Robust Estimation of the Birnbaum-Saunders Distribution

by

Joanna E. Mills

A Thesis Submitted to the
Faculty of Engineering
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

Major Subject: Applied Mathematics

.
.
.

TECHNICAL UNIVERSITY OF NOVA SCOTIA

Halifax, Nova Scotia

1997

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-31621-1

To my dear father Raymond G. Mills

Table of Contents

List of Tables	vii
List of Figures	ix
List of Symbols and Abbreviations	x
Acknowledgements	xiv
Abstract	xv
1 Introduction	1
2 The Birnbaum-Saunders Distribution	5
2.1 The Derivation	7
2.2 Some Interesting Properties	12
2.3 Maximum Likelihood Estimation	18
2.4 A Graphical Approach to Estimation	24
2.5 The Usefulness of Quantiles	26
2.5.1 Constructing Confidence Intervals	28
3 The Need for Robust Estimation	31
3.1 Robustness Concepts	32
3.2 The Optimal B -Robust Estimator	34

3.3	The Algorithm	36
3.3.1	An Effective Starting Value	38
3.3.2	Limitations of the Mean Mean Estimate	42
3.4	The Weighting Function	46
3.5	The Variance of the OBRE	48
4	Applications to Real Data	52
4.1	Aluminum Fatigue Testing	52
4.1.1	Data Set 1	53
4.1.2	Data Set 2	59
4.1.3	The Choice of c	61
4.2	Fatigue of Concrete	63
4.3	Fluid Failure Data	64
4.4	Steel Data	65
4.5	Fatigue Test Results	66
5	Simulation Study	68
5.1	Suitable Contaminated Models	68
5.2	The Implementation	70
5.3	The OBRE at the Model	71
5.3.1	Parameter Estimates	71

5.3.2	A Look at Relative Efficiencies	73
5.4	The OBRE with Contamination	75
5.4.1	Parameter Estimates	75
5.4.2	The Weights	76
5.4.3	Efficiency	78
5.5	Comments	78
6	Conclusions	80
	References	83
	Appendix A Proof of Theorem 3	88
	Appendix B Data Sets	91
	Appendix C Algorithm Implementation	95
C.1	Program Hierarchy	95
C.2	Program Listings	97

List of Tables

1	Mean (standard deviation) of weights for the OBRE ($c = 4$) for Birnbaum-Saunders data.	47
2	$p = .001$ quantiles for Data Set 1.	58
3	$p = .001$ quantiles for Data Set 2.	60
4	Parameter estimates for concrete data ($c = 4.0$).	63
5	$p = .001$ quantiles for concrete data.	64
6	Parameter estimates for steel data ($c = 4.0$).	65
7	Bias (RMSE) of estimators of the shape parameter α for Birnbaum-Saunders data.	72
8	Bias (RMSE) of estimators of the scale parameter β for Birnbaum-Saunders data.	73
9	Relative efficiencies of the OBRE ($c = 4.0$) for Birnbaum-Saunders data, $n=100$	74
10	Overall relative efficiencies for the OBRE for Birnbaum-Saunders data, $n=100$	75
11	Effects of contamination on the MLE and the OBRE, $n=100$	76
12	Mean (standard deviation) of weights for the OBRE for contaminated Birnbaum-Saunders data.	77
13	$\text{Var}(\text{MLE})/\text{Var}(\text{OBRE})$ for contaminated Birnbaum-Saunders data.	78
14	Aluminum fatigue data, lifetimes in cycles $\times 10^{-3}$ [8].	92

15	Fatigue lifetimes for aluminum, lifetimes in cycles $\times 10^{-3}$ [15].	92
16	Fatigue lifetimes for concrete, lifetimes in cycles $\times 10^{-3}$ [38].	93
17	Insulating fluid failure data, lifetimes in minutes [48].	93
18	Fatigue lifetimes for steel, lifetimes in cycles $\times 10^{-3}$ [38].	94

List of Figures

1	The $(j + 1)th$ cycle of a periodic loading.	8
2	Birnbaum-Saunders distributions for $\alpha = 0.2, 0.6$ and 1.0 , $\beta = 1.0$. . .	14
3	Birnbaum-Saunders distributions for $\alpha = 1.4$ and 1.8 , $\beta = 1.0$	15
4	Birnbaum-Saunders probability plots of the randomly generated observations from a Birnbaum-Saunders distribution with $\alpha = 0.5$, $\beta = 1.0$. . .	19
5	The EIF of a selection of estimators. The largest data value varies from $-\infty$ to $+\infty$	44
6	Normal probability plots of $\hat{\alpha}_{OBRE}$ for sample sizes of size $n = 25, 50$ and 100 , $\alpha = .2$, $\beta = 1.0$	50
7	Normal probability plots of $\hat{\beta}_{OBRE}$ for sample sizes of size $n = 25, 50$ and 100 , $\alpha = .2$, $\beta = 1.0$	51
8	The MLE and the OBRE of the Birnbaum-Saunders distribution for Data Set 1.	55
9	Quantile-quantile plot of the MLE versus the OBRE of the Birnbaum-Saunders distribution for Data Set 1.	56
10	Quantile plots of the Birnbaum-Saunders distribution for Data Set 1.	57

List of Symbols and Abbreviations

- $\mathbf{a}(\theta)$: 2×1 vector defined implicitly through (32)
- $\hat{a}$: least squares estimate of the slope of the line $y = a + bx$
- $A(\theta)$: 2×2 matrix defined implicitly through (32)
- $A^{(n)}$: n th composition of the function $A(x)$ with itself
- $A(x)$: function defined in Method 2 of Section 2.3
- $\hat{b}$: least squares estimate of the intercept of the line $y = a + bx$
- c : robustness constant
- cdf*: cumulative distribution function
- C : a random variable representing the number of cycles to failure distributed as (3)
- C_r^n : number of ways to select an unordered sample of size r from a population of size n
- D_n : cumulative distribution function of the total crack extension W_n
- $E(\cdot)$: expectation operator
- EIF*: empirical influence function
- $f(t; \alpha, \beta)$: probability density function of the Birnbaum-Saunders distribution defined in (1) [also denoted f_θ]
- $\hat{F}$: median rank defined in (22)
- F_n : empirical distribution function of the Birnbaum-Saunders distribution
- $F(t, \alpha, \beta)$: cumulative distribution function of the Birnbaum-Saunders distribution defined in (5) [also denoted $F(\alpha, \beta)$ or F_θ]
- $\mathfrak{F}$: Birnbaum-Saunders family of distributions
- g : size of groups used for jackknife estimator defined in (18)
- $g(x)$: function defined in (15) of Theorem 3
- h : reciprocal of the harmonic mean [also denoted H_n]

- $H(x)$: function defined in Method 2 of Section 2.3
- IF : influence function
- $I_{(A)}$: indicator function of event A
- $IF(t; T, F_\theta)$: influence function of T at the model F_θ , defined in (27)
- $I(\alpha)$: function defined in (19)
- $J(\theta)$: fisher information matrix of the vector θ
- $K(x)$: harmonic mean function defined in (14)
- l_i : load in the i th time step [a *load step*]
- LMS: least median of squares
- LS: least squares
- m : number of groups used for jackknife estimator defined in (18)
- $M(\Psi, F_\theta)$: 2×2 matrix defined in (29)
- M_k : 2×2 matrix defined in (34)
- MLE: maximum likelihood estimator(s)
- n : number of cycles as in (2) or number of observations in the sample as in Theorem 3
- $\mathfrak{N}(\mu, \sigma^2)$: cumulative distribution function of the normal random variable with mean μ and variance σ^2
- $P[C \leq n]$: cumulative distribution function of C defined in (4)
- pdf: probability density function
- OBRE: optimal B -robust estimator(s)
- p : a probability between zero and one
- $\hat{p}_i$: defined in (21)
- q_1, q_2 : roots of (12) of Theorem 2
- Q : 2×2 matrix defined in (42)
- r : harmonic mean defined in (13)
- r_i : i th residual (the actual value minus the fitted value)
- s : arithmetic mean defined in (13) [also denoted S_n]

- $s(t; \theta)$: score function of the Birnbaum-Saunders distribution defined in (28)
- T : Birnbaum-Saunders distributed random variable
- t_i : defined in (21)
- T_n : estimator of θ
- t_p : p th quantile of the Birnbaum-Saunders distribution, defined in (23)
- $(\underline{t}_p, \overline{t}_p)$: confidence interval for t_p , defined in (26)
- U : uniform (0,1) random variable
- $Var(\cdot)$: variance operator
- $V(\Psi, F_\theta)$: asymptotic covariance matrix corresponding to the optimal B -robust estimators of θ for the Birnbaum-Saunders distribution, defined in (41)
- $W_c(t; \theta)$: weight function for a given c and θ , defined in (31)
- W_n : random variable representing the total crack extension during the first n cycles
- X_i : random variable representing the crack extension following the i th load step of load i
- Y_j : random variable representing the crack extension during the j th cycle
- Z : standard normal random variable
- z_p : p th quantile from a standard normal distribution
- α : shape parameter of the Birnbaum-Saunders distribution
- $\hat{\alpha}_{MLE}$: maximum likelihood estimate of α defined in (16) [also denoted $\hat{\alpha}$]
- $\hat{\alpha}_{OBRE}$: optimal B -robust estimate of α
- β : location and scale parameter of the Birnbaum-Saunders distribution
- $\tilde{\beta}$: mean mean estimate of β defined in (17)
- β^* : jackknife estimator of β defined in (18)
- β_i^* : mean mean estimator excluding the i th group
- $\hat{\beta}_{MLE}$: maximum likelihood estimate of β [also denoted $\hat{\beta}$]

- β_n : maximum likelihood estimate of β corresponding to the n th iteration in Method 1 of Section 2.3
- $\hat{\beta}_{OBRE}$: optimal B -robust estimate of β
- β_0 : starting value for iterative methods of Section 2.3
- ε : prescribed epsilon for iterative methods of Section 2.3
- $\in$: belongs to
- η : precision tolerance for OBRE algorithm
- γ_1, γ_2 : constants between zero and one
- μ : mean
- ω : fixed critical crack length
- Φ : cumulative distribution function of the standard normal random variable
- ϕ : probability density function of the standard normal random variable
- θ : vector representing the distributional parameters of the Birnbaum-Saunders distribution
- Ψ : Fisher consistent score function
- $\psi(x)$: function defined in (6)
- $\rho(x)$: function defined in (8)
- σ^2 : variance
- Δt : probability measure which places all its mass in t
- $\Delta\theta$: 2×1 vector defined in Step 5 of Section 3.3
- $\xi(t)$: function defined in (5)

Acknowledgements

I wish to express my deepest gratitude to my supervisor, Dr. D. Dupuis for her invaluable advice and constant encouragement throughout my studies at the Technical University of Nova Scotia. Her office door was always open and I am sincerely thankful for her guidance, enthusiasm and patience. I am also most grateful to Dr. G. Fenton for his incisive suggestions and continual support during this course of study. My genuine appreciation to Dr. C. Field for his insightful comments all the way from sunny Australia.

To Norma Linney for her reassurance and of course spelling assistance, to Travis Blanchett for your refreshing humor, I thank you both. I heartfully thank Ming Chen for helping make the early days easier. His ability to instill confidence in the 'new kid on the block' will remain in my memory forever.

To my many friends, especially Julie Kirk, Paula Bower, Jenny Benvie and Krista Jensen who bore with me through thick and thin, thank you all from the bottom of my heart. Most importantly I wish to sincerely thank my parents. Their immense love and enduring support echoes through all I have accomplished as well as my many hopes and dreams.

Financial support provided by Dr. D. Dupuis through her Natural Sciences and Engineering Research Council grant is gratefully acknowledged. A sincere thank you to Dr. W. Phillips, head of the Department of Applied Mathematics for providing me with financial support. Lastly, I must express my appreciation to the Technical University of Nova Scotia for awarding me with a Bruce and Dorothy Rosetti Scholarship.

Abstract

The Birnbaum-Saunders family of distributions is prevalent in the engineering sciences due to its ability to effectively model fatigue life in particular and stochastic wear-out in general. This distribution can be used to model the life-lengths of structural components in critical applications such as aerospace and bridge design. Typically, the parameters of such distributions are estimated using classical procedures such as maximum-likelihood. Unfortunately these procedures are sensitive to the model deviations that often occur in practice, such as gross errors in the data or model misspecification. These types of contamination can severely distort parameter estimates and have far reaching repercussions on both engineering design and reliability estimates. Design engineers use lower quantiles obtained as functions of the parameter estimates which, if distorted, can lead to dire consequences such as structural failure in buildings and bridges or premature failure of mechanical components.

As a powerful alternative, robust techniques are developed herein for the estimation of the distributional parameters. These techniques are based on a combination of optimal B -robust and robust regression concepts that perform almost equally as well as maximum likelihood at the model and are far superior in the presence of contamination. Robust techniques also assign a weight to each observation giving an indication of the fit of the model to the data. Real examples and simulation results help demonstrate the merits of these robust estimation techniques.

1 Introduction

It is well known that a structure properly designed for a given static load may fail after some time when exposed to a load varying in time, although this variable load may never exceed nor even approach the static load for which the structure was designed. Structural failure in materials designed correctly from the static point of view, but placed in normal use under conditions involving variable loading is known as fatigue failure. *Fatigue* is the initiation and propagation of microscopic cracks into macro cracks by the repeated application of stresses [43].

Fatigue was first recognized as a cause of failure in carriage axles in the early 1800's [50]. Later, in the 1950's it was blamed for the crashes of two British Comet airliners [9]. In fact, fatigue is responsible for a great number of failures in connecting rods and crankshafts of engines, steam or gas turbine blades, connections and supports for bridges, railroad wheels and axles, and other parts subjected to variable loading. Hence fatigue cracking is a fundamental design issue for both structures and components. If fatigue cracking does occur, it can result in failure as the crack grows to a size such that the remaining cross-section is no longer adequate to carry the imposed forces.

The task of the engineer is to proportion those structural members that have a potential for failure by fatigue crack growth so that they have a life expectancy compatible with the design life of the structure. This is done with the knowledge that some probability of failure must be accepted, since in reality no structure can be designed for zero probability of failure. The design is carried out in the expectation that initial flaws will be present in all fabricated steel structures and all such members will contain residual stresses of relatively high magnitude. A concomitant feature is that it is possible to identify the size of flaws that are permissible and then to use this information as the basis for both initial and on-going inspection of the structure.

This latter feature is not yet well-incorporated into design specifications. Instead, the usual procedure is to accept permissible flaw sizes consistent with the specifications that accompany the fabrication processes, e.g., the welding specifications.

When fatigue was first recognized, compensation was provided by increasing the factor of safety which resulted in increased strength of the structure. The crudeness of this approach led to some structural components being ultraconservative while others still failed. Further increases in the factor of safety would have made designs excessively massive and, if applied to aircraft, would have resulted in airplanes that were too heavy to fly. Consequently, the aircraft industry provided the motivation and resources for tests and studies that have revealed many of the secrets of the nature of fatigue.

If a number of carefully specified structures, made as uniformly as possible, are subjected to the same kind of dynamic loading, they most likely will fail after different times. The life-length T of a structure under such normal use is, therefore, not a fixed value but a random variable. It is of considerable practical importance to know the probability distribution of the life-length T for a structure under normal use. For example, knowing this probability distribution enables us to determine a time t such that the probability of failure before this time is .001. We can then replace the structure before time t which assures that no more than one structure in a thousand will fail. In the late 1960's the Birnbaum-Saunders family of probability distributions was derived explicitly for this type of phenomenon at the Boeing Scientific Research Laboratories in Seattle, USA [6]. This has led to procedures by which parts can be designed to carry variable loads with an acceptable degree of reliability. In Section 2 we establish the merits of the Birnbaum-Saunders distribution and examine some of its interesting properties.

The Birnbaum-Saunders distribution is fully specified by two parameters one of which is a measure of skewness or *shape* and the other a measure of *location* as well as *scale*.

These population parameters are usually estimated on the basis of statistics calculated from representative samples drawn from the population.

Statistics, calculated from random samples, that are used to estimate population parameters are called estimators. Good estimators should have, insofar as possible, desirable properties of unbiasedness, consistency, and efficiency. An *unbiased estimator* is one that neither consistently underestimates nor overestimates the true value of the parameter. A *consistent* estimator is one that is unbiased and converges more closely to the true value of the population parameter as the sample size increases. An *efficient* estimator is a consistent estimator whose standard deviation is smaller than the standard deviation of any other estimator for the same population parameter. One always strives to use the best possible estimator within the context of these desirable properties.

In Sections 2.3 and 2.4 we take a detailed look at some of the methods of estimating the parameters of a Birnbaum-Saunders distribution that have been utilized to date. Such methods include maximum likelihood estimation (MLE) for which numerical iterative procedures are required due to the lack of explicit formulae. Our examination exposes the limitations of many of these methods in the presence of deviations from the assumed underlying models and establishes the necessity for *robust* statistics. Such robust statistics provide statisticians with a method to deal effectively with deviations from ideal assumptions. In Section 3, robustness concepts are developed in the context of the Birnbaum-Saunders distribution and an effective robust estimation procedure is suggested. Algorithms for its implementation are described.

Sections 4 and 5 help substantiate our robust estimation procedure using both real data and a simulation study. A number of the real examples with which we test our method have been used as bench marks when estimating parameters of Birnbaum-Saunders distributions since the late 1960s. We look at both the parameter estimates themselves as well as at lower quantiles, which are of great use to design engineers.

The simulation study investigates the effectiveness of our robust estimation procedure both at the model and in the presence of contamination. Conclusions as well as suggestions for future research are found in Section 6. All program listings and relevant data sets are provided in the appendices.

2 The Birnbaum-Saunders Distribution

A thorough literature review determined that a number of families of probability distributions have been considered for describing the behavior of life-lengths of materials under normal use. Epstein and his associates explored in detail the family of exponential distributions [26]. Normal distributions have been used extensively, sometimes after a logarithmic transformation of the original data [19]. The Weibull distribution has also received considerable attention [28] for representing fatigue life-length data at a constant stress level.

Unfortunately, these families of distributions all have limitations. The exponential distribution, for instance, has a number of desirable mathematical properties, but is greatly limited by its memoryless property. This property implies that if a material has not failed up to a time t , then the probability distribution of its future life-length $T - t$ is the same as if the material was new and had just been placed in use at time t . Clearly such a property makes the exponential distribution inadequate for describing the life-length distribution of any material which, in normal use, undergoes changes affecting its future life-length.

The normal distribution is an excellent analytic tool, but its symmetry is restrictive since many life-length distributions occurring in practical applications are markedly skewed.

As a powerful alternative we turn to the Birnbaum-Saunders family of distributions which was originally suggested in the fatigue context and has since been shown to have more general applicability. Our primary purpose is to investigate the parametric estimation problem of this representative class of life-length distributions.

In 1969 Birnbaum and Saunders [6] proposed the *Birnbaum-Saunders distribution*

with probability density function (pdf),

$$f(t; \alpha, \beta) = \frac{1}{2\alpha\beta} \left(\frac{t}{\beta}\right)^{-1/2} \left[1 + \left(\frac{t}{\beta}\right)^{-1}\right] \times \frac{1}{\sqrt{(2\pi)}} \exp \left[-\frac{1}{2\alpha^2} \left(\frac{t}{\beta} - 2 + \frac{\beta}{t}\right) \right], \quad t > 0; \alpha, \beta > 0, \quad (1)$$

to represent the distribution of lifetimes of components under specified conditions of wear-out. T is a random variable and α and β are the distributional parameters. This type of pdf is commonly referred to as a *fatigue life* distribution. Later, in 1986, Desmond [20] presented a more general derivation and showed that this pdf could be rewritten alternatively as

$$f(t; \alpha, \beta) = \frac{1}{3} \left[\left(\frac{\alpha^{-2}\beta}{2\pi t^3}\right)^{1/2} \exp \left[-\frac{1}{2\alpha^2\beta} (t - 2\beta + \beta^2 t^{-1}) \right] \right] + \frac{1}{3} \left[\left(\frac{\alpha^{-2}\beta^{-1}}{2\pi t}\right)^{1/2} \exp \left[-\frac{\beta}{2\alpha^2} (t^{-1} - 2\beta^{-1} + \beta^{-2}t) \right] \right].$$

He also put forward a plausible argument for the appropriateness of the Birnbaum-Saunders distribution in the context of fatigue by consideration of the differential-equation models commonly employed in the engineering literature to describe crack propagation.

Desmond went on to illustrate that the Birnbaum-Saunders distribution is appropriate for modeling failure modes of frequent occurrence in the engineering sciences. These failure modes arise from the deleterious effect of random stress environments being modeled as stationary continuous-time stochastic processes. The term *stress* here, is not restricted to its usual mechanical sense, but may refer to any continuously varying response process such as voltage or temperature.

In 1993 Setó, Iwase, and Oohara [47] analyzed the rainfall characteristics of the city of Hiroshima and reached the conclusion that for these data the distribution of periods

of continuous rainfall is well fit by a Birnbaum-Saunders distribution. The Birnbaum-Saunders distribution is also applicable to the analysis of airplane components exposed to randomly fluctuating stresses in flight resulting from wind gusts, or to sudden random variations in peak stresses during the take off-landing cycle [45].

2.1 The Derivation

The derivation of the Birnbaum-Saunders family of distributions from considerations of the basic characteristics of the fatigue process is not only interesting but facilitates the development of a practical interpretation of the distributional parameters. This, in turn, leads to a better overall understanding of the distributions and their applications.

In their 1969 paper Birnbaum and Saunders [6] ask us to consider standardized material specimens which are subjected to fluctuating stresses by a periodic loading. A load, in this context, is a continuous unimodal function on the unit interval, the value of which at any time gives the stress resulting in the deformation of the specimen. The oscillating deformation is caused by a sequence of loads, l_i , applied at discrete equispaced time steps such that m time steps define one *cycle* of deformation. Each load l_i , $i = 1, 2, \dots$ will be referred to as a load step. The load steps are cyclic with period $m > 1$ such that

$$l_{i+jm} = l_{i+km}$$

for any integers j and k . That is, l_i corresponds to the load in the i th time step. Hence the $(j + 1)$ th cycle consists of the load steps $(l_{jm+1}, \dots, l_{j+m})$ as illustrated in Figure 1.

The fatigue failure is assumed to be due to the initiation, growth and ultimate extension of a dominant crack. At each load step this crack is extended by some amount

Figure 1: The $(j + 1)th$ cycle of a periodic loading.

which is a random function due to the variation in the material properties, the prior load steps in the current cycle and possibly the actual crack extensions caused by prior load steps in the current cycle. Formally this assumption is:

- (1) The incremental crack extension X_i , at the end of the i th load step is a random variable with a distribution that depends only on the load steps and crack extensions preceding it *within the cycle in which it occurs*. Thus X_{jm+1} is only dependent on the current load step. X_{jm+2} depends on X_{jm+1} and l_{jm+1} and l_{jm+2} , etc. This assumption is only reasonable if the beginning of a cycle corresponds to a zero stress state; $l_{jm+0} = 0$.

Consequently the crack extension during the $(j + 1)th$ cycle is

$$Y_{j+1} = X_{jm+1} + \cdots + X_{jm+m} \quad \text{for } j = 0, 1, \dots,$$

where X_{jm+i} is the crack extension following the application of load l_{jm+i} in the $(j + 1)th$ cycle. These crack extensions are generally microscopic except near failure.

It follows from the first assumption that regardless of how much dependence exists between the successive random extensions per oscillation in each cycle, the random total crack extensions per cycle are independent.

This yields the second assumption:

- (2) The total crack extension Y_j due to the j th cycle is a random variable with mean μ and variance σ^2 for all $j = 1, 2, \dots$. Birnbaum and Saunders use the notation

$$W_n = \sum_{j=1}^n Y_j, \quad (2)$$

where W_n is a random variable representing the total crack extension during the first n cycles which has distribution function

$$D_n(w) = P[W_n \leq w], \quad \text{for } n = 1, 2, \dots$$

One can then obtain the distribution of C , the number of such cycles until failure, in the case where failure is defined as the crack length exceeding some fixed critical length ω for the first time. That is

$$\begin{aligned} P[C \leq n] &= P[W_n > \omega] \\ &= 1 - P[W_n \leq \omega] \\ &= 1 - D_n(\omega) \end{aligned}$$

so that

$$P[C \leq n] = P \left[\sum_{i=1}^n \frac{Y_i - \mu}{\sigma \sqrt{n}} > \frac{\omega - n\mu}{\sigma \sqrt{n}} \right] \quad (3)$$

and since the Y_j are independent and identically distributed random variables, using

the central limit theorem for large n , (3) above can be approximated by the normal distribution. This leads to the third assumption, that is:

(3) The distribution of C is

$$P[C \leq n] = 1 - \Phi \left[\frac{\omega - n\mu}{\sigma\sqrt{n}} \right] = \Phi \left[\frac{\mu\sqrt{n}}{\sigma} - \frac{\omega}{\sigma\sqrt{n}} \right]. \quad (4)$$

where Φ is the distribution function of the standard normal variate with zero mean and unit variance defined for $-\infty < y < \infty$ by

$$\Phi(y) = \int_{-\infty}^y \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt.$$

Birnbaum and Saunders were aware of the fact that Assumption 3 takes as exact, for the applications intended, an approximate equality.

For the physical justification of Assumption 3 they suppose that they are dealing with a large number of cycles which would describe phenomena as complex as the ground-air-ground pattern in aeronautics. It is then reasonable to assume that we are dealing with a large number of distinct phases of loading. Even though the total crack extension per cycle is the sum of random variables which are not necessarily either independent or identically distributed, the number of summands may be sufficiently large to make it reasonable to assume that the total crack extension per cycle, namely Y_j is itself approximately normal.

Secondly, given that fatigue is often measured in the thousands or even millions of cycles, such large numbers of random variables make Assumption 3 a widely practical assumption in view of its simple analytic form.

This distributional form, parametrized differently, had been previously obtained by Freudenthal and Shinozuka [29] via a heuristic derivation based on engineering con-

siderations.

Returning to (4) we can write

$$\alpha = \frac{\sigma}{\sqrt{\mu\omega}}, \quad \beta = \frac{\omega}{\mu}$$

and replace n by the nonnegative real variable $t > 0$. Denoting the continuous extension of the discrete random variable C by T , a continuous nonnegative random variable, it follows by Assumption 3 that T has the life-length distribution

$$F(t; \alpha, \beta) = \Phi[\alpha^{-1}\xi(t/\beta)] \quad \text{for } t > 0, \quad (5)$$

where

$$\alpha > 0, \quad \beta > 0,$$

and

$$\xi(t) = t^{1/2} - t^{-1/2}.$$

We have now arrived at the two-parameter family of distributions which is deemed to be a plausible model for the distribution of fatigue life. This life-length is in effect the mathematical representation of the number of cycles needed for a dominant crack in a material to reach a critical size ω at which point fatigue failure occurs.

Desmond [21] later summarized the following assumptions about the *fatigue* process upon which the Birnbaum-Saunders distribution is based:

- Fatigue failure is due to repeated application of a common cyclic stress pattern.
- Under the influence of this cyclic stress a dominant crack in the material grows until a critical size w is reached at which point fatigue failure occurs.

- The crack extensions in each cycle are random variables with the same mean and variance.
- The crack extensions in each cycle are statistically independent.
- The total extension of the crack, after a large number of cycles, is approximately normally distributed by the central limit theorem.

In so doing he noticed that in assuming that the crack length is normally distributed Birnbaum and Saunders have implied that, in principle, the crack length could take on negative values with nonzero probability. This possibility is ignored in the derivation. Desmond provided a more general derivation of the Birnbaum-Saunders distribution without assuming normality for the distribution of crack length. He found that

- A variety of distributions for crack size (many of which are on the positive real line) are possible which still result in a Birnbaum-Saunders distribution for fatigue life.
- It is possible to allow the crack increment in a given cycle to depend on the total crack size at the beginning of the cycle and still obtain a fatigue life distribution of the Birnbaum-Saunders type.

It should be noted that this later point removes what was probably a major flaw in the attempt to found the Birnbaum-Saunders model on a physical basis. Desmond also went on to derive the Birnbaum-Saunders distribution using a biological model which Cramér [16] had used to motivate the lognormal distribution.

2.2 Some Interesting Properties

We will denote the set of all life distributions of the Birnbaum-Saunders form (5) by $\mathfrak{F}$ and will let T have a distribution from this family. β is simply a multiplier and

does not affect the shape of the distribution of T but instead functions as both a location parameter (it can be shown to be the median of the distribution) and a scale parameter. In a fatigue context β corresponds, roughly, to a typical number of cycles to failure. α is a shape parameter which causes the distribution to tend to symmetry as it tends to zero. The pdfs for various α , displayed in Figures 2 and 3, reinforce this property. Here β has been taken as one without loss of generality since it is a scale parameter. One can see that as α increases the distribution shifts from being predominantly symmetrically distributed to being severely skewed to the right.

From (5) we see that $\alpha^{-1}\xi(T/\beta)$ is a standard normal random variable with zero mean and unit variance. So, if we let X be $\mathfrak{N}(0, \frac{1}{4}\alpha^2)$ we see that, in distribution,

$$2X = \xi(T/\beta).$$

If, for real x , the function ψ is defined by

$$\psi(x) = \xi^{-1}(2x) \tag{6}$$

then

$$T = \beta\psi(x). \tag{7}$$

Using elementary algebra it can then be shown that

$$\psi(x) = [\rho(x)]^2$$

where

$$\rho(x) = x + (x^2 + 1)^{1/2}. \tag{8}$$

(a) $\alpha = 0.2$ (b) $\alpha = 0.6$ (c) $\alpha = 1.0$ Figure 2: Birnbaum-Saunders distributions for $\alpha = 0.2, 0.6$ and 1.0 , $\beta = 1.0$.

(a) $\alpha = 1.4$ (b) $\alpha = 1.8$ Figure 3: Birnbaum-Saunders distributions for $\alpha = 1.4$ and 1.8 , $\beta = 1.0$.

So by (7) above

$$T = \beta[1 + 2X^2 + 2X(1 + X^2)^{1/2}],$$

where X is $\mathfrak{N}(0, \frac{1}{4}\alpha^2)$. Hence we have immediately

$$E(T) = \beta(1 + \frac{1}{2}\alpha^2), \quad (9)$$

$$E(T^2) = \beta^2(1 + 2\alpha^2 + \frac{3}{2}\alpha^4), \quad (10)$$

$$\text{Var}(T) = (\alpha\beta)^2(1 + \frac{5}{4}\alpha^2), \quad (11)$$

and we note that for fixed α the variance of T increases as the scale parameter β increases. Empirical evidence shows that this is true for the fatigue lives themselves [6]. Similarly, for fixed β the variance of T also increases as the shape parameter α increases. By noting that

$$\frac{1}{\rho(x)} = -x + (1 + x^2)^{1/2} = \rho(-x),$$

we see that, whenever $-X$ has the same distribution as X , we have in distribution,

$$\frac{1}{\rho(x)} = \rho(x), \quad \frac{1}{\psi(x)} = \psi(x).$$

This leads directly to the following theorem:

Theorem 1 *If T has a fatigue life distribution $F(\alpha, \beta)$, in $\mathfrak{F}$ then T^{-1} has the distribution $F(\alpha, \beta^{-1})$ also in $\mathfrak{F}$, and, for any real $a > 0$, the random variable aT has a distribution given by $F(\alpha, a\beta)$ in $\mathfrak{F}$. Moreover,*

$$E(T) = \beta(1 + \frac{1}{2}\alpha^2)$$

$$\text{Var}(T) = (\alpha\beta)^2(1 + \frac{5}{4}\alpha^2).$$

If Z is a standard normal variate, then

$$1 + \frac{1}{2}\alpha^2 Z^2 + \alpha Z(1 + \frac{1}{4}\alpha^2 Z^2)^{1/2}$$

has the distribution $F(\alpha, 1)$ in $\mathfrak{F}$.

For a proof of this theorem the reader is referred to [8]. This is a very useful result. It provides us with a straightforward means of generating Birnbaum-Saunders random deviates. One need only generate observations as

$$1 + \frac{1}{2}\alpha^2 z^2 + \alpha z(1 + \frac{1}{4}\alpha^2 z^2)^{1/2}$$

where α is as desired and the z are randomly generated standard normal observations. Then simply by multiplying these generated observations by the desired β one can simulate from any $F(\alpha, \beta)$ distribution.

Chang and Tang [14] recently proposed an alternative random variate generation algorithm for the Birnbaum-Saunders distribution as is detailed in the following theorem:

Theorem 2 Suppose Z is a standard normal variable and U is a uniform $(0, 1)$ random variable. Let q_1 and q_2 be the roots of the quadratic equation

$$q^2 - \beta(2 + \alpha^2 z^2)q + \beta^2 = 0, \tag{12}$$

where z is a value of Z . Then, a Birnbaum-Saunders random variable with α and β as its shape and scale parameters, respectively, is given by

$$q_1 I_{(U > 0.5)} + q_2 I_{(U < 0.5)},$$

where $I_{(A)}$ is the indicator function of event A .

As a mechanism of validating their Birnbaum-Saunders random variate generator

Tang and Chang developed *Birnbaum-Saunders probability plots* [14]. If such plots display obvious linearity there is no evidence to indicate that the generated observations do not come from a Birnbaum-Saunders distribution. Tang and Chang successfully authenticated their random variate generator [14] using these plots.

Although quite elegant, the random variate generator of Chang and Tang is complicated by the root-finding issue. Consequently, the former method is preferred due to its computational simplicity. To validate it is equally as effective a random variate generator we utilized Birnbaum-Saunders probability plots in the identical fashion as did Tang and Chang. These plots are given in Figure 4 and give no evidence to suggest otherwise in being obviously linear and improving with increasing n . These results are in close agreement with those of Tang and Chang [14] and hence we recommend and implement the former method.

2.3 Maximum Likelihood Estimation

As our main concern lies with the parametric estimation problem for the Birnbaum-Saunders family it is imperative that we examine the methods that are in frequent use today.

In their 1969 paper Birnbaum and Saunders [8] developed two iterative numerical methods for the computation of the maximum likelihood estimates (MLE) of the parameters α and β of a Birnbaum-Saunders Distribution. They also obtained a simple estimate of β , referred to as the *mean mean* estimate ($\tilde{\beta}$) and showed it to be consistent as well as virtually the same as the MLE for small values of α . In all cases, Birnbaum and Saunders recommended it as a good starting value for the iterative numerical methods which compute the MLE.

Let us begin a review of the work of Birnbaum and Saunders by defining the arithmetic

(a) $n=25$. (b) $n=50$. (c) $n=100$.

Figure 4: Birnbbaum-Saunders probability plots of the randomly generated observations from a Birnbbaum-Saunders distribution with $\alpha = 0.5$, $\beta = 1.0$. Observations t_i , $i = 1, \dots, n$ are plotted against p_i where $p_i = \Phi^{-1}(\hat{F}(t_i))\sqrt{t_i}$, Φ is the standard normal distribution and $\hat{F}(t_i) = \frac{i-3}{n+4}$ (the median rank).

and harmonic means of a given set of positive numbers $t_1, \dots, t_n$ by

$$s = \frac{1}{n} \sum_{i=1}^n t_i \quad \text{and} \quad r = \left[\frac{1}{n} \sum_{i=1}^n t_i^{-1} \right]^{-1}, \quad \text{respectively.} \quad (13)$$

The harmonic mean function K is defined by

$$K(x) = \left[\frac{1}{n} \sum_{i=1}^n (x + t_i)^{-1} \right]^{-1} \quad \text{for } x > 0. \quad (14)$$

They then state the following:

Theorem 3 *If $t_1, \dots, t_n$ is a sample of independent observations each uniquely distributed by $F(\alpha, \beta) \in \mathfrak{F}$, then the MLE $\hat{\beta}$ of β is the unique positive solution of $g(x) = 0$ where g is the function defined by*

$$g(x) = x^2 - x[2r + K(x)] + r[s + K(x)]. \quad (15)$$

Furthermore $r < \hat{\beta} < s$. We can also express the MLE $\hat{\alpha}$ of α , defined in terms of $\hat{\beta}$, by

$$\hat{\alpha} = \left(\frac{s}{\hat{\beta}} + \frac{\hat{\beta}}{r} - 2 \right)^{1/2}. \quad (16)$$

It is worthwhile examining the proof [8] of this theorem which is given in Appendix A. This leads to a better understanding of the iterative numerical methods developed to obtain the MLE. Of interest is the fact that once we have obtained the MLE of β we can use this to obtain the MLE of α directly using (16). Consequently methods of obtaining these estimates normally focus on estimating β .

Next, Birnbaum and Saunders presented two methods of finding $\hat{\beta}$ both of which are stated here without proof. The proofs are quite involved and relatively unimportant

for our purposes as these methods are standard procedures in statistics today.

Method 1

If the data satisfy the inequality

$$2s < 3r + \min(t_1, \dots, t_n),$$

then the Newton iteration procedure

$$\beta_{n+1} = \beta_n - \frac{g(\beta_n)}{g'(\beta_n)}, \quad n = 0, 1, \dots$$

will converge to $\hat{\beta}$ for all initial points $r < \beta_0 < s$, where the function g is as defined in (15).

Newton iteration procedures of this type are commonly used.

Method 2

If the data satisfies

$$2r > s,$$

then for $H = 1/2K$, where K is as defined in (14), set

$$A(x) = r + H(x) - (H^2(x) - r(s - r))^{1/2}$$

and for all initial points $r < \beta_0 < s$, as $n \rightarrow \infty$ the iteration $A^{(n)}(\beta_0)$ converges to $\hat{\beta}$.

The conditions given in the two methods are not stringent in practice but will nearly always be satisfied. The iterations are terminated when the successive iterates agree within a prescribed epsilon ϵ .

Good starting values are paramount for ensuring the convergence of most iterative methods. In the two methods just presented we are predominantly concerned with obtaining a sound starting value for β . It is hoped that this starting value (β_0) will enable us to converge to the MLE of β . From this MLE of β we can then directly obtain the MLE of α via (16). Birnbaum and Saunders took the starting value of β to be the *mean mean* estimate, that is, $\beta_0 = \tilde{\beta}$ where

$$\tilde{\beta} = (SR)^{1/2}. \quad (17)$$

Both S and R, now considered as random variables and written in the upper case, were defined in (13). One can see that the mean mean estimate for the median life β is actually the geometric mean of the harmonic and arithmetic mean of the sample lives in the data.

Extensive investigations by Birnbaum and Saunders [8] lead to the conclusion that Method 2, which computationally is the simpler, works as well for $\alpha \leq 0.5$ as the well known Newton procedure, Method 1. However, its convergence rate deteriorates considerably outside this range and it does not work at all once α becomes as large as two. This limitation and the fact that computational speed is nowhere near as great an issue today, make Method 1 our choice for computing $\hat{\beta}$.

Birnbaum and Saunders asserted the great utility of the mean mean estimator. It is a consistent estimate of β which for small α ($\alpha < 1/2$), is almost unbiased while having a variance of approximately $(\alpha\beta)^2/n$. Moreover, for $\alpha < 1/2$, the mean mean estimate is virtually the same as the MLE estimator of β . They conclude that in fatigue applications the size of α is sufficiently small to allow the use of the mean mean estimate as an estimate of β . We are therefore quite confident in the suitability of the mean mean estimate as a starting value for the two methods discussed and have implemented it accordingly.

In 1988 Ahmad [1] noted that while the mean mean estimator is asymptotically unbiased, it does have a positive bias in finite size samples. He suggested the following jackknife estimator, based on a random sample $t_1, \dots, t_n$ of size $n = mg$ (m and g being integers).

“Divide the sample into m groups $B_1, \dots, B_m$ of size g and calculate

$$\beta_i^* = \frac{\sum_{(i)} t_j}{\sum_{(i)} t_j^{-1}}, \quad \text{for } i = 1, \dots, m$$

where $\sum_{(i)}$ denotes summation over all $j = 1, \dots, n$ except $j \in B_i$. The jackknife estimator is

$$\beta^* = m\tilde{\beta} - (1 - m^{-1}) \sum_{i=1}^m \beta_i^*, \quad (18)$$

where $\tilde{\beta} = \left(\frac{\sum_{j=1}^n n t_j}{\sum_{j=1}^n n t_j^{-1}} \right)^{1/2}$ [the mean mean estimate].”

Ahmad also showed that β^* of (18), is a consistent estimator of β . He evaluated the mean square errors of both the mean mean estimator $\tilde{\beta}$ and the jackknife estimator β^* , concluding that β^* has the smaller error, although both have the same limiting distribution. We implement the mean mean estimate due to its much smaller computational requirements. This decision has no effect on our conclusions as it is only used as a starting value and either estimate will still lead to the MLE which is of primary interest.

In 1981 Engelhardt, Bain and Wright [25] showed that the Birnbaum-Saunders pdf satisfies regularity conditions that ensure that the MLE $(\hat{\alpha}, \hat{\beta})$ of (α, β) have an asymptotically joint bivariate normal distribution with mean (α, β) and asymptotic variance-covariance matrix

$$[nB_{ij}(\alpha, \beta)]^{-1}, \quad i = 1, 2, \quad j = 1, 2$$

where

$$\begin{aligned} B_{1,1}(\alpha, \beta) &= 2\alpha^{-2}, \\ B_{1,2}(\alpha, \beta) &= B_{2,1}(\alpha, \beta) = 0, \\ B_{2,2}(\alpha, \beta) &= \beta^{-2} \left\{ \frac{1}{4} + \alpha^{-2} + I(\alpha) \right\} \end{aligned}$$

with

$$I(\alpha) = 2 \int_0^\infty \left[1 + \xi^{-1}(\alpha z)^{-1} - \frac{1}{2} \right]^2 \phi(z) dz, \quad \phi(z) = \Phi'(z), \quad (19)$$

where, as before, $\xi(t) = t^{1/2} - t^{-1/2}$. Notice that $\hat{\alpha}$ and $\hat{\beta}$ are asymptotically independent of each other. Moreover, $I(\alpha)$ can be estimated once $\hat{\alpha}$ is known (numerical integration is required). The asymptotic covariance structure will be used in later sections where the variance of the MLE is required in order to compute relative efficiencies and confidence intervals.

2.4 A Graphical Approach to Estimation

In 1994 Chang and Tang [12] presented a simple graphical technique: the construction of Birnbaum-Saunders probability plots, which is analogous to the construction of quantile-quantile plots. These plots, as first introduced in Section 2.2, can be used to check for the goodness-of-fit of failure times following a Birnbaum-Saunders distribution as well as to estimate the parameters.

As pointed out by Bhattacharyya and Fries [4], statistical inferences regarding the Birnbaum-Saunders distribution are not easy to perform. For example, as seen earlier, the MLE of its parameters must be obtained by solving a rather complicated nonlinear equation.

Chang and Tang had the insight to transform and parametrize the data to result in a

pair of variables with a linear relationship. Then, the parameters could be estimated from the intercept and slope of the fitted straight line. They chose to obtain these estimates via the Method of Least Squares (LS).

The cumulative distribution function of the Birnbaum-Saunders distribution as given in (5) can be written directly as:

$$F(t; \alpha, \beta) = \Phi\left[\frac{1}{\alpha}\left(\sqrt{\frac{t}{\beta}} - \sqrt{\frac{\beta}{t}}\right)\right]; \quad t \geq 0, \alpha > 0, \beta > 0; \quad (20)$$

where α is the shape parameter and β is the scale parameter. From (20) we have

$$t = \beta + \alpha\sqrt{\beta}\sqrt{t}\Phi^{-1}(F).$$

It is very difficult, if not impossible, to derive a linear functional in t which is essential for probability plotting. Chang and Tang circumvent this problem by putting

$$p = \sqrt{t}\Phi^{-1}(F)$$

to result in a simple linear form, $y = a + bx$, where the ordinate is t and the abscissa is p ; i.e.

$$t = \beta + \alpha\sqrt{\beta}p.$$

Suppose we have a set of n failure times $t_1, t_2, \dots, t_n$. Then, by plotting t_i against p_i , where

$$\hat{p}_i = \sqrt{t_i}\Phi^{-1}[\hat{F}(t_i)], \quad (21)$$

we obtain an approximately straight line assuming the failure times follow a Birnbaum-Saunders distribution. There exists several choices for $\hat{F}(t_i)$. They use the median

rank, i.e.,

$$\hat{F}(t_i) = \frac{i - 0.3}{n + 0.4}. \quad (22)$$

Tang and Chang fit a straight line to the data, using (21), obtaining LS estimates of the intercept, $\hat{a}$, and the slope $\hat{b}$. These lead directly to parameter estimates:

$$\begin{aligned} \hat{\beta} &= \hat{a} \\ \hat{\alpha} &= \frac{\hat{b}}{\sqrt{\hat{a}}} \end{aligned}$$

One of the advantages of this method is that $\hat{\beta}$ is given effortlessly by the intercept as shown above.

As well, using standard results from regression analysis, approximate (since the assumption of normality no longer holds) confidence intervals for a and b can easily be obtained. These confidence limits can in turn be used to compute an approximate confidence interval for both α and β . We will see in Section 3.3.1 that a modification of this graphical approach to estimation can provide effective starting values for use with our robust techniques.

2.5 The Usefulness of Quantiles

When simply comparing different sets of fatigue data, estimates of the parameters of Birnbaum-Saunders distributions may be quite acceptable. However when designing an overpass an estimate of the median life-length β may be completely inadequate. Instead, the engineer would be much happier in reporting say a life-length which would enable replacing or inspecting the overpass before a time such that no more than one in a thousand would fail. Statistics of this sort are commonly known as

quantiles and are of great use to engineers for reliability evaluation.

For a number p between zero and one, the p th quantile of a distribution, t_p , is a number such that a fraction p of the distribution lies to the left of t_p and a fraction $1 - p$ of the distribution lies to the right of t_p . The median, for example is $t_{.5}$. For fatigue data, lower quantiles are of great use to design engineers in helping to avoid structural and mechanical failure. These lower quantiles facilitate ensuring that the structure, once complete, will stand up to forecasted stresses at some probability level.

The p th quantile of the Birnbaum-Saunders distribution is the solution t_p of

$$F(t_p; \alpha, \beta) - p = 0$$

where F is the cumulative distribution function of the Birnbaum-Saunders distribution from (5). Chang and Tang [13] determined that

$$\begin{aligned} t_p &= \frac{\beta}{4} \left[\alpha z_p + \sqrt{(\alpha^2 z_p^2 + 4)} \right]^2 \\ t_{1-p} &= \frac{\beta}{4} \left[\alpha z_p - \sqrt{(\alpha^2 z_p^2 + 4)} \right]^2 \end{aligned} \quad (23)$$

where $z_p = \Phi^{-1}(p)$. By the invariance property of the MLE, the above equations give the MLEs of the quantiles when α and β are replaced by their respective MLEs. We define (for use later) the optimal B -robust estimators (OBRE) of the quantiles to be the solutions of the above equations when α and β are instead replaced by their respective OBREs.

As with any other point estimate, it is important to be able to calculate the variance of a quantile. We will let T_p be a random variable corresponding to the p th quantile as given by (23). Now given that T_p is in fact a function of two random variables; $\hat{\alpha}$ and $\hat{\beta}$, we can obtain the variance of T_p by expanding the function for T_p in (23) in a

Taylor series about the estimates $\hat{\alpha}$ and $\hat{\beta}$. This leads to a first-order approximation of the variance of T_p as follows:

$$Var(T_p) \cong c_\alpha^2 Var(\alpha) + c_\beta^2 Var(\beta) + 2c_\alpha c_\beta Cov(\alpha, \beta) \quad (24)$$

where c_α and c_β are the values of the partial derivatives of (23) with respect to α and β , respectively, evaluated at $\hat{\alpha}$ and $\hat{\beta}$. In the case of the MLE it is known that $\hat{\alpha}$ and $\hat{\beta}$ are asymptotically independent [25] so (24) reduces to

$$Var(T_p) \cong c_\alpha^2 Var(\alpha) + c_\beta^2 Var(\beta). \quad (25)$$

We will see that the same cannot be said of the OBRE so we remain with (24) in this case. In either case we arrive at straightforward methods of obtaining the variance of the p th quantile.

2.5.1 Constructing Confidence Intervals

Design engineers often express interest in being able to construct confidence intervals for quantiles. Chang and Tang [13] devised a method for doing so as is given by the following theorem:

Theorem 4 *Suppose that $[\underline{\alpha}, \bar{\alpha}]$ and $[\underline{\beta}, \bar{\beta}]$ are respectively $(1 - \gamma_1)$ and $(1 - \gamma_2)$ level confidence intervals for α and β . Then a $(1 - \gamma_1 - \gamma_2)$ level confidence interval for t_p is given by*

$$(t_p, \bar{t}_p) = \begin{cases} (t_p(\underline{\alpha}, \underline{\beta}), t_p(\bar{\alpha}, \bar{\beta})) & \text{for } p > 0.5 \\ (t_p(\bar{\alpha}, \underline{\beta}), t_p(\underline{\alpha}, \bar{\beta})) & \text{for } p < 0.5 \\ (\underline{\beta}, \bar{\beta}) & \text{for } p = .5. \end{cases} \quad (26)$$

To prove this theorem Tang and Chang asked that we consider the interval

$$[t_p, \bar{t}_p] = \left[\inf_{\Lambda} t_p(\alpha, \beta), \sup_{\Lambda} t_p(\alpha, \beta) \right]$$

where $\Lambda = \{(\alpha, \beta) | \underline{\alpha} \leq \alpha \leq \bar{\alpha}, \underline{\beta} \leq \beta \leq \bar{\beta}\}$. Having previously proven that t_p is a monotone function in both α and β for fixed p [13], the equivalence of the following sets became evident:

$$\{\lambda : \underline{\alpha}(\lambda) \leq \alpha \leq \bar{\alpha}(\lambda), \underline{\beta}(\lambda) \leq \beta \leq \bar{\beta}(\lambda)\}, \quad \{\lambda : \underline{t}_p(\lambda) \leq t_p \leq \bar{t}_p(\lambda)\}$$

from which they stated that it followed that

$$\begin{aligned} P[t_p \in (\underline{t}_p, \bar{t}_p)] &= P[\underline{\alpha} \leq \alpha \leq \bar{\alpha}, \underline{\beta} \leq \beta \leq \bar{\beta}] \\ &\geq 1 - \gamma_1 - \gamma_2 \quad \text{by the Bonferroni inequality.} \end{aligned}$$

$[\underline{t}_p, \bar{t}_p]$ is actually an *outer bound* on the $1 - \gamma_1 - \gamma_2$ confidence interval as a result of applying the Bonferroni inequality. It is of interest to note that the bounds of (26) are independent of the method used to obtain the confidence intervals of the parameters. Tang and Chang evaluated the effectiveness of this method by examining the coverage probability of the confidence intervals and determined that they were quite conservative but nevertheless useful since conservative estimates are always preferred by design engineers.

Evidently, in order to utilize the results of Theorem 4 for the construction of confidence intervals for quantiles we must first construct confidence intervals for α and β . Englehardt, Bain and Wright [25] obtained confidence intervals for these parameters based on the MLE. They arrived at both large and small sample results although their work was quite limited. Both the small and large sample confidence intervals relied on the assumption that $\hat{\alpha}$ and $\hat{\beta}$ were statistically independent although this is

actually only true asymptotically.

When considering fatigue data we are normally not in a position to take full advantage of asymptotics due to the small amounts of data available. Hence, we choose to construct confidence intervals for parameters α and β using a t distribution. The limiting distribution of a t distribution is in fact the normal distribution. By using the t distribution we compensate for having to estimate the variance of the parameters while at the same time remain consistent with the asymptotic distribution of the parameters which is normal for both the MLE and OBRE. Confidence intervals obtained using this approach can still be used directly with Theorem 4 to obtain the desired confidence intervals for the quantiles. We will refer to this approach as *Method A*.

There is still another method (*Method B*) available by which to construct confidence intervals for quantiles of the Birnbaum-Saunders distribution. We first calculate the variance of t_p using (24) or (25) as applicable. Then using a t distribution we can construct the desired confidence intervals for t_p . Notice that Theorem 4 is not required here.

In order to determine which method of constructing confidence intervals is superior a thorough simulation study must be performed but this was beyond the scope of this research. It appears this work has never been done making it a strong contender for future research. Consequently, for our purposes both types of confidence intervals will be reported.

3 The Need for Robust Estimation

“A tacit hope in ignoring deviations from ideal models was that they would not matter; that statistical procedures which were optimal under the strict model would still be approximately optimal under the approximate model. Unfortunately, it turned out that this hope was often drastically wrong; even mild deviations often have much larger effects than were anticipated by most statisticians (J. Tukey, 1960, [46]).”

Estimation is an art as well as a science, whose goal is to infer information about an unknown quantity based on available data. Estimators are selected to perform well under assumed underlying conditions. However, since these conditions are never known exactly, we should choose estimators which are robust, that is, perform well under a variety of underlying conditions. In so doing we obtain estimators which are reasonably accurate and efficient in the presence of outliers and other departures from the assumptions.

Outliers and other deviations from underlying model assumptions are unfortunately a fact of life. Common examples are recording and definition errors as well as mishandling in the case of fatigue data. These deviations can have drastic effects on the MLE of the parameters of a Birnbaum-Saunders distribution. Such estimators become biased and inefficient when the model does not hold exactly.

In a wide range of research areas a great deal of progress has been made in finding procedures which are robust. In fact, the robustification of many statistical procedures is one of the major developments in statistics this century. Our goal is to show the usefulness of robust techniques for modeling life-length data following a Birnbaum-Saunders distribution.

We often refer to outliers as sources of contamination. It is important to emphasize that the procedure is not necessarily to identify outliers and immediately rid ourselves of this contamination as is often first thought. Rather, in identifying potential outliers we may obtain very useful information regarding the data, aiding engineers, for

instance, in making decisions impacting cost, materials, and manpower in the context of the Birnbaum-Saunders family. The ability to quickly identify observations of this sort is one of the great benefits of the robust estimation procedure we develop here.

3.1 Robustness Concepts

Our goal is to develop robust estimates of the parameters of a Birnbaum-Saunders distribution when such a model is assumed, at least at the onset, to be suitable. The Birnbaum-Saunders cumulative distribution function involves a 2-dimensional parameter $\theta = [\alpha, \beta]^T$. To summarize we have (from (5)):

$$F(t; \alpha, \beta) = \Phi[\alpha^{-1}\xi(t/\beta)] \quad \text{for } t > 0,$$

where

$$\alpha > 0, \quad \beta > 0,$$

and

$$\xi(t) = t^{1/2} - t^{-1/2}.$$

α and β are the shape and scale parameters, respectively.

In order to evaluate the robustness of any estimator we can appeal to the idea of influence functions originated by Hampel [32, 33] and developed later by Hampel *et al.* [34]. Let F_n represent the empirical distribution of the data putting mass $1/n$ on each observation. As estimators of θ we consider statistics $T_n = T_n(t_1, \dots, t_n)$ which can be represented, at least asymptotically, as functionals of the empirical distribution, i.e. $T_n(t_1, \dots, t_n) = T(F_n)$. The robustness of the estimator T is assessed by means

of the influence function (IF), which is defined at the model F_θ by

$$IF(t; T, F_\theta) = \lim_{\varepsilon \rightarrow 0} \left[\frac{T((1 - \varepsilon)F_\theta + \varepsilon\Delta_t) - T(F_\theta)}{\varepsilon} \right]. \quad (27)$$

Δ_t is a probability measure which puts all its mass in t . Notice that $(1 - \varepsilon)F_\theta + \varepsilon\Delta_t$ yields *good* observations from F_θ with high probability $1 - \varepsilon$ and *bad* observations from point t with small probability ε . The IF describes the effect of this small contamination ($\varepsilon\Delta_t$) at the point t on the estimate (standardized by the mass of the contamination). In fact, the linear approximation $\varepsilon IF(t; T, F_\theta)$ measures the asymptotic bias of the estimator caused by the contamination.

One of the great strengths of influence functions rests with their ability to expose the limitations of many commonly used estimators in the presence of contamination. Consequently we use influence functions to reveal the deficiencies of the well known MLE of the parameters of the Birnbaum-Saunders distribution.

The IF of a maximum likelihood function is proportional to its score function. A score function is the derivative of the natural logarithm of the density with respect to the distributional parameters. In the case of the Birnbaum-Saunders distribution the density (f_θ) from (1) is:

$$\begin{aligned} f_\theta(t) = f(t; \alpha, \beta) &= \frac{1}{\alpha\beta} \Phi'[\alpha^{-1}\xi(t/\beta)]\xi'(t/\beta) \\ &= \frac{1}{2\alpha\beta} \left(\frac{t}{\beta}\right)^{-1/2} \left[1 + \left(\frac{t}{\beta}\right)^{-1}\right] \\ &\quad \times \frac{1}{\sqrt{(2\pi)}} \exp\left[-\frac{1}{2\alpha^2} \left(\frac{t}{\beta} - 2 + \frac{\beta}{t}\right)\right], \quad t > 0; \alpha, \beta > 0 \end{aligned}$$

and the corresponding score function ($s(t; \theta) = (\partial/\partial\theta)\ln f_\theta(t)$) is:

$$s(t; \theta) \approx s(t; \alpha, \beta) = \begin{bmatrix} \frac{-\alpha^2\beta t + t^2 - 2\beta t + \beta^2}{t\alpha^3\beta} \\ \frac{1}{2} \frac{-\beta\alpha^2 t^2 + \beta^2\alpha^2 t + t^3 - \beta^2 t + \beta t^2 - \beta^3}{\beta^2\alpha^2 t(t+\beta)} \end{bmatrix}. \quad (28)$$

One can see that the MLE of the Birnbaum-Saunders distribution has an unbounded score function (28) in t of order one and hence an unbounded IF. This unboundedness permits small deviations from the model distribution to cause large changes in the MLE. Such a characteristic is most detrimental as it can lead to badly biased results.

Alternatively, estimators exhibiting the desirable robustness property of a bounded IF are more attractive. Such estimators are called *B-robust* (Bias-robust). By bounding the IF, we can ensure that small deviations from the model distribution do not cause large changes in the estimates.

3.2 The Optimal B-Robust Estimator

When searching for robust estimators it is of course our desire to find the best estimator possible. We want these estimators to be efficient as well as insensitive to small and large deviations in the data. As is often the case in real life, these goals cannot normally be reached simultaneously. Instead we must often compromise by sacrificing some degree of efficiency to increase robustness or vice versa.

Optimal robust estimators in effect provide us with a compromise. An *optimal B-robust* estimator is constructed to have maximum efficiency subject to an upper bound on the IF. This bound restricts the influence a small amount of contamination can have on the value of the estimator. That is, we sacrifice some degree of efficiency to ensure that a small amount of contamination cannot severely distort the value of our

estimators.

The OBRE is an M -estimator which is a generalization of a MLE and the solution T_n of the (system) of equation(s)

$$\sum_{i=1}^n \Psi(t_i; T_n) = 0$$

for some function $\Psi : T \times \mathbb{R}^p \rightarrow \mathbb{R}^p$. Here p represents the number of parameters in the model. The IF of an M -estimator Ψ at F_θ is given by

$$IF(t; \Psi, F_\theta) = M(\Psi, F_\theta)^{-1} \Psi(t, \theta)$$

where

$$M(\Psi, F_\theta) = - \int \frac{\partial}{\partial \theta} \Psi(t, \theta) dF_\theta(t). \quad (29)$$

The OBRE is optimal in the sense that it is the M -estimator which minimizes the trace of the asymptotic covariance matrix under the constraint that it has a bounded IF. Minimizing the trace of a covariance matrix is equivalent to maximizing efficiency.

There are several versions of the OBRE [34] each different in the way they choose to bound the influence function. Here, we use the standardized OBRE as this is currently the only one we are aware of for which an implementation algorithm exists. For a given bound c on the IF ($c \geq p^{1/2}$), estimators are defined implicitly by

$$\sum_{i=1}^n \Psi(t_i; \theta) = \sum_{i=1}^n \{s(t_i; \theta) - a(\theta)\} W_c(t_i; \theta) \quad (30)$$

where $s(t; \theta)$ is as in (28) and $W_c(t; \theta)$ is the weight function

$$W_c(t; \theta) = \min\left\{1; \frac{c}{\|A(\theta)\{s(t, \theta) - a(\theta)\}\|}\right\} \quad (31)$$

where $\|\cdot\|$ denotes the Euclidean norm. The $p \times p$ matrix $A(\theta)$ and the $p \times 1$ vector $a(\theta)$ are defined implicitly (through (30)) by

$$E\{\Psi(t; \theta)\Psi(t; \theta)^T\} = \{A(\theta)^T A(\theta)\}^{-1}, \quad (32)$$

$$E\Psi(t; \theta) = 0. \quad (33)$$

Note that the weight function W_c multiplies the score function $s(t_i, \theta)$ after it has been made Fisher consistent by subtracting $a(\theta)$. The weights will always be less than or equal to one and will be less than one when the norm of the appropriately standardized score function exceeds the cutoff c . This downweights observations which are not close to the working values of the parameters in this score function metric. For observations which are consistent with the current values of θ , the weights are one and the score function is that of the MLE. This ensures that the OBRE is efficient since it is as similar as possible to the maximum likelihood estimator for the majority of the data. The weight function gives us precisely the information needed to determine whether our model is consistent with the data. If a potentially informative observation is downweighted, we know that inferences based on the Birnbaum-Saunders distribution are likely to be flawed. Our options are to base the inferences on the points which are well fitted by the model and downweight the observations(s) or try to develop alternative models in which all the weights are close to one.

3.3 The Algorithm

We now give the details of the algorithm used for estimating the parameters of the Birnbaum-Saunders distribution via the OBRE. Victoria-Feser and Ronchetti presented the original algorithm in the context of personal income models [53] and we have altered it as necessary to apply it to the Birnbaum-Saunders distribution. The basic idea is to start with the score function $s(t; \theta)$ and to modify it so that the in-

fluence function is bounded and so that it is Fisher consistent, i.e. satisfies (32). The details for an observed sample $t_1, \dots, t_n$ are as follows:

- Step 1. Fix an initial value for the parameter $\theta = [\alpha, \beta]^T$. That is, compute initial estimates (starting values) of the parameters α and β .
- Step 2. Fix a precision tolerance η , the robustness constant c , $\mathbf{a} = \mathbf{0}$ and $\mathbf{A} = (J^{1/2}(\theta))^{-T}$ where

$$J(\theta) = \int \mathbf{s}(t, \theta), \mathbf{s}(t, \theta)^T dF_\theta(t)$$

is the Fisher information matrix.

- Step 3. Solve the following equations for $\mathbf{a}$ and $\mathbf{A}$:

$$\mathbf{A}^T \mathbf{A} = \mathbf{M}_2^{-1}$$

where

$$\mathbf{a} = \frac{\int \mathbf{s}(t, \theta) W_c(t, \theta) dF_\theta(t)}{\int W_c(t, \theta) dF_\theta(t)},$$

and

$$\mathbf{M}_k = \int \{\mathbf{s}(t, \theta) - \mathbf{a}\} \{\mathbf{s}(t, \theta) - \mathbf{a}\}^T W_c^k(t; \theta) dF_\theta(t) \quad k=1,2. \quad (34)$$

The current values of θ , $\mathbf{a}$ and $\mathbf{A}$ are used as starting values to solve the above equations.

- Step 4. Compute $\mathbf{M}_1$ using $\mathbf{a}$ and $\mathbf{A}$ from Step 3 and

$$\Delta\theta = \mathbf{M}_1^{-1} \left[\frac{1}{n} \sum_{i=1}^n \{\mathbf{s}(t_i, \theta) - \mathbf{a}\} W_c(t_i, \theta) \right].$$

- Step 5. If $\max_j |\frac{\Delta \theta_j}{\Delta_j}| > \eta$ then $\theta \rightarrow \theta + \Delta \theta$ and return to Step 3, else stop.

Note that integration can be avoided in Step 2 and in the computation of M_k (Steps 3 and 4) by replacing F_θ with its empirical distribution function. However numerical integration must be used to compute a otherwise (30) will be satisfied by all estimates. The algorithm is convergent provided the starting value of θ is near the solution.

The matrix $A(\theta)$ and the vector $a(\theta)$ can be viewed as Lagrange multipliers for the constraints resulting from a bounded influence function and Fisher consistency. The constant c is the bound on the influence function and acts as a regulator between robustness and efficiency. That is, for small c , one gains robustness but loses efficiency, and for large c , the opposite is true. The most robust estimator is achieved at $c = p^{1/2}$ where p is the number of parameters in the model and the MLE (most efficient) is obtained at $c = \infty$. In practice, c is often chosen to achieve 95% efficiency at the model.

3.3.1 An Effective Starting Value

We must now examine one of the more intricate details of the OBRE algorithm as modified for the Birnbaum-Saunders distribution. Step 1 requires starting values of the parameters α and β . The algorithm is convergent provided these starting values are near the solution. Having already established that the MLE performs poorly in the presence of contamination, one could argue that the MLE may not be robust enough for use as a starting value.

As mentioned earlier in Section 2.4, Chang and Tang proposed using LS regression to estimate the parameters of a Birnbaum-Saunders distribution. We suggest a viable alternative with which to obtain starting values for Step 1.

For many years the LS method was the regression technique of choice due to both tradition and ease of computation. However, recently there has been increasing aware-

ness of the dangers to the LS method posed by outlying observations and the need for a regression technique which effectively deals with them. This approach is referred to as *robust regression* and tries to devise estimators that are not so strongly affected by outliers. Many people think that robust regression techniques hide or ignore outliers. On the contrary, robust regression procedures expose these outliers via the large residuals (the actual value of the data point minus the corresponding fitted value) assigned. In fact all the outliers can be identified in a single step simply by running a robust regression and examining the residuals or a plot of the data and the corresponding robust fit.

A first step toward a more robust regression estimator came from Edgeworth [24] improving on a suggestion from Boscovich [24]. He argued that outliers have a very large influence on LS because the residuals r_i are squared in

$$\text{Minimize}_{\hat{\theta}} \sum_{i=1}^n r_i^2 \quad [\text{LS criterion}]. \quad (35)$$

Therefore he proposed the *least absolute values regression estimator*, determined by

$$\text{Minimize}_{\hat{\theta}} \sum_{i=1}^n |r_i|. \quad (36)$$

This is often referred to as L_1 regression. It can be shown that this method, unlike LS is robust to outliers in the y -direction, but unfortunately still does not protect against outlying x .

Various other estimators were also considered including M -estimators (as defined in Section 3.2), R -estimators and L -estimators. R -estimators are based on rank tests whereas L -estimators take the form of linear combinations of ordered sample values. Unfortunately, none performed well when contaminants made up more than 30% of the data. This is not a large concern for fatigue data since it is unlikely that the amount of contamination is anywhere near this size. If there is a large amount

of contamination in a fatigue data set it is more likely that this part of the data follows a different life-length distribution and should be handled separately or a new distribution found. Nevertheless, at this point it was decided that another look should be taken at the LS approach. Rousseeuw [46] replaced the sum in (35) by a median which can be shown to be very robust [34]. This yields the *least median of squares* (LMS) estimator, given by

$$\text{Minimize } (\text{med}_i r_i^2) \quad (37)$$

The basic principle of LMS is to fit the majority of the data, after which outliers are identified as those points that lie far away from the robust fit. It turns out that this estimator is very robust with respect to outliers in the dependent and independent variable. In the context of the Birnbaum-Saunders distribution both these variables are actually functions of the observed failure times making robustness with respect to outliers in both these variables quite desirable. LMS performs well with the maximum amount of contamination present (50%). Its limitation is its poor asymptotic efficiency. This is not of great concern in our application to the Birnbaum-Saunders distribution since the estimate is only being used as a starting value so there is some room for variance. The other limitation is the computation time of LMS which is notably higher than that of ordinary LS. However with computer power increasing rapidly and the relatively small size of fatigue data sets, this restriction is becoming less and less of an issue. LMS regression is therefore our method of choice.

Questions may be raised regarding the necessity of going further to compute the OBRE given the strong robustness properties of simply using LMS estimates of the parameters. It is important at this point to reemphasize some of the more important aspects of the OBRE. First, unlike LMS, the OBRE has the ability to assign insightful weights to the observations of concern. Secondly, unlike LMS, the OBRE enables calculation of the asymptotic covariance matrix of the estimates from which we can

obtain confidence intervals and efficiencies. We conclude LMS regression appears to be a powerful tool with which to compute starting values but much more can be learned by subsequently applying the OBRE.

Geometrically the LMS regression method corresponds to finding the narrowest strip covering half of the observations. The LMS line lies exactly at the middle of this band. We have designed programs to implement this method which are listed in Appendix C. The algorithm proceeds in the following manner:

1. Subsamples of 2 different observations are repeatedly drawn from the data set of concern.
2. For each subsample indexed by J the regression line through the 2 points is determined and the corresponding coefficients are denoted by $(\hat{\alpha}_J, \hat{\beta}_J)$.
3. For each $(\hat{\alpha}_J, \hat{\beta}_J)$ the corresponding median squared function value with respect to the whole data set is determined. This means that the value

$$\text{med}_{i=1, \dots, n} \left(t_i - (\hat{\beta}_J + \hat{\alpha}_J \sqrt{\hat{\beta}_J} \hat{p}_i) \right)^2$$

is calculated.

4. Finally, the trial estimate for which this value is minimal is retained.

We are left with the problem of deciding how many subsamples should be considered. In principle, the above procedure should be repeated for all possible subsamples of size 2, of which there will be C_2^n . However, as n increases this number (C_2^n) increases rapidly and in many applications becomes difficult to handle. Instead, in cases where n is large we will perform only a certain number of random selections. We still wish to perform enough random selections such that there is high probability that at least one of the subsamples consists of two good observations of the sample. For our purposes we have chosen to run all subsamples when the number of data points is less than

or equal to 25. After this we take 300 randomly selected pairs to be the number of subsamples [46].

It can be shown that the LMS method significantly lessens the impact of outliers. Hence our approach is to transform and parametrize the data to provide a pair of variables with a linear relationship as Chang and Tang [12] suggested. However, instead of estimating the slope and intercept of the fitted straight line via LS, we use LMS instead.

This greatly improves results in the presence of contamination since a few outliers can throw LS estimates way off and result in a poor starting value for the OBRE algorithm and in turn a small probability of convergence. The LMS estimates remain consistent with the majority of the data helping to ensure convergence of the OBRE algorithm. As well, when no outliers are present this procedure yields results which hardly differ from the LS solution.

It is also of interest to note that Chang and Tang suggest using the median rank (see (22)) when calculating $\hat{F}(t_i)$ for various failure times t_i as part of their graphical approach. This is a wise decision due to the good robustness properties associated with this estimator. We therefore follow their suggestion.

3.3.2 Limitations of the Mean Mean Estimate

We have substantiated using LMS instead of LS as part of Chang and Tang's graphical approach especially when considering fatigue data. However, we must still justify choosing the LMS estimate over the mean mean estimate [7], as suggested by Birnbaum and Saunders, as a starting value for the OBRE algorithm. We begin by appealing to the very useful concept of an empirical influence function to help illustrate the limitations of the mean mean estimate and then provide an analytical justification for the unboundedness of the influence function of the mean mean estimate.

Consider the following data set of final marks for a fourth year applied probability and statistics course offered at the Technical University of Nova Scotia:

36, 51, 53, 57, 60, 60, 67, 68, 70, 71, 72, 80, 90

There are a multitude of different location estimators we could apply to this data set, but for demonstration purposes we shall concern ourselves with the sample mean, sample median, and mean mean.

The value 90 might be considered an outlier. One way in which to compare these estimators is to pose the question "what happens to the estimate as we let the largest data value 90 move from $-\infty$ to $+\infty$?". If 90 is an outlier we are seeing what the estimates might be for other values of this observation. This forms *empirical influence functions* (EIFs) for this data set and each particular estimator. These EIFs are displayed in Figure 5. One can see that the mean and the mean mean estimate behave differently from the median. They have empirical influence functions that are unbounded. Hence we say the mean and mean mean are not robust unlike the median which bounds the influence the outlier can have. One can go further to show that as the sample size grows, the empirical influence function of each estimate approaches it's theoretical influence function.

In order to prove analytically that the mean mean estimate $\tilde{\beta}$ has an unbounded influence function we begin by writing (from (17)):

$$\tilde{\beta} = (SR)^{1/2} = \left(\frac{S}{H}\right)^{1/2} \quad (38)$$

where S and R are random variables representing the arithmetic and harmonic means, respectively, and H is a random variable representing the reciprocal of the harmonic mean. This change of variable simplifies the calculations which follow. We must consider the limiting behavior of both S and H to evaluate the behavior of the IF of

Figure 5: The EIF of a selection of estimators. The largest data value varies from $-\infty$ to $+\infty$.

$\bar{\beta}$.

We first consider the arithmetic mean S at the Birnbaum-Saunders model $F(t; \alpha, \beta)$ (from (5)). Here, $S_n = \frac{1}{n} \sum_{i=1}^n t_i$, and the corresponding functional is $S(F(t; \alpha, \beta)) = \int t dF(t; \alpha, \beta)$. Using the limit definition of the IF [34] as first introduced in Section 3.1, we can then write:

$$IF(t; S, F(t; \alpha, \beta)) = \lim_{\epsilon \rightarrow 0} \frac{S[(1 - \epsilon)F(t; \alpha, \beta) + \epsilon \Delta_t] - S(F(t; \alpha, \beta))}{\epsilon}. \quad (39)$$

where Δ_t is a probability measure which puts all its mass in t . (Note: $\int u d\Delta_t = t$). It is also true from (10) that $S(F(t; \alpha, \beta)) = \int t dF(t; \alpha, \beta) = E[T] = \beta (1 + \frac{1}{2}\alpha^2)$ so (39) becomes:

$$\begin{aligned} IF(t; S, F) &= \lim_{\epsilon \rightarrow 0} \frac{\int t d[(1 - \epsilon)F(t; \alpha, \beta) + \epsilon \Delta_t] - \beta (1 + \frac{1}{2}\alpha^2)}{\epsilon} \\ &= \lim_{\epsilon \rightarrow 0} \frac{(1 - \epsilon) \int t dF(t; \alpha, \beta) + \epsilon \int t d\Delta_t - \beta (1 + \frac{1}{2}\alpha^2)}{\epsilon} \\ &= \lim_{\epsilon \rightarrow 0} \frac{(1 - \epsilon)\beta (1 + \frac{1}{2}\alpha^2) + \epsilon t - \beta (1 + \frac{1}{2}\alpha^2)}{\epsilon} \\ &= \lim_{\epsilon \rightarrow 0} \frac{\epsilon t - \epsilon \beta (1 + \frac{1}{2}\alpha^2)}{\epsilon} \\ &= t - \beta \left(1 + \frac{1}{2}\alpha^2\right) \end{aligned}$$

which is clearly unbounded in t .

We now consider the reciprocal of the harmonic mean, denoted by H , at the Birnbaum-Saunders model. Here, $H_n = \frac{1}{n} \sum_{i=1}^n \frac{1}{t_i}$, and the corresponding functional is $H(F(t; \alpha, \beta)) = \int \frac{1}{t} dF(t; \alpha, \beta^{-1})$.

Using the limit definition of the IF [34] we can write:

$$IF(t; H, F(t; \alpha, \beta)) = \lim_{\epsilon \rightarrow 0} \frac{H[(1 - \epsilon)F(t; \alpha, \beta) + \epsilon \Delta_t] - H(F(t; \alpha, \beta))}{\epsilon} \quad (40)$$

Note that $\int \frac{1}{t} d\Delta_t = 1/t$. From Theorem 2 we have that if T is distributed as $F(t; \alpha, \beta)$ then $1/T$ must be distributed as $F(t; \alpha, \beta^{-1})$. Hence $H(F(t; \alpha, \beta)) = \int \frac{1}{t} dF(t; \alpha, \beta^{-1}) = E(1/T) = \frac{1}{\beta} (1 + \frac{1}{2}\alpha^2)$ (by (10)) so (40) becomes:

$$\begin{aligned}
 IF(t; H, F(t; \alpha, \beta)) &= \lim_{\epsilon \rightarrow 0} \frac{\int \frac{1}{t} d[(1 - \epsilon)F(t; \alpha, \beta) + \epsilon\Delta_t] - \frac{1}{\beta} (1 + \frac{1}{2}\alpha^2)}{\epsilon} \\
 &= \lim_{\epsilon \rightarrow 0} \frac{(1 - \epsilon) \int \frac{1}{t} dF(t; \alpha, \beta) + \epsilon \int \frac{1}{t} d\Delta_t - \frac{1}{\beta} (1 + \frac{1}{2}\alpha^2)}{\epsilon} \\
 &= \lim_{\epsilon \rightarrow 0} \frac{(1 - \epsilon) \int y dF(y; \alpha, \beta^{-1}) + \epsilon \int \frac{1}{t} d\Delta_t - \frac{1}{\beta} (1 + \frac{1}{2}\alpha^2)}{\epsilon} \\
 &= \lim_{\epsilon \rightarrow 0} \frac{(1 - \epsilon) \frac{1}{\beta} (1 + \frac{1}{2}\alpha^2) + \epsilon \frac{1}{t} - \frac{1}{\beta} (1 + \frac{1}{2}\alpha^2)}{\epsilon} \\
 &= \frac{1}{t} - \frac{1}{\beta} \left(1 + \frac{1}{2}\alpha^2\right)
 \end{aligned}$$

The above expression makes it evident that as $t \rightarrow \infty$, $IF(t; H, F(t; \alpha, \beta)) \rightarrow -\frac{1}{\beta} (1 + \frac{1}{2}\alpha^2)$. Returning now to (38) we see that as $t \rightarrow \infty$, H is bounded by $\rightarrow -\frac{1}{\beta} (1 + \frac{1}{2}\alpha^2)$ unlike S which possesses an unbounded influence function and therefore approaches ∞ . Consequently the influence function for (38) must also be unbounded.

The mean mean estimate, in having an unbounded influence function, is quite undesirable in the presence of contamination. This can deter convergence of the OBRE algorithm making the LMS estimate a much better choice.

3.4 The Weighting Function

We will see later that in comparing the OBRE with the MLE that we have a procedure which gives efficient estimates if the model is true but has the major advantage of providing a weight for each observation which measures the concordance of the observation with the estimated model.

In order to properly evaluate the weights assigned to a data set by the OBRE, it

Table 1: Mean (standard deviation) of weights for the OBRE ($c = 4$) for Birnbaum-Saunders data.

Rank of Weight											
α	1	2	3	4	5	...	96	97	98	99	100
0.2	.80 (.18)	.96 (.09)	1.00 (.02)	1.00 (.00)	1.00 (.00)		1.00 (.00)	1.00 (.00)	1.00 (.02)	.96 (.09)	.81 (.18)
0.6	.77 (.19)	.94 (.10)	.99 (.03)	1.00 (.01)	1.00 (.00)		1.00 (.00)	1.00 (.00)	.99 (.03)	.94 (.10)	.78 (.18)
1.0	.73 (.19)	.92 (.12)	.98 (.04)	1.00 (.01)	1.00 (.00)		1.00 (.00)	1.00 (.01)	.99 (.04)	.92 (.12)	.74 (.19)
1.4	.69 (.20)	.88 (.14)	.97 (.06)	1.00 (.02)	1.00 (.00)		1.00 (.00)	1.00 (.01)	.98 (.06)	.89 (.14)	.70 (.19)
1.8	.65 (.20)	.85 (.15)	.96 (.08)	1.00 (.03)	1.00 (.00)		1.00 (.00)	1.00 (.02)	.96 (.08)	.86 (.15)	.66 (.19)

is important to recognize that the OBRE may in fact downweight points in data generated from a Birnbaum-Saunders distribution. In order to better understand the behavior of the OBRE for Birnbaum-Saunders data we performed a simulation study. We chose samples of size 100 and generated 500 samples at each of five α values. The shape parameter was varied from 0.2 to 1.8 and the scale parameter β was set to one. Table 1 displays the results. We report the mean weight over the 500 simulations for each of the ordered observations omitting those for observations 5 through 95 which are all one. When the random number generator seed was changed (yielding 500 new random samples) and the computations performed again, no perceptible change was seen in the weights giving us confidence in the accuracy and precision of the results in Table 1. Typically the smallest 4 and largest 4 observations are downweighted with substantial downweighting occurring on at most the smallest 2 and the largest 2 observations. This provides us with a calibration for determining when the observed downweighting indicates a serious lack of fit.

3.5 The Variance of the OBRE

It is important to be able to report more than just the OBRE of the parameters of a Birnbaum-Saunders distribution. Additionally, we would like to provide variances for these estimates from which we can create confidence intervals when accompanied by a distributional assumption regarding these estimates. A routine method for computing variances is to use the asymptotic distribution of the parameters which fortunately is known in the case of the OBRE.

The OBRE can be shown to be asymptotically normally distributed provided the Fisher consistent score function Ψ is bounded and continuous. Exact regularity conditions are given in [35]. The asymptotic covariance matrix is given by

$$V(\Psi, F_\theta) = M(\Psi, F_\theta)^{-1} Q(\Psi, F_\theta) M(\Psi, F_\theta)^{-T}, \quad (41)$$

where $M(\Psi, F_\theta)$ is as in (29) and

$$Q(\Psi, F_\theta) = \int \Psi(t, \theta) \Psi(t, \theta)^T dF_\theta(t). \quad (42)$$

Note that since we have a *model* with scores, Fisher consistency implies

$$\int \Psi(t, \theta) dF_\theta(t) = 0$$

and hence we can compute $M(\Psi, F_\theta)$ as follows:

$$M(\Psi, F_\theta) = \int \Psi(t, \theta) s(t, \theta)^T dF_\theta(t)$$

The exact asymptotic covariance matrix is very difficult to compute analytically due to the implicit definition of many terms. However, estimates are obtained directly by replacing F_θ with its empirical distribution. We can obtain the variance from the

appropriate diagonal element of V from (41). We have to replace the unknown θ by our estimated $\hat{\theta}$. The integrals cannot be evaluated explicitly so we can use $M(\Psi, F_n)$ and $Q(\Psi, F_n)$, where F_n is the empirical distribution, to approximate $M(\Psi, F_\theta)$ and $Q(\Psi, F_\theta)$, respectively.

An important issue with such asymptotic results is to understand their performance in finite samples so we can have confidence in their accuracy for all sample sizes above a certain bound. The practical question is to decide whether for the situation of interest, *large* means a sample size of 50 or a sample size of 500. In realistic life-length distribution problems, it seldom occurs that the available data set is large enough to ensure that asymptotic large-sample theory may be directly applied. However Dupuis and Field [22] demonstrated that asymptotic results work quite well for moderately sized samples in the context of generalized extreme-value distributions. This suggests the possibility of obtaining similar results in the context of Birnbaum-Saunders distributions.

To help investigate the approach to normality in the case of Birnbaum-Saunders distributions we appeal to normal probability plots which are widely used and easily interpreted. In order to do so we ran 100 simulations at each of three sample sizes: 25, 50 and 100. For each simulation we calculated the OBRE and then constructed normal probability plots for both $\hat{\alpha}_{OBRE}$ and $\hat{\beta}_{OBRE}$. These plots are shown in Figures 6 and 7. One can see that as the sample size n increases the plots tend closer to a straight line pattern. However, even with $n = 25$ we get a relatively straight line. The closer a normal probability plot is to a straight line, the closer to normality. Hence we conclude that the OBRE move closer to being normally distributed with increasing sample size as expected. While estimates are asymptotically normally distributed, the approach to normality seems to occur at a relatively small sample size. Thus, we can be quite confident in its use for inference.

(a) $n=25$. (b) $n=50$. (c) $n=100$.

Figure 6: Normal probability plots of $\hat{\alpha}_{OBRE}$ for sample sizes of size $n = 25, 50$ and 100 , $\alpha = .2$, $\beta = 1.0$.

(a) $n=25$. (b) $n=50$. (c) $n=100$.

Figure 7: Normal probability plots of $\hat{\beta}_{OBRE}$ for sample sizes of size $n = 25, 50$ and 100 , $\alpha = .2$, $\beta = 1.0$.

4 Applications to Real Data

In order to illustrate the effectiveness of the OBRE for Birnbaum-Saunders distributions we apply the techniques presented in the previous sections to some real data sets from a variety of different origins. All data sets have been ordered from shortest to longest lifetime.

4.1 Aluminum Fatigue Testing

A special testing machine was designed and constructed by members of the Instrument Development Unit of the Boeing Airplane Company. This machine was used to obtain data suitable for testing the appropriateness of the Birnbaum-Saunders distribution. It simultaneously subjected six rectangular strips of material of identical size to repeated alternating stresses. Each of the six strips was clamped at the ends and deflected in the center to either side from the position of equilibrium, with constant frequency and with amplitudes which could be either constant (case of periodic loading) or random with a determined distribution. Elaborate precautions were taken to avoid clamping stresses and other extraneous damaging effects.

Two data sets were obtained, each for periodic loading with a frequency of 18 cycles per second, with maximum stress of 31,000 psi and 21,000 psi, respectively. These data sets are provided in Table 14 of Appendix B. The material used was 6061-T6 aluminum sheeting, cut parallel to the direction of rolling. For each of the maximum stresses, 17 sets of six strips were placed in the testing machine and stressed until all of them failed, thus yielding 102 observations. In both data sets one observation was lost due to extraneous reasons which were never reported. Hence in our analysis both data sets actually consist of 101 observations. The machine was equipped with a servomechanism adjusting for changes of amplitude which occur after a strip fails, and with apparatus recording the number of cycles to failure for each strip. Statistical

multisample procedures were used for each of the data sets to test the hypothesis that the six strip loading conditions in the machine were equivalent; in each case this hypothesis was comfortably accepted [7].

4.1.1 Data Set 1

For the first data set corresponding to a maximum stress per cycle of 31,000 psi, we begin by computing the MLE of the distributional parameters. We stop the iterations when the successive estimators of β agree within $\epsilon = .001$. Standard deviations are computed following the suggestions of Englehardt, Bain and Wright [25] as presented in Section 2.3. Numerical integration is required in order to calculate $I(\alpha)$ (19). The estimates and their standard deviations (in parentheses) are:

$$\begin{aligned}\hat{\alpha}_{MLE} &= .17 \text{ (.01)} \\ \hat{\beta}_{MLE} &= 131.82 \text{ (2.23)}.\end{aligned}$$

Upon computing the OBRE ($c = 4.0$) with $\eta = .05$, which would yield approximately 95% efficiency at the model (our simulation study will substantiate this), we obtain the estimates (standard deviations computed as detailed in Section 3.2 are in parentheses):

$$\begin{aligned}\hat{\alpha}_{OBRE} &= .15 \text{ (.01)} \\ \hat{\beta}_{OBRE} &= 133.00 \text{ (1.89)}.\end{aligned}$$

All weights are equal to one with the exception of the two smallest and largest observations which are assigned weights (each observation's number corresponds to its

place in Table 14):

Observation 1:	<i>weight</i> = .32
Observation 2:	<i>weight</i> = .92
Observation 100:	<i>weight</i> = .75
Observation 101:	<i>weight</i> = .53

The OBRE appears to have significantly downweighted the smallest and two largest observations suggesting that these observations are perhaps not well fit by a Birnbaum-Saunders distribution. This downweighting leads to a smaller standard deviation for $\hat{\beta}_{OBRE}$ than $\hat{\beta}_{MLE}$.

Contrary to the MLE, the OBRE seeks to model the majority of the data, i.e. it fully weights observations which are consistent with a Birnbaum-Saunders distribution. The MLE tends to be influenced by the extreme observations in the tails at the expense of the data in the center of the distribution, producing a poorer fit as suggested by Figure 8. Although Figure 8 depicts the distributions in their most familiar form, quantile plots can provide additional insight into what is happening in the tails of the distributions. In Figure 9 we plot the quantiles obtained using the MLE versus those of the OBRE. One can see that the OBRE has *tightened* the distribution by downweighting observations in either tail. That is, the OBRE yields larger lower quantiles and smaller higher quantiles than that of the MLE. It is also instructive to examine how the distributions perform strictly with regard to the data. Quantile plots of this sort are shown in Figure 10. Evidently the OBRE models the bulk of the data well showing discrepancies from a straight line pattern only in the tails. The MLE does not appear to follow quite as straight a line and shows more discrepancies in the upper tail as a result of inappropriately trying to fit a model to *all* of the data. In both the plots of Figure 10 a straight line with slope one passing through the origin is drawn to illustrate the pattern the quantiles should follow in

Figure 8: The MLE and the OBRE of the Birnbaum-Saunders distribution for Data Set 1.

theory.

In many engineering applications the extreme observations may be the ones of greatest interest. One could argue that the OBRE identifies potential problems with the extremes. The engineer is left to decide if these observations should remain part of the data set. If it is determined that these observations are special cases to be handled separately, the OBRE estimates are quite appropriate having downweighted them accordingly. However, if these observations are deemed necessary the OBRE are probably not choice estimates to use.

The question remains: what should we do if these extreme observations are a valid part of the data set? One may be tempted to proceed using the MLE, however since the OBRE has downweighted certain observations, this is indicative of an unsuitable model. The OBREs ability to enlighten us in this regard is one of its many strengths.

Figure 9: Quantile-quantile plot of the MLE versus the OBRE of the Birnbaum-Saunders distribution for Data Set 1.

Perhaps a safer alternative then, is to investigate other possible models which are capable of modeling *all* of the data well. Unfortunately, finding other more appropriate models is by no means an easy task.

Let us examine the repercussions of some of these decisions in light of the current application. Were we to use the OBRE resulting in a Birnbaum-Saunders distribution fitting the bulk of the data, we would estimate the median life-length β of the aluminum sheeting at approximately 133,000 cycles. However, in inappropriately choosing the MLE we would estimate the median life-length at only 131,000 cycles. Implicit here is the fact that this later estimate is obtained from a poorer fitting model.

The usual objective in designing structures and structural elements to resist repeated stress is to make certain that repeated loads do not cause failure during the contemplated useful life of the structure. This objective may have greater or lesser importance, depending on whether failure could result in a catastrophe or could mean only a minor repair operation. Considerations of this sort along with those of cost,

(a) MLE Quantile plot.

(b) OBRE Quantile plot.

Figure 10: Quantile plots of the Birnbaum-Saunders distribution for Data Set 1.

Table 2: $p = .001$ quantiles for Data Set 1.

	MLE	OBRE
$t_{.001}$	78.32	83.63
Method A	(77.41,79.19)	(82.73,84.53)
Method B	(77.71,78.93)	(83.24,84.01)

inspection, etc. will greatly influence estimate selection.

Using available data and the various methods of estimating and model fitting, it is usually possible to make a reasonable preliminary design on almost any structure. Design engineers often choose to work with lower quantiles which are functions of the distributional parameters as detailed in Section 2.5.

The $p = .001$ quantiles and their associated 95% confidence intervals are given in Table 2. One can see that the confidence intervals corresponding to Method A are longer than those of Method B (Methods A and B were defined in Section 2.5.1). This is consistent with Chang and Tang's [13] suggestion that their confidence intervals are quite conservative. We notice too that the confidence intervals associated with the MLE are longer than those for the OBRE. This is due to the large variances associated with the MLE as a result of the extreme observations. Regardless of which confidence intervals are used it is clear that there is a significant difference between the quantile obtained using the OBRE versus that obtained using the MLE. Using the MLE of the quantile would result in the engineer associating a .001 chance of failure with approximately 78,000 cycles. Alternatively, if the engineer used the OBRE estimate this chance of failure would correspond to 84,000 cycles. In a design context the former estimate would force the engineer to build a *stronger* structure but perhaps unnecessarily. In building a stronger structure there is no doubt increased cost, manpower and time involved.

Again we are faced with the decision of whether to use the OBRE, the MLE or neither in estimating this quantile. Generally, the best recommendation is to calculate

both estimates. If they are in agreement either estimate is suitable. Unfortunately, as is the case here, the OBRE has performed substantial downweighting and the estimates are therefore not in agreement. To reduce the probability of making a poor choice of estimate here it is best to determine whether the two smallest and largest observations should remain part of the data set. If this is the case alternative model fitting procedures should be employed. If instead they are special cases to be handled separately we would be confident reporting the OBRE of the quantile or perhaps even removing these observations from the data set and then recalculating the MLE of the quantile. However removing the observations and recalculating the MLE would give almost the same results as the OBRE as this is essentially what the OBRE is doing.

4.1.2 Data Set 2

We now move on to Data Set 2 which consists of 101 observations corresponding to a maximum stress per cycle of 21,000 psi. With $\varepsilon = .001$ we compute the MLE of the distributional parameters and obtain:

$$\begin{aligned}\hat{\alpha}_{MLE} &= .31 \text{ (.02)} \\ \hat{\beta}_{MLE} &= 1336.37 \text{ (40.76)}.\end{aligned}$$

Upon computing the OBRE ($c = 4.0$) with $\eta = .05$ we obtain the estimates

$$\begin{aligned}\hat{\alpha}_{OBRE} &= .27 \text{ (.02)} \\ \hat{\beta}_{OBRE} &= 1373.60 \text{ (47.57)}.\end{aligned}$$

All weights are not significantly different from one with the exception of the smallest one which is assigned a weight of .23. In this instance the OBRE has heavily down-weighted a single observation in the lower tail of the distribution. This is reflected in the larger β estimate obtained using the OBRE as compared to that obtained via

Table 3: $p = .001$ quantiles for Data Set 2.

	MLE	OBRE
$t_{.001}$	529.61	554.36
Method A	(519.87, 540.64)	(548.94, 572.03)
Method B	(522.49, 536.73)	(549.14, 559.58)

the MLE.

These differences in the OBRE and MLE may be very important to the engineer. In choosing $\hat{\beta}_{OBRE}$ the engineer would take the median life-length of the material to be approximately 1,374,000 cycles and perhaps make important inspection and maintenance decisions accordingly. If instead they choose the $\hat{\beta}_{MLE}$ this median life-length would be accepted as only 1,337,000 cycles and maintenance issues, for instance, would be addressed sooner. The normal approach when dealing with suspected outliers is to investigate their origin and determine whether they should remain part of the data set. In the situation at hand, if it is determined that the smallest observation is indeed an outlier the OBRE is most probably the best choice. However, if this observation is a valid part of the data set perhaps an alternative model should be investigated given that the MLE is not appropriate as these estimates have been heavily influenced by the one outlying observation. Once again this decision rests with the engineer. Regardless of the decision made the OBRE has exposed the great impact the smallest observation has on the MLE. This information, in itself is very useful and takes us a step closer to fully understanding the behavior of this data set.

It is also worthwhile taking a look at the impact such an extreme observation has on the lower quantiles. The $p = .001$ quantiles and their 95% confidence intervals are shown in Table 3. One can see that there is a difference of approximately 25,000 cycles in the two quantiles. Clearly the smallest observation does in fact influence this lower quantile. We notice too that the confidence intervals for the quantile obtained using the MLE are longer than those for the OBRE due to the larger variances

associated with the MLE. In order to ascertain which quantile is more appropriate further investigation of the smallest observation is once again required.

4.1.3 The Choice of c

Before moving on to fatigue data of a different origin it is advantageous to examine the effect of c on the OBRE when applied to real data. Let us consider 35 specimens of aluminum, all tested to failure at a stress of 26,000 psi using a specially designed testing program [15]. This data can be found in Table 15 of Appendix B. With $\epsilon = .001$, we compute the MLE of the distributional parameters and obtain:

$$\hat{\alpha}_{MLE} = .19 \text{ (.02)}$$

$$\hat{\beta}_{MLE} = 390.18 \text{ (12.28)}.$$

Upon computing the OBRE ($c = 4.0$) with $\eta = .05$ we obtain the estimates

$$\hat{\alpha}_{OBRE} = .19 \text{ (.02)}$$

$$\hat{\beta}_{OBRE} = 393.16 \text{ (13.88)}.$$

The smallest observation is assigned a weight of 0.72 indicative of some degree of downweighting. However, it still appears to provide some useful information as it has not been downweighted too severely. The OBRE and MLE are in relatively close agreement giving us confidence in these results. The standard deviations of the MLE and the OBRE are almost the same, as we would expect, given that it appears we have moved only slightly away from the model due to the smallest observation. Let us now compute the OBRE using a smaller value of c . By decreasing c to 2.0 we expect

to reduce the bias in our estimates while sacrificing some efficiency. We obtain:

$$\hat{\alpha}_{OBRE} = .18 \text{ (.02)}$$

$$\hat{\beta}_{OBRE} = 394.20 \text{ (13.32)}.$$

This time the smallest five and largest four observations have all been significantly downweighted as follows (each observation's number corresponds to its place in Table 15):

Observation 1: *weight* = .15

Observation 2: *weight* = .35

Observation 3: *weight* = .48

Observation 4: *weight* = .78

Observation 5: *weight* = .89

Observation 32: *weight* = .80

Observation 33: *weight* = .52

Observation 34: *weight* = .39

Observation 35: *weight* = .31

It appears the OBRE of α and β are moving further away from the MLE. The OBRE is doing too much downweighting and reminds us that we must pay special attention to ensure that we do not downweight observations too much. This is of special importance when dealing with data sets of small size wherein we are already restricted by the limited information available. We must remember that the data contains *information*: if we downweight *good* data we loose information.

4.2 Fatigue of Concrete

We now move on to briefly consider the fatigue of concrete. These data represent the life-lengths for concrete as reported by Holmen [38]. We examine three different sets of 15 observations all provided in Table 16 of Appendix B. Each is characterized by the ratio of applied cyclic stress to stress causing static failure. These ratios are 0.95, 0.90 and 0.825. We expect the number of cycles to failure to increase inversely with this ratio.

Results are shown in Table 4. ϵ has been taken as .001 and η as .05. The OBRE has not significantly downweighted any of the observations so we expect it to behave in a similar fashion to the MLE. Table 4 validates this in exposing the agreement between the MLE and the OBRE with regard to both the point estimates and their associated standard deviations. In Table 5 the $p = .001$ quantiles for the three sets of data are given. We expect the quantiles obtained using the OBRE to be similar to those obtained using the MLE as they are functions of parameter estimates which are similar. Both the quantiles themselves and their confidence intervals give us no reason to suspect this is not the case. Notice that each confidence interval includes the corresponding quantile computed using both the MLE and the OBRE. This suggests that there is no significant difference between the types of estimates. Also of interest is the fact that we again see that Method A results in longer confidence intervals than Method B.

Table 4: Parameter estimates for concrete data ($c = 4.0$).

Ratio	MLE		OBRE	
	α	β	α	β
.950	.52(.09)	112.12(14.55)	.51(.10)	109.00(14.00)
.900	.48(.09)	372.99(44.50)	.48(.11)	373.10(56.98)
.825	.51(.09)	2744.87(349.68)	.57(.20)	2780.68(489.01)

Table 5: $p = .001$ quantiles for concrete data.

	.950	.900	.825
MLE			
$t_{.001}$	25.75	95.54	646.02
Method A	(20.56,32.32)	(75.50,118.32)	(515.65,811.42)
Method B	(21.94,29.56)	(82.38,108.70)	(552.88,739.16)
OBRE			
$t_{.001}$	25.41	93.33	568.93
Method A	(20.20,32.62)	(71.50,124.91)	(378.28,876.69)
Method B	(21.68,29.14)	(73.35,113.31)	(427.08,710.78)

4.3 Fluid Failure Data

Next we apply the Birnbaum-Saunders distribution to some data of quite a different nature. The data shown in Table 17 of Appendix B represent 19 observation of times (in minutes) to break down of an insulating fluid under 32kV of voltage stress. We begin by computing the MLE of the distributional parameters stopping the iterations when the successive MLE of β agree within $\varepsilon = .001$. We obtain:

$$\hat{\alpha}_{MLE} = 1.90 \quad (.31)$$

$$\hat{\beta}_{MLE} = 4.82 \quad (1.41).$$

Upon computing the OBRE ($c = 4.0$), we obtain the estimates:

$$\hat{\alpha}_{OBRE} = 1.53 \quad (.22)$$

$$\hat{\beta}_{OBRE} = 7.20 \quad (1.89).$$

All weights are not significantly different from one with the exception of the smallest which is assigned a weight of 0.62. This is indicative of significant downweighting in the lower tail of the distribution. We see that $\hat{\beta}_{OBRE}$ is larger than $\hat{\beta}_{MLE}$ as a result of this downweighting. We notice too that $\hat{\alpha}_{OBRE}$ is smaller than $\hat{\alpha}_{MLE}$ indicating that the Birnbaum-Saunders distribution associated with the OBRE is less right-skewed

that that associated with the MLE; obviously a result of downweighting the smallest observation. The plots in Figure 3 of Section 2.2 help remind us of the distributional differences inherent in having $\alpha = 1.53$ instead of 1.90. In light of the small size of this data set we will not draw many conclusions from our analysis with the exception of noting the strong role that the smallest observation appears to play.

4.4 Steel Data

A constant stress level testing program was conducted using SAE (Society of Automotive Engineers) 1045 steel specimens, all from a single heat of material [15]. Three stress levels were investigated and the results for 20-specimen samples are shown in Table 18 of Appendix B. The fatigue lives are reported in the thousands of cycles corresponding to constant stress levels of 55,000 psi, 61,000 psi and 67,000 psi, respectively. Results of computing the MLE and the OBRE ($c = 4.0$) are shown in Table 6 ($\epsilon = .001$, $\eta = .05$). The estimates and their standard deviations appear to be in quite close agreement leading us to believe that all three data sets can be well-modeled by a Birnbaum-Saunders distribution.

It is insightful to examine what would happen to the results in Table 6 if an observation was misrecorded. It is not difficult to imagine such an event occurring in practice. Say, for instance, the sample at 67,000 psi was mishandled causing an improper number of cycles to failure and observation 10 of data set 3 was consequently 440 instead of 44. We are interested in whether, with such a small data set, our

Table 6: Parameter estimates for steel data ($c = 4.0$).

Stress Level (psi)	MLE		OBRE	
	α	β	α	β
55000	.58(.09)	440.36(54.67)	.56(.09)	415.72(63.24)
61000	.24(.04)	115.02(6.10)	.26(.07)	114.82(7.17)
67000	.24(.04)	42.17(2.26)	.24(.04)	41.35(2.49)

estimation procedures will still perform as predicted. The new estimates obtained in the presence of this contamination are as follows:

$$\begin{aligned}\hat{\alpha}_{MLE} &= .66 \text{ (.10)} \\ \hat{\beta}_{MLE} &= 52.60 \text{ (7.31)}.\end{aligned}$$

and

$$\begin{aligned}\hat{\alpha}_{OBRE} &= .27 \text{ (.01)} \\ \hat{\beta}_{OBRE} &= 42.07 \text{ (3.14)}.\end{aligned}$$

The contaminant observation is assigned a weight of .30. Evidently the OBRE has downweighted the contamination accordingly and provided estimates quite consistent with those obtained earlier with no contamination present. This is as both predicted and desired. On the other hand, the MLEs have been severely distorted by the contaminant and now possess large bias when compared with the results of Table 6 as well as larger standard deviations. This example easily validates the superior performance of the OBRE in the presence of contamination even with relatively small data sets.

4.5 Fatigue Test Results

The previous applications have illustrated that the interpretation of experimental results is extremely context dependent. The decision of whether to report a low quantile estimate or median estimate of life-length is crucially related to how these estimates will be used. Decisions regarding whether to use the OBRE, the MLE or recommend an alternative model rely heavily on the engineer's knowledge about the data set in question. Even so, consideration of some basic objectives may prevent

serious errors.

There are two frequent purposes in the use of fatigue test results. The first is to estimate, on the basis of available results of tests on materials, the fatigue lifetime for a structural part. Another objective is to evaluate, from comparison of results of two (or more) sets of fatigue tests, the effect of some specific variable in material, fabrication, or design.

The first objective is the more difficult to attain. In fact Collins [15] once wrote "It is, in most cases, not safe to depend directly upon laboratory test results for quantitative design of structures to withstand a specified number of cycles of repeated loading". Such comments could easily instill fear in an engineer. This emphasizes the usefulness of the OBRE even solely as a mechanism for checking the validity of the proposed model. The engineer would normally be happiest working with a lower quantile give this type of objective. Once this is verified one must still decide whether to use the OBRE, the MLE or search for a better model. Clearly the engineer's knowledge of the data is fundamental for ensuring wise decisions are made.

The second objective is concerned with the desire to compare data from two different sources, data taken at two different times, or data taken under two different conditions. In making such comparisons the usual approach is to compare the population parameters associated with the two different sets of data. Hence, in such situations the OBRE or the MLE of the distributional parameters may be completely adequate. The decision of which to use will be influenced by the presence (or absence) of contamination.

5 Simulation Study

In order to quantitatively compare the MLE and the OBRE a simulation study was performed. This study was designed to gage the effectiveness of the OBRE when the Birnbaum-Saunders model holds exactly as well as when there is contamination present. Samples of size $n = 25, 50$ and 100 were generated. Sample sizes larger than this were not considered since in practice data sets larger than this seldom exist. Fatigue data, for example, is not only expensive but very difficult to obtain.

The shape parameter of the Birnbaum-Saunders distribution was permitted to take on values between zero and two. This chosen range is consistent with that found in the literature. For instance, Birnbaum and Saunders [8] concluded that in fatigue studies $0 < \alpha \leq 0.5$ is the normal range. Experience with aircraft-engine data indicated that the case of $\alpha > 2$ is atypical in practice [45]. Chang and Tang [12] looked at $0.4 \leq \alpha \leq 2.2$. Ahmad reported on $\alpha = 0.25, 0.5$ but said he also looked at $\alpha = 1, 2$ [1]. Rieck and Nedelman [45] argued that α greater than two is unusual and in practical engineering applications they suggested that α is normally between zero and one. Given that we are primarily concerned with engineering applications our chosen range is therefore quite suitable.

Both the MLE and the OBRE of the scale parameter β are invariant under linear transformations of the data, so without loss of generality β was fixed at one.

5.1 Suitable Contaminated Models

To effectively gage our robust estimation procedure it is beneficial to simulate contamination as would likely occur in practice. This will lead to greater confidence in its practical application.

We will design contaminated models in the context of fatigue as this is the applica-

tion of primary concern. In an effort to do so we consider fabricated steel structures. Obtaining such structures involves complicated manufacturing and fabricating processes. Although fabricators and inspectors proceed as carefully as possible there is always the possibility of a machine malfunction or power outage. Such unlikely events may severely shorten the life-length of the structure of concern. It is for this reason that we shall judge the merits of our robust estimation procedure when faced with contaminant observations of much shorter life-length than those of the majority of the data set.

We must be careful that the OBRE does not downweight too much in the lower tail of the Birnbaum-Saunders distribution where the shorter life-lengths reside. This will help avoid severely downweighting flaws that are an expected result of the type of fabrication. Although the fabricator of the structure and those responsible for the fabrication inspection will attempt to minimize these flaws, it is neither practical nor economically possible to eliminate them in all cases. By selecting a reasonable value for c , the OBRE can allow for flaws that are known to exist in structures arising from surface and edge imperfections, lack of fusion, porosity and inclusions, irregularities in mill scale, and from mechanical notches due to handling, straightening, cutting, shearing, etc. In fact the weights assigned by the OBRE can help to quantify the nature and prevalence of flaws which is very beneficial to inspection design. At the same time the OBRE will still pick up on extremely outlying observations which may be of deeper interest.

It is also quite feasible that parts of a structure may exhibit a different life-length distribution than that of the bulk of the structure. This could be a consequence of incomplete mixing of components or perhaps even faulty machinery. Regardless of its origin it is worthwhile examining how our robust estimation procedure fares when faced with a mixture model. Such a model is one in which the majority of the data follows one distribution but a small percentage follows some other distribution. In

this case we intend for the OBRE to help identify the part of the structure following a different life-length distribution. If identified the corresponding part of the data set may be removed and modeled separately leading to a more accurate model of both parts of the structure.

Lastly, it is worthwhile introducing contaminants with much larger life-lengths. Such outliers would pose as those structures that just won't fail, but instead survive much longer than one would normally anticipate. These outliers could be the result of faulty machinery, ineffective testing, incorrect mixing, etc. In their presence we will see that the MLE of the median life is significantly magnified giving engineers a false sense of security. Errors like these may have serious consequences. For instance, in severely overestimating the life-length of a bridge human lives are placed at risk. Fortunately robust estimation procedures can help alleviate such problems.

5.2 The Implementation

Data were generated with the random number generator *ran1* [44] in Fortran on a DEC-Alpha according to the following distributions:

- Model 1: Noncontaminated model $\{F_{\alpha,\beta}\}$, with $\alpha = 0.2, 0.6, 1.0, 1.4$ and 1.8 and $\beta=1.0$.
- Model 2: Model with 1% of very bad contamination; the upper 1% of the observations is multiplied by 5 [$\alpha=0.5, \beta=1.0$].
- Model 3: Model with 1% of very bad contamination; the lower 1% of the observations is multiplied by $\frac{1}{5}$ [$\alpha=0.5, \beta=1.0$].
- Model 4: Mixture model with 5% contamination given by $\{0.95F_{\alpha,\beta}+0.05F_{\alpha,1\beta}\}$ [$\alpha=0.5, \beta=1.0$].

The algorithm for the computation of the OBRE was given in Section 3.3. Romberg integration was used where numerical integration was required. This method stands highly recommended [44] since it makes use of integration methods of a variety of orders. Often the interval over which integration was to be performed was divided into smaller subintervals. This increased precision by helping to avoid overlooking parts of the curve or approximating them poorly.

Some minor problems were encountered with convergence of the OBRE algorithm. In some cases, estimates oscillated between two values (never gaining any ground). The values of the parameters when the maximum number of iterations (30) was reached were taken as the estimates. In other cases, the proposed change in the parameters ($\Delta\theta$) were such that one or both of the new estimates were negative which is impossible. In such cases a procedure similar to that used in nonlinear regression was implemented. That is, $\Delta\theta$ was halved and new estimates obtained. If after ten attempts we were still unsuccessful the most recent non-negative estimates were used. These cases accounted for less than 1% of the total simulations. Removing the problematic cases, and reporting only cases of absolute convergence may have improved our results. However, our results as they stand, are already a considerable improvement over the MLE and thus all cases are included in the results presented. In the first step of the OBRE algorithm we used LMS estimates as initial values. The desirable robustness properties of these estimates can increase both the convergence rate and overall success of the OBRE algorithm.

5.3 The OBRE at the Model

5.3.1 Parameter Estimates

We first evaluate the behavior of the OBRE at the model. In doing so we wish to determine the loss of efficiency relative to the MLE which occurs when we build-in

Table 7: Bias (RMSE) of estimators of the shape parameter α for Birnbaum-Saunders data.

n	Method	α				
		0.2	0.6	1.0	1.4	1.8
25	MLE	-.004(.029)	-.013(.087)	-.023(.145)	-.035(.195)	-.046(.264)
	OBRE $c = 4$	-.010(.039)	-.033(.123)	-.050(.174)	-.085(.248)	-.120(.397)
	OBRE $c = 3$	-.007(.038)	-.025(.120)	-.032(.186)	-.063(.294)	-.096(.395)
	OBRE $c = 2$	-.006(.039)	-.016(.117)	-.035(.200)	-.062(.310)	-.143(.486)
50	MLE	-.001(.020)	-.004(.059)	-.008(.098)	-.012(.138)	-.016(.178)
	OBRE $c = 4$	-.004(.023)	-.014(.073)	-.027(.133)	-.034(.184)	-.068(.313)
	OBRE $c = 3$	-.003(.024)	-.011(.074)	-.019(.127)	-.024(.183)	-.048(.289)
	OBRE $c = 2$	-.002(.027)	-.007(.080)	-.012(.134)	-.018(.195)	-.046(.313)
100	MLE	-.001(.014)	-.003(.042)	-.005(.069)	-.008(.097)	-.010(.125)
	OBRE $c = 4$	-.003(.015)	-.008(.046)	-.016(.082)	-.026(.132)	-.030(.172)
	OBRE $c = 3$	-.002(.016)	-.007(.048)	-.013(.084)	-.020(.122)	-.029(.192)
	OBRE $c = 2$	-.002(.018)	-.005(.055)	-.010(.094)	-.012(.137)	-.010(.125)

this type of outlier protection. We judge the OBRE on the basis of the MLE since we know the MLE are asymptotically minimum variance unbiased estimators at the model.

500 simulations were run at each of three sample sizes; 25, 50 and 100. This random generation of Birnbaum-Saunders data (via the method of Birnbaum and Saunders as detailed in Section 2.2) was repeated at each of five α values and the OBRE were calculated for $c = 2, 3$ and 4 with $\eta = .05$. In Table 7 we report the bias and the root mean square error of $\hat{\alpha}_{MLE}$ and $\hat{\alpha}_{OBRE}$ for various values of c . The scale parameter remained fixed at one. The results showed no change in the first two decimal places when the random number generator seed was changed giving us confidence in the accuracy and precision of our results. Overall, $\hat{\alpha}_{MLE}$ and $\hat{\alpha}_{OBRE}$ are quite similar in behavior with the MLE showing slightly less bias and slightly smaller root mean square error. As expected, the results improve with increasing sample size n giving us confidence in the consistency of the OBRE of α .

In Table 8 we report the bias and the root mean square error of $\hat{\beta}_{MLE}$ and $\hat{\beta}_{OBRE}$ for

Table 8: Bias (RMSE) of estimators of the scale parameter β for Birnbaum-Saunders data.

n	Method	α				
		0.2	0.6	1.0	1.4	1.8
25	MLE	.000(.042)	.007(.122)	.018(.192)	.029(.246)	.038(.285)
	OBRE $c = 4$	.000(.044)	-.002(.142)	.005(.223)	.014(.318)	.028(.387)
	OBRE $c = 3$	.000(.042)	.000(.128)	.002(.208)	.020(.308)	.034(.399)
	OBRE $c = 2$	.000(.047)	.003(.140)	.009(.240)	.036(.330)	.044(.437)
50	MLE	.001(.028)	.006(.080)	.011(.124)	.012(.211)	-.021(.181)
	OBRE $c = 4$	.001(.030)	-.001(.093)	-.004(.155)	-.001(.186)	.000(.234)
	OBRE $c = 3$	.001(.031)	.000(.092)	.002(.150)	.009(.191)	.010(.237)
	OBRE $c = 2$	.001(.033)	.004(.098)	.006(.159)	.012(.157)	.016(.280)
100	MLE	-.000(.020)	.000(.057)	-.002(.089)	.004(.111)	-.006(.127)
	OBRE $c = 4$	-.000(.021)	-.002(.061)	-.005(.098)	-.007(.132)	-.005(.145)
	OBRE $c = 3$	-.001(.021)	-.002(.062)	-.005(.099)	-.007(.128)	-.007(.155)
	OBRE $c = 2$	-.001(.022)	-.002(.065)	-.003(.107)	-.003(.143)	-.002(.172)

various values of c . The results again showed no change in the first two decimal places when the random number generator seed was changed. Overall, $\hat{\beta}_{MLE}$ and $\hat{\beta}_{OBRE}$ are very similar in behavior with the MLE showing slightly less bias and slightly smaller root mean square error the majority of the time. With $n = 100$ the best OBRE results are obtained for $c = 4$. It is more difficult to draw similar conclusions with $n = 25$ or 50 especially when α is large. Even so, Table 8 suggests that β is estimated very well by both the OBRE and MLE given the small bias and RMSE inherent throughout the table. We conclude that the OBRE of β is in close competition with the MLE at the model.

5.3.2 A Look at Relative Efficiencies

A second comparison of the OBRE and the MLE at the model can be made using their asymptotic variances. In the case of the MLE, these variances were calculated using the results of Engelhardt, Bain and Wright [25] as presented in Section 2.3. This involved numerically integrating to obtain the asymptotic variance of $\hat{\beta}_{MLE}$.

Table 9: Relative efficiencies of the OBRE ($c = 4.0$) for Birnbaum-Saunders data, $n=100$.

		OBRE to MLE				
		0.2	0.6	1.0	1.4	1.8
α		88%	87%	85%	83%	80%
β		98%	96%	93%	89%	84%

The asymptotic variances for the OBRE were computed as detailed in Section 3.5 using empirical distributions when necessary.

In Table 9 we report the relative efficiencies of the OBRE to the MLE for both α and β . Again 500 samples of size 100 were generated at each of the five α values. We used a sample size of 100 having established earlier in Section 3 that asymptotics appear to come into play by this point leading to more accurate results and conclusions. One can see that there is a loss of efficiency relative to the MLE which becomes more pronounced as α becomes large. Even so, this loss of efficiency is not very substantial indicating that the OBRE is in close competition with the MLE at the model.

We are aware that the sensitivity of the OBRE to contamination depends on the choice of the bound c . This was demonstrated in Section 4.1.4 when considering aluminum data. However, we have also seen that lowering the bound c leads to an efficiency loss at the model. As a different mechanism of quantifying this loss we can measure the overall relative efficiency at the model using the ratio of the traces of the asymptotic covariances matrices of the OBRE and the MLE. These results are shown in Table 10. With $\alpha = 0.2$ the OBRE ($c = 4$) is 95% as efficient as the MLE. These overall relative efficiencies deteriorate both with decreasing c and increasing α . In a real situation, one would normally use a bound c such that 95% efficiency is achieved.

Table 10: Overall relative efficiencies for the OBRE for Birnbaum-Saunders data, $n=100$.

	OBRE to MLE				
	0.2	0.6	1.0	1.4	1.8
$c = 4$	95%	92%	90%	86%	82%
$c = 3$	89%	86%	81%	77%	72%
$c = 2$	73%	68%	62%	54%	48%

5.4 The OBRE with Contamination

5.4.1 Parameter Estimates

Before reporting the results obtained with contaminated models it is worthwhile taking a moment to ascertain what should happen. In Models 2 and 3 we introduce observations into either tail of the distribution. If these observations are accepted as part of the model they will increase the skewness of the original distribution. Figures 2 and 3 of Section 2.2 illustrate that as skewness increases α increases. Hence we expect the MLE to accept these extreme observations and consequently overestimate α . Alternatively, the OBRE, in identifying these extreme observations as outliers, should remain relatively unaffected. Our results substantiate this as can be seen in Table 11. There is a large positive bias associated with the MLE of α . The OBRE, in downweighting the contamination, does not exhibit such a large positive bias.

β , the median of the distribution, should show little change since a small number of extreme observations will have little impact on such a robust estimate. Again our hypotheses are validated by the results. With 1% contamination in the upper tail of the distribution we see a small but positive bias associated with the MLE of β . When the contamination is instead in the left tail of the distribution there is a small negative bias associated with the MLE of β . Interestingly this phenomenon is not exhibited by the OBRE of β which consistently slightly underestimates β . This is no doubt a result of the residual downweighting that has occurred. We established the

Table 11: Effects of contamination on the MLE and the OBRE, $n=100$.

Model	Contamination	Parameter	Bias		Bound c	RMSE
			MLE	OBRE		
2	1%[upper]	α	.110			.117
		β	.062			.081
		α		-.012	4.0	.051
		β		-.003		.053
		α		-.009	2.0	.052
		β		-.002		.054
3	1%[lower]	α	.110			.118
		β	-.058			.055
		α		-.011	4.0	.051
		β		-.002		.053
		α		-.008	2.0	.050
		β		-.001		.054
4	5%[mixed]	α	.297			.305
		β	-.198			.204
		α		.031	4.0	.070
		β		-.039		.067
		α		.029	2.0	.068
		β		-.036		.066

existence of such downweighting with our bench marks earlier in Section 3.3.

5.4.2 The Weights

It is also instructive to examine the weights that have been assigned in some of the contaminated models. This will lead to a better understanding of how the OBRE performs when faced with contamination.

The results are shown in Table 12. For the sake of simplicity all the weights have not been listed. However with the information provided one can see that with $c = 2.0$ too much downweighting is occurring. Similar behavior was seen in Table 1 of Section 2 when no contamination was present. Here observations 1 through 14 and 86 through 100 are actually significantly downweighted which is undesirable. The

Table 12: Mean (standard deviation) of weights for the OBRE for contaminated Birnbaum-Saunders data.

	Rank of Weight										
	1	2	3	4	5	...	96	97	98	99	100
Model 2	.80	.96	1.00	1.00	1.00		1.00	1.00	.99	.95	.08
$c = 4.0$	(.18)	(.09)	(.02)	(.00)	(.00)		(.00)	(.00)	(.03)	(.10)	(.15)
Model 2	.23	.30	.36	.42	.49		.47	.41	.35	.29	.02
$c = 2.0$	(.08)	(.09)	(.10)	(.11)	(.11)		(.11)	(.10)	(.09)	(.08)	(.00)
Model 3	.08	.95	.99	1.00	1.00		1.00	1.00	1.00	.96	.81
$c = 4.0$	(.01)	(.10)	(.03)	(.01)	(.00)		(.00)	(.00)	(.02)	(.09)	(.18)
Model 3	.02	.30	.36	.43	.48		.48	.42	.36	.30	.23
$c = 2.0$	(.00)	(.09)	(.11)	(.11)	(.11)		(.11)	(.10)	(.09)	(.08)	(.08)
Model 4	.23	.34	.45	.62	.87		1.00	1.00	1.00	.99	.90
$c = 4.0$	(.09)	(.12)	(.15)	(.19)	(.16)		(.00)	(.00)	(.02)	(.04)	(.14)
Model 4	.02	.04	.05	.07	.12		.52	.46	.39	.33	.25
$c = 2.0$	(.01)	(.02)	(.02)	(.03)	(.06)		(.11)	(.11)	(.10)	(.09)	(.09)

results with $c = 4.0$ are much better. In the case of Model 2 one sees that the largest observation (the 1% upper contamination) has been substantially downweighted as desired. In the case of Model 3 the smallest observation (the 1% lower contamination) has been substantially downweighted. The 5% contamination of Model 4 is in the lower tail of the distribution since β was reduced to one tenth of its original size for generation of this contamination. This contamination (the smallest five observations) has been effectively downweighted. Evidently the OBRE has successfully picked up the contamination and placed little importance on it when calculating the parameter estimates.

Another useful conclusion stemming from these results is the possibility of determining an appropriate value for c solely by examining the weights corresponding to a number of different c values. That is, we can select a c value which does not lead to too much downweighting nor is too lenient. Clearly $c = 4.0$ appears quite favorable here which is consistent with what we know regarding data sets of size $n = 100$ and an α value in the range of 0.5.

Table 13: $\text{Var}(\text{MLE})/\text{Var}(\text{OBRE})$ for contaminated Birnbaum-Saunders data.

Model	2 ($c = 4.0$)	2 ($c = 2.0$)	3 ($c = 4.0$)	3 ($c = 2.0$)
α	542%	287%	544%	288%
β	180%	135%	141%	104%

5.4.3 Efficiency

It is also important to ascertain that the OBRE is still competitive with regard to its variance in the presence of contamination. It is no longer appropriate to report relative efficiencies since this term is usually reserved for comparing the variance of one estimate with another known a priori to be optimal, for instance, the MLE at the model. However, here we are no longer at the model so the MLE is not necessarily optimal. Consequently, we choose to report the ratio of the variances and cautiously interpret them. The variance ratios for some of the contaminated models are given in Table 13 and make it clear that the OBRE are far less variable than the MLE in the presence of contamination. We see too that as c decreases the variances of the OBRE increase. This is as we expect since we are sacrificing efficiency for robustness by decreasing c .

5.5 Comments

The simulation results provide strong evidence in support of the OBRE. The OBRE is extremely competitive with the MLE at the model. As well, in the presence of contamination the OBRE provides good estimates with modest standard deviations based on the bulk of the data. It also provides insightful weights identifying observations lying outside the model.

The performance of the OBRE with respect to efficiency deteriorates with increasing α . However, the estimates are not severely distorted but rather need only be interpreted with less confidence due to their increased variance. Additionally, in the

normal range of $0 < \alpha < 1$ the performance of the OBRE is more than adequate for practical purposes.

By far the most concrete conclusions are drawn from consideration of results obtained with $n = 100$. This is true of many statistical procedures whose performance is enhanced by increased sample size. Even so, with smaller sample sizes the OBRE does not yield nonsensical results but merely deteriorates in a quantifiable fashion.

6 Conclusions

Statistical techniques are fundamental in describing, analyzing and comparing fatigue failure data so that the engineer may achieve the desired reliability on a rational basis. The Birnbaum-Saunders family of probability distributions can effectively model fatigue as well as many other interesting phenomena.

If data sets were free of model deviations the method of maximum likelihood would provide the most efficient unbiased estimates of a Birnbaum-Saunders distribution. Unfortunately, seldom, if ever, do we have the pleasure of dealing with data sets that can be impeccably modeled and are instead faced with a multitude of different types of contamination particularly when considering fatigue data. Whether or not this contamination is due to misrecording errors or machine flaws, we must acknowledge its existence and design estimation procedures accordingly.

By exposing the detrimental impact of contamination on the commonly used MLE, we established the great need for an alternative estimation procedure for the Birnbaum-Saunders distribution. We responded with powerful robust techniques which still give reliable parameter estimates in the presence of contamination while at the same time providing a set of weights giving an indication of the fit of the model to the data. The robustification of statistical procedures of this sort marks one of the major developments in statistics this century.

The OBRE was chosen as an effective robust estimator with which to obtain estimates of the parameters of the Birnbaum-Saunders distribution and programs were written for its implementation. Applications to real data and a thorough simulation study validated its excellent performance in giving estimates based on the bulk of the data as well as providing useful diagnostics in the presence of contamination. We also saw the benefit of using the OBRE when calculating lower quantiles which are of great use to design engineers.

The importance of selecting an effective starting value for the OBRE algorithm was emphasized. Having established that the MLE perform poorly in the presence of contamination, we investigated other possibilities for use as a starting value. Using the theory of influence functions we exposed the limitations of Birnbaum and Saunders' mean mean estimate via both a simple example and an analytical justification. Ahmad's Jackknife estimate [1] was evaluated and praised but its large computational time also made it undesirable for use as a starting value.

Another estimation procedure for the parameters of the Birnbaum-Saunders distribution was recently proposed by Chang and Tang [12]. We improved this graphical approach [12] by using least median of squares regression as an alternative to least squares regression. These estimates, although robust, were used solely as starting values for the OBRE algorithm. This was due primarily to the fact that they do not enable calculation of insightful weights nor possess a manageable asymptotic covariance structure as does the OBRE.

Our robust estimation techniques for the Birnbaum-Saunders distribution provide the engineer with a mechanism with which to identify observations lying outside the model. This can communicate much about manufacturing processes and performance in general. When in agreement with the MLE additional confidence can be placed in both types of estimates. Valuable insight can be obtained concerning the adequacy of the model for the data in the case of discrepancies.

We have seen that we must give up some degree of efficiency to gain robustness and the OBRE provides us with a powerful method of doing so. Our simulation study has determined that compromises can be made which are effective in both regards. As well, efficiency in many ways becomes a non-issue as we move away from the model since there is no established method currently available which has been shown to be optimal in such settings. The strength of the OBRE rests with its ability to indicate when we are no longer at the model and identify the probable cause via the weights

assigned.

Like many estimation techniques our robustness approach is limited by the lack of available data. Today we are capable of handling huge data sets with which we can make maximum use of asymptotic results. Unfortunately in many fatigue problems data of this sort is non-existent. For the time being we must judge the effectiveness of our robust approach through appropriately designed simulations as well as by examining how it fairs in practice after being implemented in the design context. Perhaps, in the future our approach will be of importance in the development of codes for use by engineers the world over.

Any technique which takes us closer to the *truth* regarding a data set is extremely valuable. With the great increases in computing power, computer intensive methods of this sort are taking statistics to a whole new frontier. Powerful estimation procedures like the one developed here will become an important member of every statisticians tool box.

References

- [1] Ahmad I. A. (1988). Jackknife estimation for a family of life distributions. *Journal of Statistical Computation and Simulation*, 29, 211-223.
- [2] Barlow, R.E. and Proschan, F. (1965). *Mathematical Theory of Reliability*. Wiley, New York.
- [3] Berman, S.M. (1970). Occupation times of stationary gaussian processes. *Journal of Applied Probability*, 7, 721-733.
- [4] Bhattacharyya, G.K. and Fries, A. (1982). Fatigue failure models - Birnbaum-Saunders vs. Inverse Gaussian. *IEEE Transactions on Reliability*, R-31, 439-440.
- [5] Birnbaum, Z.W., Esary, J.D., and Marshall, A.W. (1966). Stochastic characterization of wear-out for components and systems. *Annals of Mathematical Statistics*, 37, 816-825.
- [6] Birnbaum, Z.W. and Saunders, S.C. (1969). A new family of life distributions. *Journal of Applied Probability*, 6, 319-327.
- [7] Birnbaum, Z.W. and Saunders, S.C. (1958). A statistical model for life-length of materials. *Journal of the American Statistical Association*, 53, 151-160.
- [8] Birnbaum, Z.W. and Saunders, S.C. (1969). Estimation for a family of life distributions with applications to fatigue. *Journal of Applied Probability*, 6, 328-347.
- [9] Bishop, T. (1986). Fatigue and the Comet Disasters. *Sourcebook in Failure Analysis*, Menlo Park, OH: American Society for Materials, 209-215.
- [10] Bowes, W. H., Russell, L.T., Suter, G.T. (1990). *Mechanics of Engineering Materials*. BRS Publications, Ottawa.

- [11] Chang D.S., and L.C. Tang (1993). Reliability bounds and critical time for the Birnbaum-Saunders Distribution. *IEEE Transactions on Reliability*, 42(3), 464-469.
- [12] Chang D.S., and L.C. Tang (1994). Graphical analysis for Birnbaum-Saunders Distribution. *Microelectronics and Reliability*, 34(1), 17-22.
- [13] Chang D.S., and L.C. Tang (1994). Percentile bounds and tolerance limits for the Birnbaum-Saunders Distribution. *Communications in Statistics-Theory and Methods*, 23, 2853-2863.
- [14] Chang D.S., and L.C. Tang (1994). Random Number Generator for the Birnbaum-Saunders Distribution. *Computers ind. Engng*, 27(1-4), 345-348.
- [15] Collins, J. A. (1981). *Failure of Materials in Mechanical Design*. John Wiley & Sons, Inc, New York.
- [16] Cramér, H. (1951). *Mathematical Methods of Statistics*. Princeton University Press.
- [17] Cramér, H., and Leadbetter, M.R. (1967). *Stationary and Related Stochastic Processes*. Wiley, New York.
- [18] Cushny, A. R. and Peebles, A. R. (1905). The action of optical isomers. II. Hyoscines. *J. Physiol.* 32, 501-510.
- [19] Davis, D. T. (1952). Analysis of some failure data. *Journal of the American Statistical Association*, 47, 113-150.
- [20] Desmond, A.F. (1986). On the relationship between two fatigue-life models. *IEEE Transactions on Reliability*, 35(2), 167-169.
- [21] Desmond, A.F. (1985). Stochastic models of failure in random environments. *Canadian Journal of Statistics*, 13(2), 171-183.

- [22] Dupuis, D.J. and Field, C.A. (1997). Robust estimation of extremes. *Canadian Journal of Statistics*, to appear.
- [23] Dupuis, D.J. and Field, C.A. (1997). A comparison of confidence intervals for generalized extreme-value distributions. Submitted manuscript.
- [24] Edgeworth, F.Y. (1887). On observations relating to several quantities. *Hermathena*, 6, 279-285.
- [25] Engelhardt, M., Bain, L.J. and Wright, F.T. (1981). Inferences on the parameters of the Birnbaum-Saunders Fatigue Life Distribution based on maximum likelihood estimation. *Technometrics*, 23(3), 251-260.
- [26] Epstein, B. (1954). Truncated life tests in the exponential case. *Annals of Mathematical Statistics*, 25, 555-564.
- [27] Fenton, G.A. (1996). Double Precision Mathematics Fortran Library, Computer Software, Department of Applied Mathematics, Technical University of Nova Scotia.
- [28] Freudenthal, A.M., and Gumbel, E. J. (1954). Minimum life in Fatigue. *Journal of the American Statistical Association*, 49, 575-597.
- [29] Freudenthal, A.M., and Shinozuka, M. (1961). Structural safety under conditions of ultimate load failure and fatigue. Technical Report WADD-TR-61-77, Wright Air Development Division.
- [30] Freund, J.E. (1992). *Mathematical Statistics*. Prentice-Hall, New Jersey.
- [31] Grover, H. J., Gordon, S A., and Jackson, L. R. (1954). *The Fatigue of Metals and Structures*. Bureau of Aeronautics, Department of The Navy, USA.
- [32] Hampel, F.R. (1968). *Contribution to the Theory of Robust Estimation*. Ph.D. Thesis, University of California, Berkeley.

- [33] Hampel, F.R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association*. 69, 383-393.
- [34] Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., and Stahel, W.A. (1986). *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York.
- [35] Heritier, S., and Ronchetti, E.M. (1994). Robust bounded-influence tests in general parametric models. *Journal of the American Statistical Association*, 89, 897-904.
- [36] Hibbeler, R.C. (1991). *Mechanics of Materials*. Macmillan, New York.
- [37] Hogg, R.V. and Craig, A.T. (1995). *Introduction to Mathematical Statistics*. Prentice-Hall, New Jersey.
- [38] Holmen, J. O. (1979). Fatigue of concrete by constant and variable amplitude loading, Technical Paper, *The Norwegian Institute of Technology*, University of Trondheim.
- [39] Hsieh, H.K. (1990). Estimating the critical time of the inverse gaussian hazard rate. *IEEE Transactions on Reliability*, 39(3), 342-345.
- [40] Johnson, N.L., Kotz, S. and Balakrishnan, N. (1994). *Continuous Univariate Distributions. Volumes I, II*. Wiley, New York.
- [41] Johnson, R.A. and Wichern, D.W. (1982). *Applied Multivariate Statistical Analysis*. Prentice-Hall, New Jersey.
- [42] Jorgensen, B., Seshadri, V., and Whitmore, G.A. (1991). On the mixture of the inverse Gaussian distribution with its complementary reciprocal. *Scandinavian Journal of Statistics*. 18, 77-89.

- [43] Kulak, G.L., Adams, P.F. and Gilmor M.I. (1995). *Limit States Design in Structural Steel*. Canadian Institute of Steel Construction, Ontario.
- [44] Press, W.H., Teukolsky, S. A., Vetterling, W.T. and Flannery B.P. (1992). *Numerical Recipes in Fortran*. Cambridge University Press, Cambridge.
- [45] Rieck, J.R. and Nedelman, J.R. (1991). A log-linear model for the Birnbaum-Saunders Distribution. *Technometrics*, 33(1), 51-60.
- [46] Rousseeuw P. J. and Leroy, A. M., (1987). *Robust Regression and Outlier Detection*, John Wiley & Sons, Inc., USA.
- [47] Setó, S., Iwase, K., and Oohara, M. (1993). Characteristics of rainfall for a single event. *Technical Report No. 93-D2*, Department of Applied Mathematics, Hiroshima University, Hiroshima, Japan.
- [48] Soofi, E.S., Ebrahimi, N. and Habibullah, M. (1995). Information distinguishability with application to analysis of failure data. *Journal of the American Statistical Association*, 90, 657-668.
- [49] Stewart, J. (1991). *Calculus*, Brooks/Cole Publishing Company, California.
- [50] Stulen, F.B., and Schulte, W.C. (1986). Fatigue Failure Analysis and Prevention. *Sourcebook in Failure Analysis*, Menlo Park, OH: American Society for Materials, 10-19.
- [51] Vanmarke, E. (1983). *Random Fields: Analysis and Synthesis*, The MIT Press, London.
- [52] Vardeman, S.B. (1994). *Statistics for Engineering Problem Solving*, PWS Publishing Company, Boston.
- [53] Victoria-Feser, M. and Ronchetti, E.M. (1994). Robust methods for personal income models. *Canadian Journal of Statistics*, 22(2), 247-258.

Proof of Theorem 3 [8]

Consider the density for $t > 0$:

$$F'(t; \alpha, \beta) = \frac{1}{\alpha\beta} \Phi'[\alpha^{-1}\xi(t/\beta)]\xi'(t/\beta).$$

Taking the natural logarithm of the joint density of the observations $t_1, \dots, t_n$ one obtains the likelihood

$$L = -n \ln \alpha - n \ln \beta + \sum_{i=1}^n \left\{ -\frac{1}{2} \ln(2\pi) - \frac{1}{2} \alpha^{-2} \xi^2(t_i/\beta) + \ln \xi'(t_i/\beta) \right\}.$$

Then

$$-\frac{\alpha^3}{n} \frac{\partial L}{\partial \alpha} = \alpha^2 - \frac{1}{n} \sum_{i=1}^n \xi^2(t_i/\beta), \quad (\text{A.1})$$

$$\frac{\partial L}{\partial \beta} = -\frac{n}{\beta} + (\alpha\beta)^{-2} \sum_{i=1}^n t_i \xi(t_i/\beta) \xi'(t_i/\beta) - \frac{1}{\beta^2} \sum_{i=1}^n \frac{t_i \xi''(t_i/\beta)}{\xi'(t_i/\beta)}.$$

But notice that

$$\begin{aligned} \xi(t) &= t^{1/2} - t^{-1/2}, & \xi^2(t) &= t + t^{-1} - 2, \\ \xi'(t) &= \frac{t^{1/2} + t^{-1/2}}{2t} = \frac{1}{2\xi(t)}(1 - 1/t^2), \\ \frac{t\xi''(t)}{\xi'(t)} &= -1 + \frac{1}{2} \frac{t-1}{t+1} = -\frac{1}{2} - \frac{1}{t+1}. \end{aligned}$$

Hence by substitution

$$\frac{\partial L}{\partial \beta} = \frac{-n}{2\beta} + \frac{n}{2\alpha^2\beta} \left(\frac{s}{\beta} - \frac{\beta}{r} \right) + \frac{n}{K(\beta)}, \quad (\text{A.2})$$

$$\frac{2\alpha^2\beta}{n} \frac{\partial L}{\partial \beta} = -\alpha^2 + \left(\frac{s}{\beta} - \frac{\beta}{r}\right) + \frac{2\alpha^2\beta}{K(\beta)}. \quad (\text{A.3})$$

Equating (A.1) yields the equation

$$\alpha^2 = \frac{s}{\beta} + \frac{\beta}{r} - 2 \quad (\text{A.4})$$

and equating (A.3) to zero yields

$$\alpha^2 = \frac{s}{\beta} - \frac{\beta}{r} + \frac{2\beta\alpha^2}{K(\beta)}. \quad (\text{A.5})$$

Equating these two expressions for α^2 and simplifying yields

$$\frac{\beta}{r} = 1 + \frac{\beta\alpha^2}{K(\beta)}.$$

Substituting again for α^2 and simplifying it can be seen that the maximum likelihood estimate of β is the solution of $g(x) = 0$ for $0 < x < \infty$ (presuming for the present it is unique). It remains to show that $\hat{\beta}$ is unique and $r < \hat{\beta} < s$. Note that $g(0) = r[s + K(0)] = r(s + r) > 0$. One can check that $g(x) \rightarrow -\infty$ as $x \rightarrow \infty$, by seeing that $K(x)/x \rightarrow 1$ and $[x - K(x)] \rightarrow -s$ as $x \rightarrow \infty$. Then

$$\frac{g(x)}{x} = x - K(x) + r \frac{K(x)}{x} - 2r + \frac{sr}{x},$$

so $g(x)/x \rightarrow -(s + r)$ as $x \rightarrow \infty$. Now

$$g'(x) = (x - r)[1 - K'(x)] + x - r - K(x)$$

and one sees that

$$K'(x) = K^2(x) \frac{1}{n} \sum (x + t_i)^2,$$

and since $(E|X|^\nu)^{1/\nu}$ is nondecreasing in ν for any random variable X , one has $K'(x) > 1$. Therefore, since $x - K(x)$ is decreasing, one sees that $K(x) > x - r$ for $x > 0$ and $g'(x) < 0$ for $x > r$ with at most one change in sign. Thus it has been shown that $\hat{\beta}$ is unique. One now calculates

$$g(r) = r(s - r), \quad g(s) = (s - r)[s - K(s)].$$

Again by the argument above, $s > r$, therefore $g(r) > 0$. Thus there exists a unique solution to $g(x) = 0$, say $\hat{\beta}$, and $\hat{\beta} > r$. But $g(s) < 0$ if and only if

$$\frac{1}{s} > \frac{1}{K(s)} \quad \text{iff} \quad 1 > \frac{1}{n} \sum \frac{s}{s + t_i} = 1 - \frac{1}{n} \sum \frac{t_i}{s + t_i},$$

therefore $g(s) < 0$. Thus the unique solution $\hat{\beta}$ is such that $s > \hat{\beta} > r$. Having established that the unique solution of $g(x) = 0$ exists one checks that it is indeed the maximum likelihood solution. Since by substituting (A.4) into (A.2) gives

$$\frac{1}{n} \frac{\partial L}{\partial \beta} = \frac{(r - \beta)}{rs + \beta^2 - 2\beta r} + \frac{1}{K(\beta)},$$

it is sufficient to check that $(\frac{\partial L}{\partial \beta})_{\beta=r} > 0$. By substituting into the above one finds the first is $1/K(r) > 0$ and the second is $-1/s + 1/K(s) < 0$. Q.E.D.

Data Sets

Table 14: Aluminum fatigue data, lifetimes in cycles $\times 10^{-3}$ [8].

Data Set 1, maximum stress/cycle 31,000 psi							
70	90	96	97	99	100	103	104
104	105	107	108	108	108	109	109
112	112	113	114	114	114	116	119
120	120	120	121	121	123	124	124
124	124	124	128	128	129	129	130
130	130	131	131	131	131	131	132
132	132	133	134	134	134	134	134
136	136	137	138	138	138	139	139
141	141	142	142	142	142	142	142
144	144	145	146	148	148	149	151
151	152	155	156	157	157	157	157
158	159	162	163	163	164	166	166
168	170	174	196	212			

Data Set 2, maximum stress/cycle 21,000 psi							
370	706	716	746	785	797	844	855
858	886	886	930	960	988	990	1000
1010	1016	1018	1020	1055	1085	1102	1102
1108	1115	1120	1134	1140	1199	1200	1200
1203	1222	1235	1238	1252	1258	1262	1269
1270	1290	1293	1300	1310	1313	1315	1330
1355	1390	1416	1419	1420	1420	1450	1452
1475	1478	1481	1485	1502	1505	1513	1522
1522	1530	1540	1560	1567	1578	1594	1602
1604	1608	1630	1642	1674	1730	1750	1750
1763	1768	1781	1782	1792	1820	1868	1881
1890	1893	1895	1910	1923	1940	1945	2023
2100	2130	2215	2268	2440			

Table 15: Fatigue lifetimes for aluminum, lifetimes in cycles $\times 10^{-3}$ [15].

233	276	290	310	315	321
342	351	356	358	358	360
362	367	376	395	400	400
403	404	406	410	422	433
439	439	443	445	445	456
473	490	517	540	560	

Table 16: Fatigue lifetimes for concrete, lifetimes in cycles $\times 10^{-3}$ [38].

<u>.95</u>	<u>.90</u>	<u>.825</u>
37	201	1246
72	216	1258
74	226	1460
76	252	1492
83	257	2400
85	295	2410
105	311	2590
109	342	2903
120	356	3330
123	451	3590
143	457	3847
203	509	4110
206	540	4820
217	680	5560
257	1129	5598

Table 17: Insulating fluid failure data, lifetimes in minutes [48].

<u>.19</u>
.78
.96
1.31
2.78
3.16
4.15
4.67
4.85
6.50
7.35
8.01
8.27
12.06
31.75
32.52
33.91
36.71
<u>72.89</u>

Table 18: Fatigue lifetimes for steel, lifetimes in cycles $\times 10^{-3}$ [38].

Data Set 1 55,000psi	Data Set 2 61,000psi	Data Set 3 67,000psi
174	78	30
236	82	31
242	85	31
257	86	32
271	95	33
295	98	33
319	101	38
352	102	39
357	105	41
377	105	44
415	123	45
438	126	45
458	126	45
493	127	46
552	137	46
578	147	46
685	148	50
696	158	54
1390	162	64
1676	175	75

Algorithm Implementation

Program Hierarchy

SIMULATION STUDY

```

BSOBRE
  ARITHDB
  FDATE**
  FISHINFO
    SCORE1
    SCORE2
  JACOBI*
  LMSSTART
    DPHINV**
    DPHI**
    MLMSRAND
      MEDIAN
        SORT*
      MLS
        GAUSSJ*
      RANO*
  LNBLNK**
  MEANMEAN
    ARITHMETIC
    HARMONIC
  METH1
    ARITHMETIC
    G
      K
    GP
      K
      KP
      K
    HARMONIC
    MINMX**
  MIDPNT*
  QROMO*
    POLINT*
  SIMULATE
    GASDEV*
    RAN1*
  SORT*
  STEP
    COMPARE
    DENOM
      DENSITY
      WEIGHT
        SCORE1
        SCORE2
    DGJINV**
    MIDPNT*
    NUMER1
      DENSITY
      SCORE1
      WEIGHT (see above)
    NUMER2
      DENSITY
      SCORE2
      WEIGHT (see above)
    QROMO* (see above)
    SCORE1
    SCORE2
  TERM

```

APPLICATIONS TO REAL DATA

```

BSREAL
  DAUEXP
  DBUEXP
  DPHINV**
    DPHI**
  EXPQU
  FDATE**
  FISHINFO
    SCORE1
    SCORE2
  JACOBI*
  LMSSTART
    DPHINV** (see above)
    MLMSRAND
      MEDIAN
      SORT*
      MLS
      GAUSSJ*
      RANO*
  LNBLNK**
  MEANMEAN
    ARITHMETIC
    HARMONIC
  METH1
    ARITHMETIC
    G
      K
    GP
      K
      KP
      K
    HARMONIC
    MINMX**
  MIDPNT*
  QROMO*
    POLINT*
  STEP
    COMPARE
    DENOM
      DENSITY
      WEIGHT
        SCORE1
        SCORE2
    DGJINV**
    MIDPNT*
    NUMER1
      DENSITY
      SCORE1
      WEIGHT (see above)
    NUMER2
      DENSITY
      SCORE2
      WEIGHT (see above)
    QROMO* (see above)
    SCORE1
    SCORE2
  TERM

```

NOTES:

* : indicates that the program was extracted from Numerical Recipes in Fortran [44].

** : indicates that the program was extracted from G. Fenton's Mathematical Library [27].

Program Listings**INDEX**

arithdb.f	99
arithmetic.f	99
bsobre.f	100
bsreal.f	109
compare.f	115
daquexp.f	116
dbquexp.f	117
denom.f	118
density.f	118
expqu.f	119
fishinfo.f	119
G.f	121
Gp.f	121
harmonic.f	122
K.f	123
Kp.f	124
lmsstart.f	125
meanmean.f	126
median.f	126
meth1.f	127
mlmsrand.f	129
mls.f	130
numer1.f	133
numer2.f	133

score1.f	133
score2.f	134
simulate.f	135
step.f	135
stepre.f	143
term.f	152
weight.f	152

arithdb.f

```

c *****
c *
c *          function arithdb
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, TUNS, October 1996
c
c PURPOSE Calculate the arithmetic mean of an array of data
c
c      n    size of the array
c
c      p    array of data (Stored in Double Precision)
c -----
c      FUNCTION arithdb(n,p)
c      INTEGER n, i, no
c      PARAMETER (no=1000)
c      REAL*8 p(no), arithdb, zero
c      DATA zero/0.d0/
c
c      *Initialize
c      arithdb = zero
c
c      *Loop through data values
c      do 10 i=1, n
c          arithdb = arithdb + p(i)
c 10      continue
c
c      arithdb=arithdb/dbl(n)
c
c      return
c      END

```

arithmetic.f

```

c *****
c *
c *          function arithmetic
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, TUNS, October 1996
c
c PURPOSE Calculate the arithmetic mean of an array of data

```

```

c
c      n      size of the array
c
c      p      array of data (Stored in Single Precision)
c-----

```

```

      FUNCTION arithmetic(n,p)
      INTEGER n, i, no
      PARAMETER (no=1000)
      REAL p(no)
      REAL*8 arithmetic, zero
      DATA zero/0.d0/

c      *Initialize
      arithmetic = zero

c      *Loop through data values
      do 10 i=1, n
         arithmetic = arithmetic + dbble(p(i))
10    continue

      arithmetic=arithmetic/dbble(n)

      return
      END

```

bsobre.f

```

c *****
c *
c *                               Main bsobre
c *
c *****
c Single Precision Version 2.0
c Written by Joanna E. Mills, TUNS, October 1996
c
c PURPOSE  Compute the MLE and OBRE of the parameters of a
c           Birnbaum-Saunders distribution for a specified number
c           of simulations. Various summary statistics are reported.
c
c           The MLEs are obtained via iterative computing procedures
c           with the meanmean estimate as the starting value.
c
c           The OBREs are obtained using Victoria-Feser and Ronchetti's
c           algorithm [53] with LMS estimates as it's starting values.
c
c The program is executed by typing
c

```

```

c      Est datafile
c
c      where 'datafile' is the name of the input data file. An output file will
c      be created and will have the following name;
c
c      *.stats      will contain all intermediate and summary results
c
c      where '*' is the basename of the input file (note that no check is
c      performed to ensure that files do not exist prior to creating them,
c      so to preserve existing files, make sure that the data file basenames
c      are unique).
c
c      DATAFILE FORMAT
c      sim  [Desired number of simulations]
c      n    [Desired Number of Data Points]
c      ep   [Prescribed Epsilon (Stopping Criteria)]
c      max  [maximum number of iterations to be performed]
c      alph [Actual value of alpha]
c      beta [Actual value of beta]
c      c    [Desired bound on the influence function]
c
c      NO  is the maximum number of observations that can be simulated (100)
c      NS  is the maximum number of simulations that can be performed (1000)
c      NP  is the number of parameters involved in the model (2)
c-----

```

```

program bsobre
integer no, ns, np, n, max, sim, yseed, iouts, lp, iin, jd
integer narg, iargc, ld, lnblnk, i, j, k, nrot, failpdc
integer count
parameter ( no = 100)
parameter ( ns = 1000)
parameter ( np = 2)
logical t, f, header, found
character*80 DATFM, STATS, progrnm*64, rdate*24, usage*80
character*80 prog2*64
real x(no), zero, ep, alph, beta, alphimse, alph2mse
real betaimse, beta2mse
real*8 bias11(ns), bias12(ns), bias21(ns), bias22(ns)
real*8 mb11, mb12, dum, lm
real*8 mb21, mb22
real*8 Binit, meanmean, ei(np), ev(np,np), d1(np), det, temp
real*8 avals1(ns), bvals1(ns), avals2(ns), bvals2(ns), limit
real*8 we(no), avve(no), d2(np)
real*8 malph1, mbeta1, malph2, mbeta2, sdobre(np), limpr
real*8 arithdb, fin(np), sdmle(np), eff, sdve(no), eff1, eff2
real*8 avec(np), fish(np,np), amat(np,np), c, term
common/BOUND/c
common/DAT/x,n
external midpnt
data t/.true./, f/.false./ zero/0.d0/

```

```

c  *** Program name and version ***
      data progm/'This is BSOBRE(Version 1.0): '/
      data usage/'Usage: Est datafile'/
      data prog2/'End of BSOBRE: '/

c  *** Formatting statements ***
      1  format(a,a,a)
      2  format(i4)
      3  format(a,i4)
      4  format(a,i3,a,f15.9)
      5  format(a,f15.9)
      6  format(f7.2)
      7  format(f6.5)
      8  format(a,f11.5)
      9  format(f8.5)

c  *** Unit numbers used ***
      iin  = 7
      iouts = 9

c  *** Set defaults ***
      header = t
      lp      = lnblnk(progm) + 1
      call fdate(rdate)
      narg    = iargc()
      if( narg .lt. 1 ) then
         write(6,1) usage
         stop
      endif

c  *** Get data file name ***
      call getarg( 1, DATFW)
      ld = lnblnk(DATFW)

c  *** Print introduction ***
      if( header ) then
         write(6,1) progm(1:lp), rdate
         header = f
      endif

c  *** It's a data file name, check it ***
      inquire( file = DATFW, exist = found )
      if( .not. found ) then
         write(6,1)'Data file: ',DATFW(1:ld),' not found!'
         stop
      endif

c  *** Find basename (X...XX.dat) ***
      do 15 jd = ld, 1, -1
         if( DATFW(jd:jd) .eq. '.' ) go to 20
15      continue

```

```

        jd = ld + 1
20      jd = jd - 1
        STATS = DATFW(1:jd) //' .stats'
        jd = jd + 1

open(iin, file = DATFW, status = 'UNKNOWN' )
open(iouts, file = STATS, status = 'UNKNOWN' )

c***** Main Body of Program *****
c*****

c***** Read data file *****

c  *** Number of Simulations (sim) ***
      read(iin,2) sim
      write(iouts,3)'Number of simulations: ',sim

c  *** Number of Observations (n) ***
      read(iin,2) n
      write(iouts,3)'Number of observations: ',n

c  *** Prescribed Epsilon ***
      read(iin,7) ep
      write(iouts,8)'Prescribed Epsilon: ',ep

c  *** Maximum Number of Iterations ***
      read(iin,2) max
      write(iouts,3)'Maximum number of iterations: ',max

c  *** Actual Value of Alpha ***
      read(iin,9) alph
      write(iouts,8)'Actual Value of Alpha: ',alph

c  *** Actual Value of Beta ***
      read(iin,6) beta
      write(iouts,5)'Actual Value of Beta: ',beta

c                                     *** Bound on the IF (c) ***
      read(iin,6) c
      write(iouts,5)'Bound on the IF (c): ',c

c*****Initializations*****

c                                     *** Initialize random number generator ***
      yseed = -4

c                                     *** Initialize summary statistics ***
      do 30 i =1, sim
        avals1(i)=0.d0
        bvals1(i)=0.d0

```

```

        avals2(i)=0.d0
        bvals2(i)=0.d0
30  continue

c                                     *** Initialize mean standard deviations***
      do 31 i=1, np
        sdobre(i)=0.d0
        sdmle(i)=0.d0
31  continue

c                                     *** Initialize mean weights***
      do 32 i=1, n
        avwe(i)=0.d0
        sdwe(i)=0.d0
32  continue

c                                     *** Count of non-positive definite
c                                     matrices requiring another simulation
c                                     performed ***
      failpdc=0

c*****
c*****
c                                     *** Begin simulations ***  => (Start loop)

      do 90 i=1, sim
        write(iouts,3)'Simulation:',i

c                                     *** Simulate Observations (x) ***
34  call simulate(x,n,alph,beta,yseed)

c                                     *** Starting value for Method 1 ***
      Binit=meanmean(n,x)

c                                     *** Parameter Estimation via Method 1 ***
      call meth1(Binit,ep,max,avals1(i),bvals1(i),iouts)

c                                     *** Parameter Estimation via LMS ***
      call lmsstart(n,x,avals2(i),bvals2(i))

c***** Begin OBRE algorithm *****
c-----
c LMS estimates are the starting values for this OBRE algorithm.
c-----

c                                     *** Calculation of initial values for
c                                     avec and amat ***
      do 35 j =1, np
        avec(j)=0.d0
35  continue

```

```

c          *** Calculation of the fisher information
c          matrix ***
c      call fishinfo(avals2(i),bvals2(i),x,n,fish,np)

c          *** Calculation of the eigenvalues and
c          eigenvectors of the fisher information
c          matrix ***
c      call jacobi(fish,np,np,ei,ev,nrot)

c          *** Check to make sure matrix is positive
c          definite ***

c      if(ei(1) .lt. zero .or. ei(2) .lt. zero) then
c          failpdc= failpdc+1
c          write(iouts,3)'Matrix not p.d. - redoing simulation'
c          goto 34
c      endif

c          *** Take the reciprocal of the squareroot
c          of each eigenvalue ***
c      do 38 j=1,np
c          ei(j)=1.d0/dsqrt(ei(j))
38      continue

c          *** Obtain the inverse of the square-root
c          of the fisher information matrix
c          via the spectral decomposition [40] ***

c      fish(1,1)=(ei(1)*ev(1,1)**2)+(ei(2)*ev(1,2)**2)
c      fish(2,2)=(ei(1)*ev(2,1)**2)+(ei(2)*ev(2,2)**2)
c      fish(1,2)=(ei(1)*ev(1,1)*ev(2,1))+(ei(2)*ev(1,2)*ev(2,2))
c      fish(2,1)=(ei(1)*ev(2,1)*ev(1,1))+(ei(2)*ev(2,2)*ev(1,2))

c          *** Transpose to arrive at amat ***
c      do 45 j =1, np
c          do 40 k=1, np
c              amat(j,k)=fish(k,j)
40      continue
45      continue

c          *** Steps 2, 3, and 4 of algorithm ***
c      call step(n,x,avals2(i),bvals2(i),np,avec,amat,iouts,fin,we,d1)
c      *****End OBRE algorithm*****
c          *** Store final values in estimate
c          array ***
c      avals2(i)=fin(1)
c      bvals2(i)=fin(2)

c          *** Calculate mean weights ***
c      do 50 j=1, n

```



```

      avwe(j)=avwe(j)+we(j)
      sdwe(j)=sdwe(j)+we(j)**2
50  continue

c                                     *** Calculate variances of OBRE estimates ***
      do 52 j=1,np
        sdobre(j)=sdobre(j)+d1(j)
52  continue

c                                     *** Calculate variances of MLE
c                                     *** estimates [11] ***
      d2(1)=avals1(i)**2/(2.d0*dble(n))

c*Ensure integrating over appropriate range*
      dum=-.1d0
      lm=-1.d0
      count=0
53  dum=dum+.1d0
      if(abs(term(dum)).le.0.000000000000001) then
        count=count+1
      else
        goto 53
      endif
      if(count.ge.3) then
        lm=dum
        goto 54
      else
        goto 53
      endif
54  det=0.d0
      limit=0.d0
      do 55 j=1, 2*(idint(lm))+1
        limpr=limit
        limit=limit+0.5d0
        call qromo(term,limpr,limit,temp,midpnt)
        det=det+temp
55  continue

      d2(2)=bvals1(i)**2/(dble(n)*(0.25d0+avals1(i)**(-2)+2.d0*det))

      do 57 j=1,np
        sdmle(j)=sdmle(j)+d2(j)
57  continue

c                                     *** Calculate the bias of the ***
c                                     *** various estimates ***
c *OBRE*
      bias11(i)=avals2(i)-dble(alph)
      bias12(i)=bvals2(i)-dble(beta)

c *MLE*
```

```

      bias21(i)=avals1(i)-dble(alph)
      bias22(i)=bvals1(i)-dble(beta)

c          *** End simulations *** => (End loop)
90  continue
c*****
c*****

c          *** Summary Statistics ***

      alph1mse=0.0
      alph2mse=0.0
      beta1mse=0.0
      beta2mse=0.0

      malph1=arithdb(sim,avals2)
      malph2=arithdb(sim,avals1)
      mbeta1=arithdb(sim,bvals2)
      mbeta2=arithdb(sim,bvals1)

      mb11=arithdb(sim,bias11)
      mb12=arithdb(sim,bias12)
      mb21=arithdb(sim,bias21)
      mb22=arithdb(sim,bias22)

      do 95 i=1, sim
        alph1mse=alph1mse+sngl(bias11(i))**2
        alph2mse=alph2mse+sngl(bias21(i))**2
        beta1mse=beta1mse+sngl(bias12(i))**2
        beta2mse=beta2mse+sngl(bias22(i))**2
95  continue

      do 99 i=1, n
        avve(i)=avve(i)/dble(sim)
        sdwe(i)=dsqrt(sdwe(i)/dble(sim)-avve(i)**2)
99  continue

      alph1mse=sqrt(alph1mse/float(sim))
      alph2mse=sqrt(alph2mse/float(sim))
      beta1mse=sqrt(beta1mse/float(sim))
      beta2mse=sqrt(beta2mse/float(sim))

      do 100 i=1,np
        sdobre(i)=sdobre(i)/dble(sim)
        sdmle(i)=sdmle(i)/dble(sim)
100  continue

c  Calculate relative efficiency
      eff=100.d0*(sdmle(1)+sdmle(2))
      eff=eff/(sdobre(1)+sdobre(2))
      eff1=100.d0*(sdmle(1)/sdobre(1))
      eff2=100.d0*(sdmle(2)/sdobre(2))

```

```

do 105 i=1, np
  sdmls(i)=dsqrt(sdmls(i))
  sdobre(i)=dsqrt(sdobre(i))
105 continue

write(iouts,3)'SUMMARY STATISTICS'
write(iouts,3)'OBRE results:'
write(iouts,3)'-----'
write(iouts,8) 'ALPHA:',malph1
write(iouts,5) 'rmse: ',alph1mse
write(iouts,5) 'asyp. s.d.:',sdobre(1)
write(iouts,8) 'bias:',mb11
write(iouts,8) 'BETA:',mbeta1
write(iouts,5) 'rmse: ',betaimse
write(iouts,5) 'asyp. s.d.:',sdobre(2)
write(iouts,8) 'bias:',mb12
write(iouts,3)'MLE results:'
write(iouts,3)'-----'
write(iouts,8) 'ALPHA:',malph2
write(iouts,5) 'rmse: ',alph2mse
write(iouts,5) 'asyp. s.d.:',sdmls(1)
write(iouts,8) 'bias:',mb21
write(iouts,8) 'BETA:',mbeta2
write(iouts,5) 'rmse: ',beta2mse
write(iouts,5) 'asyp. s.d.:',sdmls(2)
write(iouts,8) 'bias:',mb22
write(iouts,3)
write(iouts,3)'RELATIVE EFFICIENCY'
write(iouts,3) '(OBRE relative to MLE)'
write(iouts,3) 'Alpha:'
write(iouts,6) eff1
write(iouts,3) 'Beta:'
write(iouts,6) eff2
write(iouts,3) 'Overall [trace]:'
write(iouts,6) eff
write(iouts,3)

write(iouts,3)'AVERAGE WEIGHTS'
do 110 i=1,n
  if(avve(i).ne.1.d0) then
    write(iouts,4)'Observation ',i,' -weight=',avve(i)
    write(iouts,4)'Observation ',i,' -s.d. =',sdwe(i)
  endif
110 continue

call fdate(rdate)
write(6,1) prog2(1:lp), rdate
stop
end

```

bsreal.f

```

c *****
c *
c *           Main bsreal
c *
c *****
c Single Precision Version 2.0
c Written by Joanna E. Mills, TUMS, Thu Dec 05 1996
c
c PURPOSE Compute the MLEs and OBREs of the parameters of a
c          Birnbaum-Saunders distribution for a real data set.
c
c          The MLEs are obtained via iterative computing procedures
c          with the meanmean estimate as the starting value.
c
c          The OBREs are obtained using Victoria-Feser and Ronchetti's
c          algorithm [53] with LMS estimates as it's starting values.
c
c The program is executed by typing
c
c      Est datafile
c
c where 'datafile' is the name of the input data file. An output file will
c be created and will have the following name;
c
c      *.stats      will contain all intermediate and summary results
c
c where '*' is the basename of the input file (note that no check is
c performed to ensure that files do not exist prior to creating them,
c so to preserve existing files, make sure that the data file basenames
c are unique).
c
c DATAFILE FORMAT
c   n   [Number of Data Points]
c   ep   [Prescribed Epsilon (Stopping Criteria)]
c   max  [maximum number of iterations to be performed]
c   c    [Desired bound on the influence function]
c   x1   [Data Points]
c   x2
c   .
c   .
c   .
c   xn
c
c   NO   is the maximum number of observations that can be analyzed (200)
c   NP   is the number of parameters involved in the model (2)
c -----
c      program bsreal
c      integer no, ns, np, n, max, iouts, lp, iin, jd

```

```

integer narg, iargc, ld, lnblnk, i, j, k, nrot
integer count
parameter ( no = 200)
parameter ( ns = 1000)
parameter ( np = 2)
logical t, f, header, found
character*80 DATFN, STATS, progm*64, rdate*24, usage*80
character*80 prog2*64
real x(no), zero, ep
real*8 dum, lm, interm, expqu, quant1, quant2, daquexp
real*8 dbquexp, co, dera, derb
real*8 Binit, meanmean, ei(np), ev(np,np), d1(np), det
real*8 avals1(ns), bvals1(ns), avals2(ns), bvals2(ns)
real*8 we(no), d2(np), par(np)
real*8 dphinv, sdqut1, sdqut2
real*8 fin(np)
real*8 avec(np), fish(np,np), amat(np,np), c, term
common/BOUND/c
common/PARAMS/par
common/DAT/x,n
external midpnt
data t/.true./, f/.false./ zero/0.d0/

c *** Program name and version ***
data progm/'This is BSREAL(Version 1.0): '/
data usage/'Usage: Est datafile'/
data prog2/'BSREAL completed: '/

c *** Formatting statements ***
1 format(a,a,a)
2 format(i4)
3 format(a,i4)
4 format(a,i3,a,f15.9)
5 format(a,f15.9)
6 format(f7.2)
7 format(f6.5)
8 format(a,f11.5)
9 format(f8.5)
10 format(f7.0)

c *** Unit numbers used ***
iin = 7
iouts = 9

c *** Set defaults ***
header = t
lp = lnblnk(progm) + 1
call fdate(rdate)
narg = iargc()
if( narg .lt. 1 ) then
  write(6,1) usage
  stop
endif

```

```

c  *** Get data file name ***
    call getarg( 1, DATFN)
    ld = lnblnk(DATFN)

c  *** Print introduction ***
    if( header ) then
        write(6,1) progrm(1:lp), rdate
        header = f
    endif

c  *** It's a data file name, check it ***
    inquire( file = DATFN, exist = found )
    if( .not. found ) then
        write(6,1)'Data file: ',DATFN(1:ld),' not found!'
        stop
    endif

c  *** Find basename (X...XX.dat) ***
    do 15 jd = ld, 1, -1
        if( DATFN(jd:jd) .eq. '.' ) go to 20
15    continue
        jd = ld + 1
20    jd = jd - 1
        STATS = DATFN(1:jd)//'.stats'
        jd = jd + 1

    open(iin,  file = DATFN, status = 'UNKNOWN' )
    open(iouts, file = STATS, status = 'UNKNOWN' )

c***** Main Body of Program *****
c*****

c***** Read data file *****

c  *** Number of Observations (n) ***
    read(iin,2) n
    write(iouts,3)'Number of observations: ',n

c  *** Prescribed Epsilon ***
    read(iin,7) ep
    write(iouts,8)'Prescribed Epsilon: ',ep

c  *** Maximum Number of Iterations ***
    read(iin,2) max
    write(iouts,3)'Maximum number of iterations: ',max

c                                     *** Bound on the IF (c) ***
    read(iin,6) c
    write(iouts,5)'Bound on the IF (c): ',c

```

```

      do 25 i = 1, n
        read(iin,10) x(i)
        write(iouts,10) x(i)
25      continue

c          *** Count of non-positive definite
c          matrices requiring another simulation
c          performed

c***** Initializations *****
c*****

c***** Calculate the initial value of the BS parameter vector *****

c          *** Starting values for Methods 1 and 2***
      Binit=meanmean(n,x)

c          *** Parameter Estimation via Method 1 ***
      call meth1(Binit,ep,max,avals1(1),bvals1(1),iouts)

c          *** Parameter Estimation via LMS      ***

      call lmsstart(n,x,avals2(1),bvals2(1))

c***** Begin OBRE algorithm *****
c-----
c LMS estimates are the starting values for this OBRE algorithm.
c-----

c          *** Calculation of initial values for
c          avec and amat ***
      do 35 j =1, np
        avec(j)=0.d0
35      continue

c          *** Calculation of the fisher information
c          matrix ***

      call fishinfo(avals2(1),bvals2(1),x,n,fish,np)

c          *** Calculation of the eigenvalues and
c          eigenvectors of the fisher information
c          matrix ***
      call jacobi(fish,np,np,ei,ev,nrot)

c          *** Check to make sure matrix is positive
c          definite

```

```

    if(ei(1) .lt. zero .or. ei(2) .lt. zero) then
        write(iouts,3)'Matrix not p.d.'
        stop
    endif

c                                     *** Take the reciprocal of the squareroot
c                                     of each eigenvalue
    do 38 j=1,np
        ei(j)=1.d0/dsqrt(ei(j))
38    continue

c                                     *** Obtain the inverse of the square-root
c                                     of the fisher information matrix
c                                     via the spectral decomposition
c                                     p-53 Applied Multivariate Statistical
c                                     Analysis [41]
    fish(1,1)=(ei(1)*ev(1,1)**2)+(ei(2)*ev(1,2)**2)
    fish(2,2)=(ei(1)*ev(2,1)**2)+(ei(2)*ev(2,2)**2)
    fish(1,2)=(ei(1)*ev(1,1)*ev(2,1))+(ei(2)*ev(1,2)*ev(2,2))
    fish(2,1)=(ei(1)*ev(2,1)*ev(1,1))+(ei(2)*ev(2,2)*ev(1,2))

c                                     *** Transpose to arrive at amat ***
    do 45 j =1, np
        do 40 k=1, np
            amat(j,k)=fish(k,j)
40    continue
45    continue

c                                     *** Steps 2, 3, and 4 of algorithm ***
    call stepre(n,x,avals2(1),bvals2(1),np,avec,amat,iouts,fin,we,d1,co)

*****End OBRE algorithm*****
c                                     *** Store final values in estimate
c                                     array ***
    avals2(1)=fin(1)
    bvals2(1)=fin(2)

c*****
c*****

*****Calculate MLE Covariance Structure*****

c                                     *** Calculate mean standard deviations ***
c                                     *** of MLE estimates (obtained via ***
c                                     *** Method 2), Chang and Tang [11]. ***
    d2(1)=avals1(1)**2/(2.d0*dble(n))
c*Ensure integrating over appropriate range*
    dum=-.1d0
    lm=-1.d0
    count=0

```



```

    par(1)=avals1(1)
    par(2)=avals1(2)
53  dum=dum+.1d0
    print *, 'dum',dum
    if(abs(term(dnm)).le.0.00000000000001) then
        count=count+1
    else
        goto 53
    endif
    if(count.ge.3) then
        lm=dnm
        goto 54
    else
        goto 53
    endif
54  call qromo(term,0.d0,lm,det,midpnt)
    print *, 'lm',lm
    print *, 'det',det

    d2(2)=bvals1(1)**2/(dble(n)*(0.25d0+avals1(1)**(-2)+2.d0*det))

c*****Calculate Quantile Estimates*****
    interm=dphinu(.001d0)
    quant1=expqu(interm,avals1(1),bvals1(1))
    dera=daquexp(avals1(1),bvals1(1),interm)
    derb=dbquexp(avals1(1),bvals1(1),interm)
    sdqut1=dsqrt(dera**2*d2(1) + derb**2*d2(2))
    d2(1)=dsqrt(d2(1))
    d2(2)=dsqrt(d2(2))

    quant2=expqu(interm,avals2(1),bvals2(1))
    dera=daquexp(avals2(1),bvals2(1),interm)
    derb=dbquexp(avals2(1),bvals2(1),interm)
    sdqut2=dsqrt(dera**2*d1(1) +derb**2*d1(2) +2.d0*dera*derb*co)
    d1(1)=dsqrt(d1(1))
    d1(2)=dsqrt(d1(2))
c*****SUMMARY*****
    write(iouts,3)'SUMMARY STATISTICS'
    write(iouts,3)'OBRE results:'
    write(iouts,3)'-----'
    write(iouts,8) 'ALPHA:',avals2(1)
    write(iouts,8) 'sd:',d1(1)
    write(iouts,8) 'BETA:',bvals2(1)
    write(iouts,8) 'sd:',d1(2)
    write(iouts,8) 'p=.001 Quantile:',quant2
    write(iouts,8) 'sd:',sdqut2
    write(iouts,3)
    write(iouts,3)
    write(iouts,3)'MLE results:'
    write(iouts,3)'-----'
    write(iouts,8) 'ALPHA:',avals1(1)

```

```

write(iouts,8) 'sd:',d2(1)
write(iouts,8) 'BETA:',bvals1(1)
write(iouts,8) 'sd:',d2(2)
write(iouts,8) 'p=.001 Quantile:',quant1
write(iouts,8) 'sd:',sdqut1
write(iouts,3)
write(iouts,3)
write(iouts,3)'WEIGHTS'
do 110 i=1,n
if(we(i).ne.1.d0) then
    write(iouts,4)'Observation ',i,' -weight=',we(i)
endif
110 continue

c *** Print conclusion ***
call fdate(rdate)
write(6,1) prog2(1:lp), rdate

stop
end

```

compare.f

```

c *****
c *
c *
c *
c *
c *****
c Written by Joanna E. Mills, TUMS, September 1996
c
c PURPOSE: Compares new values of a and A with previously calculated
c           values to see if we have converged to a solution
c
c           p   dimension of a (px1) and A (pxp)
c
c           av   current a vector
c
c           am   current A matrix
c
c           s   logical value indicating if solved satisfactorily
c
c           avo   a vector from previous step
c
c           amo   A matrix from previous step
c -----
c subroutine compare(p,av,avo,am,amo,s)
integer p

```

```

      real*8 av(*), avo(*), am(p,p), amo(p,p), df, rdf, zero
      logical s
      data zero/0.d0/
7      format(a)
8      format(a,f7.3)
9      format(a,f7.3,a,f7.3)

c      Maximum relative difference is 5%
      rdf=.05d0

c      Maximum absolute difference used in cases of division by zero
      df=.1d0

      s=.true.
c      *Check a vector*
      if(avo(1).ne.zero .and. avo(2).ne.zero) then
c      No division by zero expected
          if(dabs((av(1)-avo(1))/avo(1)).gt.rdf .or.
>          dabs((av(2)-avo(2))/avo(2)).gt.rdf) s=.false.
      else
          if(dabs(av(1)-avo(1)).gt.df .or.
>          dabs(av(2)-avo(2)).gt.df) s=.false.
      endif

c      *Check A matrix*
      if(amo(1,1).ne.zero.and.amo(1,2).ne.zero.and.amo(2,2).ne. zero)
> then
          if(dabs((am(1,1)-amo(1,1))/amo(1,1)).gt.rdf .or.
>          dabs((am(1,2)-amo(1,2))/amo(1,2)).gt.rdf .or.
>          dabs((am(2,2)-amo(2,2))/amo(2,2)).gt.rdf ) s=.false.

      else
          if(dabs(am(1,1)-amo(1,1)).gt.df .or.
>          dabs(am(1,2)-amo(1,2)).gt.df .or.
>          dabs(am(2,2)-amo(2,2)).gt.df) s=.false.
      endif

      return
      end

```

daquexp.f

```

c *****
c *
c *                               function daquexp
c *

```

```

c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, February 1997
c
c PURPOSE Calculate the derivative with respect to alpha of the quantile
c
c
c      c value of phinv(p) where p is the desired quantile value
c para1 estimate of alpha
c para2 estimate of beta
c -----
      real*8 function daquexp(para1,para2,c)
      real*8 para1, para2, c

      daquexp=4.d0*para1*c**2+2.d0*c*dsqrt(para1**2*c**2+4.d0)
      daquexp=daquexp+(2.d0*para1**2*c**3)*(para1**2*c**2+4.d0)**(-.5d0)
      daquexp=daquexp*para2/4.d0

      return
      end

```

dbquexp.f

```

c *****
c *
c *                               function dbquexp
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, February 1997
c
c PURPOSE Calculate the derivative with respect to beta of the quantile
c
c
c      c value of phinv(p) where p is the desired quantile value
c para1 estimate of alpha
c para2 estimate of beta
c -----
      real*8 function dbquexp(para1,para2,c)
      real*8 para1, para2, c

      dbquexp=0.5d0*para1**2*c**2+1.d0
      dbquexp=dbquexp+0.5d0*para1*c*dsqrt(para1**2*c**2+4.d0)

      return
      end

```

denom.f

```

c *****
c •
c • function denom •
c * *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, September 1996
c
c x real value at which the function is to be evaluated
c-----
      real*8 function denom(x)
      real*8 x, weight, density

      denom=weight(x)*density(x)

      return
      end

```

density.f

```

c *****
c •
c • function density •
c • *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, September 1996
c
c PURPOSE return the density for a given x corresponding to a
c          Birnbaum-Saunders distribution
c
c x real value at which the function is to be evaluated
c-----
      real*8 function density(x)
      real*8 p(2), x, inter
      real*8 pi
      common/PARAMS/p
      data pi/3.14159265d0/

```



```

c *                               subroutine fishinfo                               *
c *                               *                                                *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, TUNS, September 1996
c
c PURPOSE:  Calculates the fisher information matrix corresponding to
c           a vector of n observations from a Birnbaum Saunders
c           distribution
c
c           n   number of observations (INPUT)
c
c           y   vector containing the simulated observations
c
c           j   fisher information matrix
c
c           np  dimension of information matrix (np*np)
c
c           par1 estimated alpha value
c
c           par2 estimated beta value
c -----
c           subroutine fishinfo(par1,par2,y,n,j,np)
c           integer n, np, i, k
c           real y(*)
c           real*8 j(np,np), sco1, sco2, zero
c           real*8 par1,par2, para(2), score1, score2
c           common/PARAMS/para
c           data zero/0.d0/
c
7           format(a)
8           format(a,f7.3)
9           format(a,f7.3,a,f7.3)
c
c           para(1)=par1
c           para(2)=par2
c * initializations *
c           do 20 i=1, np
c               do 10 k=1, np
c                   j(i,k)=zero
10          continue
20          continue
c
c * Have replaced dH with its empirical distribution function.
c * This means replacing the integral with averages over the sample.
c           do 30 i = 1, n
c               sco1=score1(dble(y(i)))
c               sco2=score2(dble(y(i)))
c               j(1,1)=j(1,1)+sco1**2
c               j(1,2)=j(1,2)+sco1*sco2
c               j(2,2)=j(2,2)+sco2**2

```

```

30 continue

c * taking the average *
  j(1,1)=j(1,1)/dble(n)
  j(1,2)=j(1,2)/dble(n)
  j(2,1)=j(1,2)
  j(2,2)=j(2,2)/dble(n)
  return
end

```

G.f

```

c *****
c *
c *                               function G
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, TUHS, October 1996
c
c PURPOSE Calculate the function g(x):
c           g(x)=x**2 -x[2r+K(x)] + r[s+K(x)]
c           where K(x) is the harmonic mean function, s is the
c           arithmetic mean and r is the harmonic mean
c
c
c           x  real*8 value at which the function is to be evaluated
c-----
c
c FUNCTION G(x)
c REAL*8 G, r,s, x, K
c COMMON/MEANS/r,s
c
c G=x**2 - x*(2.d0*r+K(x)) + r*(s+K(x))
c
c return
c END

```

Gp.f

```

c *****
c *

```



```

c *                               function Gp                               *
c *                               *                                           *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, TUNS, October 1996
c
c PURPOSE Calculate the function  $g'(x)$ :
c            $g(x)=(x-r)*[1-K'(x)] +x -r -K(x)$ 
c           where  $K(x)$  is the harmonic mean function,  $K'(x)$  is the
c           derivative of the harmonic mean function and  $r$  is the harmonic
c           mean.
c
c
c           x    real*8 value at which the function is to be evaluated
c -----

```

```

FUNCTION Gp(x)
REAL*8 Gp, r,s, x, Kp, K
COMMON/MEANS/r,s

Gp = (x-r)*(1.d0-Kp(x)) +x -r -K(x)

return
END

```

harmonic.f

```

c *****
c *                               *                                           *
c *                               function harmonic                               *
c *                               *                                           *
c *****
c Single Precision Version 1.0
c Written by Joanna E. Mills, TUNS, October 1996
c
c PURPOSE Calculate the harmonic mean of an array of data
c
c           n    size of the array
c
c           p    array of data
c -----

```

```

FUNCTION harmonic(n,p)
INTEGER n, i, no
PARAMETER (no = 100)
REAL p(no)

```

```

      REAL*8 harmonic, zero
      DATA zero/0.d0/

c      *Initialize
      harmonic = zero

c      *Loop through data values
      do 10 i=1, n
         harmonic = harmonic + 1.d0/dbl(p(i))
10     continue

      harmonic=harmonic/dbl(n)

      harmonic=1.d0/harmonic

      return
      END

```

K.f

```

c *****
c *
c *                               function K
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, TUNS, October 1996
c
c PURPOSE Calculate the harmonic mean function corresponding to an
c          array of data for a given value of x
c
c          x  real*8 value at which the function is to be evaluated
c -----

      FUNCTION K(x)
      INTEGER no
      PARAMETER (no = 100)
      REAL p(no)
      REAL*8 x, K, zero
      INTEGER n, i
      COMMON/DAT/p,n
      DATA zero/0.d0/

c      *Initialize
      K = zero

```

```

c      *Loop through data values
      do 10 i=1, n
        K = K + 1.d0/(x+db1e(p(i)))
10     continue

      K=K/db1e(n)

      K=1.d0/K

      return
      END

```

Kp.f

```

c *****
c *
c *                               function Kp
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, TUMS, October 1996
c
c PURPOSE Calculate the derivative of the harmonic mean function
c          corresponding to an array of data for a given value of x
c
c          x    real*8 value at which the function is to be evaluated
c-----
c
      FUNCTION Kp(x)
      INTEGER n, i, no
      PARAMETER (no = 100)
      REAL p(no)
      REAL*8 x, K, zero, Kp
      COMMON/DAT/p,n
      DATA zero/0.d0/

c      *Initialize
      Kp = zero

c      *Loop through data values
      do 10 i=1, n
        Kp = Kp + 1.d0/(x+db1e(p(i)))**2
10     continue

      Kp=Kp/db1e(n)

      Kp=K(x)**2*Kp

```

```

return
END

```

lmsstart.f

```

c *****
c *
c *          subroutine lmsstart
c *
c *****
c Single Precision Version 1.0
c Written by Joanna E. Mills, TUMS, October 1996
c
c PURPOSE Estimate the parameters of a Birbaum-Saunders distribution
c          via the LMS - Chang Tang graphical procedure [12]
c
c          n number of data points
c
c          t array of data points
c
c          afin final alpha estimate
c
c          bfin final beta estimate
c -----
c          subroutine lmsstart(n, t, afin, bfin)
c          integer n, no, i, np, sd
c          parameter (no = 100)
c          parameter(np = 2)
c          real t(no), p(no,np), F(no), est(np)
c          real*8 afin, bfin, dphinv
c
1      format(a)
2      format(a,f15.9)
3      format(a,i3)
c
c          do 10 i=1, n
c             F(i)=(i-0.3)/(n+0.4)
c             p(i,1)=1.0
c             p(i,2)=sqrt(t(i))*sngl(dphinv(dble(F(i))))
10      continue
c
c          sd=1
c          call mlmsrand(p,t,n,np,est,sd)
c
c          bfin=dble(est(1))
c          afin=dble(est(2)/sqrt(est(1)))

```

```

return
end

```

meanmean.f

```

c *****
c *
c *          function meanmean
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, TUNS, October 1996
c
c PURPOSE  Calculates the geometric mean of the harmonic and arithmetic
c           means of an array of data
c
c           n    size of the array
c
c           p    array of data
c -----
c
c FUNCTION meanmean(n,p)
c INTEGER n, no
c PARAMETER (no = 100)
c REAL p(no)
c REAL*8 meanmean, arithmetic, harmonic
c
c meanmean=dsqrt(arithmetic(n,p)*harmonic(n,p))
c
c return
c END

```

median.f

```

c *****
c *
c *          function median
c *
c *****
c Written by Joanna E. Mills, TUNS, May 1996
c

```

```

c  PURPOSE:  Calculate the median of a vector of data
c
c          n  number of data points (INPUT)
c
c          p  vector of data (INPUT)
c
c *****

      FUNCTION median(n,p)
      INTEGER ns
      PARAMETER ( ns = 100 )
      REAL median, sp(ns), p(ns), zero
      INTEGER n, len, i
      DATA zero/0.0/

      len=n
      do 10 i=1, len
        sp(i)=p(i)
10    continue

      call sort(len,sp)
      if(MOD(len,2) .eq. zero) then
        median=(sp(len/2)+sp(len/2+1))/2.0
      else
        median=sp((len+1)/2)
      endif

      return
      END

```

meth1.f

```

c *****
c *
c *          subroutine meth1
c *
c *****
c  Single Precision Version 1.0
c  Written by Joanna E. Mills, TUMS, Wed October 1996
c
c  PURPOSE  Estimate the parameters of a Birbaum-Saunders distribution
c           via the Newton iteration procedure
c
c  init    initial value of B parameter
c
c  eps     prescribed iteration termination point
c

```

```

c      itmax  maximum number of iterations to be performed
c
c      afin  final alpha value
c
c      bfin  final beta value
c
c      iout  file to which output is to be written
c-----
      subroutine meth1(init,eps,itmax,afin, bfin,iout)
      integer no, n, iout, itmax, i
      parameter (no = 100)
      real p(no), min, max, eps
      real*8 init, r, s, arithmetic, harmonic, bold, bnew, G, Gp, anew
      real*8 afin, bfin
      COMMON/DAT/p,n
      COMMON/MEANS/r,s

1      format(a)
2      format(a,f15.9)
3      format(a,i3)

c      *Calculate means
      s = arithmetic(n,p)
      r = harmonic(n,p)

c      *Find the smallest data value
      call minmx(p,n,max,min)
c      *Test to see if data meet required conditions for this method
c      *NOTE: If this test fails it implies that convergence may not
c              occur since it cannot be validated theoretically

      if(2.d0*s .ge. (3.d0*r + dble(min))) then
c          write(iout,1) 'Method 1 may not converge - Condition not met'
      endif

c      *Begin Newton iteration procedure
      bold = init

      do 10 i=1,itmax
          bnew = bold - G(bold)/Gp(bold)
          if(dabs(bnew-bold) .lt. dble(eps)) then
              write(iout,1)'Method 1 converged to a solution'
              write(iout,3)'# Of iterations:',i
              write(iout,2)'B=',bnew
              bfin=bnew
              anew=dsqrt((s/bnew)+(bnew/r)-2)
c              write(iout,2)'A=',anew
              afin=anew
              goto 100
          endif
          bold=bnew

```

```

10    continue

100   return
      end

```

mlmsrand.f

```

c *****
c *
c *                               subroutine mlmsrand                               *
c *
c *****
c Single Precision Version 1.0
c Written by Joanna E. Mills, TUMS, May 1996
c
c PURPOSE  fit the least median of squares line to data
c
c
c      x   real matrix of size nxp
c
c      y   real vector of length n containing the observations
c
c      n   number of data points. (input)
c
c      p   number of independent variables + 1
c
c lmest   vector of estimated LMS regression coefficients (p)
c
c iseed   random number generator seed
c -----
c      subroutine mlmsrand(x,y,n,p,lmest,sd)
c      integer ns, np
c      parameter( ns = 100)
c      parameter( np = 2)
c      integer n, j, i, sd, index(np), p, k, m , l
c      real min, x(ns,np), y(ns), pars(np), yfit(ns), median, lmest(np)
c      real subx(np,np), suby(np), ran0, zero, est(ns)
c      data zero/0.0/
c
c                               Absurd initial minimum value
c      min = 9999999.9
c
c
c                               Loop through random subsamples of size p
c      do 100 i = 1, 400

```



```

c                                Get p points for the subsample
      do 20 k = 1, p
1      index(k) = nint(n*ran0(sd))
        if (index(k) .eq. 0) goto 1
        if (k .eq. 1) goto 20
        do 15 j = 1, k-1
          if (index(j) .eq. index(k)) goto 1
15      continue
20     continue

c      Place points in matrix x, vector y
      do 30 j = 1, p
        suby(j) = y(index(j))
        do 25 m = 1, p
          subx(j,m) = x(index(j),m)
25      continue
30     continue

c                                Calculate the LMS estimates for this sample
c                                ***Use LS routine**
      call mls(subx,suby,p,p,pars)

      do 45 k = 1, n
        yfit(k) = zero
        do 40 j = 1, p
          yfit(k) = yfit(k) + x(k,j)*pars(j)
40      continue
        est(k)=(y(k)-yfit(k))**2
45     continue

      if(median(n,est) .lt. min) then
        do 50 l = 1, p
          lms est(l) = pars(l)
50      continue
        min = median(n,est)
      endif
100    continue

      return
      end

```

mls.f

```

c *****
c *

```

```

c *                               subroutine mls                               *
c *                               *
c *****
c Single Precision Version 1.0
c Written by Joanna E. Mills, TUMS, May 1996
c
c PURPOSE  fit the least squares line to data
c
c   x   real matrix of size nxp
c
c   y   real vector of length n containing the observations
c
c   n   number of data points. (input)
c
c   p   number of independent variables + 1
c
c beta  vector of regression coefficients
c -----
c      subroutine mls(x,y,n,p,beta)
c      integer ns, np
c      parameter (ns =2)
c      parameter (np =2)
c      real x(np,np), y(ns), beta(np), t(np,ns)
c      real zero, tx(np,np), ty(np,1)
c      integer n, p, i , j, k
c      data zero/0.0/
c Transpose the matrix X
c      do 20 i = 1, n
c          do 10 k = 1, p
c              t(k,i) = x(i,k)
c          10 continue
c      20 continue
c Calculate X'X
c
c      do 60 i= 1, p
c          do 50 j = 1, p
c              tx(i,j) = zero
c              do 40 k = 1, n
c                  tx(i,j) = tx(i,j) + t(i,k)*x(k,j)
c              40 continue
c          50 continue
c      60 continue
c
c      Calculate X'Y
c      do 90 i= 1, p
c          ty(i,1) = zero
c          do 80 j = 1, n
c              ty(i,1) = ty(i,1) + t(i,j)*y(j)
c          80 continue

```

```
90  continue

c                                     Calculate betas!

    call gaussj(tx,p,np,ty,1,1)
c                                     NOTE - betas now ty, X'X now inverted
    do 100 i = 1,p
        beta(i) = ty(i,1)
100  continue

    return
end
```

numer1.f

```

c *****
c *
c *                      function numer1
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, September 1996
c
c      x  real value at which the function is to be evaluated
c-----
      real*8 function numer1(x)
      real*8 x, score1, weight, density

      numer1=score1(x)*weight(x)*density(x)
      return
      end

```

numer2.f

```

c *****
c *
c *                      function numer2
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, September 1996
c
c      x  real value at which the function is to be evaluated
c-----
      real*8 function numer2(x)
      real*8 x, score2, weight, density

      numer2=score2(x)*weight(x)*density(x)

      return
      end

```

score1.f

```

c *****

```

```

c *
c *
c *
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, September 1996
c
c PURPOSE to return the first element of the score function
c corresponding to a Birnbaum-Saunders distribution
c
c x real value at which the function is to be evaluated
c-----
c
c real*8 function score1(x)
c real*8 para(2), x
c common/PARAMS/para
c
c score1=-1.d0/para(1)
c score1=score1+1.d0/(para(1)**3)*(x/para(2)+para(2)/x-2.d0)
c return
c end

```

score2.f

```

C *****
C
C
C
C
C
C *****
C Double Precision Version 1.0
C Written by Joanna E. Mills, September 1996
C
C
C PURPOSE to return the second element of the score function
C          corresponding to a Birnbaum-Saunders distribution
C
C          x real value at which the function is to be evaluated
C-----
C
C      real*8 function score2(x)
C      real*8 para(2), x, mid
C      common/PARAMS/para
C
C      score2=1.d0/((2.d0*(x+para(2)))*(1.d0-(x/para(2))))
C      mid=x/((2.d0*para(1)**2*para(2)**2)*(1.d0-(x/para(2))**(-2))
C      score2=score2+mid
C      return
C      end

```

simulate.f

```

C *****
C *
C *
C *
C *
C *****
C Written by Joanna E. Mills, TUNS, October 16, 1996
C
C PURPOSE:  Simulates n observations from a Birnbaum-Saunders(alpha,beta)
C           distribution
C
C           n    desired number of observations (INPUT)
C
C           y    vector containing the simulated observations (OUTPUT)
C
C           alph  true value of alpha parameter
C
C           beta  true value of beta parameter
C
C           iseed seed for random number generator
C -----
C
C           subroutine simulate(y,n,alph,beta,iseed)
C           integer i, n, iseed
C           real y(*), gasdev, z
C           real alph, beta
C
C           do 10 i = 1, n
C             *generate a standard normal deviate
C             z=gasdev(iseed)
C             y(i)=1.0 + 0.5*alph**2*z**2+alph*z*sqrt(1.0+0.25*alph**2*z**2)
C             y(i)=beta*y(i)
10          continue
C
C           call sort(n,y)
C
C           return
C           end

```

step.f

```

C *****
C *
C *          subroutine step          *
C *
C *

```

```

c *****
c Written by Joanna E. Mills, TUNIS, October 1996
c
c PURPOSE: Performs Steps 2,3 and 4 of Victoria-Feser and Ronchetti's
c           algorithm [53]
c
c           n   number of observations
c
c           y   vector containing the simulated observations
c
c           ip1  initial alpha value of parameter vector
c
c           ip2  initial beta value of parameter vector
c
c           np   dimension of parameter vector
c
c           initav  initial value of a vector
c
c           initama  initial value of A matrix
c
c           iout   file to which output is to be written
c
c           final  vector of parameter estimates obtained (OUTPUT)
c
c           w      vector of weights corresponding to OBRE estimates (OUTPUT)
c
c           var     variances of OBRE estimates (OUTPUT)
c -----
c      subroutine step(n,y,ip1,ip2,np,initav,initama,iout,final,w,var)
c      integer iout, n, np, i, j, loop2, loop1, flag, f1,f2,f3,no, p
c      integer count1, count2, count3, indx(2), error, negcount
c      parameter(no=100)
c      parameter(p=2)
c      real y(*)
c      real*8 ip1, ip2, deltapar(p), avect(p), amatr(p,p), bound
c      real*8 w(no), dm(p), s1(no), s2(no), norm, m1(p,p), m2(p,p)
c      real*8 zero, dummy, denom, Q(p,p), M(p,p)
c      real*8 IN(p,p), V(p,p), var(p), score1, score2, scale(p)
c      real*8 numer1, numer2, bottom, top1, top2, del(p)
c      real*8 oldav(p), oldam(p,p), temp
c      real*8 initav(p), initama(p,p), lm1, lm2, lm3
c      real*8 limit, limpr
c      real*8 par(p), final(p), det, rdel1, rdel2, initinc
c      logical done
c      common/PARAMS/par
c      common/BOUND/bound
c      common/LAGRANGE/avect,amatr
c      external denom, numer1, numer2, midpnt, dgjinv
c      data zero/0.0d0/
1  format(a)

```

```

2      format(a,f7.3)
3      format(a,f15.9,a,f15.9)
4      format(a,i4)
5      format(f10.5,f15.13,f15.13,f15.13)

c      *****Initializations*****

c      *Initialize parameter values*
      par(1)=ip1
      par(2)=ip2

c      *Used in numerical integration*
      initinc=par(2)/10.d0

c      *Initialize a vector and A matrix*
      do 10 i=1,np
          avect(i)=initav(i)
          do 6 j= 1,np
              amatr(i,j)=initama(i,j)
          6      continue
      10  continue

      loop1=0
      loop2=0
c      *****Iteration Starting Point*****

c      **Store 'old' values of a and A**
      9      loop1=loop1+1
      do 15 i=1,np
          oldav(i)=avect(i)
          do 12 j= 1,np
              oldam(i,j)=amatr(i,j)
          12      continue
      15  continue

c      **Calculate the weights**

      do 30 i = 1, n
          norm=zero

          s1(i)=score1(dble(y(i)))
          s2(i)=score2(dble(y(i)))
c      **Calculate the Euclidean norm**
          do 20 j = 1, np
              dm(j)=amatr(j,1)*(s1(i)-avect(1))+amatr(j,2)*(s2(i)-avect(2))
              norm = norm + dm(j)**2
          20  continue
          norm=dsqrt(norm)
          w(i)=dmini(1.d0,bound/norm)
      30  continue

```



```

c    **Calculate M2 via averaging**
      do 40 i=1,np
        do 35 j=1,np
          m2(i,j)=zero
35      continue
40      continue
      do 50 i=1,n
        m2(1,1)=m2(1,1)+(s1(i)-avect(1))**2*(w(i)**2)
        m2(2,2)=m2(2,2)+(s2(i)-avect(2))**2*(w(i)**2)
        m2(2,1)=m2(2,1)+(s1(i)-avect(1))*(s2(i)-avect(2))*(w(i)**2)
50      continue

c    *taking the average*
      m2(1,1)=m2(1,1)/dble(n)
      m2(2,1)=m2(2,1)/dble(n)
      m2(1,2)=m2(2,1)
      m2(2,2)=m2(2,2)/dble(n)

c    **Calculate the inverse of m2**
      call dgjinv(m2,p,p,indx,scale,.true.,error)
      if(error.eq.-1) then
        write(iout,1)'Problem with M2 matrix'
        return
      endif

c    **Calculate new amatr**
      amatr(1,1)=dsqrt(m2(1,1))
c    * The result is upper triangular
      amatr(2,1)=zero
      amatr(1,2)=m2(2,1)/amatr(1,1)
      amatr(2,2)=dsqrt(m2(2,2)-amatr(1,2)**2)

c    **Calculate new avect**

c    *Compute limits of integration*
      dummy=0.d0
      lm1=-1.d0
      lm2=-1.d0
      lm3=-1.d0
      count1=0
      count2=0
      count3=0
      f1=0
      f2=0
      f3=0
      flag=0

55      if(flag.ne.-1) then
        dummy =dummy + initinc
        if(abs(denom(dummy)).le.0.0000000000001.and.f1.ne.-1) then

```

```

        count1=count1 + 1
    endif
    if(count1.ge.4.and.f1.ne.-1) then
        lm1 = dummy
        f1=-1
    endif
    if(abs( numer1(dummy)).le.0.00000000000001.and.f2.ne.-1) then
        count2=count2 + 1
    endif
    if(count2.ge.4.and.f2.ne.-1) then
        lm2 = dummy
        f2=-1
    endif
    if(abs( numer2(dummy)).le.0.00000000000001.and.f3.ne.-1) then
        count3=count3 + 1
    endif
    if(count3.ge.4.and.f3.ne.-1) then
        lm3 = dummy
        f3=-1
    endif
    if(lm1.ne.-1.d0.and.lm2.ne.-1.d0.and.lm3.ne.-1.d0) then
        flag=-1
    endif
    goto 55
endif

bottom=zero
top1=zero
top2=zero
temp=zero
limit=zero
do 56 i=1, 10*(idint(lm1))+1
    limpr=limit
    limit=limit+initinc
    call qromo(denom,limpr,limit,temp,midpnt)
    bottom=bottom+temp
56 continue

    limit=zero
    do 57 i=1, 10*(idint(lm2))+1
        limpr=limit
        limit=limit+initinc
        call qromo(numer1,limpr,limit,temp,midpnt)
        top1=top1+temp
57 continue

    limit=zero
    do 58 i=1, 20*(idint(lm3))+1
        limpr=limit
        limit=limit+initinc/2.d0
        call qromo(numer2,limpr,limit,temp,midpnt)

```

```

        top2=top2+temp
58  continue

        if(sngl(top1).eq.0.0.or.sngl(bottom).eq.0.0) then
            avect(1)=zero
        elseif(sngl(top2).eq.0.0.or.sngl(bottom).eq.0.0) then
            avect(2)=zero
        else
            avect(1)=top1/bottom
            avect(2)=top2/bottom
        endif

c      *'Check' to see if a and A close to previous
c      *values (i.e. they have converged)
      call compare(2,avect,oldav,amatr,oldam,done)
      if(done .eq. .false. .and. loop1.lt.10) goto 9
c
c                                     end of LOOP A      <=
c      **At this point we have converged to values for a and A**
c      **Beginning of Step 3      **

      if(loop1.eq.10) then
          write(iout,1)'Problem -Quit loop1, taking latest values'
          write(iout,1)'and entering loop2.'
      endif
      loop1=0
      loop2=loop2+1

c      **Calculate new weights corresponding to new a, and A**

      do 64 i = 1, n
          norm=zero
          do 63 j = 1, np
              dm(j)=amatr(j,1)*(s1(i)-avect(1))+amatr(j,2)*(s2(i)-avect(2))
              norm = norm + dm(j)**2
63      continue
c      *Euclidean norm*
          norm=dsqrt(norm)
          w(i)=dmin1(1.d0,bound/norm)
64      continue

c      **Calculate M1**
      do 70 i=1,np
          do 65 j=1,np
              m1(i,j)=zero
65      continue
70      continue

      do 80 i=1,n
          m1(1,1)=m1(1,1)+(s1(i)-avect(1))**2*w(i)

```

```

        m1(2,2)=m1(2,2)+(s2(i)-avect(2))*2*w(i)
        m1(2,1)=m1(2,1)+(s1(i)-avect(1))*(s2(i)-avect(2))*w(i)
80  continue

c  *taking the average*
    m1(1,1)=m1(1,1)/dble(n)
    m1(2,1)=m1(2,1)/dble(n)
    m1(1,2)=m1(2,1)
    m1(2,2)=m1(2,2)/dble(n)

c  **Calculate the inverse of M1**
    call dgjinv(m1,p,p,indx,scale,.true.,error)
    if(error.eq.-1) then
        write(iout,1)'Problem with M1'
    endif

c  **Calculate deltapar**
    deltapar(1)=zero
    deltapar(2)=zero
    do 85 i=1,n
        deltapar(1)=deltapar(1)+(s1(i)-avect(1))*w(i)
        deltapar(2)=deltapar(2)+(s2(i)-avect(2))*w(i)
85  continue

    do 90 i = 1,np
        deltapar(i)=deltapar(i)/dble(n)
90  continue
    del(1)=m1(1,1)*deltapar(1)+m1(1,2)*deltapar(2)
    del(2)=m1(2,1)*deltapar(1)+m1(2,2)*deltapar(2)

c  **This line allows us to exit Loop2 if we have been in it
c  too long **
    if(loop2 .gt. 10) then
        write(iout,1)'Problem -Gave Up, take latest values'
        write(iout,4)'(In loop2 too long).',loop2
        goto 110
    endif

    rdel1=del(1)/par(1)
    rdel2=del(2)/par(2)
    negcount=0

c  **Basically a 5% epsilon here**
c  **If converged, End**
    if(dabs(rdel1).lt..05d0.and.dabs(rdel2).lt..05d0) goto 110

c  **More than a 100% change, quit with latest values**
c  if(dabs(rdel1).gt.1.d0.or.dabs(rdel2).gt.1.d0) goto 92

c  **Check to see if negative estimates have been obtained**

```

```

91  if((par(1)+del(1)).lt.zero.or.(par(2)+del(2)).lt.zero) then
      write(iout,1)'Problem: Negative parameter estimates obtained'
      write(iout,1)'Half dels to see if we can converge'
      del(1)=del(1)/2.d0
      del(2)=del(2)/2.d0
      negcount=negcount + 1
      if(negcount.lt.10) then
c          *try again*
          goto 91
      else
c          *give up, stuck with negative estimates*
          goto 92
      endif
      else
c          *Remedied negative estimate problem*
c          *Go back and try again*
      par(1)=par(1)+del(1)
      par(2)=par(2)+del(2)
      write(iout,3)'Not acceptable:', par(1),',',par(2)
      goto 9
  endif

92  write(iout,3)'Problem- Did not converge, take latest values'

c                                     end of LOOP B  <=

110  print *, 'END OF LOOP2 WITH:',loop2

c  **Calculate asymptotic covariance structure**

c  *Calculate Q
  do 120 i=1,np
      do 115 j=1,np
          Q(i,j)=zero
115      continue
120  continue
  do 125 i=1,n
      Q(1,1)=Q(1,1)+(s1(i)-avect(1))**2*(w(i)**2)
      Q(2,2)=Q(2,2)+(s2(i)-avect(2))**2*(w(i)**2)
      Q(2,1)=Q(2,1)+(s1(i)-avect(1))*(s2(i)-avect(2))*(w(i)**2)
125  continue
      Q(1,1)=Q(1,1)/db1e(n)
      Q(2,1)=Q(2,1)/db1e(n)
      Q(1,2)=Q(2,1)
      Q(2,2)=Q(2,2)/db1e(n)

c  *Calculate M
  do 135 i=1,np
      do 130 j=1,np
          M(i,j)=zero
130  continue

```



```

c *
c *****
c Written by Joanna E. Mills, TUNS, January 1997
c
c PURPOSE:  Performs Steps 2,3 and 4 of Victoria-Feser and Ronchetti's
c           algorithm [53]
c
c           n   number of observations
c
c           y   vector containing the simulated observations
c
c           ip1  initial alpha value of parameter vector
c
c           ip2  initial beta value of parameter vector
c
c           np   dimension of parameter vector
c
c           iav  initial value of a vector
c
c           ima  initial value of A matrix
c
c           iout file to which output is to be written
c
c           final vector of parameter estimates obtained (OUTPUT)
c
c           w    vector of weights corresponding to OBRE estimates (OUTPUT)
c
c           var  variances of OBRE estimates (OUTPUT)
c
c           cov  covariance of OBRE estimates (OUTPUT)
c -----
c
c      subroutine stepre(n,y,ip1,ip2,np,iav,ima,iout,final,w,var,cov)
c      integer iout, n, np, i, j, loop2, loop1, flag, f1,f2,f3,no, p
c      integer count1, count2, count3, indx(2), error, negcount
c      parameter(no=200)
c      parameter(p=2)
c      real y(*)
c      real*8 ip1, ip2, deltapar(p), avect(p), amatr(p,p), bound
c      real*8 w(no), dm(p), s1(no), s2(no), norm, m1(p,p), m2(p,p)
c      real*8 zero, dummy, denom, Q(p,p), M(p,p)
c      real*8 IN(p,p), V(p,p), var(p), score1, score2, scale(p)
c      real*8 numer1, numer2, bottom, top1, top2, del(p)
c      real*8 oldav(p), oldam(p,p), temp
c      real*8 iav(p), ima(p,p), lm1, lm2, lm3
c      real*8 limit, limpr, cov
c      real*8 par(p), final(p), det, rdel1, rdel2, initinc
c      logical done
c      common/PARAMS/par
c      common/BOUND/bound
c      common/LAGRANGE/avect,amatr
c      external denom, numer1, numer2, midpnt, dgjinv

```

```

      data zero/0.0d0/

1      format(a)
2      format(a,f7.3)
3      format(a,f15.9,a,f15.9)
4      format(a,i4)
5      format(f10.5,f15.13,f15.13,f15.13)

c      *****Initializations*****

c      *Initialize parameter values*
      par(1)=ip1
      par(2)=ip2

c      *Used in numerical integration*
      initinc=par(2)/10.d0

c      *Initialize a vector and A matrix*
      do 10 i=1,np
          avect(i)=iav(i)
          do 6 j= 1,np
              amatr(i,j)=ima(i,j)
          6      continue
      10  continue

      loop1=0
      loop2=0
c      *****Iteration Starting Point*****

c      **Store 'old' values of a and A**
9      loop1=loop1+1
      do 15 i=1,np
          oldav(i)=avect(i)
          do 12 j= 1,np
              oldam(i,j)=amatr(i,j)
          12  continue
      15  continue

c      **Calculate the weights**

      do 30 i = 1, n
          norm=zero

          s1(i)=score1(dble(y(i)))

          s2(i)=score2(dble(y(i)))

c      print *, y(i), s1(i), s2(i)

c      **Calculate the Euclidean norm**

```



```

        do 20 j = 1, np
            dm(j)=amatr(j,1)*(s1(i)-avect(1))+amatr(j,2)*(s2(i)-avect(2))
            norm = norm + dm(j)**2
20      continue
        norm=dsqrt(norm)
        w(i)=dmini(1.d0,bound/norm)
30      continue

c      **Calculate M2 via averaging**
        do 40 i=1,np
            do 35 j=1,np
                m2(i,j)=zero
35      continue
40      continue
        do 50 i=1,n
            m2(1,1)=m2(1,1)+(s1(i)-avect(1))**2*(w(i)**2)
            m2(2,2)=m2(2,2)+(s2(i)-avect(2))**2*(w(i)**2)
            m2(2,1)=m2(2,1)+(s1(i)-avect(1))*(s2(i)-avect(2))*(w(i)**2)
50      continue

c      *taking the average*
        m2(1,1)=m2(1,1)/dble(n)
        m2(2,1)=m2(2,1)/dble(n)
        m2(1,2)=m2(2,1)
        m2(2,2)=m2(2,2)/dble(n)

c      **Calculate the inverse of m2**
        call dgjinv(m2,p,p,indx,scale,.true.,error)
        if(error.eq.-1) then
            write(iout,1)'Problem with M2 matrix'
            return
        endif

c      **Calculate new amatr**
        amatr(1,1)=dsqrt(m2(1,1))
c      * The result is upper triangular
        amatr(2,1)=zero
        amatr(1,2)=m2(2,1)/amatr(1,1)
        amatr(2,2)=dsqrt(m2(2,2)-amatr(1,2)**2)

c      **Calculate new avect**

c      *Compute limits of integration*
        dummy=0.d0
        lm1=-1.d0
        lm2=-1.d0
        lm3=-1.d0
        count1=0
        count2=0
        count3=0

```

```

f1=0
f2=0
f3=0
flag=0

55  if(flag.ne.-1) then
      dummy =dummy + initinc
      if(abs(denom(dummy)).le.0.0000000000001.and.f1.ne.-1) then
          count1=count1 + 1
      endif
      if(count1.ge.4.and.f1.ne.-1) then
          lm1 = dummy
          f1=-1
      endif
      if(abs( numer1(dummy)).le.0.0000000000001.and.f2.ne.-1) then
          count2=count2 + 1
      endif
      if(count2.ge.4.and.f2.ne.-1) then
          lm2 = dummy
          f2=-1
      endif
      if(abs( numer2(dummy)).le.0.0000000000001.and.f3.ne.-1) then
          count3=count3 + 1
      endif
      if(count3.ge.4.and.f3.ne.-1) then
          lm3 = dummy
          f3=-1
      endif
      if(lm1.ne.-1.d0.and.lm2.ne.-1.d0.and.lm3.ne.-1.d0) then
          flag=-1
      endif
      goto 55
  endif

  bottom=zero
  top1=zero
  top2=zero
  temp=zero
  limit=zero
  do 56 i=1, 10*(idint(lm1))+1
      limpr=limit
      limit=limit+initinc
      call qromo(denom,limpr,limit,temp,midpnt)
      bottom=bottom+temp
56  continue

  limit=zero
  do 57 i=1, 10*(idint(lm2))+1
      limpr=limit
      limit=limit+initinc
      call qromo(numer1,limpr,limit,temp,midpnt)

```

```

        top1=top1+temp
57  continue

        limit=zero
        do 58 i=1, 20*(idint(lm3))+1
            limpr=limit
            limit=limit+initinc/2.d0
            call qromo(numer2,limpr,limit,temp,midpnt)
            top2=top2+temp
58  continue

        if(sngl(top1).eq.0.0.or.sngl(bottom).eq.0.0) then
            avect(1)=zero
        elseif(sngl(top2).eq.0.0.or.sngl(bottom).eq.0.0) then
            avect(2)=zero
        else
            avect(1)=top1/bottom
            avect(2)=top2/bottom
        endif

c      *'Check' to see if a and A close to previous
c      *values (i.e. they have converged)
        call compare(2,avect,oldav,amatr,oldam,done)
        if(done .eq. .false. .and. loop1.lt.10) goto 9
c                                     end of LOOP A      <=
c      **At this point we have converged to values for a and A**
c      **Beginning of Step 3      **

        if(loop1.eq.10) then
            write(iout,1)'Problem -Quit loop1, taking latest values'
            write(iout,1)'and entering loop2.'
        endif
        loop1=0
        loop2=loop2+1

c      **Calculate new weights corresponding to new a, and A**

        do 64 i = 1, n
            norm=zero
            do 63 j = 1, np
                dm(j)=amatr(j,1)*(s1(i)-avect(1))+amatr(j,2)*(s2(i)-avect(2))
                norm = norm + dm(j)**2
63      continue
c      *Euclidean norm*
            norm=dsqrt(norm)
            w(i)=dmin1(1.d0,bound/norm)
64      continue

c      **Calculate M1**
        do 70 i=1,np

```

```

        do 65 j=1,np
            m1(i,j)=zero
65      continue
70    continue

        do 80 i=1,n
            m1(1,1)=m1(1,1)+(s1(i)-avect(1))*2*w(i)
            m1(2,2)=m1(2,2)+(s2(i)-avect(2))*2*w(i)
            m1(2,1)=m1(2,1)+(s1(i)-avect(1))*(s2(i)-avect(2))*w(i)
80    continue

c    *taking the average*
        m1(1,1)=m1(1,1)/db1e(n)
        m1(2,1)=m1(2,1)/db1e(n)
        m1(1,2)=m1(2,1)
        m1(2,2)=m1(2,2)/db1e(n)

c    **Calculate the inverse of M1**
        call dgjinv(m1,p,p,indx,scale,.true.,error)
        if(error.eq.-1) then
            write(iout,1)'Problem with M1'
        endif

c    **Calculate deltapar**
        deltapar(1)=zero
        deltapar(2)=zero
        do 85 i=1,n
            deltapar(1)=deltapar(1)+(s1(i)-avect(1))*w(i)
            deltapar(2)=deltapar(2)+(s2(i)-avect(2))*w(i)
85    continue

        do 90 i = 1,np
            deltapar(i)=deltapar(i)/db1e(n)
            write(iout,3)'deltapar2:',deltapar(2)
90    continue
        del(1)=m1(1,1)*deltapar(1)+m1(1,2)*deltapar(2)
        del(2)=m1(2,1)*deltapar(1)+m1(2,2)*deltapar(2)

c    **This line throws us out of the Loop2 if we have been in it
c    too long **
        if(loop2 .gt. 30) then
            write(iout,1)'Problem -Gave Up, take latest values'
            write(iout,4)'(In loop2 too long).',loop2
            goto 110
        endif

        rdel1=del(1)/par(1)
        rdel2=del(2)/par(2)
        write(iout,3)'rdel1:',rdel1, 'rdel2:',rdel2
        negcount=0

```

```

c      **Basically a 5% epsilon here**
c      **If converged, End**
      if(dabs(rdel1).lt..05d0.and.dabs(rdel2).lt..05d0) goto 110

c      **More than a 100% change, quit with latest values**
c      if(dabs(rdel1).gt.1.d0.or.dabs(rdel2).gt.1.d0) goto 92

c      **Check to see if negative estimates have been obtained**
91  if((par(1)+del(1)).lt.zero.or.(par(2)+del(2)).lt.zero) then
      write(iout,1)'Problem: Negative parameter estimates obtained'
      write(iout,1)'Half dels to see if we can converge'
      del(1)=del(1)/2.d0
      del(2)=del(2)/2.d0
      negcount=negcount + 1
      if(negcount.lt.10) then
c          *try again*
          goto 91
      else
c          *give up, stuck with negative estimates*
          goto 92
      endif
  else
c      *Remedied negative estimate problem*
c      *Go back and try again*
      par(1)=par(1)+del(1)
      par(2)=par(2)+del(2)
      write(iout,3)'Not acceptable:', par(1),',',par(2)
      goto 9
  endif

92  write(iout,3)'Problem- Did not converge, take latest values'

c                                     end of LOOP B  <=

110  print *, 'END OF LOOP2 WITH:',loop2
c      **Calculate asymptotic covariance structure**

c      *Calculate Q
      do 120 i=1,np
          do 115 j=1,np
              Q(i,j)=zero
115          continue
120      continue
      do 125 i=1,n
          Q(1,1)=Q(1,1)+(s1(i)-avect(1))**2*(w(i)**2)
          Q(2,2)=Q(2,2)+(s2(i)-avect(2))**2*(w(i)**2)
          Q(2,1)=Q(2,1)+(s1(i)-avect(1))*(s2(i)-avect(2))*(w(i)**2)
125      continue
      Q(1,1)=Q(1,1)/dble(n)
      Q(2,1)=Q(2,1)/dble(n)
      Q(1,2)=Q(2,1)

```

```

      Q(2,2)=Q(2,2)/dble(n)

c      *Calculate M
      do 135 i=1,np
        do 130 j=1,np
          M(i,j)=zero
130      continue
135      continue
      do 140 i=1,n
        M(1,1)=M(1,1)+(s1(i)-avect(1))*s1(i)*w(i)
        M(2,2)=M(2,2)+(s2(i)-avect(2))*s2(i)*w(i)
        M(1,2)=M(1,2)+(s1(i)-avect(1))*s2(i)*w(i)
        M(2,1)=M(2,1)+(s2(i)-avect(2))*s1(i)*w(i)
140      continue
      M(1,1)=M(1,1)/dble(n)
      M(2,1)=M(2,1)/dble(n)
      M(1,2)=M(1,2)/dble(n)
      M(2,2)=M(2,2)/dble(n)

c      *Calculate Inverse of M matrix
      call dgjinv(M,p,p,indx,scale,.true.,error)
      if(error.eq.-1) then
        write(iout,1)'Problem with MINV'
      endif

c      write(iout,1)'Inverse of M matrix:'
c      write(iout,3)'M(1,1):',M(1,1),' MINV(1,2):',M(1,2)
c      write(iout,3)'M(2,1):',M(2,1),' MINV(2,2):',M(2,2)

c      *CALCULATE V
      IN(1,1)=M(1,1)*Q(1,1)+M(1,2)*Q(2,1)
      IN(1,2)=M(1,1)*Q(1,2)+M(1,2)*Q(2,2)
      IN(2,1)=M(2,1)*Q(1,1)+M(2,2)*Q(2,1)
      IN(2,2)=M(2,1)*Q(1,2)+M(2,2)*Q(2,2)

      V(1,1)=IN(1,1)*M(1,1)+IN(1,2)*M(1,2)
      V(1,2)=IN(1,1)*M(2,1)+IN(1,2)*M(2,2)
      V(2,1)=IN(2,1)*M(1,1)+IN(2,2)*M(1,2)
      V(2,2)=IN(2,1)*M(2,1)+IN(2,2)*M(2,2)

      var(1)=V(1,1)/dble(n)
      var(2)=V(2,2)/dble(n)
      cov=V(1,2)/dble(n)

      final(1)=par(1)
      final(2)=par(2)
150      continue
      return
      end

```

term.f

```

c *****
c *
c *          function term
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, November 1996
c
c PURPOSE  to return the term function required for calculation of the
c           covariance of the mle estimate of the Beta parameter of a
c           Birnbaum-Saunders distribution.
c
c      x  real value at which the function is to be evaluated
c -----
      real*8 function term(x)
      real*8 para(2), x, pi, g
      common/PARAMS/para
      data pi/3.14159265d0/

      g=1.d0+((para(1)*x)**2/2.d0)
      g=g+para(1)*x*dsqrt(1.d0+((para(1)*x)**2/4.d0))

      term=((1.d0+g)**(-1)-0.5d0)**2/dsqrt(2.d0*pi)*dexp(-x**2/2.d0)

      return
      end

```

weight.f

```

c *****
c *
c *          function weight
c *
c *****
c Double Precision Version 1.0
c Written by Joanna E. Mills, September 1996
c
c PURPOSE  to return the weight for a given c, am, score, and av
c
c      x  real value at which the function is to be evaluated
c -----
      real*8 function weight(x)
      real*8 am(2,2), av(2), c, vector(2), x
      real*8 diff(2), norm, score1, score2

```

```
common/BOUND/c
common/LAGRANGE/av,am

diff(1)=score1(x)-av(1)
diff(2)=score2(x)-av(2)

vector(1)=am(1,1)*diff(1)+am(1,2)*diff(2)
vector(2)=am(2,1)*diff(1)+am(2,2)*diff(2)

norm = dsqrt(vector(1)**2+vector(2)**2)

weight=c/norm

if (weight .gt. 1.d0) then
    weight = 1.d0
endif
return
end
```


IMAGE EVALUATION TEST TARGET (QA-3)

APPLIED IMAGE, Inc.
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved