. J. Introduction to Robust and Quasi-Robust Statistical Methods. New York: Springer-Verlag, (1983).

Rider, P.R., "The Method of Moments Applied to a Mixture of Two Exponential Distributions." Annals of Mathematical Statistics, Vol. 32: 143-147, (1961).

Rockette, Howard, Charles Antle, and Lawrence A. Klimko. "Maximum Likelihood Estimation with the Weibull Model." Journal of the American Statistical Association, Vol. 69, No. 345: 246- 249, (March 1974).

Sahler, W. "Estimation by Minimum-Discrepancy Methods." Metrika, 16: 85-106, (1970).

Sinha, S.K., Reliability and Life Testing. Wiley Eastern Limited, New Delhi, p252, (1986).

Song, Yang. "Minimum Hellinger Distance Estimation of Parameter in the Random Censorship Model." Annal of Statistics, (1991).

Stephens, M. A. "EDF Statistics for Goodness of Fit and Some Comparisons." Journal of the American Statistical Association, Vol. 69, No. 343: 730-737, (1974).

Tan, W.Y. and W.C. Chang, "Some Comparison of the Method of the Moments and the Method of Maximum Likelihood in Estimating Parameters of a Mixture of Two Normal Densities." Journal of the American Statistical Association, Vol. 67: 702-708, (1972).

Tittering, A. F. M. Smith, U.E. Markov, Statistical Analysis of Finite Mixture Distributions; John Wiley and Sons (1985).

Wilson D. L. "Parameter Estimation and Goodness-of-fit in the Finite Mixed Exponential Distribution." A project for the Degree of Industrial Engineer, Syracuse University (1978).

Wingo, D. Royland. "Maximum Likelihood Estimation of the Parameters of the Weibull Distribution by Modified Quasilinearization." IEEE Transactions on Reliability, R-21, No. 2: 89-93, (1972).

Wirth, Niklaus. Algorithms + Data Structures = Programs. Englewood Cliffs, N.J.: Prentice-Hall, Inc., (1976).

Wolfe, J.H., "Pattern Clustering by Multivariate Mixture Analysis." Multivariate Behavioral Research, Vol. 5: 329-350, (1970).

Wolfowitz, J. "The Minimum Distance Method." Annals of Mathematical Statistics, Vol. 28: 75-88, (1957).

Wolfowitz, J. " Estimation by the Minimum Distance Method." Annals of the Institute of Statistical Mathematics, Vol. 5: 9-23, (1953).

Woodard, Whitney E. "Minimum Hellinger Distance Estimation of Mixture Proportions." Journal of Statistical Planning and Inference, (1995).

Woodward, Wayne A., Suchany, W. R., Lindsey, H. and Parr, W. C. "A Comparison of Minimum Distance and Maximum Likelihood Techniques for Proportion Estimation." (1984).

Wu, Ke. " On MDE in two-sample Scale Problem with Right Censoring." Thesis - Texas Tech, (1995).

Yang, Song; Koul, Hir. "Minimum Hellinger Distance Estimation of the Center of Symmetry with Randomly Censored Data." Metrika, (1995).

Yuan, John; Shih, Shi. "Mixed Hazard Models and their Applications." Reliability Engineering and System Safety, (1991).

Zhiliang, Ying. "Minimum Hellinger-type Distance Estimation for Censored Data." Annals of Mathematical Statistics, (1992).

Vita

Capt Mumford [redacted]. He graduated from Northwest High School in 1982 at which time he entered undergraduate studies at the US Air Force Academy in Colorado. He graduated with a Bachelor of Science degree in General Engineering in May 1986. He received his commission upon graduation.

His first assignment was at Peterson AFB as a scientific analyst for Space Systems Division. Later, he transferred to the Cheyenne Mountain Upgrade Site Activation Team (Electronic Systems Division). While at Peterson AFB, he earned a Master of Science degree in Systems Management from the University of Southern California. His second assignment was at the Air Force Operational Test and Evaluation Center as a Test Manager on the Command and Control Element for the Ballistic Missile Defense Program. In August 1995, he entered the School of Engineering, Air Force Institute of Technology.

[redacted]

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1997	Master's Thesis

4. TITLE AND SUBTITLE

Robust Parameter Estimation for the Mixed Weibull (seven parameter) including the Method of Maximum Likelihood and the Method of Minimum Distance

5. FUNDING NUMBERS

6. AUTHOR(S)

Donald A. Mumford, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology/ENS
2750 P Street AFIT/GOA/ENS/97M-12
Wright-Patterson AFB, Ohio 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GOR/ENY/97M-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Operational Test and Evaluation Center / SAL
8500 Gibson Blvd SE
Kirtland AFB, NM 23665-2778

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution is Unlimited

12b. DISTRIBUTION CODE

A

13. ABSTRACT *(Maximum 200 words)*

Robust parameter estimation is successfully applied to the Mixed Weibull (seven parameter) using the Method of Minimum Distance and the Method of Maximum Likelihood. That is, parameters can now be estimated for a mixture of two Weibull distributions where the true populations are co-located, partially co-located or highly separated. Both techniques provided very robust estimates that were far superior to current parameter estimation techniques. Sample sizes as low as ten with mixing proportions down to 0.1 were investigated.

For the MLEs, innovative bounding techniques are presented to allow consistent and correct convergence using any reasonable point estimate. The likelihood function is solved numerically as a non-linear constrained optimization using a quasi-Newton method.

Minimum Distance Estimates (over three hundred scenarios investigated) are derived for some variation or combination of the mixing proportion and the location parameter(s), individually and simultaneously (the Anderson-Darling and Cramer-von Mises statistics were used). In fact, the MDE for the mixing proportion was so effective that future researchers should consider some permanent combination.

14. SUBJECT TERMS

Parameter Estimation; Mixed Distribution; Mixed Weibull; Maximum Likelihood Estimation; Minimum Distance Estimation

15. NUMBER OF PAGES 182

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL