



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1990-09

Lower confidence interval bounds for coherent systems with cyclic components

Covington, Valerie A.

Monterey, California: Naval Postgraduate School

<https://hdl.handle.net/10945/34882>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

AD-A242 713



2

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC
ELECTE
NOV 22 1991
S C D

THESIS

LOWER CONFIDENCE INTERVAL BOUNDS FOR
COHERENT SYSTEMS WITH CYCLIC COMPONENTS

by

Valerie A. Covington

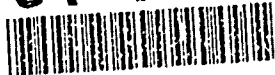
September, 1990

Thesis Advisor:

W. Max Woods

Approved for public release; distribution is unlimited.

91-15859



91 1118 086

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified			1b Restrictive Markings		
2a Security Classification Authority			3 Distribution Availability of Report		
2b Declassification/Downgrading Schedule			Approved for public release; distribution is unlimited.		
4 Performing Organization Report Number(s)			5 Monitoring Organization Report Number(s)		
6a Name of Performing Organization Naval Postgraduate School		6b Office Symbol (if applicable) OR	7a Name of Monitoring Organization Naval Postgraduate School		
6c Address (city, state, and ZIP code) Monterey, CA 93943-5000			7b Address (city, state, and ZIP code) Monterey, CA 93943-5000		
8a Name of Funding Sponsoring Organization		8b Office Symbol (if applicable)	9 Procurement Instrument Identification Number		
8c Address (city, state, and ZIP code)			10 Source of Funding Numbers		
			Program Element No	Project No	Task No
			Work Unit Accession No		
11 Title (Include security classification) LOWER CONFIDENCE INTERVAL BOUNDS FOR COHERENT SYSTEMS WITH CYCLIC COMPONENTS					
12 Personal Author(s) Valerie A. Covington					
13a Type of Report Master's Thesis		13b Time Covered From To		14 Date of Report (year, month, day) September 1990	15 Page Count 132
16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
17 Coati Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)		
Field	Group	Subgroup	reliability, lower confidence limit, coherent systems, cyclic components.		
19 Abstract (continue on reverse if necessary and identify by block number) Three lower confidence interval estimation procedures for system reliability of coherent systems with cyclic components are developed and their accuracy measured using Monte Carlo techniques. The procedures use either the Poisson approximation to the Binomial distribution, the lower Binomial confidence limit procedure, or a modified procedure using the Poisson approximation to the Binomial distribution to obtain an equation for the lower confidence limit. The accuracy of the interval estimators were evaluated using standard computer simulation methods for series, parallel, series-parallel, and Wheatstone Bridge systems. The method determined to be most accurate can be combined with similar procedures for components that have continuous failure times and applied to yield a lower confidence interval procedure for the reliability of coherent systems with cyclic and continuously operating components.					
20 Distribution/Availability of Abstract <input checked="" type="checkbox"/> unclassified unlimited <input type="checkbox"/> same as report <input type="checkbox"/> DTIC users			21 Abstract Security Classification Unclassified		
22a Name of Responsible Individual W. Max Woods			22b Telephone (include Area code) (408) 646-2768	22c Office Symbol OR-Wo	

DD FORM 1473,84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Lower Confidence Interval
Bounds for Coherent Systems
With Cyclic Components

by

Valerie A. Covington
Lieutenant, United States Navy
B.A., University of South Florida, 1975
M.Ed., West Georgia College, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1990

Author:

[Redacted]

Valerie A. Covington

Approved by:

[Redacted]

W. Max Woods, Thesis Advisor

[Redacted]

Lyn R. Whitaker, Second Reader

[Redacted]

Peter Purdue, Chairman,
Department of Operations Research

ABSTRACT

Three lower confidence interval estimation procedures for system reliability of coherent systems with cyclic components are developed and their accuracy measured using Monte Carlo techniques. The procedures use either the Poisson approximation to the Binomial distribution, the lower Binomial confidence limit procedure, or a modified procedure using the Poisson approximation to the Binomial distribution to obtain an equation for the lower confidence limit. The accuracy of the interval estimators were evaluated using standard computer simulation methods for series, parallel, series-parallel, and Wheatstone Bridge systems. The method determined to be most accurate can be combined with similar procedures for components that have continuous failure times and applied to yield a lower confidence interval procedure for the reliability of coherent systems with cyclic and continuously operating components.

Accession For	
NTIS GRAY	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

I. INTRODUCTION	1
II. PREFERRED LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY	5
A. METHODOLOGY	5
B. RESULTS	8
1. Series System	8
2. Series-Parallel Systems	10
3. Series-Parallel Systems with a 2 of 3 Parallel Component	10
4. Parallel System	12
5. Wheatstone Bridge	13
III. ALTERNATE PROCEDURE A FOR THE LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY	15
A. METHODOLOGY	15
B. RESULTS	15
IV. ALTERNATE PROCEDURE B FOR THE LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY	18
A. METHODOLOGY	18
B. RESULTS	18
V. SIMULATION	21
VI. CONCLUSIONS AND RECOMMENDATIONS	22
APPENDIX A. DISCRETE CONFIDENCE LIMIT PROPERTIES	23
APPENDIX B. INPUT PARAMETERS	25

APPENDIX C. FORTRAN CODE FOR THE PREFERRED LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY (SERIES SYSTEM AND WHEATSTONE BRIDGE SYSTEM)	29
APPENDIX D. FORTRAN CODE FOR THE PREFERRED LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY (SERIES PARALLEL SYSTEM)	43
APPENDIX E. FORTRAN CODE FOR THE PREFERRED LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY (SERIES-PARALLEL SYSTEM WITH A 2/3 COMPONENT)	58
APPENDIX F. FORTRAN CODE FOR THE PREFERRED LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY (PARALLEL SYSTEM)	73
APPENDIX G. FORTRAN CODE FOR ALTERNATE PROCEDURE A FOR ESTIMATING THE LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY	88
APPENDIX H. FORTRAN CODE FOR ALTERNATE PROCEDURE B FOR ESTIMATING THE LOWER CONFIDENCE LIMIT FOR SELECTED SYSTEMS	102
APPENDIX I. SUBROUTINES	117
LIST OF REFERENCES	121
BIBLIOGRAPHY	122
INITIAL DISTRIBUTION LIST	123

LIST OF TABLES

Table 1. SERIES SYSTEM	9
Table 2. SERIES-PARALLEL SYSTEM	10
Table 3. SERIES-PARALLEL WITH A 2 OUT OF 3 COMPONENT SYSTEM	11
Table 4. SERIES-PARALLEL WITH A 2 OUT OF 3 COMPONENT SYSTEM (CONSTANT NUMBER OF MISSION TESTS, DECREASING RELI- ABILITY)	12
Table 5. PARALLEL SYSTEM	13
Table 6. WHEATSTONE BRIDGE SYSTEM	14
Table 7. COMPARISON OF PREFERRED PROCEDURE AND ALTERNATE PROCEDURE A (SERIES SYSTEM)	16
Table 8. COMPARISON OF PREFERRED PROCEDURE AND ALTER- NATE PROCEDURE A (WHEATSTONE BRIDGE)	17
Table 9. COMPARISON OF PREFERRED PROCEDURE AND ALTERNATE PROCEDURE B (SERIES SYSTEM)	19
Table 10. COMPARISON OF PREFERRED PROCEDURE AND ALTER- NATE PROCEDURE B (WHEATSTONE BRIDGE)	20
Table 11. SERIES SYSTEM INPUT PARAMETERS	25
Table 12. SERIES-PARALLEL SYSTEM INPUT PARAMETERS	26
Table 13. SERIES-PARALLEL WITH A 2 OUT OF 3 COMPONENT SYSTEM INPUT PARAMETERS	27
Table 14. PARALLEL SYSTEM INPUT PARAMETERS	28
Table 15. WHEATSTONE BRIDGE INPUT PARAMETERS	28

LIST OF FIGURES

Figure 1. Series System	2
Figure 2. Series-Parallel System	3
Figure 3. Series Parallel With Two Out of Three Subcomponent System	3
Figure 4. Parallel System	3
Figure 5. Wheatstone Bridge	4
Figure 6. Behavior of	24

I. INTRODUCTION

Coherent systems are those systems for which the system performs if all components function, the system fails if all the components fail, replacing a failed component with a working component does not cause the system to fail and similarly replacing a working component with a failed component does not cause the system to work [Ref. 1: p. 343]. The reliability of a coherent system is not reduced when the reliability of its components is increased. Cyclic components are those whose function is measured on a pass-fail basis and whose reliability is computed from a discrete probability distribution, usually the Geometric distribution.

The problem of obtaining confidence limits on the reliability of a coherent system based on data gathered on its individual components has attracted considerable interest. Confidence bounds for the reliability of series systems have been obtained asymptotically, based on methods such as Likelihood Ratio, Maximum Likelihood, or Modified Maximum Likelihood. Asymptotic methods are inaccurate at higher percentiles unless the component sample sizes are large. Bayesian methods have been developed for this problem, but they are extremely sensitive to the selection of the prior distribution. [Ref. 2: p. 21]

Exact confidence limits have been obtained for simple systems. This solution requires that the reliability of at least one of the components has to be greater than that of the system. However, identifying such a component becomes quite complex for all but simple systems composed of no more than two or three components [Ref. 3: p. 220]. Exact methods have also been developed for series systems using asymptotic approaches and the unbiased minimum variance estimators of the probability of success, p , based on Binomial data [Ref. 4: p. 782].

Unfortunately, none of the above referenced interval estimation procedures based on discrete data can be readily used in conjunction with the data of components that have continuous failure times. Therefore it is difficult to obtain interval estimates for the reliability of complex systems that have mixtures of cyclic components and components that operate continuously. The methods developed in this thesis can be combined with similar methods that use continuous data, namely those developed by Lee [Ref. 5]. The combination of these methods may provide interval estimation for the reliability of systems with cyclic and continuously operating components.

In this thesis, three procedures that provide lower confidence limits for the reliability of coherent systems with cyclic components were analyzed. These procedures use only discrete data. There is a problem with using component data to establish system reliability, especially for a system that has quite a bit of redundancy. Even though the tested components fail individually and their estimated reliabilities are moderate, had these components been assembled into a system, the system could very well have worked. In such a case, the system has a very high degree of reliability and methods that work well in estimating the more moderate component reliability will not work well in estimating the system reliability. Thus, each procedure that we study has modifications to accommodate component test data which when assembled into systems would exhibit zero system failures, one system failure, or more than one system failure, i.e.

- no component failures occur or only components fail that are redundant in the system, so that no system failure could occur if all the components were combined to form systems,
- exactly one component fails that would result in a system failure or redundant components in the system fail in a quantity, so that no more than one system failure could occur if the components were combined to form systems (this modification is explained in more detail in the following chapters),
- for any component, i , more than one component fails out of n_i tested which would lead to more than one system failure.

The systems analyzed in this thesis are as follows:

- all components arranged in series (see Figure 1)

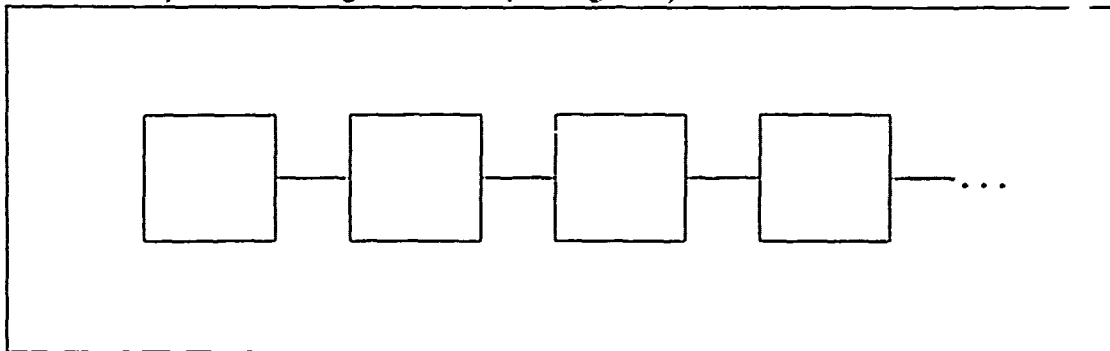


Figure 1. Series System

- all components arranged in series with at least one component consisting of two parallel subcomponents (see Figure 2)

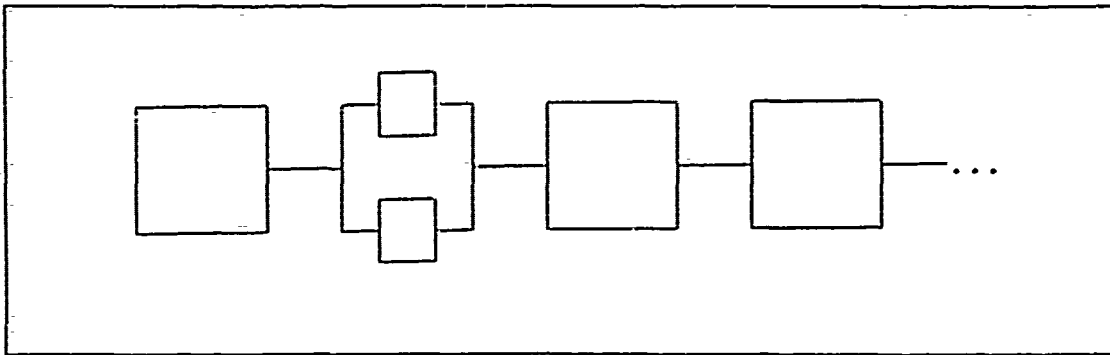


Figure 2. Series-Parallel System

- all components arranged in series with at least one component consisting of two parallel subcomponents and at least one component in a two out of three subcomponent parallel arrangement (see Figure 3)

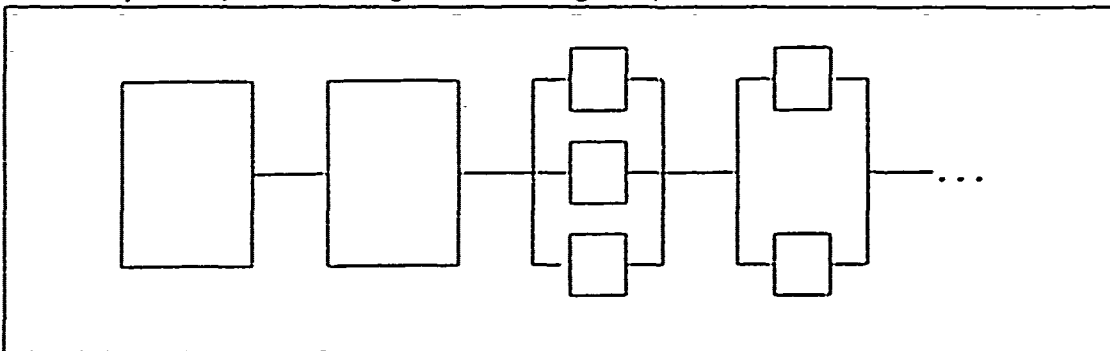


Figure 3. Series Parallel With Two Out of Three Subcomponent System

- a system with all components in a parallel arrangement (see Figure 4)

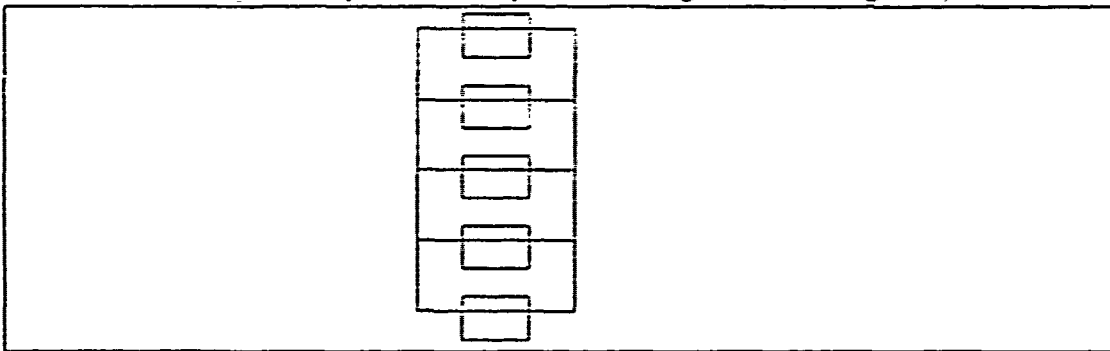


Figure 4. Parallel System

- a five component redundant system commonly referred to as a Wheatstone Bridge (see Figure 5)

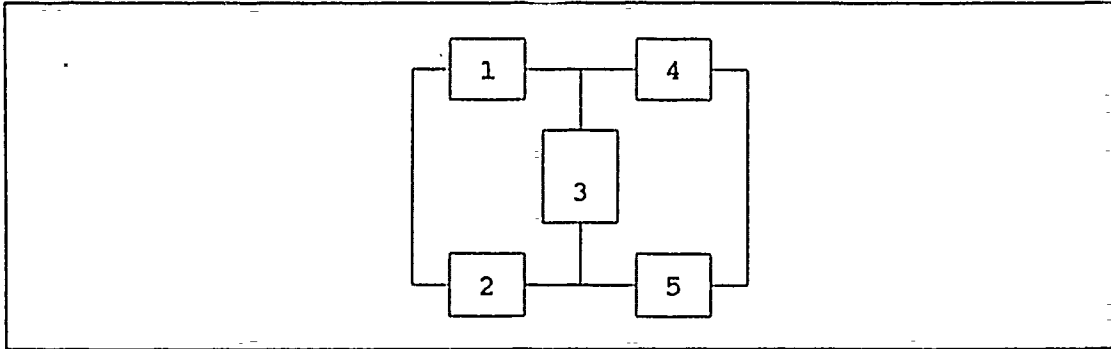


Figure 5. Wheatstone Bridge

II. PREFERRED LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY

A. METHODOLOGY

The procedure selected to determine the lower confidence limit of relatively simple complex systems uses a method employed by Lomnicki [Ref. 6: p. 109] and extended by Myhre and others [Ref. 2: p. 213]. It uses the Poisson approximation to the Binomial distribution when numerous component failures occur that would result in multiple system failures if the tested components were assembled into systems. In cases where zero system failures occur or only one system failure could result from the failed components, the Poisson approximation appears to be conservative and the lower confidence limit is computed using the Binomial distribution directly.

Suppose a coherent system has k different types of components that are statistically independent. Test data is available on each type of component. The n_i tests for component i are assumed to be independent Bernoulli trials with probability of failure, q_i , for each test. Therefore the number of failures, F_i , in these n_i tests has a Binomial distribution.

Any of the k components, say component m , can be chosen as a base component and q_i expressed as a fraction or multiple, a_i , of q_m . That is

$$q_i = a_i q_m \quad (2.1)$$

Suppose it is appropriate to assume that the probability distribution of the number of failures of each component, F_i , can be approximated by the Poisson distribution with mean $n_i q_i$. If $F = \sum_{i=1}^k F_i$, then from Equation (2.1), the distribution of F is approximately Poisson with parameter $q_m \sum_{i=1}^k n_i a_i$. Consequently the mean of F , $E(F)$, is given by

$$E(F) = q_m \sum_{i=1}^k n_i a_i \quad (2.2)$$

The system reliability, R_s , can be defined as a function of the unreliability, q_m and the associated values $a_i = \frac{q_i}{q_m}$. That is,

$$R_s = h(q_m, a_1, a_2, \dots, a_k) \quad (2.3)$$

By definition, h is non-increasing in q_m , because the system is coherent. If a_1, a_2, \dots, a_k are known, an approximate lower confidence limit, $R_{s,L(\alpha)}$, may be obtained from an upper confidence limit, $\hat{q}_{m,U(\alpha)}$, for q_m by the equation

$$\hat{R}_{s,L(\alpha)} = h(\hat{q}_{m,U(\alpha)}, a_1, a_2, \dots, a_k) \quad (2.4)$$

When appropriate, the upper confidence limit, $\hat{q}_{m,U(\alpha)}$, may be obtained from the well known upper confidence limit for the mean, λ , of a Poisson distribution, namely, if F is distributed POISSON(λ) then

$$\lambda_{U(\alpha)} = \frac{X_{\alpha, 2(1+F)}^2}{2} \quad (2.5)$$

where $X_{\alpha, 2(1+F)}^2$ is the $(1 - \alpha)$ quantile of a Chi-square probability distribution with degrees of freedom equivalent to $2(1 + F)$, where F is the number of system failures. From Equation (2.2), substitution of $q_m \sum_{i=1}^k n_i a_i$ for λ into Equation (2.5) gives

$$q_{m,U(\alpha)} = \frac{X_{\alpha, 2(1+F)}^2}{2 \sum_{i=1}^k n_i a_i} \quad (2.6)$$

If the a_i are unknown then an approximate upper confidence limit, $\hat{q}_{m,U(\alpha)}$, is given by

$$\hat{q}_{m,U(\alpha)} = \frac{X_{\alpha, 2(1+F)}^2}{2 \sum_{i=1}^k n_i \hat{a}_i} \quad (2.7)$$

where \hat{a}_i is an estimator for a_i , $i = 1, 2, \dots, k$.

The Poisson approximation to the Binomial distribution is conservative when all F_i equal zero or redundant components of the system fail in such a way that results in zero system failures. In this case, let n^* represent the number of system tests equivalent to testing each component n_i times, $i = 1, \dots, k$. Then the distribution of s , the number of system tests that would have been successful, is approximately Binomial(n^*, R_i). If s is distributed BINOMIAL(n^*, R_i) then we can use the binomial lower confidence limit

$$\hat{R}_{s,L(\alpha)} = n^* \sqrt{\alpha} \quad (2.8)$$

to compute the lower confidence limit for system reliability. The following two methods for the calculation of the number of equivalent system tests, n^* , were selected and each applied separately using Equation (2.8).

$$n_1^* = \min(n_1, n_2, \dots, n_k) \quad (2.9)$$

$$n_2^* = \frac{\sum_{i=1}^k n_i}{k} \quad (2.10)$$

where n_i is the number of tests of component i .

Some instances of component failures could only result in one equivalent system failure if all of the tested components were combined into complete systems. In this case, we again define n^* and treat n^* system trials with one failure. The resulting lower confidence limit, $\hat{R}_{s, L(\alpha)}$, for system reliability is the solution for $p = (1 - q)$ in the equation

$$\alpha = \sum_{j=n^*-1}^{n^*} \binom{n^*}{j} (1-q)^j q^{n^*-j} \quad (2.11)$$

In the Wheatstone Bridge case, two or more component failures among the tested components could result in one system failure if the components are assembled into Wheatstone systems. In this case we set $n^* = n_1^*$ for one interval procedure and $n^* = n_2^*$ for the second interval procedure.

In a series system, n^* is equal to the number of tests performed on the failed single component, because the reliability of the system is determined largely by the reliability of the least reliable component. Since it is difficult to solve for p in Equation (2.11), an equivalent equation using the Snedecor F distribution is used. Thus,

$$\hat{R}_{s, L(\alpha)} = \frac{s}{s + (f + 1) F_{(1-\alpha), 2(f+1), 2s}} \quad (2.12)$$

where s is the number of system successes, f is the number of system failures, and $F_{(1-\alpha), 2(f+1), 2s}$ is the α quantile of the Snedecor F distribution with $2(f + 1)$ and $2s$ degrees of freedom [Ref. 7: p. 43].

B. RESULTS

The accuracy of this procedure was evaluated using computer simulations for each of the following systems described in Section A:

- series systems
- series systems where the second component is composed of two parallel subcomponents
- series systems where the second component is composed of two parallel subcomponents and the fourth component consists of a two of three subcomponent parallel arrangement
- parallel systems
- Wheatstone Bridge

Groups of test data were generated where the parameters, q_i and n_i , were chosen to control the expected number of failures, $E[F] = \sum_{i=1}^k n_i q_i$. Confidence levels of 0.20 and 0.05 were used in each case. A total of 1000 replications were generated for each set of parameter values. Each replication produced one value of $\hat{R}_{r,L(\alpha)}$. These 1000 values, $\hat{R}_{r,L(\alpha)}$, were ordered and used to get the simulated probability distribution of $\hat{R}_{r,L(\alpha)}$. The simulation procedures are described in Chapter IV. The 80th and 95th percentile point of the simulated probability distribution of $\hat{R}_{r,L(\alpha)}$ was compared to R_r for determining the accuracy of the procedure. This comparison is made because $\hat{R}_{r,L(\alpha)}$ is the lower $100(1 - \alpha)$ percentile confidence limit for R_r if $1 - \alpha = P(\hat{R}_{r,L(\alpha)} \leq R_r)$. This equation states that R_r is the $100(1 - \alpha)$ percentile point of the probability distribution of $\hat{R}_{r,L(\alpha)}$. The "true confidence level" is the percentile point of the simulated distribution corresponding to the true value of R_r .

The parameter values $n_1 q_1, n_2 q_2, \dots, n_k q_k$ determine a case number and are labeled as such in the tables that describe the simulation results. A summary table that provides the parameter values, q_i and n_i , is given in Appendix B.

All tables report the 80th and 95th percentile points of the simulated distribution of $\hat{R}_{r,L(\alpha)}$ and appear in the tables under the column labeled $\hat{R}_{r,L(\alpha)}$.

1. Series System

By definition, the reliability, R_s , for a series system of k independent components is

$$R_s = \prod_{i=1}^k (1 - q_i) = \prod_{i=1}^k (1 - a_i q_m) \quad (2.13)$$

The corresponding lower confidence limit is given by

$$\hat{R}_{s,L(\alpha)} = \prod_{i=1}^k (1 - \hat{a}_i \hat{q}_{m,U(x)}) \quad (2.14)$$

These formulae are used to calculate the reliability when at least two components fail. If zero components fail Equation (2.8) is used and if one component fails Equation (2.12) is used. The results are presented in Table 1. In Table 1, column 1 of $\hat{R}'_{s,L(\alpha)}$ is calculated using Equation (2.10) and column 2 is calculated using Equation (2.9) for n^* when the component failures equate to zero system failures.

Table 1. SERIES SYSTEM

Case	# Compts	$E[F]$	R_s	α Level	$\hat{R}'_{s,L(\alpha)}$		True Confidence Level	
1	5	1.03	.95572	.20	.91447	.90345	75	75
				.05	.85140	.85140	100	100
2	10	1.2	.93206	.20	.88491	.88491	100	100
				.05	.85140	.85140	100	100
3	5	4.9	.95034	.20	.95361	.95361	76	76
				.05	.95969	.95969	90	90
4	10	4.9	.85951	.20	.85369	.85369	85.2	85.2
				.05	.86718	.86089	94.5	94.5
5	5	5.74	.95084	.20	.95036	.95036	76.3	76.3
				.05	.95779	.95779	91.1	91.1
6	10	5.75	.85951	.20	.85910	.85910	80.1	80.1
				.05	.88114	.88072	92.5	92.6
7	5	10.5	.85828	.20	.87174	.87174	72.4	72.4
				.05	.88366	.88366	86.6	86.6
8	10	10.5	.85951	.20	.87137	.87137	73.3	73.3
				.05	.89049	.89049	86.4	86.4

In some cases the true confidence level equals 100. This is not uncommon for confidence intervals based on discrete data and is further explained in Appendix A.

2. Series-Parallel Systems

A modification was made to the series systems to form the series-parallel system. The second component in the series was modified to consist of two parallel subcomponents of equal unreliability, q_2 . For the second component to fail both subcomponents must fail. The unreliability of component two is q_2^2 . Since component two is in series with the other components the corresponding lower confidence limit is calculated using Equation (2.8) when the equivalent number of system failures is zero, Equation (2.12) when the number of system failures is one, and Equation (2.14) is used in all other cases. The results are presented in Table 2. In Table 2, column 1 of $\hat{R}'_{s,L(s)}$ is calculated using Equation (2.10) and column 2 is calculated using Equation (2.9) for n^* when the component failures equate to zero system failures.

Table 2. SERIES-PARALLEL SYSTEM

Case	# Compts	$E[F]$	R_s	α Level	$\hat{R}'_{s,L(s)}$		True Confidence Level	
9	5	1.03	.95572	.20	.91447	.90345	75	75
				.05	.85140	.85140	100	100
10	10	2.72	.93206	.20	.95770	.95770	64.5	64.5
				.05	.95344	.95344	74.4	64.5
11	5	4.9	.85828	.20	.85369	.85369	83.5	83.5
				.05	.86685	.86089	94.2	94.3
12	10	4.9	.85951	.20	.85546	.85546	83.1	83.1
				.05	.86685	.86685	94.1	94.1
13	5	9.79	.95084	.20	.95321	.95321	76.8	76.8
				.05	.95779	.96029	88.6	88.6
14	10	10.5	.85951	.20	.85943	.86943	75.2	75.2
				.05	.88606	.88606	87.6	87.6

Note the accuracy of the lower confidence interval is somewhat reasonable when the expected number of failures exceeds 4.9 in both the series and series-parallel systems.

3. Series-Parallel Systems with a 2 of 3 Parallel Component

Another modification was made to the five and ten component series systems. The third component in the series was modified and now consists of three parallel subcomponents of equal unreliability, q_3 . For this component to fail, two or three parallel components must fail. The unreliability, q_3 , of component three is $\binom{3}{2}(1 - q_3)q_3^2 + q_3^3$.

The fourth component of each system is composed of two parallel subcomponents as defined in the series-parallel system. Since these components are in series with the other components, the corresponding lower-confidence limit is calculated using the same series of equations as the series-parallel system. The results are presented in Table 3. The term n^* is computed from Equation (2.10) when the equivalent number of system failures is zero.

Table 3. SERIES-PARALLEL WITH A 2 OUT OF 3 COMPONENT SYSTEM

Case	# Compts	$E[F]$	R_s	α Level	$\hat{R}'_{s,L(\alpha)}$	True Confidence Level
15	5	.88	.96525	.20	.91447	100
				.05	.85140	100
16	10	2.6	.94136	.20	.95770	63.7
				.05	.95344	85.7
17	5	3.21	.96035	.20	.95937	80.4
				.05	.96877	88.5
18	10	4.6	.88945	.20	.88582	80.9
				.05	.91735	89.5
19	5	6.17	.96035	.20	.96126	78.8
				.05	.96886	90.5
20	10	8.37	.88945	.20	.89734	76.7
				.05	.91817	87.8

Another series of simulations were conducted on these systems. For each case previously run, new cases were defined by decreasing the unreliability of selected components without changing the number of component tests. These changes result in a decrease in the number of expected failures and an increase in the reliability of the system. The results are presented in Table 4. The term n^* is computed from Equation (2.10) when the equivalent number of system failures is zero.

Table 4. SERIES-PARALLEL WITH A 2 OUT OF 3 COMPONENT SYSTEM (CONSTANT NUMBER OF MISSION TESTS, DECREASING RELIABILITY)

Case	# Compts	$E[F]$	R_s	α Level	$\hat{R}_{L(\alpha)}$	True Confidence Level
21	10	.51	.99136	.20	.97661	100
				.05	.95690	100
22	10	1.68	.96836	.20	.97661	72.3
				.05	.95690	96.3
23	10	6.57	.95430	.20	.95497	77.2
				.05	.96169	91.3
24	10	9.90	.90249	.20	.90515	78.9
				.05	.91486	91.6

Tables 3 and 4 indicate that for $E(F) > 5$ the lower confidence limits are reasonably accurate.

4. Parallel System

The accuracy of the lower confidence procedure was evaluated when it was applied to a five component system with each component, i , in parallel with the others. The unreliabilities, q , of each component are equal and the unreliability of the system is defined as q^5 . For the system to fail all five components must fail. The results of the computer simulations are presented in Table 5. The term n^* is computed from Equation (2.10) when the equivalent number of system failures is zero. If n of these components are tested and four or fewer failures occur, then no system failures would have occurred had these components been assembled into systems. The lower confidence limit is calculated using Equation (2.8) when the equivalent number of system failures is zero, Equation (2.12) when the number of system failures is one, and Equation (2.14), with $k = 1$, in all other cases.

Table 5. PARALLEL SYSTEM

Case	# Compts	$E[F]$	R_s	α Level	$\hat{R}_{s, U(\alpha)}$	True Confidence Level
25	5	1	.99000	.20	.98403	100
				.05	.97049	100
26	5	4	.96000	.20	.95770	91.1
				.05	.95344	98.3
27	5	6	.94000	.20	.93370	83.9
				.05	.93838	98.2
28	5	9	.94000	.20	.94012	77.6
				.05	.94002	93.6

5. Wheatstone Bridge

By definition, the reliability, R_s , for the Wheatstone Bridge, in Figure 5, with five independent components is

$$R_s = p_1 p_4 + p_2 p_5 + p_1 p_2 p_5 + p_2 p_3 p_4 - p_1 p_2 p_3 p_4 - p_1 p_2 p_4 p_5 - p_1 p_3 p_4 p_5 - p_2 p_3 p_4 p_5 + 2 p_1 p_2 p_3 p_4 p_5 \quad (2.16)$$

where $p_i = (1 - q_i)$. In terms of q_i , Equation (2.16) becomes

$$\begin{aligned} R_s &= h(q_m, a_1, \dots, a_5) \\ &= 1 - q_m^2(a_1 a_2 + a_4 a_5) - q_m^3(a_1 a_3 a_5 + a_2 a_3 a_4) \\ &\quad + q_m^4(a_1 a_2 a_3 a_4 + a_1 a_2 a_3 a_5 + a_1 a_2 a_4 a_5 + a_1 a_3 a_4 a_5 + a_2 a_3 a_4 a_5) \\ &\quad - 2 q_m^5(a_1 a_2 a_3 a_4 a_5) \end{aligned} \quad (2.17)$$

[Ref. 2: p. 215]. By substituting $\hat{a}_{\hat{q}_{m, U(\alpha)}}$ for q_i , the corresponding lower confidence limit is obtained.

$$\hat{R}_{s, U(\alpha)} = h(\hat{q}_{m, U(\alpha)}, \hat{a}_1, \dots, \hat{a}_5) \quad (2.18)$$

The reliability of the Wheatstone Bridge system is normally high due to the redundancy of the system. This system experiences zero system failures in the following five different failure patterns.

- $F_1 = F_2 = F_3 = F_4 = F_5 = 0$
- $F_1 = F_3 = F_5 = 0$
- $F_1 = F_4 = 0$
- $F_2 = F_5 = 0$
- $F_2 = F_3 = F_4 = 0$

where F_i is the number of failures of component i , among its n_i tests. Any other failure patterns will produce one or more system failures. The lower confidence limit is calculated using Equation (2.8) when the equivalent number of system failures is zero, Equation (2.12) when the number of system failures is one, and Equation (2.18) in all other cases. The results of the computer simulations are presented in Table 6. In Table 6, column 1 of $\hat{R}'_{s,L(t)}$ is calculated using Equation (2.10) and column 2 is calculated using Equation (2.9) for n^* when the component failures equate to zero system failures.

Table 6. WHEATSTONE BRIDGE SYSTEM

Case	# Compts	$E[F]$	R_s	α Level	$\hat{R}'_{s,L(t)}$		True Confidence Level	
29	5	5.75	.99776	.20	.99658	.99658	88.6	88.6
				.05	.99766	.99766	95.4	95.4
30	5	6	.99976	.20	.99960	.99960	91.4	91
				.05	.99966	.99966	98.3	98.3
31	5	.5	.99977	.20	.91447	.72478	100	100
				.05	.98678	.98678	100	100
32	5	.5	.99977	.20	.92622	.72478	100	100
				.05	.99339	.99339	100	100

The system reliability values in cases 31 and 32 are too large for these interval estimation procedures to be accurate using the sample sizes given in Table 15 in Appendix B. Also, more than 1000 replications would be needed to assess the accuracy of any lower confidence limit procedure for system reliability when the true system reliability is as large as it is in these cases.

III. ALTERNATE PROCEDURE A FOR THE LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY

A. METHODOLOGY

This procedure determines the lower confidence limit for simple and complex systems using only the Poisson approximation to the Binomial distribution. It is used extensively by Bellini [Ref. 8: p.4-6].

This procedure uses the principles outlined in Chapter II of this thesis except in the cases where there are zero or one system failure. In the case where there are zero system failures (no components fail) the estimated unreliability, \hat{q}_i , becomes zero because $\hat{q}_i = \frac{F_i}{N_i}$ where F_i is the number of failures in n_i mission tests of component i . Therefore, the value of \hat{a}_i , the estimated value of a_i , becomes zero and Equation (2.7) becomes undefined. When this occurs the estimated lower confidence limit of the system reliability, $\hat{R}_{s,L(x)}$, is defined as

$$\hat{R}_{s,L(x)} = 1 - \frac{\chi_{\alpha, 2}^2}{2n^*} \quad (3.1)$$

where n^* is defined as in Equation (2.10).

If only component, m , fails the value of \hat{a}_m is equal to 1. The \hat{a}_i for all other components are zero, and the lower confidence limit, $\hat{R}_{s,L(x)}$, is defined as

$$\hat{R}_{s,L(x)} = 1 - \frac{\chi_{\alpha, 2(1+r)}^2}{2n_m} \quad (3.2)$$

where n_m is the number of mission tests of the failed component.

B. RESULTS

The accuracy of this procedure was evaluated only for series and Wheatstone Bridge systems. Testing was limited because a comparison of results of the three procedures, discussed in this thesis, indicated this procedure to be less accurate in determining the lower confidence limit for the reliability of simple systems. This observation is noted in those systems experiencing zero system failures or one system failure. Selected results illustrating the accuracy of this procedure compared to the accuracy of the preferred

procedure" (Chapter II) are indicated in Tables 7 and 8. The term n^* is computed from Equation (2.10) when the equivalent number of system failures is zero.

Table 7. COMPARISON OF PREFERRED PROCEDURE AND ALTERNATE PROCEDURE A (SERIES SYSTEM)

Case	R_i	α Level	Preferred Procedure		Alternate Procedure	
			$\hat{R}'_{i,L(\alpha)}$	True Confidence Level	$\hat{R}'_{i,L(\alpha)}$	True Confidence Level
1	.95572	.20	.91447	75	.83140	100
		.05	.85140	100	.92356	100
2	.93206	.20	.88491	100	.83140	100
		.05	.85140	100	.92356	100
3	.95084	.20	.95361	76	.95361	76
		.05	.95969	90	.95969	90
4	.85951	.20	.85369	85.2	.85119	85.5
		.05	.86718	94.5	.85083	95.2
5	.95084	.20	.95036	76.3	.95036	80.3
		.05	.95779	91.1	.96359	89.6
6	.85951	.20	.85910	80.1	.86910	80
		.05	.88114	92.5	.88072	92.9
7	.85828	.20	.87174	72.4	.87174	72.4
		.05	.88366	86.6	.88366	86.6
8	.85951	.20	.87137	73.3	.87137	73.3
		.05	.89049	86.4	.89049	86.4

Table 8. COMPARISON OF PREFERRED PROCEDURE AND ALTERNATE PROCEDURE A (WHEATSTONE BRIDGE)

Case	R_s	α Level	Preferred Procedure		Alternate Procedure	
			$\hat{R}'_{s,L(\alpha)}$	True Confidence Level	$\hat{R}'_{s,L(\alpha)}$	True Confidence Level
29	.99776	.20	.99658	88.6	1.000	63.6
		.05	.99766	95.4	1.000	70.4
30	.99976	.20	.99960	91.4	1.000	66.2
		.05	.99966	98.3	1.000	73.1
31	.99977	.20	.91447	100	.72478	91.8
		.05	.98678	100	1.000	91.8
32	.99977	.20	.92622	100	.99549	93.7
		.05	.99339	100	1.000	93.7

This alternate procedure produced lower confidence limits that were more conservative than those produced by the "preferred procedure".

IV. ALTERNATE PROCEDURE B FOR THE LOWER CONFIDENCE LIMIT FOR SYSTEM RELIABILITY

A. METHODOLOGY

The purpose of this alternate procedure was to construct a procedure that accommodates zero component failures in a different manner than that employed in other procedures. This procedure redefines the estimated value, \hat{a}_i , of a_i , and in so doing uses the Poisson approximation to the Binomial distribution.

Suppose a component, i , undergoes n_i tests where there is a probability of success, p_i , on each component test. Let F_i be the number of failures in these n_i tests. We define $F = \sum_{i=1}^k F_i$.

The 50 percent lower binomial confidence limit for the reliability of component i , $\hat{p}_{i,L}$, can be determined by using its number of failures, F_i , and number of tests, n_i . The upper Binomial confidence limit, $\hat{q}_{i,U}$, is equal to $1 - \hat{p}_{i,L}$. In this alternate procedure we define \hat{a}_i by

$$\hat{a}_i = \frac{\hat{q}_{i,U}}{\hat{q}_{m,U}} \quad (4.1)$$

where $\hat{q}_{m,U} = \max(\hat{q}_{1,U}, \hat{q}_{2,U}, \dots, \hat{q}_{k,U})$. Note that the index will be determined by the data and \hat{a}_i is well defined even if no components fail. The probability distribution of F_i is approximated by the Poisson distribution. The estimated upper confidence limit, $\hat{q}_{m,U(s)}$, can be calculated using Equation (2.7) and the system reliability lower confidence limit, $\hat{R}_{s,L(s)}$, can be obtained from Equation (2.4) where \hat{a}_i is defined by Equation (4.1).

This procedure is used without exception, regardless of the number of system failures.

B. RESULTS

This procedure was evaluated only on series and Wheatstone Bridge systems. Evaluations were limited because a comparison of the results with other procedures discussed in this thesis, indicated this procedure to be less accurate. Selected results illustrating the accuracy of this procedure compared to the accuracy of the "preferred procedure" (Chapter II) are indicated in Tables 9 and 10. The term n^* is computed for Equation (2.10) when the equivalent number of system failures is zero.

Table 9. COMPARISON OF PREFERRED PROCEDURE AND ALTERNATE PROCEDURE B (SERIES SYSTEM)

Case	R_i	α Level	Preferred Procedure		Alternate Procedure	
			$\hat{R}'_{i,L(\alpha)}$	True Confidence Level	$\hat{R}'_{i,L(\alpha)}$	True Confidence Level
1	.95572	.20	.91447	75	.97788	37.1
		.05	.85140	100	.99489	7.4
2	.93206	.20	.88491	100	.97197	70.3
		.05	.85140	100	.99350	33.2
3	.95084	.20	.95361	76	.97294	25.7
		.05	.95969	90	.98732	9.7
4	.85951	.20	.85369	85.2	.93224	26.7
		.05	.86718	94.5	.97896	8.3
5	.95084	.20	.95036	76.3	.97497	30.1
		.05	.95779	91.1	.99155	9.2
6	.85951	.20	.85910	80.1	.91470	39
		.05	.88114	92.5	.97405	7.1
7	.85828	.20	.87174	72.4	.89848	44.5
		.05	.88366	86.6	.94607	19.4
8	.85951	.20	.87137	73.3	.88412	56.3
		.05	.89049	86.4	.93886	24.9

Cases 4, 5, and 6 in Table 9 clearly illustrate a more accurate "preferred procedure" for a series system.

Table 10. COMPARISON OF PREFERRED PROCEDURE AND ALTERNATE PROCEDURE B (WHEATSTONE BRIDGE)

Case	R_s	α Level	Preferred Procedure		Alternate Procedure	
			$\hat{R}'_{s,L(\alpha)}$	True Confidence Level	$\hat{R}'_{s,L(\alpha)}$	True Confidence Level
29	.99776	.20	.99658	88.6	98123	90.1
		.05	.99766	95.4	98235	97.6
30	.99976	.20	.99960	91.4	99810	92.0
		.05	.99966	98.3	99856	99.1
31	.99977	.20	.91447	100	.90478	91.8
		.05	.98678	100	.99900	91.8
32	.99977	.20	.92622	100	90932	100
		.05	.99339	100	99221	100

In the case of the Wheatstone Bridge, Table 10 illustrates that the "preferred procedure" is more accurate than the "alternate procedure".

V. SIMULATION

Standard simulation techniques are used to determine the accuracy of the lower confidence limit procedures. Some of the basic simulation programs were developed by Bellini [Ref. 8: Appendix A]. Each program was modified to incorporate the necessary mathematical formulae that are needed to define the lower confidence limit for a particular procedure.

Input parameters needed to run the computer programs are

- k - number of components in the system
- \bar{n} - vector of component tests (n_1, n_2, \dots, n_k)
- \bar{q} - vector of component unreliabilities (q_1, q_2, \dots, q_k)
- α - level of confidence

Note that the value of system reliability, R_s , is determined by the vector \bar{q} .

The NON-IMSL random number generator, SRND, was used to simulate the success or failure of each test of the k components. From this data, the values of \hat{q}_i , \hat{a}_i , $\hat{q}_{m, U(\alpha)}$, and $\hat{R}_{r, L(\alpha)}$ are calculated. Each scenario was replicated 1000 times to generate 1000 ordered values of $\hat{R}_{r, L(\alpha)}$. The $1000(1 - \alpha)^{\text{th}}$ ordered value, $\hat{R}_{r, L(\alpha), 1000(1-\alpha)}$ from smallest to largest denotes the $100(1 - \alpha)^{\text{th}}$ percentile point of the probability distribution of $\hat{R}_{r, L(\alpha)}$. If the lower confidence limit procedure is exact, $\hat{R}_{r, L(\alpha), 1000(1-\alpha)}$ should equal R_s . The percentiles in all of the tables are the values of $\hat{R}_{r, L(\alpha), 1000(1-\alpha)}$. A true confidence limit is then determined by finding the element of the vector of replications which is closest to R_s and noting its index number, j . The true confidence level is then calculated to be $\frac{j}{1000} \times 100$.

A system report is generated and reports to the analyst the following information.

- q_i - unreliability of each component, i
- a_i - fraction of unreliability of base component, m
- n_i - number of mission tests for each component, i
- R_s - true system reliability
- $\hat{R}_{r, L(\alpha)}$ - estimated lower confidence limit for the $100(1 - \alpha)$ percentile
- difference between R_s and $\hat{R}_{r, L(\alpha)}$
- true confidence level

VI. CONCLUSIONS AND RECOMMENDATIONS

The accuracies of three approximate interval estimation procedures, based on discrete component data, for the reliability of coherent systems were analyzed in this thesis. Computer simulations were used to perform this analysis. Each interval estimation method employs ratios of component failure rate estimates when two or more different component types have at least one failure. This specific ratio feature is needed to extend this work to more complex systems with mixtures of cyclic components and components whose failure times have a continuous probability distribution.

The simulations reveal that the method labeled the "preferred method" in this thesis appears to be reasonably accurate if four or more failures are expected to occur among all components tested. However, any general interpretation of this type is not warranted at this time. The potential for error can be significant if several components have relatively small samples (less than 15) and zero failures. Zero failures joined with small sample sizes will always be the bane of classical interval estimation procedures.

The ratio procedure does allow the possible use of information extraneous to the data. Previous test programs on similar hardware operating under similar environments, as that present for the current test data, might be used to modify the component failure rate estimates or perhaps only the ratios of the failure rate estimates. So called "off the shelf" hardware purchased in accordance with existing DOD specifications would be prime candidates for this type of failure rate modification. Supplementing current test data with other existing "similar" data has become more common as resources for reliability demonstration testing has been reduced.

APPENDIX A. DISCRETE CONFIDENCE LIMIT PROPERTIES

Equations for confidence limits on parameters of discrete probability distributions are not exact. If $\hat{p}_{L(\alpha)}$ is the lower $100(1 - \alpha)$ percent confidence limit for the parameter p in the Binomial distribution, then $\hat{p}_{L(\alpha)}$ is defined so that

$$P(\hat{p}_{L(\alpha)} < p) \geq 1 - \alpha \quad (A1)$$

If the parameter p , is the probability of success on each trial in a sample of size n and s is the observed number of successes then $\hat{p}_{L(\alpha)}$ is the solution for p in the equation

$$\sum_{j=s}^n \binom{n}{j} p^j (1-p)^{n-j} = \alpha \quad (A2)$$

if $s > 0$, and $\hat{p}_{L(\alpha)} = 0$ if $s = 0$. Specifically, suppose $s = n$ then in Equation (A2), $\hat{p}_{L(\alpha)} = \sqrt[n]{\alpha}$. This is the largest value of $\hat{p}_{L(\alpha)}$. Consequently if the true value of p is greater than $\sqrt[n]{\alpha}$ then $P(\hat{p}_{L(\alpha)} \leq p) = 1$. This has important implications when analyzing computer simulations of confidence limit procedures based on discrete data to assess their accuracy. If the value of p used to generate the data on the computer is greater than $\sqrt[n]{\alpha}$, then all of the $\hat{p}_{L(\alpha)}$ values will be smaller than p and the analysis will show the procedure has confidence level 100. This is to be expected when evaluating these confidence interval procedures for some choices of sample sizes and parameter values.

The exact value of $P(\hat{p}_{L(\alpha)} \leq p)$ depends on the sample size, n , and the true value of p . For fixed n , the possible values of s are $0, 1, 2, \dots, n$. Each value of s yields a specific value of $\hat{p}_{L(\alpha)}$, say $p(s)$. Consequently,

$$P(\hat{p}_{L(\alpha)} = p(s) | p) = P(S = s | p) \quad (A3)$$

and

$$P(\hat{p}_{L(\alpha)} < p(s) | p) = P(S < s | p) \quad (A4)$$

If the true value of p equals $\hat{p}_{L(\alpha)}(s)$ for some s then, the probability in Equation (A4) has the value $1 - \alpha$ because this value of p satisfies Equation (A2). Consequently if the true value of p equals any of the values $p(n), p(n-1), \dots, p(1)$, then $P(\hat{p}_{L(\alpha)} < p) = 1 - \alpha$.

For all other values of p , $P(\hat{p}_{L(\alpha)} < p) > 1 - \alpha$. Figure 6 is a sketch of the behavior of this phenomena for $\alpha = .10$ and $n = 5$.

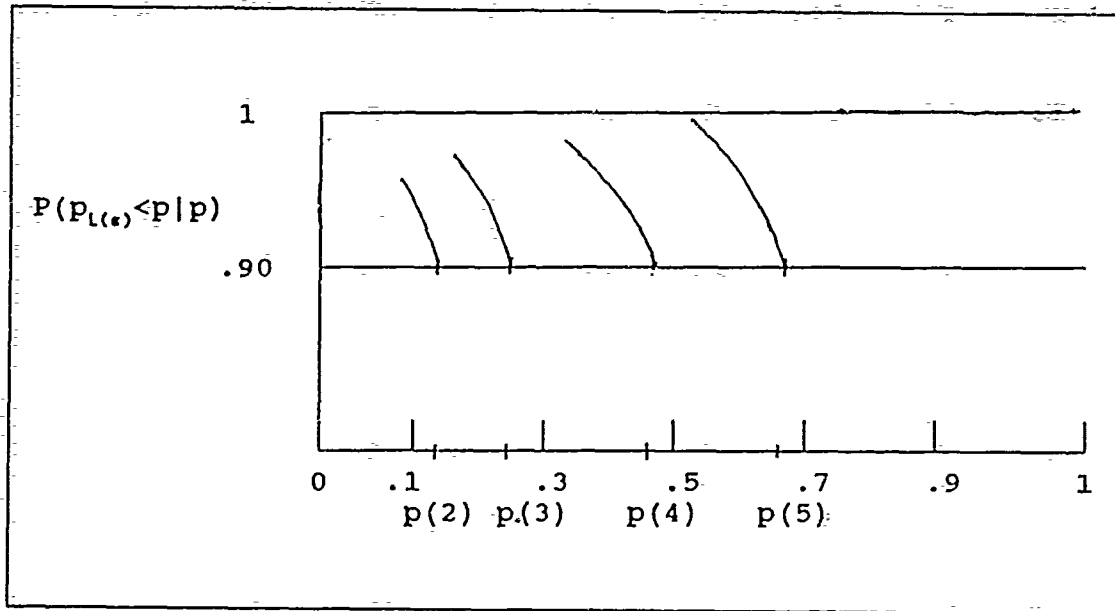


Figure 6. Behavior of $P(\hat{p}_{L(\alpha)} < p)$

APPENDIX B. INPUT PARAMETERS

Table 11. SERIES SYSTEM INPUT PARAMETERS

Case	Components										
		1	2	3	4	5	6	7	8	9	10
1	q_i	.02	.01	.005	.005	.005					
	n_i	30	25	20	10	5					
2	q_i	.02	.01	.005	.005	.005	.005	.005	.005	.005	.005
	n_i	30	25	20	15	10	5	5	5	5	5
3	q_i	.01	.005	.003	.008	.025					
	n_i	200	400	720	265	80					
4	q_i	.005	.01	.015	.02	.025	.005	.01	.015	.02	.025
	n_i	100	50	30	25	20	100	50	30	25	20
5	q_i	.01	.005	.003	.008	.025					
	n_i	150	80	240	265	50					
6	q_i	.005	.01	.015	.02	.025	.005	.01	.015	.02	.025
	n_i	150	20	20	50	25	150	20	20	50	25
7	q_i	.01	.02	.03	.04	.05					
	n_i	300	20	20	100	50					
8	q_i	.005	.01	.015	.02	.025	.005	.01	.015	.02	.025
	n_i	300	20	20	100	50	300	20	20	100	50

Table 12. SERIES-PARALLEL SYSTEM INPUT PARAMETERS

Case	Components										
		1	2	3	4	5	6	7	8	9	10
9	q_i	.02	.01	.005	.005	.005					
	n_i	30	25	20	10	5					
10	q_i	.02	.01	.005	.005	.005	.005	.005	.005	.005	.005
	n_i	100	30	15	10	10	10	10	10	10	10
11	q_i	.01	.02	.03	.04	.05					
	n_i	100	50	30	25	20					
12	q_i	.005	.01	.015	.02	.025	.005	.01	.015	.02	.025
	n_i	100	50	30	25	20	100	50	30	25	20
13	q_i	.01	.005	.003	.008	.025					
	n_i	200	400	720	265	80					
14	q_i	.005	.01	.015	.02	.025	.005	.01	.015	.02	.025
	n_i	300	20	20	100	50	300	20	20	100	50

Table 13. SERIES-PARALLEL WITH A 2 OUT OF 3 COMPONENT SYSTEM INPUT PARAMETERS

Case	Components										
		1	2	3	4	5	6	7	8	9	10
15	q_i	.02	.01	7E-5	2E-5	.005					
	n_i	30	25	20	10	5					
16	q_i	.02	.01	7E-5	2E-5	.005	.005	.005	.005	.005	.005
	n_i	100	30	15	10	10	10	10	10	10	10
17	q_i	.01	.005	2E-5	6E-5	.025					
	n_i	150	80	240	265	50					
18	q_i	.005	.01	7E-5	4E-5	.025	.005	.01	.015	.02	.025
	n_i	150	20	20	50	25	150	20	20	50	25
19	q_i	.01	.005	2E-5	6E-5	.025					
	n_i	200	400	720	265	80					
20	q_i	.005	.01	7E-5	4E-5	.025	.005	.01	.015	.02	.025
	n_i	300	20	20	100	50	300	20	20	100	50
21	q_i	1E-4	.002	7E-5	1E-4	.001	.001	.001	.001	.002	4E-4
	n_i	200	100	60	50	40	40	40	40	50	60
22	q_i	.001	.002	7E-5	.000	.005	.005	.005	.005	.006	.003
	n_i	200	100	60	50	40	40	40	40	50	60
23	q_i	.025	.02	2E-4	2E-4	.001	.001	.001	.001	.002	.005
	n_i	200	100	60	50	40	40	40	40	50	60
24	q_i	.025	.015	7E-5	4E-4	.005	.005	.005	.005	.006	.035
	n_i	200	100	60	50	40	40	40	40	50	60

Table 14. PARALLEL SYSTEM INPUT PARAMETERS

Case	Components										
		1	2	3	4	5	6	7	8	9	10
25	q_i	.01									
	n_i	100									
26	q_i	.04									
	n_i	30									
27	q_i	.06									
	n_i	100									
28	q_i	.06									
	n_i	150									

Table 15. WHEATSTONE BRIDGE INPUT PARAMETERS

Case	Components										
		1	2	3	4	5	6	7	8	9	10
29	q_i	.01	.02	.03	.04	.05					
	n_i	150	20	20	50	25					
30	q_i	.01	.005	.003	.008	.025					
	n_i	150	80	240	265	50					
31	q_i	.02	.01	.005	.005	.005					
	n_i	30	25	20	10	5					
32	q_i	.02	.01	.005	.005	.005					
	n_i	50	30	10	10	5					

APPENDIX C. FORTRAN CODE FOR THE PREFERRED LOWER
CONFIDENCE LIMIT FOR SYSTEM RELIABILITY (SERIES SYSTEM
AND WHEATSTONE BRIDGE SYSTEM)

PROGRAM ZFYSCN

```

*****
*
*   TITLE: BINOMIAL INTERVAL ESTIMATION PROCEDURE
*           ZERO FAILURES ALLOWED; NO SCALING
*   AUTHOR: E. F. BELLINI, LT, USN
*   MODIFIED BY: LT VALERIE A. COVINGTON, USN   (MAR 90)
*   DATE: NOV 89
*
*   THIS PROGRAM COMPUTES THE TRUE CONFIDENCE LEVEL FOR THE ESTIMATE
*   RELIABILITY OF A SERIES AND BRIDGE SYSTEM GIVEN THE RELIABILITY
*   OF THEIR COMPONENTS
*
*   IN ITS PRESENT CONFIGURATION THIS PROGRAM IS SET UP TO RUN 12
*   TIMES EACH TIME PRODUCING 1000 REPLICATIONS USING A DIFFERENT
*   SET OF INPUT DATA.  RUN THE PROGRAM FROM CMS BY TYPING 'B1 EXEC'.
*   THE REXX EXEC PROGRAM
*   'B1' CALLS THE INPUT FILES TO BE READ AND NAMES THE 12 OUTPUT
*   FILES RESULTING FROM THE 12 CONSECUTIVE RUNS.  BY EDITING THE
*   INDEX COUNTERS I, J, K OF THE 'B1' EXEC ONE CAN RUN ANY USER-
*   SPECIFIC RUN FROM JUST ONE RUN TO ALL 12.
*
*   VARIABLES USED
*
*   AHATI : WEIGHT ESTIMATES FOR EACH COMPONENT
*   AI    : INPUT WEIGHTS FOR EACH COMPONENT
*   ALFA  : LEVELS OF SIGNIFICANCE
*   BIGF  : TOTAL NO. OF FAILURES FOR EACH REPLICATION
*   CHISQ : CHI-SQUARE RANDOM VARIABLE VALUE
*   C1C15 : FORMAT LABEL
*   DEGFR : DEGREES OF FREEDOM
*   DELBRG : DIFFERENCE FOR BRIDGE SYSTEM
*   DELSTR : DIFFERENCE FOR SERIES SYSTEM- CLOSED FORM
*   DELTAR : DIFFERENCE FOR SERIES SYSTEM
*   DIFF  : DIFFERENCE (TRUE REL. - ESTIMATED REL. )
*   EPS   : SMALL QUANTITY(CONSTANT)
*   ERROR : PARAMETER FOR IMSL ROUTINE
*   FAILS : COUNTS NO. OF REPLICATIONS WITH AT LST. 1 FAILURE
*   FI    : NO. OF FAILURES FOR EACH COMPONENT(ALL MISSION TST)
*   FLAG  : 1 IF ALL COMP. HAVE SAME NO. OF MISSION TESTS
*   INC   : INCREMENT STEP SIZE FOR ROUTINE USMNMX
*   KEY1  : ARRAY OF INDECES FOR ROUTINE SHSORT
*   KEY2  : ARRAY OF INDECES FOR ROUTINE SHSORT
*   KEY3  : ARRAY OF INDECES FOR ROUTINE SHSORT
*   KEY4  : ARRAY OF INDECES FOR ROUTINE SHSORT
*   KK    : ARRAY SIZING PARAMETER FOR THE MAX NO OF COMPONENTS
*   LOOP  : COUNTS NO. OF REPLICATION PERFORMED
*   MAXALF : MAX NO. OF SIGNIFICANT LEVELS DESIRED(ARRAY SIZING)

```

```

*      MAXREP : MAX NO. OF REPLICATIONS      *
*      MAXRUN : MAX NO. OF PROGRAM ITERATIONS ALLOWED *
*      MSTRQ  : MASTER UNRELIABILITY(USED WITH AI'S TO CALC. QI'S) *
*      MULT   : MULTIPLIER FOR RANDOM NO. GENERATOR SRND *
*      N      : NO. OF MISSION TEST FOR EACH COMPONENT *
*      NIMAX  : MAX NO. OF MISSION TESTS      *
*      NIMIM  : MIN NO. OF MISSION TESTS      *
*      NINDX  : INDEX NO. OF MAX NO. OF MISSION TESTS *
*      NIREAL : NO. OF MISSION TESTS TRANSFORMED TO REAL *
*      NMAX   : MAX NO. OF MISSION TESTS FOR OUTPUT CONTROL *
*      NPRNT  : FLAG FOR DETAILED REPORT OUTPUT *
*      PRNT   : SAME AS ABOVE(PARAMETER) *
*      QHATI  : UNRELIABILITY ESTIMATES FOR EACH COMPONENT *
*      QHTMAX : LARGEST QHATI *
*      QHTUPR : UPPER LIMIT ON SYSTEM UNRELIABILITY *
*      QI     : INPUT UNRELIABILITY FOR EACH COMPONENT *
*      QINDX  : INDEX *
*      QUANTL : QUANTILE *
*      REPSHD : REPLICATIONS HEADING FORMAT NUMBER *
*      RHTSTR : SERIES SYSTEM RELIABILITY ESTIMATE(CLOSED FORM) *
*      RS     : TRUE SERIES SYSTEM RELIABILITY *
*      RSBRDG : TRUE BRIDGE SYSTEM RELIABILITY *
*      RSHAT  : SERIES SYSTEM RELIABILITY ESTIMATE *
*      RSHTBR : BRIDGE SYSTEM RELIABILITY ESTIMATE *
*      SEED   : PARAMETER *
*      SELCTA : SIGNIFICANCE LEVEL SELECTION *
*      SELCTB : SIGNIFICANCE LEVEL SELECTION *
*      SORT   : PARAMETER FOR ROUTINE SRND *
*      SUMNAI : SUM OF THE PRODUCT OF NI'S AND AI'S *
*      TEMP   : TEMPORARY ARRAY *
*      TOTREP : TOTAL NUMBER OF PROGRAM ITERATIONS *
*      TRANBR : TEMPORARY ARRAY *
*      TRANSQ : TEMPORARY ARRAY *
*      TRANSR : TEMPORARY ARRAY *
*      TRIALS : BERNOULLI TRIALS ARRAY (2-DIM) *
*      TRNSTR : TEMPORARY ARRAY *
*      TRUQNT : TRUE QUANTILE *
*      UNIRV  : UNIFORM RANDOM DEVIATES (2-DIM) *
*      ZFAILS : TOTAL NUMBER OF REPLICATIONS WITH ZERO FAILURES *
*      ZFPREP : NO. OF COMPNTS. WITH ZERO FAILURES PER REPLICATION *
*
*****

```

```

PARAMETER (KK=10,MAXALF=2,NPRNT=0)
PARAMETER (MAXREP=1000, MAXRUN=2000, EPS=.000001)
REAL*4 UNIRV(15,1000),TEMP(1000),QI(KK),AI(KK),AHATI(KK)
REAL*4 QHATI(KK), NMAX, NNMAX, QHTMAX, CHISQR(5,5), ALFA(MAXALF)
REAL*4 DF(5),AALFA(5),SUMNAI,RSHAT(MAXALF,MAXREP),RS
REAL*4 KEY1(MAXREP),KEY2(MAXREP),KEY3(MAXREP),TRNSTR(MAXREP)
REAL*4 DEGFR(MAXREP), QHTUPR(MAXALF,MAXREP),CHISQ(MAXALF,MAXREP)
REAL*4 QUPA1(MAXREP), QUPA2(MAXREP),RHTSTR(MAXALF,MAXREP)
REAL*4 DELTAR(MAXALF), TRANSQ(MAXREP),TRANSR(MAXREP),DIFF(MAXREP)
REAL*4 DELSTR(MAXALF),NIMIN,NIMAX,NIREAL(KK)
REAL*4 RSHTBR(MAXALF,MAXREP),DELBRG(MAXALF),KEY4(MAXREP)
REAL*4 TRANBR(MAXREP),RSBRDG ,MSTRQ
REAL*4 ZFPREP

```


REAL*4 AVGN,SUC,STUD

INTEGER SEED, MULT, SORT, TRIALS(15,1000), BIGF, FI(KK), N(KK)
INTEGER NINDX, QINDX, ERROR, REPS, SELCTA, SELCTK, TOTREP
INTEGER C1C15, REPSHD, SELCTB, ALF, FLAG, LOOP, PRNT
INTEGER QUANTL(MAXALF), TRUQNT(MAXALF), ZFAILS, FAILS, INC
INTEGER NTEST, FCT, BFLAG

CHARACTER*8 LOOPSO(MAXREP)

DATA SEED/123457/, MULT/1/, INC/1/
DATA AALFA/.01,.05,.9,.95,.99/, DF/1,5,10,30,40/
DATA ALFA/.20,.050/
DATA SORT/0/

ASSIGN 8 TO C1C15
ASSIGN 9 TO REPSHD

* CALL COMPRS
PRNT = NPRNT

DO 12 I=1, KK
 AI(I) = 9999.
 N(I) = 99999999
12 CONTINUE

READ(03,*)K,MSTRQ

DO 11 I=1, K
 READ(03,*) AI(I), N(I)
11 CONTINUE

IF(K.NE.5) THEN
 WRITE(1,('WARNING: BRIDGE STRUCTURE ''
+''ONLY USES THE FIRST 5 COMPONENTS'''))
 ELSE
 END IF

***INITIALIZE THE QHTUPR ARRAY OF UNRELIABILITY REPLICATIONS, ***
* RSHAT ARRAY OF ESTIMATE SYSTEM RELIABILITY REPLICATIONS *
* AND RHTSTR ARRAY OF EST. SYST. REL. FOR A SERIES SYST WHEN *
***ALL THE COMPONENT MISSION TESTS ARE EQUAL IN NUMBER ***

DO 172 ALF=1, MAXALF
 DO 173 REPS=1, MAXREP
 QHTUPR(ALF, REPS) = 0.
 RSHAT(ALF, REPS) = 0.
 RHTSTR(ALF, REPS) = 0.
 RSHIBR(ALF, REPS) = 0.
 LOOPSO(REPS)='*****'
173 CONTINUE
172 CONTINUE

SET FLAG TO 1 IF ALL COMPONENTS HAVE SAME NO. OF MISSION TESTS

```

FLAG=1
DO 50 I=1,K -1
  IF((N(I) - N(I+1)).NE.0) THEN
    FLAG=0
  ELSE
    END IF
50 CONTINUE
PRINT *, 'FLAG IS:', FLAG

***MAIN PROGRAM OUTER LOOP START(EVERY LOOP IS ONE REPLICATION)***

ZFPREP = 0.
ZFAILS = 0
FAILS = 0
TOTREP = 0
LOOP = 0
10 IF(LOOP.LT.MAXREP) THEN
  LOOP = LOOP + 1
  IF(TOTREP.LT.MAXRUN) THEN
    TOTREP = TOTREP + 1
  SELCTA = 1
  SELCTB = 2

***FILL ARRAY KEY(REPS) WITH INTEGERS 1 TO K TO BE USED AS OUTPUT***
***OF THE SUBROUTINE SHSORT***

DO 95 REPS=1, MAXREP
  KEY1(REPS) = REPS
  KEY2(REPS) = REPS
  KEY3(REPS) = REPS
  KEY4(REPS) = REPS
95 CONTINUE

***CALCULATE NMAX NOT TO PRINT LONGER THAN THE MAX SAMPLE SIZE***

***CALCULATE THE MAXIMUM NO. OF TRIALS AND ITS INDEX NO.***

CALL IMAX(N,K,NMAX,NINDX)

***CALCULATE THE QI'S FROM THE GIVEN MASTER Q AND THE AI'S***

DO 115 I=1, K
  QI(I) = MSTRQ * AI(I)
115 CONTINUE

DO 120 I=1,15
  DO 125 J=1,500
    UNIRV(I,J) = 999.
    TRIALS(I,J) = 99999
125 CONTINUE
120 CONTINUE

*** DRAW UNIFORM (0,1) RV'S AND CONVERT TO BERNOULLI TRIALS***

DO 130 I=1, K

```

```

CALL SRND(SEED, TEMP, N(I), MULT, SORT)
DO 135 J=1, N(I)
  UNIRV(I,J) = TEMP(J)
  IF (UNIRV(I,J).LE. 1 - QI(I)) THEN
    TRIALS(I,J) = 0
  ELSE
    TRIALS(I,J) = 1
  END IF
135 CONTINUE
130 CONTINUE

```

CALCULATE THE NO. OF FAILURES FOR EACH COMPONENT

```

DO 150 I=1, K
  FI(I) = 0
150 CONTINUE
IONECT = 0

```

CALCULATE THE F SUB I'S AND THE GRAND TOTAL NO. OF FAILURES

```

BIGF = 0
DO 155 I=1, K
  DO 160 J=1, N(I)
    FI(I) = FI(I) + TRIALS(I,J)
160 CONTINUE
  IF(FI(I).EQ.0) THEN
    ZFPREP = ZFPREP + 1
  ELSE
    END IF

```

CALCULATE THE QHAT SUB I'S: F SUB I'S DIVIDED BY N SUB I'S

```

QHATI(I) = REAL(FI(I)) / N(I)
BIGF = BIGF + FI(I)
155 CONTINUE

```

COUNTS NUMBER OF COMPONENTS THAT HAVE FAILED

```

DO 156 I=1,K
  IF (FI(I) .NE. 0) IONECT=IONECT+1
156 CONTINUE

```

```

CALL CPARE(FI,K,BFLAG)

```

CASE WHERE NO COMPONENTS HAVE ANY FAILURES

```

AVGN=0.0
DO 200 I=1,K
  AVGN=AVGN+REAL(N(I))
200 CONTINUE
AVGN=AVGN/REAL(K)
IF(BIGF.EQ.0) THEN
  LOOPSO(LOOP)=' *ZERO* '
  ZFAILS = ZFAILS + 1
  DO 205 ALF=1, MAXALF
    RSHAT(ALF,LOOP)= ALFA(ALF)**(1./AVGN)

```

```

        IF(FLAG.EQ.1) THEN
            RHTSTR(ALF,LOOP)=ALFA(ALF)**(1./N(1))
        ELSE
            END IF
        IF (BFLAG.EQ.0) THEN
            RSHTBR(ALF,LOOP)=ALFA(ALF)**(1./AVGN)
        ENDIF
205     CONTINUE
        DEGFR(LOOP) = 2.
        GO TO 10
    ELSE
        FAILS = FAILS + 1
    END IF

```

COUNTS NUMBER OF COMPONENTS THAT FAIL RECORDS NO. COMPT TESTS

```

    FCT=0
    DO 202 I=1,K
        IF (FI(I) .NE. 0) THEN
            FCT=FCT+1
            NTEST=N(I)
        ENDIF
202     CONTINUE

```

FIND THE MAX OF THE INDIVIDUAL COMPONENT UNRELIABILITIES

```

    CALL RMAX(QHATI, K, QHTMAX, QINDX)

```

CALCULATE THE AHAT SUB I'S (WEIGHT ESTIMATES)

IF COMPONENT HAS NO FAILURES AHAT SUB I IS ZERO

```

    SUMNAI = 0.
    DO 165 I=1, K
        AHATI(I) = QHATI(I) / QHTMAX
        SUMNAI = SUMNAI + N(I) * AHATI(I)
165     CONTINUE

```

1 COMPONENT FAILURE SERIES SYSTEM

```

    IF (FCT.EQ.1) THEN
        LOOPSO(LOOP)='*ONECF*'
        DO 305 ALF=1,MAXALF
            SUC=REAL(NTEST-BIGF)
            STUD=FIN(1.-ALFA(ALF),2.*(REAL(BIGF)+1.),2.*SUC)
            RSHAT(ALF,LOOP)=SUC/(SUC+(REAL(BIGF)+1.)*STUD)
            IF (FLAG.EQ.1) THEN
                RHTSTR(ALF,LOOP)=RSHAT(ALF,LOOP)
            ELSE
                ENDIF
        ENDIF
305     CONTINUE
    ENDIF

```

CALCULATE 1 REPLICATION OF UPPR ALFA C.L. ON SYSTEM RELIABILITY

```

    DEGFR(LOOP) = 2 * (1 + BIGF)

```

```

    DO 170 ALF=1, MAXALF

```

```

CALL MDCHI(1 - ALFA(ALF),DEGFR(LOOP),CHISQ(ALF,LOOP), ERROR)
QHTUPR(ALF,LOOP) = CHISQ(ALF,LOOP) / (2 * SUMNAI)
IF(FLAG.EQ.1) THEN
    RHTSTR(ALF,LOOP) = 1 -(CHISQ(ALF,LOOP) / REAL(2*N(1)))
ELSE
    END IF

***CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR COMPNTS. IN SERIES***

IF (FCT .NE. 1) THEN
    CALL RHTSR(S(QHTUPR(ALF,LOOP)), AHATI,K, RSHAT(ALF,LOOP))
ENDIF

***CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR BRIDGE STRUCTURE***

***IF NO SYSTEM FAILURE AND BRIDGE SYSTEM***

IF (BFLAG .EQ. 0) THEN
    RSHTBR(ALF,LOOP)=ALFA(ALF)**(1./AVGN)
ENDIF

***IF MORE THAN 1 SYSTEM FAILURE AND BRIDGE SYSTEM***

IF (BFLAG .EQ. 2) THEN
    CALL RHTBRG(QHTUPR(ALF,LOOP),AHATI,K,RSHTBR(ALF,LOOP))
ENDIF

***EXACTLY 1 SYSTEM FAILURE AND BRIDGE SYSTEM***

IF (BFLAG .EQ. 1) THEN
    SUC=REAL(AVGN-1)
    STUD=FIN(1.-ALFA(ALF),2.*2.,2.*SUC)
    RSHTBR(ALF,LOOP)=SUC/(SUC+2.*STUD)
ENDIF
170 CONTINUE

***THIS ELSE AND ENDIF ARE FOR THE TEST AGAINST MAXRUN***

ELSE
    WRITE(1,(' ',/'PROGRAM EXCEEDED THE MAX NO. OF RUNS',
+' ALLOWED OF: ',I6) ) TOTREP
    GOTO 9999

END IF
GOTO 10
END IF

C WRITE(2,('UNSORTED RSHAT 1 IS: ',/10(F8.5))' )
C +(RSHAT(1,LOOP), LOOP=1, MAXREP)
C WRITE(2,('UNSORTED RSHAT 2 IS: ',/10(F8.5))' )
C +(RSHAT(2,LOOP), LOOP=1, MAXREP)
C IF(FLAG.EQ.1) THEN
C WRITE(2,('UNSORTED RHTSTR 1 IS: ',/10(F8.5))' )
C +(RHTSTR(1,LOOP), LOOP=1, MAXREP)
C WRITE(2,('UNSORTED RHTSTR 2 IS: ',/10(F8.5))' )
C +(RHTSTR(2,LOOP), LOOP=1, MAXREP)
C ELSE

```

```

C     END IF
C     IF(K.EQ.5) THEN
C       WRITE(2,('UNSORTED RSHTBR 1 IS:',/10(F8.5)))
C     +(RSHTBR(1,LOOP), LOOP=1, MAXREP)
C       WRITE(2,('UNSORTED RSHTBR 2 IS:',/10(F8.5)))
C     +(RSHTBR(2,LOOP), LOOP=1, MAXREP)
C     ELSE
C     END IF
C     WRITE (2,('ZERO AND ONE FAILURE REPS:',/10(A8)))
C     + (LOOPS0(LOOP),LOOP=1,MAXREP)

```

SORT THE ARRAYS OF SYSTEM UNRELIABILITIES(1 FOR EACH CONF. LEVEL)

```

      DO 700 ALF=1, MAXALF
        DO 800 REPS=1, MAXREP
          TRANSQ(REPS) = QHTUPR(ALF,REPS)
          TRANSR(REPS) = RSHAT(ALF,REPS)
          TRNSTR(REPS) = RHTSTR(ALF,REPS)
          TRANBR(REPS) = RSHTBR(ALF,REPS)
800      CONTINUE
          CALL SHSORT(TRANSQ,KEY1,MAXREP)
          CALL SHSORT(TRANSR,KEY2,MAXREP)
          CALL SHSORT(TRNSTR,KEY3,MAXREP)
          CALL SHSORT(TRANBR,KEY4,MAXREP)
          DO 900 REPS=1, MAXREP
            QHTUPR(ALF,REPS) = TRANSQ(REPS)
            RSHAT(ALF,REPS) = TRANSR(REPS)
            RHTSTR(ALF,REPS) = TRNSTR(REPS)
            RSHTBR(ALF,REPS) = TRANBR(REPS)
900      CONTINUE
700    CONTINUE

```

PRINT OUTPUT REPORT HEADINGS

```

      WRITE(1,6666)
      WRITE(1,6667) MAXREP
      WRITE(1,6668) K
      WRITE(1,6669)
      IF(K.EQ.5) THEN
        WRITE(1,6699)
      ELSE
      END IF
      WRITE(1,6670) MSTRQ
      WRITE(1,6671)
      WRITE(1,C1C15)
      WRITE(1,3334) AI
      WRITE(1,0007)
      WRITE(1,C1C15)
      WRITE(1,3334) QI
      WRITE(1,0005)
      WRITE(1,C1C15)
      WRITE(1,3335) N
      WRITE(1,6674)

```

COMPUTE THE VALUE RS OF THE TRUE SYSTEM REL. FNCTN. (SERIES SYSTEM)
 *** AND FOR THE 5-COMPONENT BRIDGE STRUCTURE***

```

CALL RSR(S(QI,K,RS)
WRITE(1,(' '),'////' 'THE TRUE SERIES SYSTEM ' ',
+'RELIABILITY VALUE IS: ',T51,F8.5)') RS
CALL RBRIDG(QI,K,RSBRDG)
IF(K.EQ.5) THEN
WRITE(1,(' '),'////' 'THE TRUE BRIDGE STRUCTURE ' ',
+'RELIABILITY VALUE IS: ',T51,F8.5)') RSBRDG
ELSE
END IF
WRITE(1,6675)

```

COMPUTE THE DIFFERENCE 'DELTAR' BTWN. RS AND RSHAT OF THE THEO
 RETICAL QUANTILE GIVEN BY ALFA(MUST USE SORTED RSHAT ARRAY)

```

IF(FLAG.EQ.1) THEN
WRITE(1,5755)
ELSE
END IF
DO 450 ALF=1, MAXALF
  QUANTL(ALF) = MAXREP * (1 - ALFA(ALF))
  DELTAR(ALF) = RS - RSHAT(ALF,QUANTL(ALF))
  DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
  IF(FLAG.EQ.1) THEN
    DELSTR(ALF) = RS - RHTSTR(ALF,QUANTL(ALF))
    WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
    WRITE(1,5656) RHTSTR(ALF,QUANTL(ALF))
    WRITE(1,5657) DELSTR(ALF)
  ELSE
  END IF
  IF(K.EQ.5) THEN
    DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
    WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
    WRITE(1,5666) RSHTBR(ALF,QUANTL(ALF))
    WRITE(1,5667) DELBRG(ALF)
  ELSE
  END IF
  WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
  WRITE(1,5556) RSHAT(ALF,QUANTL(ALF))
  WRITE(1,5557) DELTAR(ALF)
450 CONTINUE
PRINT *, 'QUANTL(1) IS: ', QUANTL(1)
PRINT *, 'QUANTL(2) IS: ', QUANTL(2)

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RSHAT ***** ***

```

WRITE(1,6676)
DO 400 ALF=1,MAXALF
  TRUQNT(ALF) = 0
  DO 500 REPS=1, MAXREP
    DIFF(REPS) = RS - RSHAT(ALF,REPS)
500 CONTINUE
  DO 600 REPS=1, MAXREP
    IF(ABS(DIFF(REPS)).LE.EPS) THEN
      TRUQNT(ALF) = REPS

```

```

        WRITE(1,(' ' ' ',/'TRUE CONFIDENCE LIMIT IS: ',
+         F8.4)')
+         (TRUQNT(ALF) / REAL(MAXREP)) * 100.
        GO TO 620
    ELSEIF(DIFF(REPS).LT.0.) THEN
        TRUQNT(ALF) = REPS
        GO TO 610
    ELSE
    END IF
600 CONTINUE
610 IF(TRUQNT(ALF).EQ.0.) THEN
    WRITE(1,4443) ALFA(ALF)
    WRITE(1,(' ' ' ',/'THE SMALLEST',
+     ' ' DIFFERENCE BETWEEN RS AND RSHAT IS: ',F10.5)') DIFF(
+     MAXREP)
    ELSEIF(TRUQNT(ALF).EQ.1.) THEN
    WRITE(1,4442) ALFA(ALF)
    WRITE(1,(' ' ' ',/'ALL RSHAT',
+     ' ' ARE GREATER THAN RS''))
    ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+     THEN
    WRITE(1,4444) ALFA(ALF),
+     (TRUQNT(ALF) / REAL(MAXREP)) * 100.
    WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF))
    WRITE(1,4446)
    ELSE
+     WRITE(1,4444) ALFA(ALF),
+     ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
    WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF)-1)
    WRITE(1,4447)
620 END IF
400 CONTINUE

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ****RSHTBR (BRIDGE) **** ***

```

    IF(K.EQ.5) THEN
    DO 401 ALF=1,MAXALF
    TRUQNT(ALF) = 0
    DO 501 REPS=1, MAXREP
    DIFF(REPS) = RSBRDG - RSHTBR(ALF,REPS)
501 CONTINUE
    DO 601 REPS=1, MAXREP
    IF(ABS(DIFF(REPS)).LE.EPS) THEN
    TRUQNT(ALF) = REPS
    WRITE(1,(' ' ' ',/'TRUE CONFIDENCE LIMIT IS: ',
+     F8.4)')
+     (TRUQNT(ALF) / REAL(MAXREP)) * 100.
    GO TO 621
    ELSEIF(DIFF(REPS).LT.0.) THEN
    TRUQNT(ALF) = REPS
    GO TO 611
    ELSE
    END IF
601 CONTINUE

```



```

611      IF(TRUQNT(ALF).EQ.0.) THEN
          WRITE(1,4443) ALFA(ALF)
          WRITE(1,(' ' ' ',/'THE SMALLEST' ',
+         ' ' DIFFERENCE BETWEEN RSBRDG AND RSHTBR IS: ' ',
+         F10.5)') DIFF(MAXREP)
          ELSEIF(TRUQNT(ALF).EQ.1.) THEN
          WRITE(1,4442) ALFA(ALF)
          WRITE(1,(' ' ' ',/'ALL RSHTBR' ',
+         ' ' ARE GREATER THAN RSBRDG' '))
          ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+         THEN
          WRITE(1,4444) ALFA(ALF),
+         (TRUQNT(ALF) / REAL(MAXREP)) * 100.
          WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF))
          WRITE(1,4446)
          ELSE
+         WRITE(1,4444) ALFA(ALF),
+         ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
          WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF)-1)
          WRITE(1,4447)
621      END IF
401      CONTINUE
          ELSE
          END IF

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** RHTSTR ***

```

          IF(FLAG.EQ.1) THEN
          DO 4400 ALF=1,MAXALF
          TRUQNT(ALF) = 0
          DO 5500 REPS=1, MAXREP
          DIFF(REPS) = RS - RHTSTR(ALF,REPS)
5500      CONTINUE
          DO 6600 REPS=1, MAXREP
          IF(ABS(DIFF(REPS)).LE.EPS) THEN
          TRUQNT(ALF) = REPS
          WRITE(1,(' ' ' ',/'TRUE CONFIDENCE LIMIT IS: ' ',
+         F8.4)')
+         (TRUQNT(ALF) / REAL(MAXREP)) * 100.
          GO TO 6620
          ELSEIF(DIFF(REPS).LT.0.) THEN
          TRUQNT(ALF) = REPS
          GO TO 6610
          ELSE
          END IF
6600      CONTINUE
6610      IF(TRUQNT(ALF).EQ.0.) THEN
          WRITE(1,4443) ALFA(ALF)
          WRITE(1,(' ' ' ',/'THE SMALLEST' ',
+         ' ' DIFFERENCE BETWEEN RS AND RHTSTR IS: ' ',
+         F9.5)') DIFF(MAXREP)
          ELSEIF(TRUQNT(ALF).EQ.1.) THEN
          WRITE(1,4442) ALFA(ALF)
          WRITE(1,(' ' ' ',/'ALL RHTSTR' ',

```

```

+      ' ARE GREATER THAN RS''')
+      ELSEIF(ABS(DIFF(TRUQNT(ALF))). LE. ABS(DIFF(TRUQNT(ALF) - 1)))
+      THEN
+          WRITE(1,4444) ALFA(ALF),
+          (TRUQNT(ALF) / REAL(MAXREP)) * 100.
+          WRITE(1,4448) RHTSTR(ALF, TRUQNT(ALF))
+          WRITE(1,4446)
+      ELSE
+          WRITE(1,4444) ALFA(ALF),
+          ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
+          WRITE(1,4448) RHTSTR(ALF, TRUQNT(ALF)-1)
+          WRITE(1,4447)
6620      END IF
4400 CONTINUE
      ELSE
      END IF

```

PRINT THE ARRAYS PERTINENT TO THE OUPUT OF EACH REPLICATION

```

      IF(PRNT.EQ.1) THEN
      I = 1
185  WRITE(1,REPSHD) ALFA(SELCTA), ALFA(SELCTA),
+ALFA(SELCTB),ALFA(SELCTB),ALFA(SELCTA),ALFA(SELCTA),ALFA(SELCTB),
+ALFA(SELCTB)
175  IF(I.GE.(MAXREP + 1)) THEN
      GOTO 180
      ELSE
+      IF( (I.EQ.71).OR.(I.EQ.211).OR.(I.EQ.351).OR.(I.EQ.491).OR.
+      (I.EQ.631).OR.(I.EQ.771).OR.(I.EQ.911).OR.(I.EQ.1051) ) THEN
+          I = I + 70
+          WRITE(1,(''+'))
+          GOTO 185
      ELSE
+          WRITE(1,3336) I, INT(DEGFR(I)), CHISQ(1,I), QHTUPR(1,I),
+          CHISQ(2,I), QHTUPR(2,I)
+          END IF
+          IF(I + 70.LE.MAXREP) THEN
+          WRITE(1,3337) I+70,INT(DEGFR(I+70)),CHISQ(1,I+70),
+          QHTUPR(1,I+70),CHISQ(2,I+70),QHTUPR(2,I+70)
+          ELSE
+          END IF
+          I = I + 1
+          GOTO 175
180  END IF
      ELSE
      ENDIF
9999 WRITE(1,(''THE TOTAL NO OF REPS WAS:'' ,I8)) TOTREP
      WRITE(1,(''THE TOTAL NO OF EFFECTIVE REPS WAS:'' ,I8)) LOOP
      WRITE(1,(''THE TOTAL NO OF NO FAILURE RUNS WAS:'' ,I8)) ZFAILS
      WRITE(1,(''AVERAGE NO. OF COMPONENTS PER REPLICATION WITH '' ,
+'' NO FAILURES:'' ,F5.2)) ZFPREP / MAXREP
      WRITE(1,(''THE TOTAL NO OF RUNS WITH FAILURES WAS:'' ,I8)) FAILS
0008 FORMAT (/ 3X,'C 1',5X,'C 2',
+5X,'C 3',5X,'C 4',5X,'C 5',5X,'C 6',5X,'C 7',5X,
+'C 8',5X,'C 9',5X,'C 10',4X,'C 11',4X,

```

```

      + 'C 12',4X,'C 13',4X,'C 14',4X,'C 15')
0009 FORMAT(/1X,'REP NO',2X,'DF',1X,'CHISQR(',F4.3,')',1X,
      + 'QHTUPR(',F4.3,')',1X,'CHISQR(',F4.3,')',1X,'QHTUPR(',F4.3,')',
      + 2X,'REP NO',2X,'DF',1X,'CHISQR(',F4.3,')',1X,
      + 'QHTUPR(',F4.3,')',1X,'CHISQR(',F4.3,')',1X,'QHTUPR(',F4.3,')'//)
0001 FORMAT (///'UNIFORM RANDOM DEVIATES ARE: ')
0002 FORMAT (///'BERNOULLI TRIALS ARE: ')
0003 FORMAT (///'TOTAL NO. OF FAILURES FOR EACH COMPONENT: ')
0004 FORMAT (///'ESTIMATED UNRELIABILITY FOR EACH COMPONENT: ')
0005 FORMAT (///'TOTAL NUMBER OF MISSION TESTS: ')
0006 FORMAT (///'ESTIMATED WEIGHTS FOR EACH COMPONENT: ')
0007 FORMAT (///'Q I FOR EACH COMPONENT: ')
1111 FORMAT (15F8.5)

2222 FORMAT (/1X,15(I4,4X))
3333 FORMAT (/1X,15(I4,4X))
3334 FORMAT (/15F8.5)
3335 FORMAT (/1X,15(I4,4X))
3336 FORMAT (T3,I4,T9,I3,T13,F11.5,T27,F8.5,T39,F11.5,T53,F8.5)
3337 FORMAT ('+',T67,I4,T73,I3,T77,F11.5,T91,F8.5,T103,F11.5,T117,F8.5)
4442 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
      + 'LIMIT IS: ',T50,' 00.000 ')
4443 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
      + 'LIMIT IS: ',T50,' 100.0000 ')
4444 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
      + 'LIMIT IS: ',T50,F8.4)
4445 FORMAT (' ',/'THE RSHAT VALUE CLOSEST TO RS IS: ',T51,F8.5)
4446 FORMAT (' ',/'(FIRST NEGATIVE DIFFERENCE)')
4447 FORMAT (' ',/'(ELEMENT PRECEEDING FIRST NEGATIVE DIFFERENCE)')
4448 FORMAT (' ',/'THE RHTSTR VALUE CLOSEST TO RS IS: ',T51,F8.5)
4449 FORMAT (' ',/'THE RSHTBR VALUE CLOSEST TO RSBRDG IS: ',T51,F8.5)
5555 FORMAT (' ',///'THE ',I4,'(1-',F4.3,') QUANTILE IS: ',T49,F8.3)
5556 FORMAT (' ',/'THE VALUE OF RSHAT FOR THAT QUANTILE IS: ',T51,F8.5)
5557 FORMAT (' ',/'THE DIFFERENCE(RS - RSHAT) IS: ',T51,F8.5)
5656 FORMAT (' ',/'THE VALUE OF RHTSTR FOR THAT QUANTILE IS: ',T51,F8.5)
5666 FORMAT (' ',/'THE VALUE OF RSHTBR FOR THAT QUANTILE IS: ',T51,F8.5)
5657 FORMAT (' ',/'THE DIFFERENCE(RS - RHTSTR) IS: ',T51,F8.5)
5667 FORMAT (' ',/'THE DIFFERENCE(RS - RSHTBR) IS: ',T51,F8.5)
5755 FORMAT (' ',///'SINCE THE NO. OF MISSION TESTS IS THE SAME FOR',
      + ' ALL COMPONENTS THE CLOSED FORM SERIES SYSTEM RELIABILITY ',
      + ''RHTSTR' IS COMPUTED')
6666 FORMAT ('+', '*****',
      + '***** RUN INPUT SETTINGS *****',
      + '*****')
6667 FORMAT (' ',/'NUMBER OF REPLICATIONS: ',T50,I4)
6668 FORMAT (' ',/'NUMBER OF COMPONENTS: ',T50,I4)
6669 FORMAT (' ',/'SYSTEM RELIABILITY FUNCTION: ',T50,'SERIES')
6699 FORMAT (' ',/'SYSTEM RELIABILITY FUNCTION: ',T50,'BRIDGE')
6670 FORMAT (' ',/'MASTER UNRELIABILITY USED: ',T50,F8.5)
6671 FORMAT (' ',/'INPUT WEIGHTS(A SUB I'S): ')
6674 FORMAT ('+',///'*****',
      + '**R U N R E S U L T S*****',
      + '*****')
6675 FORMAT ('+',///'*****',
      + ' ESTIMATE ERRORS *****',
      + '*****')

```

```
6676 FORMAT ('+',//'*****',  
+ ' TRUE CONFIDENCE LIMITS *****',  
+ '*****')  
END
```

APPENDIX D. FORTRAN CODE FOR THE PREFERRED LOWER
CONFIDENCE LIMIT FOR SYSTEM RELIABILITY (SERIES PARALLEL
SYSTEM)

PROGRAM ZFYSCN

```

*****
*
* TITLE: BINOMIAL INTERVAL ESTIMATION PROCEDURE
* ZERO FAILURES ALLOWED; NO SCALING
* AUTHOR: E. F. BELLINI, LT, USN
* MODIFIED BY: LT VALERIE A. COVINGTON, USN (MAR 90)
* DATE: NOV 89
*
* THIS PROGRAM COMPUTES THE TRUE CONFIDENCE LEVEL FOR THE ESTIMATE
* RELIABILITY OF A SERIES AND BRIDGE SYSTEM GIVEN THE RELIABILITY
* OF THEIR COMPONENTS
*
* IN ITS PRESENT CONFIGURATION THIS PROGRAM IS SET UP TO RUN 12
* TIMES EACH TIME PRODUCING 1000 REPLICATIONS USING A DIFFERENT
* SET OF INPUT DATA. RUN THE PROGRAM FROM CMS BY TYPING 'B1 EXEC'.
* THE REXX EXEC PROGRAM
* 'B1' CALLS THE INPUT FILES TO BE READ AND NAMES THE 12 OUTPUT
* FILES RESULTING FROM THE 12 CONSECUTIVE RUNS. BY EDITING THE
* INDEX COUNTERS I, J, K OF THE 'B1' EXEC ONE CAN RUN ANY USER-
* SPECIFIC RUN FROM JUST ONE RUN TO ALL 12.
*
* VARIABLES USED
*
* AHATI : WEIGHT ESTIMATES FOR EACH COMPONENT
* AI : INPUT WEIGHTS FOR EACH COMPONENT
* ALFA : LEVELS OF SIGNIFICANCE
* BIGF : TOTAL NO. OF FAILURES FOR EACH REPLICATION
* CHISQ : CHI-SQUARE RANDOM VARIABLE VALUE
* C1C15 : FORMAT LABEL
* DEGFR : DEGREES OF FREEDOM
* DELBRG : DIFFERENCE FOR BRIDGE SYSTEM
* DELSTR : DIFFERENCE FOR SERIES SYSTEM- CLOSED FORM
* DELTAR : DIFFERENCE FOR SERIES SYSTEM
* DIFF : DIFFERENCE (TRUE REL. - ESTIMATED REL. )
* EPS : SMALL QUANTITY(CONSTANT)
* ERROR : PARAMETER FOR IMSL ROUTINE
* FAILS : COUNTS NO. OF REPLICATIONS WITH AT LST. 1 FAILURE
* FI : NO. OF FAILURES FOR EACH COMPONENT(ALL MISSION TST)*
* FLAG : 1 IF ALL COMP. HAVE SAME NO. OF MISSION TESTS
* INC : INCREMENT STEP SIZE FOR ROUTINE USMNMX
* KEY1 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY2 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY3 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY4 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KK : ARRAY SIZING PARAMETER FOR THE MAX NO OF COMPONENTS*
* LOOP : COUNTS NO. OF REPLICATION PERFORMED
* MAXALF : MAX NO. OF SIGNIFICANT LEVELS DESIRED(ARRAY SIZING)*

```

```

*      MAXREP : MAX NO. OF REPLICATIONS                                *
*      MAXRUN : MAX NO. OF PROGRAM ITERATIONS ALLOWED                  *
*      MSTRQ  : MASTER UNRELIABILITY USED WITH AI'S TO CALC. QI'S)    *
*      MULT   : MULTIPLIER FOR RANDOM NO. GENERATOR SRND              *
*      N      : NO. OF MISSION TEST FOR EACH COMPONENT                 *
*      NIMAX  : MAX NO. OF MISSION TESTS                               *
*      NIMIM  : MIN NO. OF MISSION TESTS                              *
*      NINDX  : INDEX NO. OF MAX NO. OF MISSION TESTS                 *
*      NIREAL : NO. OF MISSION TESTS TRANSFORMED TO REAL              *
*      NMAX   : MAX NO. OF MISSION TESTS FOR OUTPUT CONTROL          *
*      NPRNT  : FLAG FOR DETAILED REPORT OUTPUT                       *
*      PRNT   : SAME AS ABOVE(PARAMETER)                              *
*      QHATI  : UNRELIABILITY ESTIMATES FOR EACH COMPONENT           *
*      QHTMAX : LARGEST QHATI                                          *
*      QHTUPR : UPPER LIMIT ON SYSTEM UNRELIABILITY                  *
*      QI     : INPUT UNRELIABILITY FOR EACH COMPONENT                *
*      QINDX  : INDEX                                                 *
*      QUANTL : QUANTILE                                              *
*      REPSHD : REPLICATIONS HEADING FORMAT NUMBER                   *
*      RHTSTR : SERIES SYSTEM RELIABILITY ESTIMATE(CLOSED FORM)      *
*      RS     : TRUE SERIES SYSTEM RELIABILITY                       *
*      RSBRDG : TRUE BRIDGE SYSTEM RELIABILITY                       *
*      RSHAT  : SERIES SYSTEM RELIABILITY ESTIMATE                    *
*      RSHTBR : BRIDGE SYSTEM RELIABILITY ESTIMATE                   *
*      SEED   : PARAMETER                                             *
*      SELCTA : SIGNIFICANCE LEVEL SELECTION                          *
*      SELCTB : SIGNIFICANCE LEVEL SELECTION                          *
*      SORT   : PARAMETER FOR ROUTINE SRND                            *
*      SUMNAI : SUM OF THE PRODUCT OF NI'S AND AI'S                   *
*      TEMP   : TEMPORARY ARRAY                                       *
*      TOTREP : TOTAL NUMBER OF PROGRAM ITERATIONS                    *
*      TRANBR : TEMPORARY ARRAY                                       *
*      TRANSQ : TEMPORARY ARRAY                                       *
*      TRANSR : TEMPORARY ARRAY                                       *
*      TRIALS : BERNOULLI TRIALS ARRAY (2-DIM)                        *
*      TRNSTR : TEMPORARY ARRAY                                       *
*      TRUQNT : TRUE QUANTILE                                         *
*      UNIRV  : UNIFORM RANDOM DEVIATES (2-DIM)                       *
*      ZFAILS : TOTAL NUMBER OF REPLICATIONS WITH ZERO FAILURES      *
*      ZFPREP : NO. OF COMPNTS. WITH ZERO FAILURES PER REPLICATION  *
*
*****

```

```

PARAMETER (KK=10,MAXALF=2,NPRNT=0)
PARAMETER (MAXREP=1000, MAXRUN=2000, EPS=.000001)
REAL*4 UNIRV(15,1000),TEMP(1000),QI(KK),AI(KK),AHATI(KK)
REAL*4 QHATI(KK), NMAX, NNMAX, QHTMAX, CHISQR(5,5), ALFA(MAXALF)
REAL*4 DF(5),AALFA(5),SUMNAI,RSHAT(MAXALF,MAXREP),RS
REAL*4 KEY1(MAXREP),KEY2(MAXREP),KEY3(MAXREP),TRNSTR(MAXREP)
REAL*4 DEGFR(MAXREP), QHTUPR(MAXALF,MAXREP),CHISQ(MAXALF,MAXREP)
REAL*4 QUPA1(MAXREP), QUPA2(MAXREP),RHTSTR(MAXALF,MAXREP)
REAL*4 DELTAR(MAXALF), TRANSQ(MAXREP),TRANSR(MAXREP),DIFF(MAXREP)
REAL*4 DELSTR(MAXALF),NIMIN,NIMAX,NIREAL(KK)
REAL*4 RSHTBR(MAXALF,MAXREP),DELBRG(MAXALF),KEY4(MAXREP)
REAL*4 TRANBR(MAXREP),RSBRDG ,MSTRQ
REAL*4 ZFPREP

```

```

REAL*4 AVGN,SUC,STUD,PTEMP(10),FDEG1,FDEG2,P3,S3,S4

INTEGER SEED, MULT, SORT, TRIALS(15,1000), BIGF, FI(KK), N(KK)
INTEGER NINDX, QINDX, ERROR, REPS, SELCTA, SELCTK, TOTREP
INTEGER C1C15, REPSHD, SELCTB, ALF, FLAG, LOOP, PRNT
INTEGER QUANTL(MAXALF), TRUQNT(MAXALF), ZFAILS, FAILS, INC
INTEGER NTEST, FCT, HFI

CHARACTER*8 LOOPSO(MAXREP)

DATA SEED/123457/, MULT/1/, INC/1/
DATA AALFA/.01,.05,.9,.95,.99/, DF/1,5,10,30,40/
DATA ALFA/.20,.050/
DATA SORT/0/

ASSIGN 8 TO C1C15
ASSIGN 9 TO REPSHD

* CALL COMPRS
  PRNT = NPRNT

DO 12 I=1, KK
  AI(I) = 9999.
  N(I) = 99999999
12 CONTINUE

  READ(03,*) K, MSTRQ

DO 11 I=1, K
  READ(03,*) AI(I), N(I)
11 CONTINUE

  IF(K.NE.5) THEN
    WRITE(1,('WARNING: BRIDGE STRUCTURE ',
+ 'ONLY USES THE FIRST 5 COMPONENTS'))
  ELSE
    END IF

***INITIALIZE THE QHTUPR ARRAY OF UNRELIABILITY REPLICATIONS, ***
* RSHAT ARRAY OF ESTIMATE SYSTEM RELIABILITY REPLICATIONS *
* AND RHTSTR ARRAY OF EST. SYST. REL. FOR A SERIES SYST WHEN *
***ALL THE COMPONENT MISSION TESTS ARE EQUAL IN NUMBER ***

DO 172 ALF=1, MAXALF
  DO 173 REPS=1, MAXREP
    QHTUPR(ALF, REPS) = 0.
    RSHAT(ALF, REPS) = 0.
    RHTSTR(ALF, REPS) = 0.
    RSHTR(ALF, REPS) = 0.
    LOOPSO(REPS)='*****'
  173 CONTINUE
172 CONTINUE

***SET FLAG TO 1 IF ALL COMPONENTS HAVE SAME NO. OF MISSION TESTS***

```

```

FLAG=1
DO 50 I=1,K -1
  IF((N(I) - N(I+1)).NE.0) THEN
    FLAG=0
  ELSE
    END IF
50 CONTINUE
PRINT *, 'FLAG IS:', FLAG

***MAIN PROGRAM OUTER LOOP START(EVERY LOOP IS ONE REPLICATION)***

ZFPREP = 0.
ZFAILS = 0
FAILS = 0
TOTREP = 0
LOOP = 0
10 IF(LOOP.LT.MAXREP) THEN
  LOOP = LOOP + 1
  IF(TOTREP.LT.MAXRUN) THEN
    TOTREP = TOTREP + 1
  SELCTA = 1
  SELCTB = 2

***FILL ARRAY KEY(REPS) WITH INTEGERS 1 TO K TO BE USED AS OUTPUT***
***OF THE SUBROUTINE SHSORT***

DO 95 REPS=1, MAXREP
  KEY1(REPS) = REPS
  KEY2(REPS) = REPS
  KEY3(REPS) = REPS
  KEY4(REPS) = REPS
95 CONTINUE

***CALCULATE NMAX NOT TO PRINT LONGER THAN THE MAX SAMPLE SIZE***

***CALCULATE THE MAXIMUM NO. OF TRIALS AND ITS INDEX NO.***

CALL IMAX(N,K,NMAX,NINDX)

***CALCULATE THE QI'S FROM THE GIVEN MASTER Q AND THE AI'S***

DO 115 I=1, K
  QI(I) = MSTRQ * AI(I)
115 CONTINUE
C
  S3=QI(3)
  S4=QI(1)**.50
DO 120 I=1,15
  DO 125 J=1,500
    UNIRV(I,J) = 999.
    TRIALS(I,J) = 99999
125 CONTINUE
120 CONTINUE

*** DRAW UNIFORM (0,1) RV'S AND CONVERT TO BERNOULLI TRIALS***
DO 130 I=1, K

```



```

        CALL SRND(SEED, TEMP, N(I), MULT, SORT)
        DO 135 J=1, N(I)
            UNIRV(I,J) = TEMP(J)
            IF (UNIRV(I,J).LE. 1 - QI(I)) THEN
                TRIALS(I,J) = 0
            ELSE
                TRIALS(I,J) = 1
            END IF
135  CONTINUE
130  CONTINUE

```

CALCULATE THE NO. OF FAILURES FOR EACH COMPONENT

```

        DO 150 I=1, K
            FI(I) = 0
150  CONTINUE
        IONECT = 0

```

CALCULATE THE F SUB I'S AND THE GRAND TOTAL NO. OF FAILURES

```

        BIGF = 0
        DO 155 I=1, K
            DO 160 J=1, N(I)
                FI(I) = FI(I) + TRIALS(I,J)
160  CONTINUE
            IF (FI(I) .EQ. 0) THEN
                ZFPREP=ZFPREP+1
            ELSE
                ENDIF
            BIGF = BIGF + FI(I)

            QHATI(I) = REAL(FI(I)) / N(I)
155  CONTINUE

```

CHANGES FOR SERIES PARALLEL SYSTEM (COMPONENT 2)

```

        IF (FI(2) .EQ. 0) THEN
            ZFPREP=ZFPREP - 1
        ENDIF
        BIGF = BIGF - FI(2)
        FI(1) = 0
        DO 161 I=1,N(2)
            HFI=0
            DO 162 J=1,2
                CALL SRND(SEED, PTEMP(J), N(2), MULT, SORT)
                IF (PTEMP(J) .GT. 1-S4) THEN
                    HFI = HFI + 1
                ENDIF
162  CONTINUE
            IF (HFI .EQ. 2) THEN
                FI(2) = FI(2) + 1
            ENDIF
161  CONTINUE
            BIGF = BIGF + FI(2)
            IF (FI(2) .EQ. 0) THEN
                ZFPREP = ZFPREP + 1

```

```

ELSE
ENDIF
G IF (FI(3) .EQ. 0) THEN
C ZFPREP=ZFPREP - 1
C
C ENDIF
C BIGF = BIGF - FI(3)
C FI(3) = 0
C DO 163 I=1,N(3)
C HFI=0
C DO 164 J=1,3
C CALL SRND(SEED, PTEMP(J), N(3), MULT, SORT)
C IF (PTEMP(J) .GT. 1-S3) THEN
C HFI = HFI + 1
C ENDIF
C164 CONTINUE
C IF (HFI .GE. 2) THEN
C FI(3) = FI(3) + 1
C ENDIF
C163 CONTINUE
C BIGF = BIGF + FI(3)
C IF (FI(3) .EQ. 0) THEN
C ZFPREP = ZFPREP + 1
C ELSE
C ENDIF
C DO 19 I=1,K
C QI(I) = MSTRQ * AI(I)
C QHATI(I) = REAL(FI(I)) / N(I)

19 CONTINUE
C QI(4) = QI(4)**2
C P3 = 1. -QI(3)
C QI(3) = 1. -((3.*P3**2*QI(3))+(P3**3))
***COUNTS NUMBER OF COMPONENTS THAT HAVE FAILED***

DO 156 I=1,K
IF (FI(I) .NE. 0) IONECT=IONECT+1
156 CONTINUE

***CASE WHERE NO COMPONENTS HAVE ANY FAILURES***

IF(BIGF.EQ.0) THEN
LOOPS0(LOOP)=' *ZERO* '
ZFAILS = ZFAILS + 1
AVGN=0.0
DO 200 I=1,K
AVGN=AVGN+REAL(N(I))
200 CONTINUE
AVGN=AVGN/REAL(K)
DO 205 ALF=1, MAXALF
RSHAT(ALF,LOOP)= ALFA(ALF)**(1./AVGN)
IF(FLAG.EQ.1) THEN
RHTSTR(ALF,LOOP)=ALFA(ALF)**(1./N(1))
ELSE
END IF
* PRINT *,LOOP', LOOP, 'RSHAT', RSHAT(ALF,LOOP)
205 CONTINUE

```

```

                DEGFR(LOOP) = 2.
                GO TO 10
ELSE
    FAILS = FAILS + 1
END IF

***COUNTS NUMBER OF COMPONENTS THAT FAIL RECORDS NO. COMPT TESTS***
    FCT=0
    DO 202 I=1,K
        IF (FI(I) .NE. 0) THEN
            FCT=FCT+1
            NTEST=N(I)
        ENDIF
    202 CONTINUE

***FIND THE MAX OF THE INDIVIDUAL COMPONENT UNRELIABILITIES***

    CALL RMAX(QHATI, K, QHTMAX, QINDX)
    IF (LOOP .EQ. 1) THEN
        ENDIF

***CALCULATE THE AHAT SUB I'S (WEIGHT ESTIMATES)***
***IF COMPONENT HAS NO FAILURES AHAT SUB I IS ZERO***

    SUMNAI = 0.
    DO 165 I=1, K
        AHATI(I) = QHATI(I) / QHTMAX
        SUMNAI = SUMNAI + N(I) * AHATI(I)
    165 CONTINUE

***1 COMPONENT FAILURE SERIES SYSTEM***

    IF (FCT .EQ. 1) THEN
        LOOPS0(LOOP)=' *ONECF* '
        DO 305 ALF=1,MAXALF
            SUC=REAL(NTEST-BIGF)
            FDEG1=2.*(REAL(BIGF)+1.)
            FDEG2=2.*SUC
            STUD=FIN(1.-ALFA(ALF),FDEG1,FDEG2)
            RSHAT(ALF,LOOP)=SUC/(SUC+(REAL(BIGF)+1.)*STUD)
*           PRINT *, 'SUC=', SUC, 'FAIL=', BIGF, 'NTEST=', NTEST
*           PRINT *, 'FIN=', STUD
*           PRINT *, 'ALFA=', ALFA(ALF), 'RSHAT=', RSHAT(ALF, LOOP)
            IF (FLAG .EQ. 1) THEN
                RHTSTR(ALF, LOOP)=RSHAT(ALF, LOOP)
            ELSE
                ENDIF
        305 CONTINUE
    ENDIF

***CALCULATE 1 REPLICATION OF UPPR ALFA C. L. ON SYSTEM RELIABILITY***

    DEGFR(LOOP) = 2 * (1 + BIGF)

    DO 170 ALF=1, MAXALF
        CALL MDCHI(1 - ALFA(ALF), DEGFR(LOOP), CHISQ(ALF, LOOP), ERROR)

```

```

QHTUPR(ALF,LOOP) = CHISQ(ALF,LOOP) / (2 * SUMNAI)
IF(FLAG.EQ.1) THEN
    RHTSTR(ALF,LOOP) = 1 -(CHISQ(ALF,LOOP) / REAL(2*N(1)))
ELSE
    END IF

```

CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR COMPNTS. IN SERIES

```

IF (FCT .NE. 1) THEN
    CALL RHTSR(S(QHTUPR(ALF,LOOP), AHATI,K, RSHAT(ALF,LOOP))
ENDIF

```

CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR BRIDGE STRUCTURE

```

IF (IONECT .NE. 1) THEN
    CALL RHTBRG(QHTUPR(ALF,LOOP),AHATI,K,RSHTBR(ALF,LOOP))
ENDIF

```

170 CONTINUE

EXACTLY 1 COMPONENT FAILS AND REDUNDANT COMPONENT

```

IF ((IONECT .EQ. 1) .AND. (K .EQ.5)) THEN
    DO 207 I=1, K
        NIREAL(I) = REAL(N(I))
207    CONTINUE
    CALL USMNMX(NIREAL,K,INC,NIMIN,NIMAX)
    DO 206 ALF=1,MAXALF
        RSHTBR(ALF,LOOP)=ALFA(ALF)**(1./NIMIN)
206    CONTINUE
ENDIF

```

THIS ELSE AND ENDIF ARE FOR THE TEST AGAINST MAXRUN

```

ELSE
    WRITE(1,(' '),'PROGRAM EXCEEDED THE MAX NO. OF RUNS',
+' ALLOWED OF:',I6) TOTREP
    GOTO 9999

    END IF
    GOTO 10
    END IF

```

```

C    WRITE(2,(''UNSORTED RSHAT 1 IS:',/10(F8.5))')
C    +(RSHAT(1,LOOP), LOOP=1, MAXREP)
C    WRITE(2,(''UNSORTED RSHAT 2 IS:',/10(F8.5))')
C    +(RSHAT(2,LOOP), LOOP=1, MAXREP)
C    IF(FLAG.EQ.1) THEN
C        WRITE(2,(''UNSORTED RHTSTR 1 IS:',/10(F8.5))')
C        +(RHTSTR(1,LOOP), LOOP=1, MAXREP)
C        WRITE(2,(''UNSORTED RHTSTR 2 IS:',/10(F8.5))')
C        +(RHTSTR(2,LOOP), LOOP=1, MAXREP)
C    ELSE
C        END IF
C    IF(K.EQ.5) THEN
C        WRITE(2,(''UNSORTED RSHTBR 1 IS:',/10(F8.5))')
C        +(RSHTBR(1,LOOP), LOOP=1, MAXREP)

```

```

C      WRITE(2,('UNSORTED RSHTBR 2 IS: ',/10(F8.5)))
C      +(RSHTBR(2,LOOP), LOOP=1, MAXREP)
C      ELSE
C      END IF
C      WRITE (2,('ZERO AND ONE FAILURE REPS: ',/10(A8)))
C      + (LOOPS0(Loop),Loop=1,MAXREP)

```

SORT THE ARRAYS OF SYSTEM UNRELIABILITIES(1 FOR EACH CONF. LEVEL)

```

      DO 700 ALF=1, MAXALF
        DO 800 REPS=1, MAXREP
          TRANSQ(REPS) = QHTUPR(ALF,REPS)
          TRANSR(REPS) = RSHAT(ALF,REPS)
          TRNSTR(REPS) = RHTSTR(ALF,REPS)
          TRANBR(REPS) = RSHTBR(ALF,REPS)
800      CONTINUE
          CALL SHSORT(TRANSQ,KEY1,MAXREP)
          CALL SHSORT(TRANSR,KEY2,MAXREP)
          CALL SHSORT(TRNSTR,KEY3,MAXREP)
          CALL SHSORT(TRANBR,KEY4,MAXREP)
          DO 900 REPS=1, MAXREP
            QHTUPR(ALF,REPS) = TRANSQ(REPS)
            RSHAT(ALF,REPS) = TRANSR(REPS)
            RHTSTR(ALF,REPS) = TRNSTR(REPS)
            RSHTBR(ALF,REPS) = TRANBR(REPS)
900      CONTINUE
700      CONTINUE

```

PRINT OUTPUT REPORT HEADINGS

```

      WRITE(1,6666)
      WRITE(1,6667) MAXREP
      WRITE(1,6668) K
      WRITE(1,6669)
      IF(K.EQ.5) THEN
        WRITE(1,6699)
      ELSE
      END IF
      WRITE(1,6670) MSTRQ
      WRITE(1,6671)
      WRITE(1,C1C15)
      WRITE(1,3334) AI
      WRITE(1,0007)
      WRITE(1,C1C15)
      WRITE(1,3334) QI
      WRITE(1,0005)
      WRITE(1,C1C15)
      WRITE(1,3335) N
      WRITE(1,6674)

```

COMPUTE THE VALUE RS OF THE TRUE SYSTEM REL. FNCTN. (SERIES SYSTEM)
 *** AND FOR THE 5-COMPONENT BRIDGE STRUCTURE***

```

      CALL RSR(S(QI,K,RS)

```

```

WRITE(1,(' ' ' ',////'THE TRUE SERIES SYSTEM ' ',
+'RELIABILITY VALUE IS: ',T51,F8.5)') RS
CALL RBRIDG(QI,K,RSBRDG)
IF(K.EQ.5) THEN
WRITE(1,(' ' ' ',////'THE TRUE BRIDGE STRUCTURE ' ',
+'RELIABILITY VALUE IS: ',T51,F8.5)') RSBRDG
ELSE
END IF
WRITE(1,6675)

```

COMPUTE THE DIFFERENCE 'DELTA' BTWN. RS AND RSHAT OF THE THEO
 RETICAL QUANTILE GIVEN BY ALFA(MUST USE SORTED RSHAT ARRAY)

```

IF(FLAG.EQ.1) THEN
WRITE(1,5755)
ELSE
END IF
DO 450 ALF=1, MAXALF
  QUANTL(ALF) = MAXREP * (1 - ALFA(ALF))
  DELTAR(ALF) = RS - RSHAT(ALF,QUANTL(ALF))
  DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
  IF(FLAG.EQ.1) THEN
    DELSTR(ALF) = RS - RHTSTR(ALF,QUANTL(ALF))
    WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
    WRITE(1,5656) RHTSTR(ALF,QUANTL(ALF))
    WRITE(1,5657) DELSTR(ALF)
  ELSE
  END IF
  IF(K.EQ.5) THEN
    DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
    WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
    WRITE(1,5666) RSHTBR(ALF,QUANTL(ALF))
    WRITE(1,5667) DELBRG(ALF)
  ELSE
  END IF
  WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
  WRITE(1,5556) RSHAT(ALF,QUANTL(ALF))
  WRITE(1,5557) DELTAR(ALF)
450 CONTINUE
PRINT *, 'QUANTL(1) IS: ', QUANTL(1)
PRINT *, 'QUANTL(2) IS: ', QUANTL(2)

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RSHAT ***** ***

```

WRITE(1,6676)
DO 400 ALF=1,MAXALF
  TRUQNT(ALF) = 0
  DO 500 REPS=1, MAXREP
    DIFF(REPS) = RS - RSHAT(ALF,REPS)
500 CONTINUE
  DO 600 REPS=1, MAXREP
    IF(ABS(DIFF(REPS)).LE.EPS) THEN
      TRUQNT(ALF) = REPS
      WRITE(1,(' ' ' ',/'TRUE CONFIDENCE LIMIT IS: ',
+          F8.4)')

```

```

+          (TRUQNT(ALF) / REAL(MAXREP)) * 100.
          GO TO 620
        ELSEIF(DIFF(REPS).LT.0.) THEN
          TRUQNT(ALF) = REPS
          GO TO 610
        ELSE
          END IF
600      CONTINUE
610      IF(TRUQNT(ALF).EQ.0.) THEN
          WRITE(1,4443) ALFA(ALF)
          WRITE(1,(' ',' ',/'THE SMALLEST',
+          ' ' DIFFERENCE BETWEEN RS AND RSHAT IS: ',F10.5)') DIFF(
+          MAXREP)
        ELSEIF(TRUQNT(ALF).EQ.1.) THEN
          WRITE(1,4442) ALFA(ALF)
          WRITE(1,(' ',' ',/'ALL RSHAT',
+          ' ' ARE GREATER THAN RS''')')
        ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+          THEN
          WRITE(1,4444) ALFA(ALF),
          (TRUQNT(ALF) / REAL(MAXREP)) * 100.
          WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF))
          WRITE(1,4446)
        ELSE
+          WRITE(1,4444) ALFA(ALF),
          ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
          WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF)-1)
          WRITE(1,4447)
620      END IF
400      CONTINUE

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RSHTBR (BRIDGE) ***** ***

```

IF(K.EQ.5) THEN
DO 401 ALF=1,MAXALF
TRUQNT(ALF) = 0
DO 501 REPS=1, MAXREP
DIFF(REPS) = RSBRDG - RSHTBR(ALF,REPS)
501  CONTINUE
DO 601 REPS=1, MAXREP
IF(ABS(DIFF(REPS)).LE.EPS) THEN
TRUQNT(ALF) = REPS
WRITE(1,(' ',' ',/'TRUE CONFIDENCE LIMIT IS: ',
+          F8.4)')
+          (TRUQNT(ALF) / REAL(MAXREP)) * 100.
          GO TO 621
        ELSEIF(DIFF(REPS).LT.0.) THEN
          TRUQNT(ALF) = REPS
          GO TO 611
        ELSE
          END IF
601  CONTINUE
611  IF(TRUQNT(ALF).EQ.0.) THEN
          WRITE(1,4443) ALFA(ALF)

```

```

        WRITE(1,(' ' ' ',/'THE SMALLEST',
+         ' ' DIFFERENCE BETWEEN RSBRDG AND RSHTBR IS: ',
+         F10.5)') DIFF(MAXREP)
        ELSEIF(TRUQNT(ALF).EQ.1.) THEN
            WRITE(1,4442) ALFA(ALF)
            WRITE(1,(' ' ' ',/'ALL RSHTBR',
+             ' ' ARE GREATER THAN RSBRDG'))')
        ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+         THEN
            WRITE(1,4444) ALFA(ALF),
+             (TRUQNT(ALF) / REAL(MAXREP)) * 100.
            WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF))
            WRITE(1,4446)
        ELSE
+         WRITE(1,4444) ALFA(ALF),
            ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
            WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF)-1)
            WRITE(1,4447)
621     END IF
401     CONTINUE
        ELSE
            END IF

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RHTSTR ***** ***

```

IF(FLAG.EQ.1) THEN
DO 4400 ALF=1,MAXALF
TRUQNT(ALF) = 0
    DO 5500 REPS=1, MAXREP
        DIFF(REPS) = RS - RHTSTR(ALF,REPS)
5500    CONTINUE
        DO 6600 REPS=1, MAXREP
            IF(ABS(DIFF(REPS)).LE.EPS) THEN
                TRUQNT(ALF) = REPS
                WRITE(1,(' ' ' ',/'TRUE CONFIDENCE LIMIT IS: ',
+                 F8.4)')
                (TRUQNT(ALF) / REAL(MAXREP)) * 100.
                GO TO 6620
            ELSEIF(DIFF(REPS).LT.0.) THEN
                TRUQNT(ALF) = REPS
                GO TO 6610
            ELSE
                END IF
6600    CONTINUE
6610    IF(TRUQNT(ALF).EQ.0.) THEN
            WRITE(1,4443) ALFA(ALF)
            WRITE(1,(' ' ' ',/'THE SMALLEST',
+             ' ' DIFFERENCE BETWEEN RS AND RHTSTR IS: ',
+             F9.5)') DIFF(MAXREP)
            ELSEIF(TRUQNT(ALF).EQ.1.) THEN
                WRITE(1,4442) ALFA(ALF)
                WRITE(1,(' ' ' ',/'ALL RHTSTR',
+                 ' ' ARE GREATER THAN RS'))')
            ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))

```



```

+     THEN
+       WRITE(1,4444) ALFA(ALF),
+       (TRUQNT(ALF) / REAL(MAXREP)) * 100.
+       WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF))
+       WRITE(1,4446)
+     ELSE
+       WRITE(1,4444) ALFA(ALF),
+       ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
+       WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF)-1)
+       WRITE(1,4447)
6620     END IF
4400 CONTINUE
      ELSE
      END IF

```

PRINT THE ARRAYS PERTINENT TO THE OUPUT OF EACH REPLICATION

```

      IF(PRNT.EQ.1) THEN
      I = 1
185  WRITE(1,REPSHD) ALFA(SELCTA), ALFA(SELCTA),
+ALFA(SELCTB),ALFA(SELCTB),ALFA(SELCTA),ALFA(SELCTA),ALFA(SELCTB),
+ALFA(SELCTB)
175  IF(I.GE.(MAXREP + 1)) THEN
      GOTO 180
      ELSE
+     IF( (I.EQ.71).OR.(I.EQ.211).OR.(I.EQ.351).OR.(I.EQ.491).OR.
+     (I.EQ.631).OR.(I.EQ.771).OR.(I.EQ.911).OR.(I.EQ.1051) ) THEN
+       I = I + 70
+       WRITE(1,(''+'))
+       GOTO 185
      ELSE
+     WRITE(1,3336) I, INT(DEGFR(I)), CHISQ(1,I), QHTUPR(1,I),
+     CHISQ(2,I), QHTUPR(2,I)
+     END IF
+     IF(I + 70.LE.MAXREP) THEN
+       WRITE(1,3337) I+70,INT(DEGFR(I+70)),CHISQ(1,I+70),
+       QHTUPR(1,I+70),CHISQ(2,I+70),QHTUPR(2,I+70)
+     ELSE
+     END IF
+     I = I + 1
+     GOTO 175
180  END IF
      ELSE
      ENDIF
9999 WRITE(1,(''THE TOTAL NO OF REPS WAS:'' ,I8)') TOTREP
      WRITE(1,(''THE TOTAL NO OF EFFECTIVE REPS WAS:'' ,I8)') LOOP
      WRITE(1,(''THE TOTAL NO OF NO FAILURE RUNS WAS:'' ,I8)') ZFAILS
      WRITE(1,(''AVERAGE NO. OF COMPONENTS PER REPLICATION WITH '' ,
+ ''NO FAILURES:'' ,F5.2)') ZFPREP / MAXREP
      WRITE(1,(''THE TOTAL NO OF RUNS WITH FAILURES WAS:'' ,I8)') FAILS
0008 FORMAT (/ 3X,'C 1',5X,'C 2',
+5X,'C 3',5X,'C 4',5X,'C 5',5X,'C 6',5X,'C 7',5X,
+ 'C 8',5X,'C 9',5X,'C 10',4X,'C 11',4X,
+ 'C 12',4X,'C 13',4X,'C 14',4X,'C 15')
0009 FORMAT(/1X,'REP NO',2X,'DF',1X,'CHISQR(',F4.3,')',1X,

```

```

+ 'QHTUPR(' ,F4.3,')' ,1X, 'CHISQR(' ,F4.3,')' ,1X, 'QHTUPR(' ,F4.3,')' ,
+2X, 'REP NO' ,2X, 'DF' ,1X, 'CHISQR(' ,F4.3,')' ,1X,
+ 'QHTUPR(' ,F4.3,')' ,1X, 'CHISQR(' ,F4.3,')' ,1X, 'QHTUPR(' ,F4.3,')' //)
0001 FORMAT (///'UNIFORM RANDOM DEVIATES ARE: ')
0002 FORMAT (///'BERNOULLI TRIALS ARE: ')
0003 FORMAT (///'TOTAL NO. OF FAILURES FOR EACH COMPONENT: ')
0004 FORMAT (///'ESTIMATED UNRELIABILITY FOR EACH COMPONENT: ')
0005 FORMAT (///'TOTAL NUMBER OF MISSION TESTS: ')
0006 FORMAT (///'ESTIMATED WEIGHTS FOR EACH COMPONENT: ')
0007 FORMAT (///'Q I FOR EACH COMPONENT: ')
1111 FORMAT (15F8.5)

2222 FORMAT (/1X,15(I4,4X))
3333 FORMAT (/1X,15(I4,4X))
3334 FORMAT (/15F8.5)
3335 FORMAT (/1X,15(I4,4X))
3336 FORMAT (T3,I4,T9,I3,T13,F11.5,T27,F8.5,T39,F11.5,T53,F8.5)
3337 FORMAT ('+',T67,I4,T73,I3,T77,F11.5,T91,F8.5,T103,F11.5,T117,F8.5)
4442 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+'LIMIT IS: ',T50,' 00.000 ')
4443 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+'LIMIT IS: ',T50,' 100.0000 ')
4444 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+'LIMIT IS: ',T50,F8.4)
4445 FORMAT (' ',/'THE RSHAT VALUE CLOSEST TO RS IS: ',T51,F8.5)
4446 FORMAT (' ',/'(FIRST NEGATIVE DIFFERENCE)')
4447 FORMAT (' ',/'(ELEMENT PRECEEDING FIRST NEGATIVE DIFFERENCE)')
4448 FORMAT (' ',/'THE RHTSTR VALUE CLOSEST TO RS IS: ',T51,F8.5)
4449 FORMAT (' ',/'THE RSHTBR VALUE CLOSEST TO RSBRDG IS: ',T51,F8.5)
5555 FORMAT (' ',///'THE ',I4,'(1-',F4.3,') QUANTILE IS: ',T49,F8.3)
5556 FORMAT (' ',/'THE VALUE OF RSHAT FOR THAT QUANTILE IS: ',T51,F8.5)
5557 FORMAT (' ',/'THE DIFFERENCE(RS - RSHAT) IS: ',T51,F8.5)
5656 FORMAT (' ',/'THE VALUE OF RHTSTR FOR THAT QUANTILE IS: ',T51,F8.5)
5666 FORMAT (' ',/'THE VALUE OF RSHTBR FOR THAT QUANTILE IS: ',T51,F8.5)
5657 FORMAT (' ',/'THE DIFFERENCE(RS - RHTSTR) IS: ',T51,F8.5)
5667 FORMAT (' ',/'THE DIFFERENCE(RS - RSHTBR) IS: ',T51,F8.5)
5755 FORMAT (' ',///'SINCE THE NO. OF MISSION TESTS IS THE SAME FOR',
+' ALL COMPONENTS THE CLOSED FORM SERIES SYSTEM RELIABILITY ',
+' 'RHTSTR' IS COMPUTED')
6666 FORMAT ('+', '*****',
+'***** RUN INPUT SETTINGS *****',
+'*****')
6667 FORMAT (' ',/'NUMBER OF REPLICATIONS: ',T50,I4)
6668 FORMAT (' ',/'NUMBER OF COMPONENTS: ',T50,I4)
6669 FORMAT (' ',/'SYSTEM RELIABILITY FUNCTION: ',T50,'SERIES')
6699 FORMAT (' ',/'SYSTEM RELIABILITY FUNCTION: ',T50,'BRIDGE')
6670 FORMAT (' ',/'MASTER UNRELIABILITY USED: ',T50,F8.5)
6671 FORMAT (' ',/'INPUT WEIGHTS(A SUB I'S): ')
6674 FORMAT ('+',///'*****',
+'**R U N R E S U L T S*****',
+'*****')
6675 FORMAT ('+',///'*****',
+' ESTIMATE ERRORS *****',
+'*****')
6676 FORMAT ('+',///'*****',
+' TRUE CONFIDENCE LIMITS *****',
+'*****')

```

+ '*****')
-END

**APPENDIX E. FORTRAN CODE FOR THE PREFERRED LOWER
CONFIDENCE LIMIT FOR SYSTEM RELIABILITY (SERIES-PARALLEL
SYSTEM WITH A 2/3 COMPONENT**

PROGRAM ZFYSCN

```

*****
*
* TITLE: BINOMIAL INTERVAL ESTIMATION PROCEDURE
*         ZERO FAILURES ALLOWED; NO SCALING
* AUTHOR: E. F. BELLINI, LT, USN
* MODIFIED BY: LT VALERIE A. COVINGTON, USN (MAR 90)
*         DATE: NOV 89
*
* THIS PROGRAM COMPUTES THE TRUE CONFIDENCE LEVEL FOR THE ESTIMATE
* RELIABILITY OF A SERIES AND BRIDGE SYSTEM GIVEN THE RELIABILITY
* OF THEIR COMPONENTS
*
* IN ITS PRESENT CONFIGURATION THIS PROGRAM IS SET UP TO RUN 12
* TIMES EACH TIME PRODUCING 1000 REPLICATIONS USING A DIFFERENT
* SET OF INPUT DATA. RUN THE PROGRAM FROM CMS BY TYPING 'B1 EXEC'.
* THE REXX EXEC PROGRAM
* 'B1' CALLS THE INPUT FILES TO BE READ AND NAMES THE 12 OUTPUT
* FILES RESULTING FROM THE 12 CONSECUTIVE RUNS. BY EDITING THE
* INDEX COUNTERS I, J, K OF THE 'B1' EXEC ONE CAN RUN ANY USER-
* SPECIFIC RUN FROM JUST ONE RUN TO ALL 12.
*
* VARIABLES USED
*
* AHATI : WEIGHT ESTIMATES FOR EACH COMPONENT
* AI : INPUT WEIGHTS FOR EACH COMPONENT
* ALFA : LEVELS OF SIGNIFICANCE
* BIGF : TOTAL NO. OF FAILURES FOR EACH REPLICATION
* CHISQ : CHI-SQUARE RANDOM VARIABLE VALUE
* C1C15 : FORMAT LABEL
* DEGFR : DEGREES OF FREEDOM
* DELBRG : DIFFERENCE FOR BRIDGE SYSTEM
* DELSTR : DIFFERENCE FOR SERIES SYSTEM- CLOSED FORM
* DELTAR : DIFFERENCE FOR SERIES SYSTEM
* DIFF : DIFFERENCE (TRUE REL. - ESTIMATED REL.)
* EPS : SMALL QUANTITY(CONSTANT)
* ERROR : PARAMETER FOR IMSL ROUTINE
* FAILS : COUNTS NO. OF REPLICATIONS WITH AT LST. 1 FAILURE
* FI : NO. OF FAILURES FOR EACH COMPONENT(ALL MISSION TST)
* FLAG : 1 IF ALL COMP. HAVE SAME NO. OF MISSION TESTS
* INC : INCREMENT STEP SIZE FOR ROUTINE USMNMX
* KEY1 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY2 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY3 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY4 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KK : ARRAY SIZING PARAMETER FOR THE MAX NO OF COMPONENTS
* LOOP : COUNTS NO. OF REPLICATION PERFORMED
* MAXALF : MAX NO. OF SIGNIFICANT LEVELS DESIRED(ARRAY SIZING)

```

```

*      MAXREP : MAX NO. OF REPLICATIONS                      *
*      MAXRUN : MAX NO. OF PROGRAM ITERATIONS ALLOWED        *
*      MSTRQ  : MASTER UNRELIABILITY(USED WITH AI'S TO CALC. QI'S) *
*      MULT   : MULTIPLIER FOR RANDOM NO. GENERATOR SRND     *
*      N      : NO. OF MISSION TEST FOR EACH COMPONENT        *
*      NIMAX  : MAX NO. OF MISSION TESTS                      *
*      NIMIN  : MIN NO. OF MISSION TESTS                     *
*      NINDX  : INDEX NO. OF MAX NO. OF MISSION TESTS       *
*      NIREAL : NO. OF MISSION TESTS TRANSFORMED TO REAL     *
*      NMAX   : MAX NO. OF MISSION TESTS FOR OUTPUT CONTROL  *
*      NPRNT  : FLAG FOR DETAILED REPORT OUTPUT              *
*      PRNT   : SAME AS ABOVE(PARAMETER)                     *
*      QHATI  : UNRELIABILITY ESTIMATES FOR EACH COMPONENT   *
*      QHTMAX : LARGEST QHATI                                 *
*      QHTUPR : UPPER LIMIT ON SYSTEM UNRELIABILITY         *
*      QI     : INPUT UNRELIABILITY FOR EACH COMPONENT       *
*      QINDX  : INDEX                                         *
*      QUANTL : QUANTILE                                       *
*      REPSHD : REPLICATIONS HEADING FORMAT NUMBER          *
*      RHTSTR : SERIES SYSTEM RELIABILITY ESTIMATE(CLOSED FORM) *
*      RS     : TRUE SERIES SYSTEM RELIABILITY               *
*      RSBRDG : TRUE BRIDGE SYSTEM RELIABILITY              *
*      RSHAT  : SERIES SYSTEM RELIABILITY ESTIMATE           *
*      RSHTBR : BRIDGE SYSTEM RELIABILITY ESTIMATE           *
*      SEED   : PARAMETER                                     *
*      SELCTA : SIGNIFICANCE LEVEL SELECTION                 *
*      SELCTB : SIGNIFICANCE LEVEL SELECTION                 *
*      SORT   : PARAMETER FOR ROUTINE SRND                   *
*      SUMNAI : SUM OF THE PRODUCT OF NI'S AND AI'S          *
*      TEMP   : TEMPORARY ARRAY                              *
*      TOTREP : TOTAL NUMBER OF PROGRAM ITERATIONS           *
*      TRANBR : TEMPORARY ARRAY                              *
*      TRANSQ : TEMPORARY ARRAY                              *
*      TRANSR : TEMPORARY ARRAY                              *
*      TRIALS : BERNOULLI TRIALS ARRAY (2-DIM)               *
*      TRNSTR : TEMPORARY ARRAY                              *
*      TRUQNT : TRUE QUANTILE                                 *
*      UNIRV  : UNIFORM RANDOM DEVIATES (2-DIM)             *
*      ZFAILS : TOTAL NUMBER OF REPLICATIONS WITH ZERO FAILURES *
*      ZFPREP : NO. OF COMPNTS. WITH ZERO FAILURES PER REPLICATION *
*

```

```

PARAMETER (KK=10,MAXALF=2,NPRNT=0)
PARAMETER (MAXREP=1000, MAXRUN=2000, EPS=.000001)
REAL*4 UNIRV(15,1000),TEMP(1000),QI(KK),AI(KK),QHATI(KK)
REAL*4 QHATI(KK), NMAX, NMAX, QHTMAX, CHISQR(5,5), ALFA(MAXALF)
REAL*4 DF(5),AALFA(5),SUMNAI,RSHAT(MAXALF,MAXREP),RS
REAL*4 KEY1(MAXREP),KEY2(MAXREP),KEY3(MAXREP),TRNSTR(MAXREP)
REAL*4 DEGFR(MAXREP), QHTUPR(MAXALF,MAXREP),CHISQ(MAXALF,MAXREP)
REAL*4 QUPA1(MAXREP), QUPA2(MAXREP),RHTSTR(MAXALF,MAXREP)
REAL*4 DELTAR(MAXALF), TRANSQ(MAXREP),TRANSR(MAXREP),DIFF(MAXREP)
REAL*4 DELSTR(MAXALF),NIMIN,NIMAX,NIREAL(KK)
REAL*4 RSHTBR(MAXALF,MAXREP),DELBRG(MAXALF),KEY4(MAXREP)
REAL*4 TRANBR(MAXREP),RSBRDG,MSTRQ
REAL*4 ZFPREP

```

```

REAL*4 AVGN,SUC,STUD,PTEMP(10),FDEG1,FDEG2,P3,S3,S4

INTEGER SEED, MULT, SORT, TRIALS(15,1000), BIGF, FI(KK), N(KK)
INTEGER NINDX, QINDX, ERROR, REPS, SELCTA, SELCTK, TOTREP
INTEGER C1C15, REPSHD, SELCTB, ALF, FLAG, LOOP, PRNT
INTEGER QUANTL(MAXALF), TRUQNT(MAXALF), ZFAILS, FAILS, INC
INTEGER NTEST, FCT, HFI

CHARACTER*8 LOOPSO(MAXREP)

DATA SEED/123457/, MULT/1/, INC/1/
DATA AALFA/.01,.05,.9,.99/, DF/1,5,10,30,40/
DATA ALFA/.20,.050/
DATA SORT/0/

ASSIGN 8 TO C1C15
ASSIGN 9 TO REPSHD

* CALL COMPRS
  PRNT = NPRNT

  DO 12 I=1, KK
    AI(I) = 9999.
    N(I) = 99999999
  12 CONTINUE

  READ(03,*) K, MSTRQ

  DO 11 I=1, K
    READ(03,*) AI(I), N(I)
  11 CONTINUE

  IF(K.NE.5) THEN
    WRITE(1,('WARNING: BRIDGE STRUCTURE ',
+ 'ONLY USES THE FIRST 5 COMPONENTS'))
  ELSE
    END IF

***INITIALIZE THE QHTUPR ARRAY OF UNRELIABILITY REPLICATIONS, ***
* RSHAT ARRAY OF ESTIMATE SYSTEM RELIABILITY REPLICATIONS *
* AND RHTSTR ARRAY OF EST. SYST. REL. FOR A SERIES SYST WHEN *
***ALL THE COMPONENT MISSION TESTS ARE EQUAL IN NUMBER ***

  DO 172 ALF=1, MAXALF
    DO 173 REPS=1, MAXREP
      QHTUPR(ALF, REPS) = 0.
      RSHAT(ALF, REPS) = 0.
      RHTSTR(ALF, REPS) = 0.
      RSHTBR(ALF, REPS) = 0.
      LOOPSO(REPS) = ' ****'
    173 CONTINUE
  172 CONTINUE

***SET FLAG TO 1 IF ALL COMPONENTS HAVE SAME NO. OF MISSION TESTS****

```

```

FLAG=1
DO 50 I=1,K -1
    IF((N(I) - N(I+1)).NE.0) THEN
        FLAG=0
    ELSE
        END IF
50 CONTINUE
PRINT *, 'FLAG IS:', FLAG

***MAIN PROGRAM OUTER LOOP START(EVERY LOOP IS ONE REPLICATION)***

ZFPREP = 0.
ZFAILS = 0
FAILS = 0
TOTREP = 0
LOOP = 0
10 IF(LOOP.LT.MAXREP) THEN
    LOOP = LOOP + 1
    IF(TOTREP.LT.MAXRUN) THEN
        TOTREP = TOTREP + 1
    SELCTA = 1
    SELCTB = 2

***FILL ARRAY KEY(REPS) WITH INTEGERS 1 TO K TO BE USED AS OUTPUT***
***OF THE SUBROUTINE SHSORT***

DO 95 REPS=1, MAXREP
    KEY1(REPS) = REPS
    KEY2(REPS) = REPS
    KEY3(REPS) = REPS
    KEY4(REPS) = REPS
95 CONTINUE

***CALCULATE NMAX NOT TO PRINT LONGER THAN THE MAX SAMPLE SIZE***

***CALCULATE THE MAXIMUM NO. OF TRIALS AND ITS INDEX NO.***

CALL IMAX(N,K,NMAX,NINDX)

***CALCULATE THE QI'S FROM THE GIVEN MASTER Q AND THE AI'S***

DO 115 I=1, K
    QI(I) = MSTRQ * AI(I)
115 CONTINUE
    S3=QI(3)
    S4=QI(4)
DO 120 I=1,15
    DO 125 J=1,500
        UNIRV(I,J) = 999.
        TRIALS(I,J) = 99999
125 CONTINUE
120 CONTINUE

*** DRAW UNIFORM (0,1) RV'S AND CONVERT TO BERNOULLI TRIALS***
DO 130 I=1, K

```

```

        CALL SRND(SEED, TEMP, N(I), MULT, SORT)
        DO 135 J=1, N(I)
            UNIRV(I,J) = TEMP(J)
            IF (UNIRV(I,J).LE. 1 - QI(I)) THEN
                TRIALS(I,J) = 0
            ELSE
                TRIALS(I,J) = 1
            END IF
135    CONTINUE
130    CONTINUE

```

CALCULATE THE NO. OF FAILURES FOR EACH COMPONENT

```

        DO 150 I=1, K
            FI(I) = 0
150    CONTINUE
        IONECT = 0

```

CALCULATE THE F SUB I'S AND THE GRAND TOTAL NO. OF FAILURES

```

        BIGF = 0
        DO 155 I=1, K
            DO 160 J=1, N(I)
                FI(I) = FI(I) + TRIALS(I,J)
160        CONTINUE
            IF (FI(I) .EQ. 0) THEN
                ZFPREP=ZFPREP+1
            ELSE
                ENDIF
            BIGF = BIGF + FI(I)

            QHATI(I) = REAL(FI(I)) / N(I)
155    CONTINUE

```

Change for Series-Parallel System with 2 out of 3 component

```

        IF (FI(4) .EQ. 0) THEN
            ZFPREP=ZFPREP - 1
        ENDIF
        BIGF = BIGF - FI(4)
        FI(4) = 0
        DO 161 I=1,N(4)
            HFI=0
            DO 162 J=1,2
                CALL SRND(SEED, PTEMP(J), N(4), MULT, SORT)
                IF (PTEMP(J) .GT. 1-S4) THEN
                    HFI = HFI + 1
                ENDIF
162        CONTINUE
            IF (HFI .EQ. 2) THEN
                FI(4) = FI(4) + 1
            ENDIF
161    CONTINUE
        BIGF = BIGF + FI(4)
        IF (FI(4) .EQ. 0) THEN

```



```

        ZFPREP = ZFPREP + 1
    ELSE
    ENDIF
    IF (FI(3) .EQ. 0) THEN
        ZFPREP=ZFPREP - 1
    ENDIF
    BIGF = BIGF - FI(3)
    FI(3) = 0
    DO 163 I=1,N(3)
        HFI=0
        DO 164 J=1,3
            CALL SRND(SEED, PTEMP(J), N(3), MULT, SORT)
            IF (PTEMP(J) .GT. 1-S3) THEN
                HFI = HFI + 1
            ENDIF
164    CONTINUE
            IF (HFI .GE. 2) THEN
                FI(3) = FI(3) + 1
            ENDIF
163    CONTINUE
        BIGF = BIGF + FI(3)
        IF (FI(3) .EQ. 0) THEN
            ZFPREP = ZFPREP + 1
        ELSE
        ENDIF
        DO 19 I=1,K
            QI(I) = MSTRQ * AI(I)
            QHATI(I) = REAL(FI(I)) / N(I)
19    CONTINUE
            QI(4) = QI(4)**2
            P3 = 1. -QI(3)
            QI(3) = 1. -((3.*P3**2*QI(3))+(P3**3))
            ***COUNTS NUMBER OF COMPONENTS THAT HAVE FAILED***
            DO 156 I=1,K
                IF (FI(I) .NE. 0) IONECT=IONECT+1
156    CONTINUE
            ***CASE WHERE NO COMPONENTS HAVE ANY FAILURES***
            IF(BIGF.EQ.0) THEN
                LOOPSO(LOOP)=' *ZERO* '
                ZFAILS = ZFAILS + 1
                AVGN=0.0
                DO 200 I=1,K
                    AVGN=AVGN+REAL(N(I))
200    CONTINUE
                AVGN=AVGN/REAL(K)
                DO 205 ALF=1, MAXALF
                    RSHAT(ALF,LOOP)= ALFA(LOOP)**(1./AVGN)
                    IF(FLAG.EQ.1) THEN
                        RHTSTR(ALF,LOOP)=ALFA(LOOP)**(1./N(1))
                    ELSE
                    END IF
                    PRINT *,LOOP', LOOP, 'RSHAT', RSHAT(ALF,LOOP)

```

```

205     CONTINUE
           DEGFR(LOOP) = 2.
           GO TO 10
ELSE
           FAILS = FAILS + 1
END IF

***COUNTS NUMBER OF COMPONENTS THAT FAIL RECORDS NO. COMPT TESTS***
FCT=0
DO 202 I=1,K
           IF (FI(I) .NE. 0) THEN
               FCT=FCT+1
               NTEST=N(I)
           ENDIF
202     CONTINUE

***FIND THE MAX OF THE INDIVIDUAL COMPONENT UNRELIABILITIES***
           CALL RMAX(QHATI, K, QHTMAX, QINDX)

***CALCULATE THE AHAT SUB I'S (WEIGHT ESTIMATES)***
***IF COMPONENT HAS NO FAILURES AHAT SUB I IS ZERO***

SUMNAI = 0.
DO 165 I=1, K
           AHATI(I) = QHATI(I) / QHTMAX
           SUMNAI = SUMNAI + N(I) * AHATI(I)
165     CONTINUE

***1 COMPONENT FAILURE SERIES SYSTEM***

IF (FCT .EQ. 1) THEN
           LOOPS0(LOOP)=' *ONECF* '
           DO 305 ALF=1,MAXALF
               SUC=REAL(NTEST-BIGF)
               FDEG1=2.*(REAL(BIGF)+1.)
               FDEG2=2.*SUC
               STUD=FIN(1. -ALFA(ALF),FDEG1,FDEG2)
               RSHAT(ALF,LOOP)=SUC/(SUC+(REAL(BIGF)+1.)*STUD)
*           PRINT *, 'SUC=', SUC, 'FAIL=', BIGF, 'NTEST=', NTEST
*           PRINT *, 'FIN=', STUD
*           PRINT *, 'ALFA=', ALFA(ALF), 'RSHAT=', RSHAT(ALF,LOOP)
           IF (FLAG .EQ. 1) THEN
               RHTSTR(ALF,LOOP)=RSHAT(ALF,LOOP)
           ELSE
               ENDIF
305     CONTINUE
           ENDIF

***CALCULATE 1 REPLICATION OF UPPR ALFA C.L. ON SYSTEM RELIABILITY***

DEGFR(LOOP) = 2 * (1 + BIGF)

DO 170 ALF=1, MAXALF
           CALL MDCHI(1 - ALFA(ALF),DEGFR(LOOP),CHISQ(ALF,LOOP), ERROR)
           QHTUPR(ALF,LOOP) = CHISQ(ALF,LOOP) / (2 * SUMNAI)

```

```

        IF(FLAG.EQ.1) THEN
            RHTSTR(ALF,LOOP) = 1 -(CHISQ(ALF,LOOP) / REAL(2*N(1)))
        ELSE
            END IF

***CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR COMPNTS. IN SERIES***

        IF (FCT .NE. 1) THEN
            CALL RHTSR(S(QHTUPR(ALF,LOOP), AHATI,K, RSHAT(ALF,LOOP))
            ENDIF

***CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR BRIDGE STRUCTURE***

        IF (IONECT .NE. 1) THEN
            CALL RHTBRG(QHTUPR(ALF,LOOP),AHATI,K,RSHTBR(ALF,LOOP))
        ENDIF
170 CONTINUE

***EXACTLY 1 COMPONENT FAILS AND REDUNDANT COMPONENT***

        IF ((IONECT .EQ. 1) .AND. (K .EQ. 5)) THEN
            DO 207 I=1, K
                NIREAL(I) = REAL(N(I))
207 CONTINUE
            CALL USMMX(NIREAL,K,INC,NIMIN,NIMAX)
            DO 206 ALF=1,MAXALF
                RSHTBR(ALF,LOOP)=ALFA(ALF)**(1./NIMIN)
206 CONTINUE
            ENDIF

***THIS ELSE AND ENDIF ARE FOR THE TEST AGAINST MAXRUN***

        ELSE
            WRITE(1,(' '),'PROGRAM EXCEEDED THE MAX NO. OF RUNS',
+' ' ALLOWED OF: ',I6) TOTREP
            GOTO 9999

            END IF
            GOTO 10
            END IF

C WRITE(2,(' 'UNSORTED RSHAT 1 IS: ',/10(F8.5))')
C +(RSHAT(1,LOOP), LOOP=1, MAXREP)
C WRITE(2,(' 'UNSORTED RSHAT 2 IS: ',/10(F8.5))')
C +(RSHAT(2,LOOP), LOOP=1, MAXREP)
C IF(FLAG.EQ.1) THEN
C WRITE(2,(' 'UNSORTED RHTSTR 1 IS: ',/10(F8.5))')
C +(RHTSTR(1,LOOP), LOOP=1, MAXREP)
C WRITE(2,(' 'UNSORTED RHTSTR 2 IS: ',/10(F8.5))')
C +(RHTSTR(2,LOOP), LOOP=1, MAXREP)
C ELSE
C END IF
C IF(K.EQ.5) THEN
C WRITE(2,(' 'UNSORTED RSHTBR 1 IS: ',/10(F8.5))')
C +(RSHTBR(1,LOOP), LOOP=1, MAXREP)
C WRITE(2,(' 'UNSORTED RSHTBR 2 IS: ',/10(F8.5))')

```

```

C   +(RSHTBR(2,LOOP), LOOP=1, MAXREP)
C   ELSE
C   END IF
C   WRITE (2,('ZERO AND ONE FAILURE REPS:',/10(A8)))
C   + (LOOPS0(LOOP),LOOP=1,MAXREP)

```

SORT THE ARRAYS OF SYSTEM UNRELIABILITIES(1 FOR EACH CONF. LEVEL)

```

      DO 700 ALF=1, MAXALF
        DO 800 REPS=1, MAXREP
          TRANSQ(REPS) = QHTUPR(ALF,REPS)
          TRANSR(REPS) = RSHAT(ALF,REPS)
          TRNSTR(REPS) = RHTSTR(ALF,REPS)
          TRANBR(REPS) = RSHTBR(ALF,REPS)
800    CONTINUE
        CALL SHSORT(TRANSQ,KEY1,MAXREP)
        CALL SHSORT(TRANSR,KEY2,MAXREP)
        CALL SHSORT(TRNSTR,KEY3,MAXREP)
        CALL SHSORT(TRANBR,KEY4,MAXREP)
        DO 900 REPS=1, MAXREP
          QHTUPR(ALF,REPS) = TRANSQ(REPS)
          RSHAT(ALF,REPS) = TRANSR(REPS)
          RHTSTR(ALF,REPS) = TRNSTR(REPS)
          RSHTBR(ALF,REPS) = TRANBR(REPS)
900    CONTINUE
700  CONTINUE

```

PRINT OUTPUT REPORT HEADINGS

```

      WRITE(1,6666)
      WRITE(1,6667) MAXREP
      WRITE(1,6668) K
      WRITE(1,6669)
      IF(K.EQ.5) THEN
        WRITE(1,6699)
      ELSE
      END IF
      WRITE(1,6670) MSTRQ
      WRITE(1,6671)
      WRITE(1,C1C15)
      WRITE(1,3334) AI
      WRITE(1,0007)
      WRITE(1,C1C15)
      WRITE(1,3334) QI
      WRITE(1,0005)
      WRITE(1,C1C15)
      WRITE(1,3335) N
      WRITE(1,6674)

```

COMPUTE THE VALUE RS OF THE TRUE SYSTEM REL. FNCTN. (SERIES SYSTEM)
 *** AND FOR THE 5-COMPONENT BRIDGE STRUCTURE***

```

      CALL RSRS(QI,K,RS)
      WRITE(1,(' ',////'THE TRUE SERIES SYSTEM ',

```

```

+'RELIABILITY VALUE IS: ',T51,F8.5') RS
CALL RBRIDG(QI,K,RSBRDG)
IF(K.EQ.5) THEN
WRITE(1,(' ',,////'THE TRUE BRIDGE STRUCTURE ',
+'RELIABILITY VALUE IS: ',T51,F8.5') RSBRDG
ELSE
END IF
WRITE(1,6675)

```

COMPUTE THE DIFFERENCE 'DELTAR' BTWN. RS AND RSHAT OF THE THEO
 RETICAL QUANTILE GIVEN BY ALFA(MUST USE SORTED RSHAT ARRAY)

```

IF(FLAG.EQ.1) THEN
WRITE(1,5755)
ELSE
END IF
DO 450 ALF=1, MAXALF
  QUANTL(ALF) = MAXREP * (1 - ALFA(ALF))
  DELTAR(ALF) = RS - RSHAT(ALF,QUANTL(ALF))
  DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
  IF(FLAG.EQ.1) THEN
    DELSTR(ALF) = RS - RHTSTR(ALF,QUANTL(ALF))
    WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
    WRITE(1,5656) RHTSTR(ALF,QUANTL(ALF))
    WRITE(1,5657) DELSTR(ALF)
  ELSE
  END IF
  IF(K.EQ.5) THEN
    DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
    WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
    WRITE(1,5666) RSHTBR(ALF,QUANTL(ALF))
    WRITE(1,5667) DELBRG(ALF)
  ELSE
  END IF
  WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
  WRITE(1,5556) RSHAT(ALF,QUANTL(ALF))
  WRITE(1,5557) DELTAR(ALF)
450 CONTINUE
PRINT *, 'QUANTL(1) IS: ', QUANTL(1)
PRINT *, 'QUANTL(2) IS: ', QUANTL(2)

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RSHAT ***** ***

```

WRITE(1,6676)
DO 400 ALF=1,MAXALF
  TRUQNT(ALF) = 0
  DO 500 REPS=1, MAXREP
    DIFF(REPS) = RS - RSHAT(ALF,REPS)
500 CONTINUE
  DO 600 REPS=1, MAXREP
    IF(ABS(DIFF(REPS)).LE.EPS) THEN
      TRUQNT(ALF) = REPS
      WRITE(1,(' ',,/'TRUE CONFIDENCE LIMIT IS: ',
      F8.4)')
      + (TRUQNT(ALF) / REAL(MAXREP)) * 100.
    +
  END IF

```

```

        GO TO 620
    ELSEIF(DIFF(REPS).LT.0.) THEN
        TRUQNT(ALF) = REPS
        GO TO 610
    ELSE
    END IF
600 CONTINUE
610 IF(TRUQNT(ALF).EQ.0.) THEN
    WRITE(1,4443) ALFA(ALF)
    WRITE(1,(' ',' ',/'THE SMALLEST',
+      '' DIFFERENCE BETWEEN RS AND RSHAT IS: ',F10.5)') DIFF(
+      MAXREP)
    ELSEIF(TRUQNT(ALF).EQ.1.) THEN
    WRITE(1,4442) ALFA(ALF)
    WRITE(1,(' ',' ',/'ALL RSHAT',
+      '' ARE GREATER THAN RS'''))
    ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+      THEN
+      WRITE(1,4444) ALFA(ALF),
+      (TRUQNT(ALF) / REAL(MAXREP)) * 100.
    WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF))
    WRITE(1,4446)
    ELSE
+      WRITE(1,4444) ALFA(ALF),
+      ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
    WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF)-1)
    WRITE(1,4447)
620 END IF
400 CONTINUE

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RSHTBR (BRIDGE) ***** ***

```

    IF(K.EQ.5) THEN
    DO 401 ALF=1,MAXALF
    TRUQNT(ALF) = 0
    DO 501 REPS=1, MAXREP
    DIFF(REPS) = RSBRDG - RSHTBR(ALF,REPS)
501 CONTINUE
    DO 601 REPS=1, MAXREP
    IF(ABS(DIFF(REPS)).LE.EPS) THEN
    TRUQNT(ALF) = REPS
    WRITE(1,(' ',' ',/'TRUE CONFIDENCE LIMIT IS: ',
+      F8.4)')
+      (TRUQNT(ALF) / REAL(MAXREP)) * 100.
    GO TO 621
    ELSEIF(DIFF(REPS).LT.0.) THEN
    TRUQNT(ALF) = REPS
    GO TO 611
    ELSE
    END IF
601 CONTINUE
611 IF(TRUQNT(ALF).EQ.0.) THEN
    WRITE(1,4443) ALFA(ALF)
    WRITE(1,(' ',' ',/'THE SMALLEST',

```

```

+         '' DIFFERENCE BETWEEN RSBRDG AND RSHTBR IS: '' ,
+         F10.5)') DIFF(MAXREP)
ELSEIF(TRUQNT(ALF).EQ.1.) THEN
    WRITE(1,4442) ALFA(ALF)
    WRITE(1,('' '' ,/'' ALL RSHTBR'' ,
+         '' ARE GREATER THAN RSBRDG''))'')
ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+     THEN
+         WRITE(1,4444) ALFA(ALF),
+         (TRUQNT(ALF) / REAL(MAXREP)) * 100.
+         WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF))
+         WRITE(1,4446)
ELSE
+         WRITE(1,4444) ALFA(ALF),
+         ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
+         WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF)-1)
+         WRITE(1,4447)
621     END IF
401     CONTINUE
    ELSE
    END IF

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 ***** RHTSTR *****

```

IF(FLAG.EQ.1) THEN
DO 4400 ALF=1,MAXALF
TRUQNT(ALF) = 0
DO 5500 REPS=1, MAXREP
    DIFF(REPS) = RS - RHTSTR(ALF,REPS)
5500 CONTINUE
DO 6600 REPS=1, MAXREP
    IF(ABS(DIFF(REPS)).LE.EPS) THEN
        TRUQNT(ALF) = REPS
        WRITE(1,('' '' ,/'' TRUE CONFIDENCE LIMIT IS: '' ,
+         F8.4)')
+         (TRUQNT(ALF) / REAL(MAXREP)) * 100.
        GO TO 6620
    ELSEIF(DIFF(REPS).LT.0.) THEN
        TRUQNT(ALF) = REPS
        GO TO 6610
    ELSE
    END IF
6600 CONTINUE
6610 IF(TRUQNT(ALF).EQ.0.) THEN
    WRITE(1,4443) ALFA(ALF)
    WRITE(1,('' '' ,/'' THE SMALLEST'' ,
+         '' DIFFERENCE BETWEEN RS AND RHTSTR IS: '' ,
+         F9.5)') DIFF(MAXREP)
ELSEIF(TRUQNT(ALF).EQ.1.) THEN
    WRITE(1,4442) ALFA(ALF)
    WRITE(1,('' '' ,/'' ALL RHTSTR'' ,
+         '' ARE GREATER THAN RS''))'')
+     ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+     THEN

```

```

+       WRITE(1,4444) ALFA(ALF),
        (TRUQNT(ALF) / REAL(MAXREP)) * 100.
        WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF))
        WRITE(1,4446)
      ELSE
+       WRITE(1,4444) ALFA(ALF),
        ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
        WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF)-1)
        WRITE(1,4447)
6620     END IF
4400 CONTINUE
      ELSE
      END IF

```

PRINT THE ARRAYS PERTINENT TO THE OUPUT OF EACH REPLICATION

```

      IF(PRNT.EQ.1) THEN
        I = 1
185  WRITE(1,REPSHD) ALFA(SELCTA), ALFA(SELCTA),
+ALFA(SELCTB),ALFA(SELCTB),ALFA(SELCTA),ALFA(SELCTA),ALFA(SELCTB),
+ALFA(SELCTB)
175  IF(I.GE.(MAXREP + 1)) THEN
        GOTO 180
      ELSE
+       IF( (I.EQ.71).OR.(I.EQ.211).OR.(I.EQ.351).OR.(I.EQ.491).OR.
        (I.EQ.631).OR.(I.EQ.771).OR.(I.EQ.911).OR.(I.EQ.1051) ) THEN
          I = I + 70
          WRITE(1,(''+'))
          GOTO 185
        ELSE
+       WRITE(1,3336) I, INT(DEGFR(I)), CHISQ(1,I), QHTUPR(1,I),
        CHISQ(2,I), QHTUPR(2,I)
        END IF
        IF(I + 70.LE.MAXREP) THEN
+       WRITE(1,3337) I+70,INT(DEGFR(I+70)),CHISQ(1,I+70),
        QHTUPR(1,I+70),CHISQ(2,I+70),QHTUPR(2,I+70)
        ELSE
        END IF
        I = I + 1
        GOTO 175
180  END IF
      ELSE
      ENDIF
9999 WRITE(1,(''THE TOTAL NO OF REPS WAS:'' ,I8)') TOTREP
      WRITE(1,(''THE TOTAL NO OF EFFECTIVE REPS WAS:'' ,I8)') LOOP
      WRITE(1,(''THE TOTAL NO OF NO FAILURE RUNS WAS:'' ,I8)') ZFAILS
      WRITE(1,(''AVERAGE NO. OF COMPONENTS PER REPLICATION WITH '' ,
+''NO FAILURES:'' ,F5.2)') ZFPREP / MAXREP
      WRITE(1,(''THE TOTAL NO OF RUNS WITH FAILURES WAS:'' ,I8)') FAILS
0008 FORMAT (/ 3X,'C 1',5X,'C 2',
+5X,'C 3',5X,'C 4',5X,'C 5',5X,'C 6',5X,'C 7',5X,
+'C 8',5X,'C 9',5X,'C 10',4X,'C 11',4X,
+'C 12',4X,'C 13',4X,'C 14',4X,'C 15')
0009 FORMAT(/1X,'REP NO',2X,'DF',1X,'CHISQR(',F4.3,')',1X,
+'QHTUPR(',F4.3,')',1X,'CHISQR(',F4.3,')',1X,'QHTUPR(',F4.3,')',

```



```

+2X, 'REP NO', 2X, 'DF', 1X, 'CHISQR(', F4.3, ')', 1X,
+'QHTUPR(', F4.3, ')', 1X, 'CHISQR(', F4.3, ')', 1X, 'QHTUPR(', F4.3, ')')/'
0001 FORMAT (///'UNIFORM RANDOM DEVIATES ARE: ')
0002 FORMAT (///'BERNOULLI TRIALS ARE: ')
0003 FORMAT (///'TOTAL NO. OF FAILURES FOR EACH COMPONENT: ')
0004 FORMAT (///'ESTIMATED UNRELIABILITY FOR EACH COMPONENT: ')
0005 FORMAT (///'TOTAL NUMBER OF MISSION TESTS: ')
0006 FORMAT (///'ESTIMATED WEIGHTS FOR EACH COMPONENT: ')
0007 FORMAT (///'Q I FOR EACH COMPONENT: ')
1111 FORMAT (15F8.5)

2222 FORMAT (/1X, 15(I4, 4X))
3333 FORMAT (/1X, 15(I4, 4X))
3334 FORMAT (/15F8.5)
3335 FORMAT (/1X, 15(I4, 4X))
3336 FORMAT (T3, I4, T9, I3, T13, F11.5, T27, F8.5, T39, F11.5, T53, F8.5)
3337 FORMAT ('+', T67, I4, T73, I3, T77, F11.5, T91, F8.5, T103, F11.5, T117, F8.5)
4442 FORMAT (' ', ///'THE RESULTING (1 - ', F4.3, ') CONFIDENCE ',
+'LIMIT IS: ', T50, ' 00.000 ')
4443 FORMAT (' ', ///'THE RESULTING (1 - ', F4.3, ') CONFIDENCE ',
+'LIMIT IS: ', T50, ' 100.0000 ')
4444 FORMAT (' ', ///'THE RESULTING (1 - ', F4.3, ') CONFIDENCE ',
+'LIMIT IS: ', T50, F8.4)
4445 FORMAT (' ', /'THE RSHAT VALUE CLOSEST TO RS IS: ', T51, F8.5)
4446 FORMAT (' ', /'(FIRST NEGATIVE DIFFERENCE)')
4447 FORMAT (' ', /'(ELEMENT PRECEDING FIRST NEGATIVE DIFFERENCE)')
4448 FORMAT (' ', /'THE RHTSTR VALUE CLOSEST TO RS IS: ', T51, F8.5)
4449 FORMAT (' ', /'THE RSHTBR VALUE CLOSEST TO RSBRDG IS: ', T51, F8.5)
5555 FORMAT (' ', ///'THE ', I4, '(1-', F4.3, ') QUANTILE IS: ', T49, F8.3)
5556 FORMAT (' ', /'THE VALUE OF RSHAT FOR THAT QUANTILE IS: ', T51, F8.5)
5557 FORMAT (' ', /'THE DIFFERENCE(RS - RSHAT) IS: ', T51, F8.5)
5656 FORMAT (' ', /'THE VALUE OF RHTSTR FOR THAT QUANTILE IS: ', T51, F8.5)
5666 FORMAT (' ', /'THE VALUE OF RSHTBR FOR THAT QUANTILE IS: ', T51, F8.5)
5657 FORMAT (' ', /'THE DIFFERENCE(RS - RHTSTR) IS: ', T51, F8.5)
5667 FORMAT (' ', /'THE DIFFERENCE(RS - RSHTBR) IS: ', T51, F8.5)
5755 FORMAT (' ', ///'SINCE THE NO. OF MISSION TESTS IS THE SAME FOR',
+' ALL COMPONENTS THE CLOSED FORM SERIES SYSTEM RELIABILITY ',
+' 'RHTSTR' IS COMPUTED')
6666 FORMAT ('+', '*****',
+'***** RUN INPUT SETTINGS *****',
+'*****')
6667 FORMAT (' ', ///'NUMBER OF REPLICATIONS: ', T50, I4)
6668 FORMAT (' ', ///'NUMBER OF COMPONENTS: ', T50, I4)
6669 FORMAT (' ', ///'SYSTEM RELIABILITY FUNCTION: ', T50, 'SERIES')
6699 FORMAT (' ', ///'SYSTEM RELIABILITY FUNCTION: ', T50, 'BRIDGE')
6670 FORMAT (' ', ///'MASTER UNRELIABILITY USED: ', T50, F8.5)
6671 FORMAT (' ', ///'INPUT WEIGHTS(A SUB I'S): ')
6674 FORMAT ('+', ///'*****',
+'**R U N R E S U L T S*****',
+'*****')
6675 FORMAT ('+', ///'*****',
+' ESTIMATE ERRORS *****',
+'*****')
6676 FORMAT ('+', ///'*****',
+' TRUE CONFIDENCE LIMITS *****',
+'*****')

```

END

APPENDIX F. FORTRAN CODE FOR THE PREFERRED LOWER
CONFIDENCE LIMIT FOR SYSTEM RELIABILITY (PARALLEL
SYSTEM)

PROGRAM ZFYSCN

```

*****
*
*   TITLE: BINOMIAL INTERVAL ESTIMATION PROCEDURE
*           ZERO FAILURES ALLOWED; NO SCALING
*   AUTHOR: E. F. BELLINI, LT, USN
*   MODIFIED BY: LT VALERIE A. COVINGTON, USN   (MAR 90)
*   DATE: NOV 89
*
*   THIS PROGRAM COMPUTES THE TRUE CONFIDENCE LEVEL FOR THE ESTIMATE
*   RELIABILITY OF A SERIES AND BRIDGE SYSTEM GIVEN THE RELIABILITY
*   OF THEIR COMPONENTS
*
*   IN ITS PRESENT CONFIGURATION THIS PROGRAM IS SET UP TO RUN 12
*   TIMES EACH TIME PRODUCING 1000 REPLICATIONS USING A DIFFERENT
*   SET OF INPUT DATA.  RUN THE PROGRAM FROM CMS BY TYPING 'B1 EXEC'.
*   THE REXX EXEC PROGRAM
*   'B1' CALLS THE I'  FILES TO BE READ AND NAMES THE 12 OUTPUT
*   FILES RESULTING  THE 12 CONSECUTIVE RUNS.  BY EDITING THE
*   INDEX COUNTERS 1,  K OF THE 'B1' EXEC ONE CAN RUN ANY USER-
*   SPECIFIC RUN FROM JUST ONE RUN TO ALL 12.
*
*   VARIABLES USED
*
*   AHATI : WEIGHT ESTIMATES FOR EACH COMPONENT
*   AI    : INPUT WEIGHTS FOR EACH COMPONENT
*   ALFA  : LEVELS OF SIGNIFICANCE
*   BIGF  : TOTAL NO. OF FAILURES FOR EACH REPLICATION
*   CHISQ : CHI-SQUARE RANDOM VARIABLE VALUE
*   C1C15 : FORMAT LABEL
*   DEGR  : DEGREES OF FREEDOM
*   DELBRG : DIFFERENCE FOR BRIDGE SYSTEM
*   DELSTR : DIFFERENCE FOR SERIES SYSTEM- CLOSED FORM
*   DELTAR : DIFFERENCE FOR SERIES SYSTEM
*   DIFF  : DIFFERENCE (TRUE REL. - ESTIMATED REL. )
*   EPS   : SMALL QUANTITY(CONSTANT)
*   ERROR : PARAMETER FOR IMSL ROUTINE
*   FAILS : COUNTS NO. OF REPLICATIONS WITH AT LST. 1 FAILURE
*   FI    : NO. OF FAILURES FOR EACH COMPONENT(ALL MISSION TST)*
*   FLAG  : 1 IF ALL COMP. HAVE SAME NO. OF MISSION TESTS
*   INC   : INCREMENT STEP SIZE FOR ROUTINE USMNMX
*   KEY1  : ARRAY OF INDECES FOR ROUTINE SHSORT
*   KEY2  : ARRAY OF INDECES FOR ROUTINE SHSORT
*   KEY3  : ARRAY OF INDECES FOR ROUTINE SHSORT
*   KEY4  : ARRAY OF INDECES FOR ROUTINE SHSORT
*   KK    : ARRAY SIZING PARAMETER FOR THE MAX NO OF COMPONENTS*
*   LOOP  : COUNTS NO. OF REPLICATION PERFORMED
*   MAXALF : MAX NO. OF SIGNIFICANT LEVELS DESIRED(ARRAY SIZING)*

```

```

*      MAXREP : MAX NO. OF REPLICATIONS *
*      MAXRUN : MAX NO. OF PROGRAM ITERATIONS ALLOWED *
*      MSTRQ  : MASTER UNRELIABILITY(USED WITH AI'S TO CALC. QI'S) *
*      MULT   : MULTIPLIER FOR RANDOM NO. GENERATOR SRND *
*      N      : NO. OF MISSION TEST FOR EACH COMPONENT *
*      NIMAX  : MAX NO. OF MISSION TESTS *
*      NIMIM  : MIN NO. OF MISSION TESTS *
*      NINDX  : INDEX NO. OF MAX NO. OF MISSION TESTS *
*      NIREAL : NO. OF MISSION TESTS TRANSFORMED TO REAL *
*      NMAX   : MAX NO. OF MISSION TESTS FOR OUTPUT CONTROL *
*      NPRNT  : FLAG FOR DETAILED REPORT OUTPUT *
*      PRNT   : SAME AS ABOVE(PARAMETER) *
*      QHATI  : UNRELIABILITY ESTIMATES FOR EACH COMPONENT *
*      QHTMAX : LARGEST QHATI *
*      QHTUPR : UPPER LIMIT ON SYSTEM UNRELIABILITY *
*      QI     : INPUT UNRELIABILITY FOR EACH COMPONENT *
*      QINDX  : INDEX *
*      QUANTL : QUANTILE *
*      REPSHD : REPLICATIONS HEADING FORMAT NUMBER *
*      RHTSTR : SERIES SYSTEM RELIABILITY ESTIMATE(CLOSED FORM) *
*      RS     : TRUE SERIES SYSTEM RELIABILITY *
*      RSBRDG : TRUE BRIDGE SYSTEM RELIABILITY *
*      RSHAT  : SERIES SYSTEM RELIABILITY ESTIMATE *
*      RSHTBR : BRIDGE SYSTEM RELIABILITY ESTIMATE *
*      SEED   : PARAMETER *
*      SELCTA : SIGNIFICANCE LEVEL SELECTION *
*      SELCTB : SIGNIFICANCE LEVEL SELECTION *
*      SORT   : PARAMETER FOR ROUTINE SRND *
*      SUMNAI : SUM OF THE PRODUCT OF NI'S AND AI'S *
*      TEMP   : TEMPORARY ARRAY *
*      TOTREP : TOTAL NUMBER OF PROGRAM ITERATIONS *
*      TRANBR : TEMPORARY ARRAY *
*      TRANSQ : TEMPORARY ARRAY *
*      TRANSR : TEMPORARY ARRAY *
*      TRIALS : BERNOULLI TRIALS ARRAY (2-DIM) *
*      TRNSTR : TEMPORARY ARRAY *
*      TRUQNT : TRUE QUANTILE *
*      UNIRV  : UNIFORM RANDOM DEVIATES (2-DIM) *
*      ZFAILS : TOTAL NUMBER OF REPLICATIONS WITH ZERO FAILURES *
*      ZFPREP : NO. OF COMPNTS. WITH ZERO FAILURES PER REPLICATION *
*
*****

```

```

PARAMETER (KK=10,MAXALF=2,NPRNT=0)
PARAMETER (MAXREP=1000, MAXRUN=2000, EPS=.000001)
REAL*4 UNIRV(15,1000),TEMP(1000),QI(KK),AI(KK),AHATI(KK)
REAL*4 QHATI(KK), NMAX, NNMAX, QHTMAX, CHISQR(5,5), ALFA(MAXALF)
REAL*4 DF(5),AALFA(5),SUMNAI,RSHAT(MAXALF,MAXREP),RS
REAL*4 KEY1(MAXREP),KEY2(MAXREP),KEY3(MAXREP),TRNSTR(MAXREP)
REAL*4 DEGFR(MAXREP), QHTUPR(MAXALF,MAXREP),CHISQ(MAXALF,MAXREP)
REAL*4 QUPA1(MAXREP), QUPA2(MAXREP),RHTSTR(MAXALF,MAXREP)
REAL*4 DELTAR(MAXALF), TRANSQ(MAXREP),TRANSR(MAXREP),DIFF(MAXREP)
REAL*4 DELSTR(MAXALF),NIMIN,NIMAX,NIREAL(KK).
REAL*4 RSHTBR(MAXALF,MAXREP),DELBRG(MAXALF),KEY4(MAXREP)
REAL*4 TRANBR(MAXREP),RSBRDG,MSTRQ
REAL*4 ZFPREP

```

```

REAL*4 AVGN,SUC,STUD,PTEMP(10),FDEG1,FDEG2,P3,S3,S4

INTEGER SEED, MULT, SORT, TRIALS(15,1000), BIGF, FI(KK), N(KK)
INTEGER NINDX, QINDX, ERROR, REPS, SELCTA, SELCTK, TOTREP
INTEGER C1C15, REPSHD, SELCTB, ALF, FLAG, LOOP, PRNT
INTEGER QUANTL(MAXALF), TRUQNT(MAXALF), ZFAILS, FAILS, INC
INTEGER NTEST, FCT, HFI

CHARACTER*8 LOOPS0(MAXREP)

DATA SEED/123457/, MULT/1/, INC/1/
DATA AALFA/.01,.05,.9,.95,.99/, DF/1,5,10,30,40/
DATA ALFA/.20,.050/
DATA SORT/0/

ASSIGN 8 TO C1C15
ASSIGN 9 TO REPSHD

* CALL COMPRS
  PRNT = NPRNT

  DO 12 I=1, KK
    AI(I) = 9999.
    N(I) = 99999999
  12 CONTINUE

  READ(03,*)K,MSTRQ

  DO 11 I=1, K
    READ(03,*) AI(I),N(I)
  11 CONTINUE

  IF(K.NE.5) THEN
    WRITE(1,('WARNING: BRIDGE STRUCTURE ',
+'ONLY USES THE FIRST 5 COMPONENTS'))
  ELSE
    END IF

***INITIALIZE THE QHTUPR ARRAY OF UNRELIABILITY REPLICATIONS, ***
* RSHAT ARRAY OF ESTIMATE SYSTEM RELIABILITY REPLICATIONS *
* AND RHTSTR ARRAY OF EST. SYST. REL. FOR A SERIES SYST WHEN *
***ALL THE COMPONENT MISSION TESTS ARE EQUAL IN NUMBER ***

  DO 172 ALF=1, MAXALF
    DO 173 REPS=1, MAXREP
      QHTUPR(ALF, REPS) = 0.
      RSHAT(ALF, REPS) = 0.
      RHTSTR(ALF, REPS) = 0.
      RSHTBR(ALF, REPS) = 0.
      LOOPS0(REPS)='*****'
    173 CONTINUE
  172 CONTINUE

***SET FLAG TO 1 IF ALL COMPONENTS HAVE SAME NO. OF MISSION TESTS***

```

```

FLAG=1
DO 50 I=1,K -1
    IF((N(I) - N(I+1)).NE.0) THEN
        FLAG=0
    ELSE
        END IF
50 CONTINUE
PRINT *, 'FLAG IS: ', FLAG

```

MAIN PROGRAM OUTER LOOP START(EVERY LOOP IS ONE REPLICATION)

```

ZFPREP = 0.
ZFAILS = 0
FAILS = 0
TOTREP = 0
LOOP = 0
10 IF(LOOP.LT.MAXREP) THEN
    LOOP = LOOP + 1
    IF(TOTREP.LT.MAXRUN) THEN
        TOTREP = TOTREP + 1
    SELCTA = 1
    SELCTB = 2

```

FILL ARRAY KEY(REPS) WITH INTEGERS 1 TO K TO BE USED AS OUTPUT
 ***OF THE SUBROUTINE SHSORT ***

```

DO 95 REPS=1, MAXREP
    KEY1(REPS) = REPS
    KEY2(REPS) = REPS
    KEY3(REPS) = REPS
    KEY4(REPS) = REPS
95 CONTINUE

```

CALCULATE NMAX NOT TO PRINT LONGER THAN THE MAX SAMPLE SIZE

CALCULATE THE MAXIMUM NO. OF TRIALS AND ITS INDEX NO.

```
CALL IMAX(N,K,NMAX,NINDX)
```

CALCULATE THE QI'S FROM THE GIVEN MASTER Q AND THE AI'S

```

DO 115 I=1, K
    QI(I) = MSTRQ * AI(I)
115 CONTINUE
C    S3=QI(3)
    S4=QI(1)**.20
DO 120 I=1,15
    DO 125 J=1,500
        UNIRV(I,J) = 999.
        TRIALS(I,J) = 99999
125 CONTINUE
120 CONTINUE

```

: DRAW UNIFORM (0,1) RV'S AND CONVERT TO BERNOULLI TRIALS
 DO 130 I=1, K

```

        CALL SRND(SEED, TEMP, N(I), MULT, SORT)
        DO 135 J=1, N(I)
            UNIRV(I,J) = TEMP(J)
            IF (UNIRV(I,J).LE. 1 - QI(I)) THEN
                TRIALS(I,J) = 0
            ELSE
                TRIALS(I,J) = 1
            END IF
135  CONTINUE
130  CONTINUE

```

CALCULATE THE NO. OF FAILURES FOR EACH COMPONENT

```

        DO 150 I=1, K
            FI(I) = 0
150  CONTINUE
        IONECT = 0

```

CALCULATE THE F SUB I'S AND THE GRAND TOTAL NO. OF FAILURES

```

        BIGF = 0
        DO 155 I=1, K
            DO 160 J=1, N(I)
                FI(I) = FI(I) + TRIALS(I,J)
160  CONTINUE
            IF (FI(I).EQ. 0) THEN
                ZFPREP=ZFPREP+1
            ELSE
                ENDIF
            BIGF = BIGF + FI(I)

```

```

        QHATI(I) = REAL(FI(I)) / N(I)
155  CONTINUE

```

Changes for parallel system

```

        IF (FI(1) .EQ. 0) THEN
            ZFPREP=ZFPREP - 1
        ENDIF
        BIGF = BIGF - FI(1)
        FI(1) = 0
        DO 161 I=1,N(1)
            HFI=0
            DO 162 J=1,5
                CALL SRND(SEED, PTEMP(J), N(1), MULT, SORT)
                IF (PTEMP(J) .GT. 1-S4) THEN
                    HFI = HFI + 1
                ENDIF
162  CONTINUE
            IF (HFI .EQ. 5) THEN
                FI(1) = FI(1) + 1
            ENDIF
161  CONTINUE
        BIGF = BIGF + FI(1)
        IF (FI(1) .EQ. 0) THEN

```

```

        ZFPREP = ZFPREP + 1
    ELSE
    ENDIF
C     IF (FI(3) .EQ. 0) THEN
C         ZFPREP=ZFPREP - 1
C     ENDIF
C     BIGF = BIGF - FI(3)
C     FI(3) = 0
C     DO 163 I=1,N(3)
C         HFI=0
C         DO 164 J=1,3
C             CALL SRND(SEED, PTEMP(J), N(3), MULT, SORT)
C             IF (PTEMP(J) .GT. 1-S3) THEN
C                 HFI = HFI + 1
C             ENDIF
C164     CONTINUE
C         IF (HFI .GE. 2) THEN
C             FI(3) = FI(3) + 1
C         ENDIF
C163     CONTINUE
C     BIGF = BIGF + FI(3)
C     IF (FI(3) .EQ. 0) THEN
C         ZFPREP = ZFPREP + 1
C     ELSE
C     ENDIF
    DO 19 I=1,K
        QI(I) = MSTRQ * AI(I)
        QHATI(I) = REAL(FI(I)) / N(I)

19     CONTINUE
C     QI(4) = QI(4)**2
C     P3 = 1. -QI(3)
C     QI(3) = 1. -((3.*P3**2*QI(3))+(P3**3))
***COUNTS NUMBER OF COMPONENTS THAT HAVE FAILED***

    DO 156 I=1,K
        IF (FI(I) .NE. 0) IONECT=IONECT+1
156     CONTINUE

***CASE WHERE NO COMPONENTS HAVE ANY FAILURES***

    IF(BIGF.EQ.0) THEN
        LOOPSO(LOOP)=' *ZERO* '
        ZFAILS = ZFAILS + 1
        AVGN=0.0
        DO 200 I=1,K
            AVGN=AVGN+REAL(N(I))
200     CONTINUE
        AVGN=AVGN/REAL(K)
        DO 205 ALF=1, MAXALF
            RSHAT(ALF,LOOP)= ALFA(ALF)**(1./AVGN)
            IF(FLAG.EQ.1) THEN
                RHTSTR(ALF,LOOP)=ALFA(ALF)**(1./N(1))
            ELSE
            END IF
*         PRINT *,LOOP',LOOP,' RSHAT',RSHAT(ALF,LOOP)

```



```

205      CONTINUE
          DEGFR(LOOP) = 2.
          GO TO 10
      ELSE
          FAILS = FAILS + 1
      END IF

***COUNTS NUMBER OF COMPONENTS THAT FAIL RECORDS NO. COMPT TESTS***
      FCT=0
      DO 202 I=1,K
          IF (FI(I) .NE. 0) THEN
              FCT=FCT+1
              NTEST=N(I)
          ENDIF
      202  CONTINUE

***FIND THE MAX OF THE INDIVIDUAL COMPONENT UNRELIABILITIES***

      CALL RMAX(QHATI, K, QHTMAX, QINDEX)
      IF (LOOP .EQ. 1) THEN
          ENDIF

***CALCULATE THE AHAT SUB I'S (WEIGHT ESTIMATES)***
***IF COMPONENT HAS NO FAILURES AHAT SUB I IS ZERO***

      SUMNAI = 0.
      DO 165 I=1, K
          AHATI(I) = QHATI(I) / QHTMAX
          SUMNAI = SUMNAI + N(I) * AHATI(I)
      165  CONTINUE

***1 COMPONENT FAILURE SERIES SYSTEM***

      IF (FCT .EQ. 1) THEN
          LOOPSO(LOOP)=' *ONECF* '
          DO 305 ALF=1,MAXALF
              SUC=REAL(NTEST-BIGF)
              FDEG1=2.*(REAL(BIGF)+1.)
              FDEG2=2.*SUC
              STUD=FIN(1.-ALFA(ALF),FDEG1,FDEG2)
              RSHAT(ALF,LOOP)=SUC/(SUC+(REAL(BIGF)+1.)*STUD)
              PRINT *, 'SUC=', SUC, 'FAIL=', BIGF, 'NTEST=', NTEST
              PRINT *, 'FIN=', STUD
              PRINT *, 'ALFA=', ALFA(ALF), 'RSHAT=', RSHAT(ALF, LOOP)
              IF (FLAG .EQ. 1) THEN
                  RHTSTR(ALF, LOOP)=RSHAT(ALF, LOOP)
              ELSE
                  ENDIF
          305  CONTINUE
      ENDIF

***CALCULATE 1 REPLICATION OF UPPR ALFA C.L. ON SYSTEM RELIABILITY***

      DEGFR(LOOP) = 2 * (1 + BIGF)

      DO 170 ALF=1, MAXALF

```

```

CALL MDCHI(1 - ALFA(ALF),DEGFR(LOOP),CHISQ(ALF,LOOP), ERROR)
QHTUPR(ALF,LOOP) = CHISQ(ALF,LOOP) / (2 * SUMNAI)
IF(FLAG.EQ.1) THEN
    RHTSTR(ALF,LOOP) = 1 -(CHISQ(ALF,LOOP) / REAL(2*N(1)))
ELSE
END IF

```

CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR COMPNTS. IN SERIES

```

IF (FCT .NE. 1) THEN
CALL RHTSR(S(QHTUPR(ALF,LOOP), AHATI,K, RSHAT(ALF,LOOP))
ENDIF

```

CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR BRIDGE STRUCTURE

```

IF (IONECT .NE. 1) THEN
CALL RHTBRG(QHTUPR(ALF,LOOP),AHATI,K,RSHTBR(ALF,LOOP))
ENDIF

```

170 CONTINUE

EXACTLY 1 COMPONENT FAILS AND REDUNDANT COMPONENT

```

IF ((IONECT .EQ. 1) .AND. (K .EQ. 5)) THEN
DO 207 I=1, K
    NIREAL(I) = REAL(N(I))
207 CONTINUE
CALL USMNMX(NIREAL,K,INC,NIMIN,NIMAX)
DO 206 ALF=1,MAXALF
    RSHTBR(ALF,LOOP)=ALFA(ALF)**(1./NIMIN)
206 CONTINUE
ENDIF

```

THIS ELSE AND ENDIF ARE FOR THE TEST AGAINST MAXRUN

```

ELSE
WRITE(1,(' ',I6),/'PROGRAM EXCEEDED THE MAX NO. OF RUNS',
+' ALLOWED OF: ',I6) TOTREP
GOTO 9999

```

```

END IF
GOTO 10
END IF

```

```

C WRITE(2,('UNSORTED RSHAT 1 IS: ',/10(F8.5)))
C +(RSHAT(1,LOOP), LOOP=1, MAXREP)
C WRITE(2,('UNSORTED RSHAT 2 IS: ',/10(F8.5)))
C +(RSHAT(2,LOOP), LOOP=1, MAXREP)
C IF(FLAG.EQ.1) THEN
C WRITE(2,('UNSORTED RHTSTR 1 IS: ',/10(F8.5)))
C +(RHTSTR(1,LOOP), LOOP=1, MAXREP)
C WRITE(2,('UNSORTED RHTSTR 2 IS: ',/10(F8.5)))
C +(RHTSTR(2,LOOP), LOOP=1, MAXREP)
C ELSE
C END IF
C IF(K.EQ.5) THEN
C WRITE(2,('UNSORTED RSHTBR 1 IS: ',/10(F8.5)))

```

```

C   +(RSHTBR(1,LOOP), LOOP=1, MAXREP)
C       WRITE(2,('UNSORTED RSHTBR 2 IS:',/10(F8.5)))
C   +(RSHTBR(2,LOOP), LOOP=1, MAXREP)
C   ELSE
C   END IF
C   WRITE (2,('ZERO AND ONE FAILURE REPS:',/10(A8)))
C   + (LOOPS0(LOOP),LOOP=1,MAXREP)

```

SORT THE ARRAYS OF SYSTEM UNRELIABILITIES(1 FOR EACH CONF. LEVEL)

```

      DO 700 ALF=1, MAXALF
        DO 800 REPS=1, MAXREP
          TRANSQ(REPS) = QHTUPR(ALF,REPS)
          TRANSR(REPS) = RSHAT(ALF,REPS)
          TRNSTR(REPS) = RHTSTR(ALF,REPS)
          TRANBR(REPS) = RSHTBR(ALF,REPS)
800    CONTINUE
        CALL SHSORT(TRANSQ,KEY1,MAXREP)
        CALL SHSORT(TRANSR,KEY2,MAXREP)
        CALL SHSORT(TRNSTR,KEY3,MAXREP)
        CALL SHSORT(TRANBR,KEY4,MAXREP)
        DO 900 REPS=1, MAXREP
          QHTUPR(ALF,REPS) = TRANSQ(REPS)
          RSHAT(ALF,REPS) = TRANSR(REPS)
          RHTSTR(ALF,REPS) = TRNSTR(REPS)
          RSHTBR(ALF,REPS) = TRANBR(REPS)
900    CONTINUE
700  CONTINUE

```

PRINT OUTPUT REPORT HEADINGS

```

      WRITE(1,6666)
      WRITE(1,6667) MAXREP
      WRITE(1,6668) K
      WRITE(1,6669)
      IF(K.EQ.5) THEN
        WRITE(1,6699)
      ELSE
      END IF
      WRITE(1,6670) MSTRQ
      WRITE(1,6671)
      WRITE(1,C1C15)
      WRITE(1,3334) AI
      WRITE(1,0007)
      WRITE(1,C1C15)
      WRITE(1,3334) QI
      WRITE(1,0005)
      WRITE(1,C1C15)
      WRITE(1,3335) N
      WRITE(1,6674)

```

COMPUTE THE VALUE RS OF THE TRUE SYSTEM REL. FNCTN. (SERIES SYSTEM)
 *** AND FOR THE 5-COMPONENT BRIDGE STRUCTURE***

```

CALL RSR(S(QI,K,RS)
WRITE(1,(' ' ' ',////'THE TRUE SERIES SYSTEM ' ',
+'RELIABILITY VALUE IS: ',T51,F8.5')) RS
CALL RBRIDG(QI,K,RSBRDG)
IF(K.EQ.5) THEN
WRITE(1,(' ' ' ',////'THE TRUE BRIDGE STRUCTURE ' ',
+'RELIABILITY VALUE IS: ',T51,F8.5')) RSBRDG
ELSE
END IF
WRITE(1,6675)

```

COMPUTE THE DIFFERENCE 'DELTAR' BTWN. RS AND RSHAT OF THE THEO
 RETICAL QUANTILE GIVEN BY ALFA(MUST USE SORTED RSHAT ARRAY)

```

IF(FLAG.EQ.1) THEN
WRITE(1,5755)
ELSE
END IF
DO 450 ALF=1, MAXALF
  QUANTL(ALF) = MAXREP * (1 - ALFA(ALF))
  DELTAR(ALF) = RS - RSHAT(ALF,QUANTL(ALF))
  DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
  IF(FLAG.EQ.1) THEN
    DELSTR(ALF) = RS - RHTSTR(ALF,QUANTL(ALF))
    WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
    WRITE(1,5656) RHTSTR(ALF,QUANTL(ALF))
    WRITE(1,5657) DELSTR(ALF)
  ELSE
  END IF
  IF(K.EQ.5) THEN
    DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
    WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
    WRITE(1,5666) RSHTBR(ALF,QUANTL(ALF))
    WRITE(1,5667) DELBRG(ALF)
  ELSE
  END IF
  WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
  WRITE(1,5556) RSHAT(ALF,QUANTL(ALF))
  WRITE(1,5557) DELTAR(ALF)
450 CONTINUE
PRINT *, 'QUANTL(1) IS: ', QUANTL(1)
PRINT *, 'QUANTL(2) IS: ', QUANTL(2)

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RSHAT ***** ***

```

WRITE(1,6676)
DO 400 ALF=1,MAXALF
  TRUQNT(ALF) = 0
  DO 500 REPS=1, MAXREP
    DIFF(REPS) = RS - RSHAT(ALF,REPS)
500 CONTINUE
  DO 600 REPS=1, MAXREP
    IF(ABS(DIFF(REPS)).LE.EPS) THEN
      TRUQNT(ALF) = REPS
      WRITE(1,(' ' ' ',/'TRUE CONFIDENCE LIMIT IS: ',

```

```

+           F8.4)')
+           (TRUQNT(ALF) / REAL(MAXREP)) * 100.
           GO TO 620
ELSEIF(DIFF(REPS).LT. 0.) THEN
           TRUQNT(ALF) = REPS
           GO TO 610
ELSE
           END IF
600 CONTINUE
610 IF(TRUQNT(ALF).EQ. 0.) THEN
           WRITE(1,4443) ALFA(ALF)
           WRITE(1,(' ', ' ', /'THE SMALLEST',
+           ' ' DIFFERENCE BETWEEN RS AND RSHAT IS: ', F10.5)') DIFF(
+           MAXREP)
ELSEIF(TRUQNT(ALF).EQ. 1.) THEN
           WRITE(1,4442) ALFA(ALF)
           WRITE(1,(' ', ' ', /'ALL RSHAT',
+           ' ' ARE GREATER THAN RS'))')
ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE. ABS(DIFF(TRUQNT(ALF) - 1)))
+           THEN
+           WRITE(1,4444) ALFA(ALF),
+           (TRUQNT(ALF) / REAL(MAXREP)) * 100.
+           WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF))
+           WRITE(1,4446)
ELSE
+           WRITE(1,4444) ALFA(ALF),
+           ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
+           WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF)-1)
+           WRITE(1,4447)
620 END IF
400 CONTINUE

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
*** ***** RSHTBR (BRIDGE) ***** ***

```

IF(K.EQ. 5) THEN
DO 401 ALF=1, MAXALF
TRUQNT(ALF) = 0
DO 501 REPS=1, MAXREP
           DIFF(REPS) = RSBRDG - RSHTBR(ALF, REPS)
501 CONTINUE
DO 601 REPS=1, MAXREP
           IF(ABS(DIFF(REPS)).LE. EPS) THEN
+           TRUQNT(ALF) = REPS
+           WRITE(1,(' ', ' ', /'TRUE CONFIDENCE LIMIT IS: ',
+           F8.4)')
+           (TRUQNT(ALF) / REAL(MAXREP)) * 100.
           GO TO 621
ELSEIF(DIFF(REPS).LT. 0.) THEN
           TRUQNT(ALF) = REPS
           GO TO 611
ELSE
           END IF
601 CONTINUE
611 IF(TRUQNT(ALF).EQ. 0.) THEN

```

```

        WRITE(1,4443) ALFA(ALF)
        WRITE(1,(' ' ' ',/' 'THE SMALLEST' ' ',
+         ' ' DIFFERENCE BETWEEN RSBRDG AND RSHTBR IS: ' ',
+         F10.5)') DIFF(MAXREP)
        ELSEIF(TRUQNT(ALF).EQ.1.) THEN
        WRITE(1,4442) ALFA(ALF)
        WRITE(1,(' ' ' ',/' 'ALL RSHTBR' ' ',
+         ' ' ARE GREATER THAN RSBRDG' ' '))
        ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+         THEN
        WRITE(1,4444) ALFA(ALF),
+         (TRUQNT(ALF) / REAL(MAXREP)) * 100.
        WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF))
        WRITE(1,4446)
        ELSE
+         WRITE(1,4444) ALFA(ALF),
        ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
        WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF)-1)
        WRITE(1,4447)
621     END IF
401     CONTINUE
        ELSE
        END IF

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RHTSTR ***** ***

```

IF(FLAG.EQ.1) THEN
DO 4400 ALF=1,MAXALF
TRUQNT(ALF) = 0
DO 5500 REPS=1, MAXREP
DIFF(REPS) = RS - RHTSTR(ALF,REPS)
5500 CONTINUE
DO 6600 REPS=1, MAXREP
IF(ABS(DIFF(REPS)).LE.EPS) THEN
TRUQNT(ALF) = REPS
WRITE(1,(' ' ' ',/' 'TRUE CONFIDENCE LIMIT IS: ' ',
+         F8.4)')
+         (TRUQNT(ALF) / REAL(MAXREP)) * 100.
GO TO 6620
ELSEIF(DIFF(REPS).LT.0.) THEN
TRUQNT(ALF) = REPS
GO TO 6610
ELSE
END IF
6600 CONTINUE
6610 IF(TRUQNT(ALF).EQ.0.) THEN
WRITE(1,4443) ALFA(ALF)
WRITE(1,(' ' ' ',/' 'THE SMALLEST' ' ',
+         ' ' DIFFERENCE BETWEEN RS AND RHTSTR IS: ' ',
+         F9.5)') DIFF(MAXREP)
ELSEIF(TRUQNT(ALF).EQ.1.) THEN
WRITE(1,4442) ALFA(ALF)
WRITE(1,(' ' ' ',/' 'ALL RHTSTR' ' ',
+         ' ' ARE GREATER THAN RS' ' '))

```

```

+ ELSEIF(ABS(DIFF(TRUQNT(ALF))). LE. ABS(DIFF(TRUQNT(ALF) - 1)))
+ THEN
+ WRITE(1,4444) ALFA(ALF),
+ (TRUQNT(ALF) / REAL(MAXREP)) * 100.
+ WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF))
+ WRITE(1,4446)
+ ELSE
+ WRITE(1,4444) ALFA(ALF),
+ ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
+ WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF)-1)
+ WRITE(1,4447)
6620 END IF
4400 CONTINUE
ELSE
END IF

```

PRINT THE ARRAYS PERTINENT TO THE OUPUT OF EACH REPLICATION

```

IF(PRNT.EQ.1) THEN
I = 1
185 WRITE(1,REPSHD) ALFA(SELCTA), ALFA(SELCTA),
+ALFA(SELCTB),ALFA(SELCTB),ALFA(SELCTA),ALFA(SELCTA),ALFA(SELCTB),
+ALFA(SELCTB)
175 IF(I.GE.(MAXREP + 1)) THEN
GOTO 180
ELSE
IF( (I.EQ.71).OR.(I.EQ.211).OR.(I.EQ.351).OR.(I.EQ.491).OR.
+ (I.EQ.631).OR.(I.EQ.771).OR.(I.EQ.911).OR.(I.EQ.1051) ) THEN
I = I + 70
WRITE(1,'(''+''')')
GOTO 185
ELSE
WRITE(1,3336) I, INT(DEGFR(I)), CHISQ(1,I), QHTUPR(1,I),
+ CHISQ(2,I), QHTUPR(2,I)
END IF
IF(I + 70.LE.MAXREP) THEN
+ WRITE(1,3337) I+70,INT(DEGFR(I+70)),CHISQ(1,I+70),
+ QHTUPR(1,I+70),CHISQ(2,I+70),QHTUPR(2,I+70)
ELSE
END IF
I = I + 1
GOTO 175
180 END IF
ELSE
ENDIF
9999 WRITE(1,'(''THE TOTAL NO OF REPS WAS:'' ,I8)') TOTREP
WRITE(1,'(''THE TOTAL NO OF EFFECTIVE REPS WAS:'' ,I8)') LOOP
WRITE(1,'(''THE TOTAL NO OF NO FAILURE RUNS WAS:'' ,I8)') ZFAILS
WRITE(1,'(''AVERAGE NO. OF COMPONENTS PER REPLICATION WITH '' ,
+ ''NO FAILURES:'' ,F5.2)') ZFPREP / MAXREP
WRITE(1,'(''THE TOTAL NO OF RUNS WITH FAILURES WAS:'' ,I8)') FAILS
0008 FORMAT (/ 3X,'C 1',5X,'C 2',
+5X,'C 3',5X,'C 4',5X,'C 5',5X,'C 6',5X,'C 7',5X,
+'C 8',5X,'C 9',5X,'C 10',4X,'C 11',4X,
+'C 12',4X,'C 13',4X,'C 14',4X,'C 15')

```

```

0009 FORMAT(/1X,'REP NO',2X,'DF',1X,'CHISQR(',F4.3,')',1X,
+'QHTUPR(',F4.3,')',1X,'CHISQR(',F4.3,')',1X,'QHTUPR(',F4.3,')',
+2X,'REP NO',2X,'DF',1X,'CHISQR(',F4.3,')',1X,
+'QHTUPR(',F4.3,')',1X,'CHISQR(',F4.3,')',1X,'QHTUPR(',F4.3,')'//)
0001 FORMAT (///'UNIFORM RANDOM DEVIATES ARE: ')
0002 FORMAT (///'BERNOULLI TRIALS ARE: ')
0003 FORMAT (///'TOTAL NO. OF FAILURES FOR EACH COMPONENT: ')
0004 FORMAT (///'ESTIMATED UNRELIABILITY FOR EACH COMPONENT: ')
0005 FORMAT (///'TOTAL NUMBER OF MISSION TESTS: ')
0006 FORMAT (///'ESTIMATED WEIGHTS FOR EACH COMPONENT: ')
0007 FORMAT (///'Q I FOR EACH COMPONENT: ')
1111 FORMAT (15F8.5)

2222 FORMAT (/1X,15(I4,4X))
3333 FORMAT (/1X,15(I4,4X))
3334 FORMAT (/15F8.5)
3335 FORMAT (/1X,15(I4,4X))
3336 FORMAT (T3,I4,T9,I3,T13,F11.5,T27,F8.5,T39,F11.5,T53,F8.5)
3337 FORMAT ('+',T67,I4,T73,I3,T77,F11.5,T91,F8.5,T103,F11.5,T117,F8.5)
4442 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+'LIMIT IS: ',T50,' 00.000 ')
4443 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+'LIMIT IS: ',T50,' 100.0000 ')
4444 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+'LIMIT IS: ',T50,F8.4)
4445 FORMAT (' ',/'THE RSHAT VALUE CLOSEST TO RS IS: ',T51,F8.5)
4446 FORMAT (' ',/'(FIRST NEGATIVE DIFFERENCE)')
4447 FORMAT (' ',/'(ELEMENT PRECEEDING FIRST NEGATIVE DIFFERENCE)')
4448 FORMAT (' ',/'THE RHTSTR VALUE CLOSEST TO RS IS: ',T51,F8.5)
4449 FORMAT (' ',/'THE RSHTBR VALUE CLOSEST TO RSBRDG IS: ',T51,F8.5)
5555 FORMAT (' ',///'THE ',I4,'(1-',F4.3,') QUANTILE IS: ',T49,F8.3)
5556 FORMAT (' ',/'THE VALUE OF RSHAT FOR THAT QUANTILE IS: ',T51,F8.5)
5557 FORMAT (' ',/'THE DIFFERENCE(R3 - RSHAT) IS: ',T51,F8.5)
5656 FORMAT (' ',/'THE VALUE OF RHTSTR FOR THAT QUANTILE IS: ',T51,F8.5)
5666 FORMAT (' ',/'THE VALUE OF RSHTBR FOR THAT QUANTILE IS: ',T51,F8.5)
5657 FORMAT (' ',/'THE DIFFERENCE(RS - RHTSTR) IS: ',T51,F8.5)
5667 FORMAT (' ',/'THE DIFFERENCE(RS - RSHTBR) IS: ',T51,F8.5)
5755 FORMAT (' ',///'SINCE THE NO. OF MISSION TESTS IS THE SAME FOR',
+' ALL COMPONENTS THE CLOSED FORM SERIES SYSTEM RELIABILITY ',
+' 'RHTSTR' IS COMPUTED')
6666 FORMAT ('+', '*****',
+'***** RUN INPUT SETTINGS *****',
+'*****')
6667 FORMAT (' ',///'NUMBER OF REPLICATIONS: ',T50,I4)
6668 FORMAT (' ',///'NUMBER OF COMPONENTS: ',T50,I4)
6669 FORMAT (' ',///'SYSTEM RELIABILITY FUNCTION: ',T50,'SERIES')
6699 FORMAT (' ',///'SYSTEM RELIABILITY FUNCTION: ',T50,'BRIDGE')
6670 FORMAT (' ',///'MASTER UNRELIABILITY USED: ',T50,F8.5)
6671 FORMAT (' ',///'INPUT WEIGHTS(A SUB I'S): ')
6674 FORMAT ('+',///'*****',
+'**R U N R E S U L T S*****',
+'*****')
6675 FORMAT ('+',///'*****',
+' ESTIMATE ERRORS *****',
+'*****')
6676 FORMAT ('+',///'*****',

```


+ ' TRUE CONFIDENCE LIMITS *****',
+ '*****')
END

**APPENDIX G. FORTRAN CODE FOR ALTERNATE PROCEDURE A
FOR ESTIMATING THE LOWER CONFIDENCE LIMIT FOR SYSTEM
RELIABILITY**

PROGRAM ZFYSCN

```

*****
*
* TITLE: BINOMIAL INTERVAL ESTIMATION PROCEDURE
*         ZERO FAILURES ALLOWED; NO SCALING
* AUTHOR: E. F. BELLINI, LT, USN
*         DATE: NOV 89
*
* THIS PROGRAM COMPUTES THE TRUE CONFIDENCE LEVEL FOR THE ESTIMATE
* RELIABILITY OF A SERIES AND BRIDGE SYSTEM GIVEN THE RELIABILITY
* OF THEIR COMPONENTS
*
* IN ITS PRESENT CONFIGURATION THIS PROGRAM IS SET UP TO RUN 12
* TIMES EACH TIME PRODUCING 1000 REPLICATIONS USING A DIFFERENT
* SET OF INPUT DATA. RUN THE PROGRAM FROM CMS BY TYPING 'B1 EXEC'.
* THE REXX EXEC PROGRAM
* 'B1' CALLS THE INPUT FILES TO BE READ AND NAMES THE 12 OUTPUT
* FILES RESULTING FROM THE 12 CONSECUTIVE RUNS. BY EDITING THE
* INDEX COUNTERS I, J, K OF THE 'B1' EXEC ONE CAN RUN ANY USER-
* SPECIFIC RUN FROM JUST ONE RUN TO ALL 12.
*
* VARIABLES USED
*
* AHATI : WEIGHT ESTIMATES FOR EACH COMPONENT
* AI    : INPUT WEIGHTS FOR EACH COMPONENT
* ALFA  : LEVELS OF SIGNIFICANCE
* BIGF  : TOTAL NO. OF FAILURES FOR EACH REPLICATION
* CHISQ : CHI-SQUARE RANDOM VARIABLE VALUE
* C1C15 : FORMAT LABEL
* DEGFR : DEGREES OF FREEDOM
* DELBRG : DIFFERENCE FOR BRIDGE SYSTEM
* DELSTR : DIFFERENCE FOR SERIES SYSTEM- CLOSED FORM
* DELTAR : DIFFERENCE FOR SERIES SYSTEM
* DIFF  : DIFFERENCE (TRUE REL. - ESTIMATED REL.)
* EPS   : SMALL QUANTITY(CONSTANT)
* ERROR : PARAMETER FOR IMSL ROUTINE
* FAILS : COUNTS NO. OF REPLICATIONS WITH AT LST. ? FAILURE
* FI    : NO. OF FAILURES FOR EACH COMPONENT(ALL MISSION TST)*
* FLAG  : 1 IF ALL COMP. HAVE SAME NO. OF MISSION TESTS
* INC   : INCREMENT STEP SIZE FOR ROUTINE USMMX
* KEY1  : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY2  : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY3  : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY4  : ARRAY OF INDECES FOR ROUTINE SHSORT
* KK    : ARRAY SIZING PARAMETER FOR THE MAX NO OF COMPONENTS*
* LOOP  : COUNTS NO. OF REPLICATION PERFORMED
* MAXALF : MAX NO. OF SIGNIFICANT LEVELS DESIRED(ARRAY SIZING)*
* MAXREP : MAX NO. OF REPLICATIONS
*

```

```

*      MAXRUN : MAX NO. OF PROGRAM ITERATIONS ALLOWED      *
*      MSTRQ  : MASTER UNRELIABILITY(USED WITH AI'S TO CALC. QI'S) *
*      MULT   : MULTIPLIER FOR RANDOM NO. GENERATOR SRND    *
*      N      : NO. OF MISSION TEST FOR EACH COMPONENT      *
*      NIMAX  : MAX NO. OF MISSION TESTS                    *
*      NIMIM  : MIN NO. OF MISSION TESTS                    *
*      NINDX  : INDEX NO. OF MAX NO. OF MISSION TESTS      *
*      NIREAL : NO. OF MISSION TESTS TRANSFORMED TO REAL    *
*      NMAX   : MAX NO. OF MISSION TESTS FOR OUTPUT CONTROL *
*      NPRNT  : FLAG FOR DETAILED REPORT OUTPUT             *
*      PRNT   : SAME AS ABOVE(PARAMETER)                    *
*      QHATI  : UNRELIABILITY ESTIMATES FOR EACH COMPONENT  *
*      QHTMAX : LARGEST QHATI                                *
*      QHTUPR : UPPER LIMIT ON SYSTEM UNRELIABILITY         *
*      QI     : INPUT UNRELIABILITY FOR EACH COMPONENT      *
*      QINDX  : INDEX                                        *
*      QUANTL : QUANTILE                                     *
*      REPSHD : REPLICATIONS HEADING FORMAT NUMBER         *
*      RHTSTR : SERIES SYSTEM RELIABILITY ESTIMATE(CLOSED FORM) *
*      RS     : TRUE SERIES SYSTEM RELIABILITY              *
*      RSDRDG : TRUE BRIDGE SYSTEM RELIABILITY             *
*      RSHAT  : SERIES SYSTEM RELIABILITY ESTIMATE         *
*      RSHTBR : BRIDGE SYSTEM RELIABILITY ESTIMATE         *
*      SEED   : PARAMETER                                    *
*      SELCTA : SIGNIFICANCE LEVEL SELECTION                *
*      SELCTB : SIGNIFICANCE LEVEL SELECTION                *
*      SORT   : PARAMETER FOR ROUTINE SRND                 *
*      SUMNAI : SUM OF THE PRODUCT OF NI'S AND AI'S        *
*      TEMP   : TEMPORARY ARRAY                             *
*      TOTREP : TOTAL NUMBER OF PROGRAM ITERATIONS         *
*      TRANBR : TEMPORARY ARRAY                             *
*      TRANSQ : TEMPORARY ARRAY                             *
*      TRANSR : TEMPORARY ARRAY                             *
*      TRIALS : BERNOULLI TRIALS ARRAY (2-DIM)              *
*      TRNSTR : TEMPORARY ARRAY                             *
*      TRUQNT : TRUE QUANTILE                                *
*      UNIRV  : UNIFORM RANDOM DEVIATES (2-DIM)           *
*      ZFAILS : TOTAL NUMBER OF REPLICATIONS WITH ZERO FAILURES *
*      ZFPREP : NO. OF COMPNTS. WITH ZERO FAILURES PER REPLICATION *
*
*****

```

```

PARAMETER (KK=10,MAXALF=2,NPRNT=0)
PARAMETER (MAXREP=1000, MAXRUN=2000, EPS=.000001)
REAL*4 UNIRV(15,1000),TEMP(1000),QI(KK),AI(KK),AHATI(KK)
REAL*4 QHATI(KK), NMAX, NNMAX, QHTMAX, CHISQR(5,5), ALFA(MAXALF)
REAL*4 DF(5),AALFA(5),SUMNAI,RSHAT(MAXALF,MAXREP),RS
REAL*4 KEY1(MAXREP),KEY2(MAXREP),KEY3(MAXREP),TRNSTR(MAXREP)
REAL*4 DEGFR(MAXREP), QHTUPR(MAXALF,MAXREP),CHISQ(MAXALF,MAXREP)
REAL*4 QUPA1(MAXREP), QUPA2(MAXREP),RHTSTR(MAXALF,MAXREP)
REAL*4 DELTAR(MAXALF), TRANSQ(MAXREP),TRANSR(MAXREP),DIFF(MAXREP)
REAL*4 DELSTR(MAXALF),NIMIN,NIMAX,NIREAL(KK)
REAL*4 RSHTBR(MAXALF,MAXREP),DELBRG(MAXALF),KEY4(MAXREP)
REAL*4 TRANBR(MAXREP), RSDRDG ,MSTRQ
REAL*4 ZFPREP

```

```
INTEGER SEED, MULT, SORT, TRIALS(15,1000), BIGF, FI(KK), N(KK)
INTEGER NINDX, QINDX, ERROR, REPS, SELCTA, SELCTK, TOTREP
INTEGER C1C15, REPSHD, SELCTB, ALF, FLAG, LOOP, PRNT
INTEGER QUANTL(MAXALF), TRUQNT(MAXALF), ZFAILS, FAILS, INC
```

```
DATA SEED/123457/, MULT/1/, INC/1/
DATA AALFA/.01,.05,.9,.95,.99/, DF/1,5,10,30,40/
DATA ALFA/.20,.050/
DATA SORT/0/
```

```
ASSIGN 8 TO C1C15
ASSIGN 9 TO REPSHD
```

```
* CALL COMPRS
  PRNT = NPRNT
```

```
DO 12 I=1, KK
  AI(I) = 9999.
  N(I) = 99999999
12 CONTINUE
```

```
READ(03,*)K, MSTRQ
```

```
DO 11 I=1, K
  READ(03,*) AI(I), N(I)
11 CONTINUE
```

```
IF(K.NE.5) THEN
  WRITE(1,('WARNING: BRIDGE STRUCTURE ',
+ 'ONLY USES THE FIRST 5 COMPONENTS'))
  ELSE
  END IF
```

```
***// INITIALIZE THE QHTUPR ARRAY OF UNRELIABILITY REPLICATIONS, //***
* RSHAT ARRAY OF ESTIMATE SYSTEM RELIABILITY REPLICATIONS *
* AND RHTSTR ARRAY OF EST. SYST. REL. FOR A SERIES SYST WHEN *
***// ALL THE COMPONENT MISSION TESTS ARE EQUAL IN NUMBER //***
```

```
DO 172 ALF=1, MAXALF
  DO 173 REPS=1, MAXREP
    QHTUPR(ALF, REPS) = 0.
    RSHAT(ALF, REPS) = 0.
    RHTSTR(ALF, REPS) = 0.
    RSHTBR(ALF, REPS) = 0.
173 CONTINUE
172 CONTINUE
```

```
***// SET FLAG TO 1 IF ALL COMPONENTS HAVE SAME NO. OF MISSION TESTS****
```

```
FLAG=1
DO 50 I=1, K - 1
  IF((N(I) - N(I+1)).NE.0) THEN
    FLAG=0
  ELSE
  END IF
```

```

50 CONTINUE
   PRINT *, 'FLAG IS: ', FLAG
***// MAIN PROGRAM OUTER LOOP START(EVERY LOOP IS ONE REPLICATION)//***

   ZFPREP = 0.
   ZFAILS = 0
   FAILS = 0
   TOTREP = 0
   LOOP = 0
10  IF(LOOP.LT.MAXREP) THEN
      LOOP = LOOP + 1
      IF(TOTREP.LT.MAXRUN) THEN
         TOTREP = TOTREP + 1

         SELCTA = 1
         SELCTB = 2
***// FILL ARRAY KEY(REPS) WITH INTEGERS 1 TO K TO BE USED AS OUTPUT
***// OF THE SUBROUTINE SHSORT
         DO 95 REPS=1, MAXREP
            KEY1(REPS) = REPS
            KEY2(REPS) = REPS
            KEY3(REPS) = REPS
            KEY4(REPS) = REPS
95  CONTINUE

***// CALCULATE NMAX NOT TO PRINT LONGER THAN THE MAX SAMPLE SIZE

***// CALCULATE THE MAXIMUM NO. OF TRIALS AND ITS INDEX NO. //***

         CALL IMAX(N,K,NMAX,NINDX)

***// CALCULATE THE QI'S FROM THE GIVEN MASTER Q AND THE AI'S

         DO 115 I=1, K
            QI(I) = MSTRQ * AI(I)
115  CONTINUE

         DO 120 I=1,15
            DO 125 J=1,500
               UNIRV(I,J) = 999.
               TRIALS(I,J) = 99999
125  CONTINUE
120  CONTINUE

***// DRAW UNIFORM (0,1) RV'S AND CONVERT TO BERNOULLI TRIALS //***

         DO 130 I=1, K
            CALL SRND(SEED, TEMP, N(I), MULT, SORT)
            DO 135 J=1, N(I)
               UNIRV(I,J) = TEMP(J)
               IF (UNIRV(I,J).LE. 1 - QI(I)) THEN
                  TRIALS(I,J) = 0
               ELSE
                  TRIALS(I,J) = 1
               END IF
135  CONTINUE

```

```

130 CONTINUE

***// CALCULATE THE NO. OF FAILURES FOR EACH COMPONENT //***

DO 150 I=1, K
    FI(I) = 0
150 CONTINUE

***// CALCULATE THE F SUB I'S AND THE GRAND TOTAL NO. OF FAILURES
BIGF = 0
DO 155 I=1, K
    DO 160 J=1, N(I)
        FI(I) = FI(I) + TRIALS(I,J)
160 CONTINUE
    IF(FI(I).EQ.0) THEN
        ZFPREP = ZFPREP + 1
    ELSE
        END IF
***// CALCULATE THE QHAT SUB I'S: F SUB I'S DIVIDED BY N SUB I'S
    QHATI(I) = REAL(FI(I)) / N(I)
    BIGF = BIGF + FI(I)
155 CONTINUE

***// CASE WHERE NO COMPONENTS HAVE ANY FAILURES //***

IF(BIGF.EQ.0) THEN
    ZFAILS = ZFAILS + 1
    DO 200 I=1, K
        NIREAL(I) = REAL(N(I))
200 CONTINUE
    CALL USMNMX(NIREAL,K,INC,NIMIN,NIMAX)
    DO 205 ALF=1, MAXALF
        CALL MDCHI(1 - ALFA(ALF),2.,CHISQ(ALF,LOOP),ERROR)
        RSHAT(ALF,LOOP)= 1 - (CHISQ(ALF,LOOP) / REAL(2 * NIMIN))
    IF(FLAG.EQ.1) THEN
        RHTSTR(ALF,LOOP)=1-(CHISQ(ALF,LOOP) / REAL(2 * N(1)))
    ELSE
        END IF
205 CONTINUE
    IF(PRNT.EQ.1) THEN
        WRITE(1,0007)
        WRITE(1,C1C15)
        WRITE(1,3334) QI
        WRITE(1,0001)
        WRITE(1,C1C15)
        DO 141 J=1,NMAX
            WRITE(1,1111) (UNIRV(I,J), I=1, K)
141 CONTINUE
        WRITE(1,0002)
        WRITE(1,C1C15)
        DO 146 J=1,NMAX
            WRITE(1,2222) (TRIALS(I,J), I=1, K)
146 CONTINUE
        WRITE(1,0003)
        WRITE(1,C1C15)

```

```

WRITE(1,3333) FI
WRITE(1,0005)
WRITE(1,C1C15)
WRITE(1,3335) N
WRITE(1,0004)
WRITE(1,C1C15)
WRITE(1,3334) QHATI
WRITE(1,'(/' 'THE MAXIMUM Q HAT SUB I IS: ' ', T40, F8.5)') QHTMAX
WRITE(1,'(/' 'THE MAXI Q HAT SUB I IS ELMNT NO.: ' ', T40, I5)') QINDX
WRITE(1,'(/' 'THE GRAND TOTAL NO. OF FAILURES IS: ' ', T40, I5)') BIGF
ELSE
ENDIF
      DEGFR(LOOP) = 2.
      GO TO 10
ELSE
      FAILS = FAILS + 1
END IF

***// FIND THE MAX OF THE INDIVIDUAL COMPONENT UNRELIABILITIES
CALL RMAX(QHATI, K, QHTMAX, QINDX)

***// PRINT THE RESULT OF THE MAIN OPERATING ELEMENTS OF THE PROGRAM

IF(PRNT.EQ.1) THEN
WRITE(1,0007)
WRITE(1,C1C15)
WRITE(1,3334) QI
WRITE(1,0001)
WRITE(1,C1C15)
DO 140 J=1,NMAX
      WRITE(1,1111) (UNIRV(I,J), I=1, K)
CONTINUE
140  WRITE(1,0002)
      WRITE(1,C1C15)
DO 145 J=1,NMAX
      WRITE(1,2222) (TRIALS(I,J), I=1, K)
CONTINUE
145  WRITE(1,0003)
      WRITE(1,C1C15)
WRITE(1,3333) FI
WRITE(1,0005)
WRITE(1,C1C15)
WRITE(1,3335) N
WRITE(1,0004)
WRITE(1,C1C15)
WRITE(1,3334) QHATI
WRITE(1,'(/' 'THE MAXIMUM Q HAT SUB I IS: ' ', T40, F8.5)') QHTMAX
WRITE(1,'(/' 'THE MAXI Q HAT SUB I IS ELMNT NO.: ' ', T40, I5)') QINDX
WRITE(1,'(/' 'THE GRAND TOTAL NO. OF FAILURES IS: ' ', T40, I5)') BIGF
ELSE
ENDIF

***// CALCULATE THE AHAT SUB I'S (WEIGHT ESTIMATES)

SUMNAI = 0.
DO 165 I=1, K

```

```

      AHATI(I) = QHATI(I) / QHTMAX
      SUMNAI = SUMNAI + N(I) * AHATI(I)
165  CONTINUE
      IF(PRNT.EQ.1) THEN
          WRITE(1,0006)
          WRITE(1,C1C15)
          WRITE(1,3334) AHATI
      ELSE
      END IF

***// CALCULATE 1 REPLICATION OF UPPR ALFA C.L. ON SYSTEM RELIABILITY

      DEGFR(LOOP) = 2 * (1 + BIGF)

      DO 170 ALF=1, MAXALF
          CALL MDCHI(1 - ALFA(ALF),DEGFR(LOOP),CHISQ(ALF,LOOP), ERROR)
          QHTUPR(ALF,LOOP) = CHISQ(ALF,LOOP) / (2 * SUMNAI)
          IF(FLAG.EQ.1) THEN
              RHTSTR(ALF,LOOP) = 1 -(CHISQ(ALF,LOOP) / REAL(2*N(1)))
          ELSE
          END IF
      *   + (ALF,LOOP), ALFA(ALF)
***// CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR COMPNTS. IN SERIES

          CALL RHTSRS(QHTUPR(ALF,LOOP), AHATI,K, RSHAT(ALF,LOOP))
      *   +T40,F8.5)' ) RSHAT(ALF,LOOP)

***// CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR BRIDGE STRUCTURE ***

          CALL RHTBRG(QHTUPR(ALF,LOOP),AHATI,K,RSHTBR(ALF,LOOP))
170  CONTINUE

***// THIS ELSE AND ENDIF ARE FOR THE TEST AGAINST MAXRUN *****
      ELSE
          WRITE(1,(' '),'PROGRAM EXCEEDED THE MAX NO. OF RUNS',
+ ' ALLOWED OF: ',I6) ) TOTREP
          GOTO 9999

          END IF
          GOTO 10
          END IF
          WRITE(2,(''UNSORTED RSHAT 1 IS: ',/10(F8.5))')
+ (RSHAT(1,LOOP), LOOP=1, MAXREP)
          WRITE(2,(''UNSORTED RSHAT 2 IS: ',/10(F8.5))')
+ (RSHAT(2,LOOP), LOOP=1, MAXREP)
          IF(FLAG.EQ.1) THEN
              WRITE(2,(''UNSORTED RHTSTR 1 IS: ',/10(F8.5))')
+ (RHTSTR(1,LOOP), LOOP=1, MAXREP)
              WRITE(2,(''UNSORTED RHTSTR 2 IS: ',/10(F8.5))')
+ (RHTSTR(2,LOOP), LOOP=1, MAXREP)
          ELSE
          END IF
          IF(K.EQ.5) THEN
              WRITE(2,(''UNSORTED RSHTBR 1 IS: ',/10(F8.5))')
+ (RSHTBR(1,LOOP), LOOP=1, MAXREP)
              WRITE(2,(''UNSORTED RSHTBR 2 IS: ',/10(F8.5))')

```



```
+ (RSHTBR(2, LOOP), LOOP=1, MAXREP)
ELSE
END IF
```

```
***// SORT THE ARRAYS OF SYSTEM UNRELIABILITIES(1 FOR EACH CONF. LEVEL)
```

```
DO 700 ALF=1, MAXALF
  DO 800 REPS=1, MAXREP
    TRANSQ(REPS) = QHTUPR(ALF, REPS)
    TRANSR(REPS) = RSHAT(ALF, REPS)
    TRNSTR(REPS) = RHTSTR(ALF, REPS)
    TRANBR(REPS) = RSHTBR(ALF, REPS)
800  CONTINUE
    CALL SHSORT(TRANSQ, KEY1, MAXREP)
    CALL SHSORT(TRANSR, KEY2, MAXREP)
    CALL SHSORT(TRNSTR, KEY3, MAXREP)
    CALL SHSORT(TRANBR, KEY4, MAXREP)
    DO 900 REPS=1, MAXREP
      QHTUPR(ALF, REPS) = TRANSQ(REPS)
      RSHAT(ALF, REPS) = TRANSR(REPS)
      RHTSTR(ALF, REPS) = TRNSTR(REPS)
      RSHTBR(ALF, REPS) = TRANBR(REPS)
900  CONTINUE
700  CONTINUE
```

```
***// PRINT OUTPUT REPORT HEADINGS *****
```

```
WRITE(1, 6666)
WRITE(1, 6667) MAXREP
WRITE(1, 6668) K
WRITE(1, 6669)
IF(K.EQ.5) THEN
  WRITE(1, 6699)
ELSE
END IF
WRITE(1, 6670) MSTRQ
WRITE(1, 6671)
WRITE(1, C1C15)
WRITE(1, 3334) AI
WRITE(1, 0007)
WRITE(1, C1C15)
WRITE(1, 3334) QI
WRITE(1, 0005)
WRITE(1, C1C15)
WRITE(1, 3335) N
WRITE(1, 6674)
C WRITE(2, 6666)
C WRITE(2, 6667) MAXREP
C WRITE(2, 6668) K
C WRITE(2, 6669)
C IF(K.EQ.5) THEN
C   WRITE(1, 6699)
C ELSE
C END IF
C WRITE(2, 6670) MSTRQ
```

```

C WRITE(2,6671)
C WRITE(2,C1C15)
C WRITE(2,3334) AI
C WRITE(2,0007)
C WRITE(2,C1C15)
C WRITE(2,3334) QI
C WRITE(2,0005)
C WRITE(2,C1C15)
C WRITE(2,3335) N
C WRITE(2,6674)
C
C WRITE(2,(''SORTED RSHAT 1 IS: ',/10(F8.5))')
C +(RSHAT(1,REPS), REPS=1, MAXREP)
C WRITE(2,(''SORTED RSHAT 2 IS: ',/10(F8.5))')
C +(RSHAT(2,REPS), REPS=1, MAXREP)
C IF(FLAG.EQ.1) THEN
C WRITE(2,(''SORTED RHTSTR 1 IS: ',/10(F8.5))')
C +(RHTSTR(1,REPS), REPS=1, MAXREP)
C WRITE(2,(''SORTED RHTSTR 2 IS: ',/10(F8.5))')
C +(RHTSTR(2,REPS), REPS=1, MAXREP)
C ELSE
C END IF
C IF(K.EQ.5) THEN
C WRITE(2,(''SORTED RSHTBR 1 IS: ',/10(F8.5))')
C +(RSHTBR(1,REPS), REPS=1, MAXREP)
C WRITE(2,(''SORTED RSHTBR 2 IS: ',/10(F8.5))')
C +(RSHTBR(2,REPS), REPS=1, MAXREP)
C ELSE
C END IF
C
***// COMPUTE THE VALUE RS OF THE TRUE SYSTEM REL. FNCTN.(SERIES SYSTEM)
***// AND FOR THE 5-COMPONENT BRIDGE STRUCTURE
CALL RSR(S,QI,K,RS)
WRITE(1,('' ',////'THE TRUE SERIES SYSTEM ',
+'RELIABILITY VALUE IS: ',T51,F8.5)') RS
CALL RBRIDG(QI,K,RSBRDG)
IF(K.EQ.5) THEN
WRITE(1,('' ',////'THE TRUE BRIDGE STRUCTURE ',
+'RELIABILITY VALUE IS: ',T51,F8.5)') RSBRDG
ELSE
END IF
WRITE(1,6675)
C
***// COMPUTE THE DIFFERENCE 'DELTAR' BTWN. RS AND RSHAT OF THE THEO
***// RETICAL QUANTILE GIVEN BY ALFA(MUST USE SORTED RSHAT ARRAY)
IF(FLAG.EQ.1) THEN
WRITE(1,5755)
ELSE
END IF
DO 450 ALF=1, MAXALF
QUANTL(ALF) = MAXREP * (1 - ALFA(ALF))
DELTAR(ALF) = RS - RSHAT(ALF,QUANTL(ALF))
DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
IF(FLAG.EQ.1) THEN
DELSTR(ALF) = RS - RHTSTR(ALF,QUANTL(ALF))

```

```

WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
WRITE(1,5656) RHTSTR(ALF,QUANTL(ALF))
WRITE(1,5657) DELSTR(ALF)
ELSE
END IF
IF(K.EQ.5) THEN
DELBRG(ALF) = RSBRDG - RSHTBR(ALF,QUANTL(ALF))
WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
WRITE(1,5666) RSHTBR(ALF,QUANTL(ALF))
WRITE(1,5667) DELBRG(ALF)
ELSE
END IF
WRITE(1,5555) MAXREP, ALFA(ALF), REAL(QUANTL(ALF))
WRITE(1,5556) RSHAT(ALF,QUANTL(ALF))
WRITE(1,5557) DELTAR(ALF)
450 CONTINUE
PRINT *, 'QUANTL(1) IS:', QUANTL(1)
PRINT *, 'QUANTL(2) IS:', QUANTL(2)

***// FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE //***
***//          ***** RSHAT ***** //***
WRITE(1,6676)
DO 400 ALF=1,MAXALF
TRUQNT(ALF) = 0
DO 500 REPS=1, MAXREP
DIFF(REPS) = RS - RSHAT(ALF,REPS)
500 CONTINUE
DO 600 REPS=1, MAXREP
IF(ABS(DIFF(REPS)).LE.EPS) THEN
TRUQNT(ALF) = REPS
WRITE(1,(' ','',/'TRUE CONFIDENCE LIMIT IS:',
F8.4)')
+
+ (TRUQNT(ALF) / REAL(MAXREP)) * 100.
GO TO 620
ELSEIF(DIFF(REPS).LT.0.) THEN
TRUQNT(ALF) = REPS
GO TO 610
ELSE
END IF
600 CONTINUE
610 IF(TRUQNT(ALF).EQ.0.) THEN
WRITE(1,4443) ALFA(ALF)
WRITE(1,(' ','',/'THE SMALLEST',
'' DIFFERENCE BETWEEN RS AND RSHAT IS:',F10.5)') DIFF(
+
+ MAXREP)
ELSEIF(TRUQNT(ALF).EQ.1.) THEN
WRITE(1,4442) ALFA(ALF)
WRITE(1,(' ','',/'ALL RSHAT',
'' AR. GREATER THAN RS''))
+
+ ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
THEN
+
+ WRITE(1,4444) ALFA(ALF),
+ (TRUQNT(ALF) / REAL(MAXREP)) * 100.
WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF))
WRITE(1,4446)
ELSE

```

```

        WRITE(1,4444) ALFA(ALF),
+       ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
        WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF)-1)
        WRITE(1,4447)
620      END IF
400     CONTINUE

***// FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE //***
***// ***** RSHTBR (BRIDGE) ***** //***

IF(K.EQ.5) THEN
DO 401 ALF=1,MAXALF
TRUQNT(ALF) = 0
DO 501 REPS=1, MAXREP
DIFF(REPS) = RSBRDG - RSHTBR(ALF,REPS)
501    CONTINUE
DO 601 REPS=1, MAXREP
IF(ABS(DIFF(REPS)).LE.EPS) THEN
TRUQNT(ALF) = REPS
WRITE(1,(' ', ' ', /' 'TRUE CONFIDENCE LIMIT IS: ',
+       F8.4)')
+       (TRUQNT(ALF) / REAL(MAXREP)) * 100.
GO TO 621
ELSEIF(DIFF(REPS).LT.0.) THEN
TRUQNT(ALF) = REPS
GO TO 611
ELSE
END IF
601    CONTINUE
611    IF(TRUQNT(ALF).EQ.0.) THEN
WRITE(1,4443) ALFA(ALF)
WRITE(1,(' ', ' ', /' 'THE SMALLEST',
+       ' ' DIFFERENCE BETWEEN RSBRDG AND RSHTBR IS: ',
+       F10.5)') DIFF(MAXREP)
ELSEIF(TRUQNT(ALF).EQ.1.) THEN
WRITE(1,4442) ALFA(ALF)
WRITE(1,(' ', ' ', /' 'ALL RSHTBR',
+       ' ' ARE GREATER THAN RSBRDG')')
ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+       THEN
WRITE(1,4444) ALFA(ALF),
+       ((TRUQNT(ALF) / REAL(MAXREP)) * 100.
WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF))
WRITE(1,4446)
ELSE
+       WRITE(1,4444) ALFA(ALF),
+       ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF)-1)
WRITE(1,4447)
621      END IF
401     CONTINUE
ELSE
END IF

```

// FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE //
 // ** RHTSTR ***** //***

```

IF(FLAG.EQ.1) THEN
DO 4400 ALF=1,MAXALF
TRUQNT(ALF) = 0
  DO 5500 REPS=1, MAXREP
    DIFF(REPS) = RS - RHTSTR(ALF,REPS)
5500  CONTINUE
  DO 6600 REPS=1, MAXREP
    IF(ABS(DIFF(REPS)).LE.EPS) THEN
      TRUQNT(ALF) = REPS
      WRITE(1,(' ', ' ', /'TRUE CONFIDENCE LIMIT IS: ',
+          F8.4)')
      +          ((TRUQNT(ALF) / REAL(MAXREP)) * 100.
          GO TO 6620
    ELSEIF(DIFF(REPS).LT.0.) THEN
      TRUQNT(ALF) = REPS
      GO TO 6610
    ELSE
      END IF
6600  CONTINUE
6610  IF(TRUQNT(ALF).EQ.0.) THEN
      WRITE(1,4443) ALFA(ALF)
      WRITE(1,(' ', ' ', /'THE SMALLEST ',
+          ' ' DIFFERENCE BETWEEN RS AND RHTSTR IS: ',
+          F9.5)') DIFF(MAXREP)
      ELSEIF(TRUQNT(ALF).EQ.1.) THEN
      WRITE(1,4442) ALFA(ALF)
      WRITE(1,(' ', ' ', /'ALL RHTSTR ',
+          ' ' ARE GREATER THAN RS ' '))
      ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
      THEN
      WRITE(1,4444) ALFA(ALF),
      +          ((TRUQNT(ALF) / REAL(MAXREP)) * 100.
      WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF))
      WRITE(1,4446)
      ELSE
      WRITE(1,4444) ALFA(ALF),
      +          ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
      WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF)-1)
      WRITE(1,4447)
6620  END IF
4400 CONTINUE
      ELSE
      END IF

```

// PRINT THE ARRAYS PERTINENT TO THE OUPUT OF EACH REPLICATION **

```

IF(PRNT.EQ.1) THEN
  I = 1
185 WRITE(1,REPSHD) ALFA(SELCTA), ALFA(SELCTA),
+ALFA(SELCTB),ALFA(SELCTB),ALFA(SELCTA),ALFA(SELCTA),ALFA(SELCTB),
+ALFA(SELCTB)
175 IF(I.GE.(MAXREP + 1)) THEN
  GOTO 180

```

```

ELSE
+   IF( (I.EQ. 71).OR.(I.EQ. 211).OR.(I.EQ. 351).OR.(I.EQ. 491).OR.
      (I.EQ. 631).OR.(I.EQ. 771).OR.(I.EQ. 911).OR.(I.EQ. 1051) ) THEN
      I = I + 70
      WRITE(1,(''+'))
      GOTO 185
ELSE
+   WRITE(1,3336) I, INT(DEGFR(I)), CHISQ(1,I), QHTUPR(1,I),
      CHISQ(2,I), QHTUPR(2,I)
      END IF
      IF(I + 70.LE. MAXREP) THEN
+     WRITE(1,3337) I+70,INT(DEGFR(I+70)),CHISQ(1,I+70),
      QHTUPR(1,I+70),CHISQ(2,I+70),QHTUPR(2,I+70)
      ELSE
      END IF
      I = I + 1
      GOTO 175
180 END IF
      ELSE
      ENDIF
9999 WRITE(1,(''THE TOTAL NO OF REPS WAS: ',I8)) TOTREP
      WRITE(1,(''THE TOTAL NO OF EFFECTIVE REPS WAS: ',I8)) LOOP
      WRITE(1,(''THE TOTAL NO OF NO FAILURE RUNS WAS: ',I8)) ZFAILS
      WRITE(1,(''AVERAGE NO. OF COMPONENTS PER REPLICATION WITH ',
+ 'NO FAILURES: ',F5.2)) ZFPREP / MAXREP
0008 FORMAT (/ 3X,'C 1',5X,'C 2',
+5X,'C 3',5X,'C 4',5X,'C 5',5X,'C 6',5X,'C 7',5X,
+ 'C 8',5X,'C 9',5X,'C 10',4X,'C 11',4X,
+ 'C 12',4X,'C 13',4X,'C 14',4X,'C 15')
0009 FORMAT(/1X,'REP NO',2X,'DF',1X,'CHISQR(',F4.3,')',1X,
+ 'QHTUPR(',F4.3,')',1X,'CHISQR(',F4.3,')',1X,'QHTUPR(',F4.3,')',
+2X,'REP NO',2X,'DF',1X,'CHISQR(',F4.3,')',1X,
+ 'QHTUPR(',F4.3,')',1X,'CHISQR(',F4.3,')',1X,'QHTUPR(',F4.3,')'/)
0001 FORMAT (///'UNIFORM RANDOM DEVIATES ARE: ')
0002 FORMAT (///'BERNOULLI TRIALS ARE: ')
0003 FORMAT (///'TOTAL NO. OF FAILURES FOR EACH COMPONENT: ')
0004 FORMAT (///'ESTIMATED UNRELIABILITY FOR EACH COMPONENT: ')
0005 FORMAT (///'TOTAL NUMBER OF MISSION TESTS: ')
0006 FORMAT (///'ESTIMATED WEIGHTS FOR EACH COMPONENT: ')
0007 FORMAT (///'Q I FOR EACH COMPONENT: ')
1111 FORMAT (15F8.5)

2222 FORMAT (/1X,15(I4,4X))
3333 FORMAT (/1X,15(I4,4X))
3334 FORMAT (/15F8.5)
3335 FORMAT (/1X,15(I4,4X))
3336 FORMAT (T3,I4,T9,I3,T13,F11.5,T27,F8.5,T39,F11.5,T53,F8.5)
3337 FORMAT ('+',T67,I4,T73,I3,T77,F11.5,T91,F8.5,T103,F11.5,T117,F8.5)
4442 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+ 'LIMIT IS: ',T50,'00.000 ')
4443 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+ 'LIMIT IS: ',T50,'100.0000')
4444 FORMAT (' ',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+ 'LIMIT IS: ',T50,F8.4)

```

```

4445 FORMAT (' ',/'THE RSHAT VALUE CLOSEST TO RS IS: ',T51,F8.5)
4446 FORMAT (' ',/'(FIRST NEGATIVE DIFFERENCE)')
4447 FORMAT (' ',/'(ELEMENT PRECEEDING FIRST NEGATIVE DIFFERENCE)')
4448 FORMAT (' ',/'THE RHTSTR VALUE CLOSEST TO RS IS: ',T51,F8.5)
4449 FORMAT (' ',/'THE RSHTBR VALUE CLOSEST TO RSBRDG IS: ',T51,F8.5)
5555 FORMAT (' ',///'THE ',I4,'(1-',F4.3,') QUANTILE IS: ',T49,F8.3)
5556 FORMAT (' ',/'THE VALUE OF RSHAT FOR THAT QUANTILE IS: ',T51,F8.5)
5557 FORMAT (' ',/'THE DIFFERENCE(RS - RSHAT) IS: ',T51,F8.5)
5656 FORMAT (' ',/'THE VALUE OF RHTSTR FOR THAT QUANTILE IS: ',T51,F8.5)
5666 FORMAT (' ',/'THE VALUE OF RSHTBR FOR THAT QUANTILE IS: ',T51,F8.5)
5657 FORMAT (' ',/'THE DIFFERENCE(RS - RHTSTR) IS: ',T51,F8.5)
5667 FORMAT (' ',/'THE DIFFERENCE(RS - RSHTBR) IS: ',T51,F8.5)
5755 FORMAT (' ',///'SINCE THE NO. OF MISSION TESTS IS THE SAME FOR',
+ ' ALL COMPONENTS THE CLOSED FORM SERIES SYSTEM RELIABILITY ',
+ ' 'RHTSTR' IS COMPUTED')
6666 FORMAT ('+', '*****',
+ '***** RUN INPUT SETTINGS *****',
+ '*****')
6667 FORMAT (' ',///'NUMBER OF REPLICATIONS: ',T50,I4)
6668 FORMAT (' ',///'NUMBER OF COMPONENTS: ',T50,I4)
6669 FORMAT (' ',///'SYSTEM RELIABILITY FUNCTION: ',T50,'SERIES')
6699 FORMAT (' ',///'SYSTEM RELIABILITY FUNCTION: ',T50,'BRIDGE')
6670 FORMAT (' ',///'MASTER UNRELIABILITY USED: ',T50,F8.5)
6671 FORMAT (' ',///'INPUT WEIGHTS(A SUB I'S): ')
6674 FORMAT ('+',///'*****',
+ '**R U N R E S U L T *****',
+ '*****')
6675 FORMAT ('+',///'*****',
+ ' ESTIMATE ERRORS *****',
+ '*****')
6676 FORMAT ('+',///'*****',
+ ' TRUE CONFIDENCE LIMITS *****',
+ '*****')
END

```

**APPENDIX H. FORTRAN CODE FOR ALTERNATE PROCEDURE B
FOR ESTIMATING THE LOWER CONFIDENCE LIMIT FOR SELECTED
SYSTEMS**

PROGRAM ZFYSCN

```

*****
*
* TITLE: BINOMIAL INTERVAL ESTIMATION PROCEDURE
* ZERO FAILURES ALLOWED; NO SCALING
* AUTHOR: E. F. BELLINI, LT, USN
* MODIFIED BY: LT VALERIE A. COVINGTON, USN (MAR 90)
* DATE: NOV 89
*
* THIS PROGRAM COMPUTES THE TRUE CONFIDENCE LEVEL FOR THE ESTIMATE
* RELIABILITY OF A SERIES AND BRIDGE SYSTEM GIVEN THE RELIABILITY
* OF THEIR COMPONENTS
*
* IN ITS PRESENT CONFIGURATION THIS PROGRAM IS SET UP TO RUN 12
* TIMES EACH TIME PRODUCING 1000 REPLICATIONS USING A DIFFERENT
* SET OF INPUT DATA. RUN THE PROGRAM FROM CMS BY TYPING 'B1 EXEC'.
* THE REXX EXEC PROGRAM
* 'B1' CALLS THE INPUT FILES TO BE READ AND NAMES THE 12 OUTPUT
* FILES RESULTING FROM THE 12 CONSECUTIVE RUNS. BY EDITING THE
* INDEX COUNTERS I, J, K OF THE 'B1' EXEC ONE CAN RUN ANY USER-
* SPECIFIC RUN FROM JUST ONE RUN TO ALL 12.
*
* VARIABLES USED
*
* AHATI : WEIGHT ESTIMATES FOR EACH COMPONENT
* AI : INPUT WEIGHTS FOR EACH COMPONENT
* ALFA : LEVELS OF SIGNIFICANCE
* BIGF : TOTAL NO. OF FAILURES FOR EACH REPLICATION
* CHISQ : CHI-SQUARE RANDOM VARIABLE VALUE
* C1C15 : FORMAT LABEL
* DEGFR : DEGREES OF FREEDOM
* DELBRG : DIFFERENCE FOR BRIDGE SYSTEM
* DELSTR : DIFFERENCE FOR SERIES SYSTEM- CLOSED FORM
* DELTAR : DIFFERENCE FOR SERIES SYSTEM
* DIFF : DIFFERENCE (TRUE REL. - ESTIMATED REL. )
* EPS : SMALL QUANTITY(CONSTANT)
* ERROR : PARAMETER FOR IMSL ROUTINE
* FAILS : COUNTS NO. OF REPLICATIONS WITH AT LST. 1 FAILURE
* FI : NO. OF FAILURES FOR EACH COMPONENT(ALL MISSION TST)*
* FLAG : 1 IF ALL COMP. HAVE SAME NO. OF MISSION TESTS
* INC : INCREMENT STEP SIZE FOR ROUTINE USMMX
* KEY1 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY2 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY3 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KEY4 : ARRAY OF INDECES FOR ROUTINE SHSORT
* KK : ARRAY SIZING PARAMETER FOR THE MAX NO OF COMPONENTS*
* LOOP : COUNTS NO. OF REPLICATION PERFORMED
* MAXALF : MAX NO. OF SIGNIFICANT LEVELS DESIRED(ARRAY SIZING)*

```



```

*      MAXREP : MAX NO. OF REPLICATIONS *
*      MAXRUN : MAX NO. OF PROGRAM ITERATIONS ALLOWED *
*      MSTRQ  : MASTER UNRELIABILITY(USED WITH AI'S TO CALC. QI'S) *
*      MULT   : MULTIPLIER FOR RANDOM NO. GENERATOR SRND *
*      N      : NO. OF MISSION TEST FOR EACH COMPONENT *
*      NIMAX  : MAX NO. OF MISSION TESTS *
*      NIMIM  : MIN NO. OF MISSION TESTS *
*      NINDX  : INDEX NO. OF MAX NO. OF MISSION TESTS *
*      NIREAL : NO. OF MISSION TESTS TRANSFORMED TO REAL *
*      NMAX   : MAX NO. OF MISSION TESTS FOR OUTPUT CONTROL *
*      NPRNT  : FLAG FOR DETAILED REPORT OUTPUT *
*      PRNT   : SAME AS ABOVE(PARAMETER) *
*      QHATI  : UNRELIABILITY ESTIMATES FOR EACH COMPONENT *
*      QHTMAX : LARGEST QHATI *
*      QHTUPR : UPPER LIMIT ON SYSTEM UNRELIABILITY *
*      QI     : INPUT UNRELIABILITY FOR EACH COMPONENT *
*      QINDX  : INDEX *
*      QUANTL : QUANTILE *
*      REPSHD : REPLICATIONS HEADING FORMAT NUMBER *
*      RHTSTR : SERIES SYSTEM RELIABILITY ESTIMATE(CLOSED FORM) *
*      RS     : TRUE SERIES SYSTEM RELIABILITY *
*      RSRDGD : TRUE BRIDGE SYSTEM RELIABILITY *
*      RSHAT  : SERIES SYSTEM RELIABILITY ESTIMATE *
*      RSHTBR : BRIDGE SYSTEM RELIABILITY ESTIMATE *
*      SEED   : PARAMETER *
*      SELCTA : SIGNIFICANCE LEVEL SELECTION *
*      SELCTB : SIGNIFICANCE LEVEL SELECTION *
*      SORT   : PARAMETER FOR ROUTINE SRND *
*      SUMNAI : SUM OF THE PRODUCT OF NI'S AND AI'S *
*      TEMP   : TEMPORARY ARRAY *
*      TOTREP : TOTAL NUMBER OF PROGRAM ITERATIONS *
*      TRANBR : TEMPORARY ARRAY *
*      TRANSQ : TEMPORARY ARRAY *
*      TRANSR : TEMPORARY ARRAY *
*      TRIALS : BERNOULLI TRIALS ARRAY (2-DIM) *
*      TRNSTR : TEMPORARY ARRAY *
*      TRUQNT : TRUE QUANTILE *
*      UNIRV  : UNIFORM RANDOM DEVIATES (2-DIM) *
*      ZFAILS : TOTAL NUMBER OF REPLICATIONS WITH ZERO FAILURES *
*      ZFPREP : NO. OF COMPNTS. WITH ZERO FAILURES PER REPLICATION *
*
*****

```

```

PARAMETER (KK=10,MAXALF=2,NPRNT=0)
PARAMETER (MAXREP=1000, MAXRUN=2000, EPS=.000001)
REAL*4 UNIRV(15,1000),TEMP(1000),QI(KK),AI(KK),QHATI(KK)
REAL*4 QHATI(KK),NMAX,NNMAX,QHTMAX,CHISQR(5,5),ALFA(MAXALF)
REAL*4 DF(5),AALFA(5),SUMNAI,RSHAT(MAXALF,MAXREP),RS
REAL*4 KEY1(MAXREP),KEY2(MAXREP),KEY3(MAXREP),TRNSTR(MAXREP)
REAL*4 DEGFR(MAXREP),QHTUPR(MAXALF,MAXREP),CHISQ(MAXALF,MAXREP)
REAL*4 QUPA1(MAXREP),QUPA2(MAXREP),RHTSTR(MAXALF,MAXREP)
REAL*4 DELTAR(MAXALF),TRANSQ(MAXREP),TRANSR(MAXREP),DIFF(MAXREP)
REAL*4 DELSTR(MAXALF),NIMIN,NIMAX,NIREAL(KK)
REAL*4 RSHTBR(MAXALF,MAXREP),DELBGR(MAXALF),KEY4(MAXREP)
REAL*4 TRANBR(MAXREP),RSBRDG,MSTRQ
REAL*4 ZFPREP

```

```
REAL*4 RSHATI(KK),SI(KK),QHATIU(KK)
REAL*4 MXQHAT,RI(KK),SUMRN
REAL*4 QHATMU(MAXALF),CHIVAL(MAXALF),P(KK),PTMP(10)
```

```
INTEGER SEED, MULT, SORT, TRIALS(15,1000), BIGF, FI(KK), N(KK)
INTEGER NINDX, QINDX, ERROR, REPS, SELCTA, SELCTK, TOTREP
INTEGER C1C15, REPSHD, SELCTB, ALF, FLAG, LOOP, PRNT, HFI
INTEGER QUANTL(MAXALF), TRUQNT(MAXALF), ZFAILS, FAILS, INC
```

```
CHARACTER*8 LOOPS0(MAXREP)
```

```
DATA SEED/123457/, MULT/1/, INC/1/
DATA AALFA/.01,.05,.9,.95,.99/, DF/1,5,10,30,40/
DATA ALFA/.20,.050/
DATA SORT/0/
```

```
ASSIGN 8 TO C1C15
ASSIGN 9 TO REPSHD
```

```
* CALL COMPRS
  PRNT = NPRNT
```

```
DO 12 I=1, KK
  AI(I) = 9999.
  N(I) = 99999999
12 CONTINUE
```

```
READ(03,*)K,MSTRQ
```

```
DO 11 I=1,K
  READ(03,*) AI(I),N(I)
11 CONTINUE
```

```
IF(K.NE.5) THEN
  WRITE(1,('WARNING: BRIDGE STRUCTURE ',
+'ONLY USES THE FIRST 5 COMPONENTS'))
  ELSE
  END IF
```

```
***INITIALIZE THE QHTUPR ARRAY OF UNRELIABILITY REPLICATIONS, ***
* RSHAT ARRAY OF ESTIMATE SYSTEM RELIABILITY REPLICATIONS *
* AND RHTSTR ARRAY OF EST. SYST. REL. FOR A SERIES SYST WHEN *
***ALL THE COMPONENT MISSION TESTS ARE EQUAL IN NUMBER ***
```

```
DO 172 ALF=1,MAXALF
  DO 173 REPS=1,MAXREP
    QHTUPR(ALF,REPS) = 0.
    RSHAT(ALF,REPS) = 0.
    RHTSTR(ALF,REPS) = 0.
    RSHTBR(ALF,REPS) = 0.
    LOOPS0(REPS)='*****'
173 CONTINUE
172 CONTINUE
```

```
***SET FLAG TO 1 IF ALL COMPONENTS HAVE SAME NO. OF MISSION TESTS***
```

```

FLAG=1
DO 50 I=1,K -1
    IF((N(I) - N(I+1)).NE.0) THEN
        FLAG=0
    ELSE
        END IF
50 CONTINUE
PRINT *, 'FLAG IS: ', FLAG

```

MAIN PROGRAM OUTER LOOP START(EVERY LOOP IS ONE REPLICATION)

```

ZFPREP = 0.
ZFALS = 0
FAILS = 0
TOTREP = 0
LOOP = 0
10 IF(LOOP.LT.MAXREP) THEN
    LOOP = LOOP + 1
    IF(TOTREP.LT.MAXRUN) THEN
        TOTREP = TOTREP + 1
    SELCTA = 1
    SELCTB = 2

```

FILL ARRAY KEY(REPS) WITH INTEGERS 1 TO K TO BE USED AS OUTPUT
 OF THE SUBROUTINE SHSORT

```

DO 95 REPS=1, MAXREP
    KEY1(REPS) = REPS
    KEY2(REPS) = REPS
    KEY3(REPS) = REPS
    KEY4(REPS) = REPS
95 CONTINUE

```

CALCULATE NMAX NOT TO PRINT LONGER THAN THE MAX SAMPLE SIZE

CALCULATE THE MAXIMUM NO. OF TRIALS AND ITS INDEX NO.

```
CALL IMAX(N,K,NMAX,NINDX)
```

CALCULATE THE QI'S FROM THE GIVEN MASTER Q AND THE AI'S

```

DO 115 I=1, K
    QI(I) = MSTRQ * AI(I)
115 CONTINUE

DO 120 I=1,15
    DO 125 J=1,500
        UNIRV(I,J) = 999.
        TRIALS(I,J) = 99999
125 CONTINUE
120 CONTINUE

```

*** DRAW UNIFORM (0,1) RV'S AND CONVERT TO BERNOULLI TRIALS***

```

DO 130 I=1, K
  CALL SRND(SEED, TEMP, N(I), MULT, SORT)
  DO 135 J=1, N(I)
    UNIRV(I,J) = TEMP(J)
    IF (UNIRV(I,J).LE. 1 - QI(I)) THEN
      TRIALS(I,J) = 0
    ELSE
      TRIALS(I,J) = 1
    END IF
  135 CONTINUE
130 CONTINUE

```

CALCULATE THE NO. OF FAILURES FOR EACH COMPONENT

```

DO 150 I=1, K
  FI(I) = 0
150 CONTINUE
IONECT = 0

```

CALCULATE THE F SUB I'S AND THE GRAND TOTAL NO. OF FAILURES

```

BIGF = 0
DO 155 I=1, K
  DO 160 J=1, N(I)
    FI(I) = FI(I) + TRIALS(I,J)
  160 CONTINUE
  IF(FI(I).EQ.0) THEN
    ZFPREP = ZFPREP + 1
  ELSE
    END IF

```

CALCULATE THE QHAT SUB I'S: F SUB I'S DIVIDED BY N SUB I'S

```

QHATI(I) = REAL(FI(I)) / N(I)
BIGF = BIGF + FI(I)
155 CONTINUE
IF (FI(2) .EQ. 0) THEN
  ZFPREP=ZFPREP - 1
ELSE
ENDIF
BIGF=BIGF-FI(2)
FI(2) = 0
DO 161 I=2,N(2)
  HFI=0
  DO 163 J=1,2
    CALL SRND(SEED, PTEMP(J), N(2), MULT, SORT)
    IF (PTEMP(J) .GT. 1-QI(2)+*.5) THEN
      HFI= HFI+1
    ENDIF
  163 CONTINUE
  IF (HFI .EQ. 2) THEN
    FI(2) = FI(2) +1
  ENDIF
161 CONTINUE
QHATI(2) = REAL(FI(2)) / N(2)
BIGF= BIGF + FI(2)
IF (FI(2) .EQ. 0) THEN

```

```
ZFPREP = ZFPREP + 1
ENDIF
```

```
***COUNTS NUMBER OF COMPONENTS THAT HAVE FAILED***
```

```
DO 136 I=1,K
  IF (FI(I) .NE. 0) IONECT=IONECT+1
136 CONTINUE
```

```
***CASE WHERE NO COMPONENTS HAVE ANY FAILURES***
```

```
IF(BIGF.EQ.0) THEN
  LOOPS0(LOOP)=' *ZERO* '
```

```
*** SERIES ESTIMATE MODIFICATION (NO. OF FAILURES IRRELEVANT)***
```

```
DO 152 I=1,K
  SI(I) = REAL(N(I))-REAL(FI(I))
  RSHATI(I)=SI(I)/(SI(I)+(REAL(FI(I))+1.))
  + * FIN(.50,2.*(REAL(FI(I))+1.),2*SI(I))
  QHATIU(I)=1.-RSHATI(I)
152 CONTINUE
```

```
MXQHAT=0.
DO 154 I=1,K
  IF (QHATIU(I) .GT. MXQHAT) THEN
    MXQHAT= QHATIU(I)
  ENDIF
154 CONTINUE
```

```
DO 156 ALF=1,MAXALF
  CALL MDCHI(ALFA(ALF),2*(1.+REAL(BIGF)),CHIVAL(ALF),ERROR)
  SUMRN=0.
  DO 157 I=1,K
    RI(I)=QHATIU(I)/MXQHAT
    SUMRN=SUMRN+RI(I)*REAL(N(I))
  157 CONTINUE
  QHATMU(ALF)=CHIVAL(ALF)/(2*SUMRN)
156 CONTINUE
```

```
DO 158 ALF=1,MAXALF
  RSHAT(ALF,LOOP)=1.
  DO 159 I=1,K
    RSHAT(ALF,LOOP)=RSHAT(ALF,LOOP)*(1.-(RI(I)*QHATMU(ALF)))
    IF(FLAG.EQ.1) THEN
      RHTSTR(ALF,LOOP) = RSHAT(ALF,LOOP)
    ELSE
      END IF
  159 CONTINUE
158 CONTINUE
```

```
***CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR BRIDGE STRUCTURE***
```

```
DO 141 ALF=1, MAXALF
  DO 142 I=1,K
```

```

          P(I)=1-RI(I)*QHATMU(ALF)
142      CONTINUE
          RSHATR(ALF,LOOP)=P(1)*P(4)+P(2)*P(5)+P(1)*P(3)*
C          P(5)+P(2)*P(3)*P(4)-P(1)*P(2)*P(3)*P(4)-P(1)*
C          P(2)*P(3)*P(5)-P(1)*P(2)*P(4)*P(5)-P(1)*P(3)*
C          P(4)*P(5)-P(2)*P(3)*P(4)*P(5)+2*P(1)*P(2)*
C          P(3)*P(4)*P(5)
141  CONTINUE

```

```

          ZFAILS = ZFAILS + 1
          DEGFR(LOOP) = 2.
          GO TO 10
ELSE
          FAILS = FAILS + 1
END IF

```

FIND THE MAX OF THE INDIVIDUAL COMPONENT UNRELIABILITIES

```
CALL RMAX(QHATI, K, QHTMAX, QINDEX)
```

CALCULATE THE AHAT SUB I'S (WEIGHT ESTIMATES)

```

SUMNAI = 0.
DO 165 I=1, K
    AHATI(I) = QHATI(I) / QHTMAX
    SUMNAI = SUMNAI + N(I) * AHATI(I)
165  CONTINUE

```

1 FAILURE ONLY SERIES SYST.

```

IF (IONECT.EQ. 1) THEN
    LOOPS0(LOOP)='*ONE*'
ENDIF

```

CALCULATE 1 REPLICATION OF UPPR ALFA C.L. ON SYSTEM RELIABILITY

```
DEGFR(LOOP) = 2 * (1 + BIGF)
```

SERIES ESTIMATE MODIFICATION (NO. OF FAILURES IRRELEVANT)

```

DO 162 I=1,K
    SI(I) = REAL(N(I))-REAL(FI(I))
    RSHATI(I)=SI(I)/(SI(I)+(REAL(FI(I))+1.)
+    * FAN(.50,2.*(REAL(FI(I))+1.),2*SI(I)))
    QHATIU(I)=1.-RSHATI(I)
162  CONTINUE

    MXQHAT=0.
    DO 164 I=1,K
        IF (QHATIU(I) .GT. MXQHAT) THEN
            MXQHAT = QHATIU(I)
        ENDIF
164  CONTINUE

```

```

DO 166 ALF=1,MAXALF
CALL MDCHI(ALFA(ALF),2*(1.+REAL(BIGF)),CHIVAL(ALF),ERROR)
SUMRN=0.
DO 167 I=1,K
RI(I)=QHATIU(I)/MXQHAT
SUMRN=SUMRN+RI(I)*REAL(N(I))
167 CONTINUE
QHATMU(ALF)=CHIVAL(ALF)/(2*SUMRN)
166 CONTINUE

DO 168 ALF=1,MAXALF
RSHAT(ALF,LOOP)=1.
DO 169 I=1,K
RSHAT(ALF,LOOP)=RSHAT(ALF,LOOP)*(1.-(RI(I)*QHATMU(ALF)))
IF(FLAG.EQ.1) THEN
RHTSTR(ALF,LOOP) = RSHAT(ALF,LOOP)
ELSE
END IF
169 CONTINUE
168 CONTINUE

```

CALCULATE VALUE OF THE SYSTEM RELIABILITY FOR BRIDGE STRUCTURE

```

DO 170 ALF=1, MAXALF
DO 206 I=1,K
P(I)=1-RI(I)*QHATMU(ALF)
206 CONTINUE
RSHTBR(ALF,LOOP)=P(1)*P(4)+P(2)*P(5)+P(1)*P(3)*
C P(5)+P(2)*P(3)*P(4)-P(1)*P(2)*P(3)*P(4)-P(1)*
C P(2)*P(3)*P(5)-P(1)*P(2)*P(4)*P(5)-P(1)*P(3)*
C P(4)*P(5)-P(2)*P(3)*P(4)*P(5)+2*P(1)*P(2)*
C P(3)*P(4)*P(5)
170 CONTINUE

```

THIS ELSE AND ENDIF ARE FOR THE TEST AGAINST MAXRUN

```

ELSE
WRITE(1,(' : ' / 'PROGRAM EXCEEDED THE MAX NO. OF RUNS',
+' ALLOWED OF: ',I6) ) TOTREP
GOTO 9999

```

```

END IF
GOTO 10
END IF

```

```

WRITE(2,('UNSORTED RSHAT 1 IS: ',/10(F8.5))' )
+(RSHAT(1,LOOP), LOOP=1, MAXREP)
WRITE(2,('UNSORTED RSHAT 2 IS: ',/10(F8.5))' )
+(RSHAT(2,LOOP), LOOP=1, MAXREP)
IF(FLAG.EQ.1) THEN
WRITE(2,('UNSORTED RHTSTR 1 IS: ',/10(F8.5))' )
+(RHTSTR(1,LOOP), LOOP=1, MAXREP)
WRITE(2,('UNSORTED RHTSTR 2 IS: ',/10(F8.5))' )
+(RHTSTR(2,LOOP), LOOP=1, MAXREP)

```

```

ELSE
END IF
IF(K.EQ.5) THEN
WRITE(2,('UNSORTED RSHTBR 1 IS: ',/10(F8.5))')
+(RSHTBR(1,LOOP), LOOP=1, MAXREP)
WRITE(2,('UNSORTED RSHTBR 2 IS: ',/10(F8.5))')
+(RSHTBR(2,LOOP), LOOP=1, MAXREP)
ELSE
END IF
WRITE(2,('ZERO AND ONE FAILURE REPS: ',/10(A8))')
+ (LOOPS0(LOOP), LOOP=1, MAXREP)

```

SORT THE ARRAYS OF SYSTEM UNRELIABILITIES(1 FOR EACH CONF. LEVEL)

```

DO 700 ALF=1, MAXALF
DO 800 REPS=1, MAXREP
TRANSQ(REPS) = QHTUPR(ALF, REPS)
TRANSR(REPS) = RSHAT(ALF, REPS)
TRNSTR(REPS) = RHTSTR(ALF, REPS)
TRANBR(REPS) = RSHTBR(ALF, REPS)
800 CONTINUE
CALL SHSORT(TRANSQ, KEY1, MAXREP)
CALL SHSORT(TRANSR, KEY2, MAXREP)
CALL SHSORT(TRNSTR, KEY3, MAXREP)
CALL SHSORT(TRANBR, KEY4, MAXREP)
DO 900 REPS=1, MAXREP
QHTUPR(ALF, REPS) = TRANSQ(REPS)
RSHAT(ALF, REPS) = TRANSR(REPS)
RHTSTR(ALF, REPS) = TRNSTR(REPS)
RSHTBR(ALF, REPS) = TRANBR(REPS)
900 CONTINUE
700 CONTINUE

```

PRINT OUTPUT REPORT HEADINGS

```

WRITE(1,6666)
WRITE(1,6667) MAXREP
WRITE(1,6668) K
WRITE(1,6669)
IF(K.EQ.5) THEN
WRITE(1,6699)
ELSE
END IF
WRITE(1,6670) MSTRQ
WRITE(1,6671)
WRITE(1,C1C15)
WRITE(1,3334) AI
WRITE(1,0007)
WRITE(1,C1C15)
WRITE(1,3334) QI
WRITE(1,0005)
WRITE(1,C1C15)
WRITE(1,3335) N
WRITE(1,6674)

```



```

IF(ABS(DIFF(REPS)).LE.EPS) THEN
  TRUQNT(ALF) = REPS
  WRITE(1,(' ','',/'TRUE CONFIDENCE LIMIT IS:','
+      F8.4)')
+      (TRUQNT(ALF) / REAL(MAXREP)) * 100.
  GO TO 620
ELSEIF(DIFF(REPS).LT.0.) THEN
  TRUQNT(ALF) = REPS
  GO TO 610
ELSE
  END IF
600 CONTINUE
610 IF(TRUQNT(ALF).EQ.0.) THEN
  WRITE(1,4443) ALFA(ALF)
  WRITE(1,(' ','',/'THE SMALLEST','
+      '' DIFFERENCE BETWEEN RS AND RSHAT IS:','F10.5)') DIFF(
+      MAXREP)
  ELSEIF(TRUQNT(ALF).EQ.1.) THEN
  WRITE(1,4442) ALFA(ALF)
  WRITE(1,(' ','',/'ALL RSHAT','
+      '' ARE GREATER THAN RS''))
  ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+  THEN
  WRITE(1,4444) ALFA(ALF),
+      (TRUQNT(ALF) / REAL(MAXREP)) * 100.
  WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF))
  WRITE(1,4446)
  ELSE
+      WRITE(1,4444) ALFA(ALF),
+      ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
  WRITE(1,4445) RSHAT(ALF,TRUQNT(ALF)-1)
  WRITE(1,4447)
620 END IF
400 CONTINUE

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RSHTBR (BRIDGE) ***** ***

```

IF(K.EQ.5) THEN
  DO 401 ALF=1,MAXALF
  TRUQNT(ALF) = 0
  DO 501 REPS=1, MAXREP
  DIFF(REPS) = RSBRDG - RSHTBR(ALF,REPS)
501 CONTINUE
  DO 601 REPS=1, MAXREP
  IF(ABS(DIFF(REPS)).LE.EPS) THEN
  TRUQNT(ALF) = REPS
  WRITE(1,(' ','',/'TRUE CONFIDENCE LIMIT IS:','
+      F8.4)')
+      (TRUQNT(ALF) / REAL(MAXREP)) * 100.
  GO TO 621
  ELSEIF(DIFF(REPS).LT.0.) THEN
  TRUQNT(ALF) = REPS
  GO TO 611
  ELSE

```

```

        END IF
601 CONTINUE
611 IF(TRUQNT(ALF).EQ.0.) THEN
        WRITE(1,4443) ALFA(ALF)
        WRITE(1,(' ' ' ',/'THE SMALLEST',
+         '' DIFFERENCE BETWEEN RSBRDG AND RSHTBR IS: '' ,
+         F10.5)') DIFF(MAXREP)
        ELSEIF(TRUQNT(ALF).EQ.1.) THEN
        WRITE(1,4442) ALFA(ALF)
        WRITE(1,(' ' ' ',/'ALL RSHTBR'',
+         '' ARE GREATER THAN RSBRDG''))
        ELSEIF(ABS(DIFF(TRUQNT(ALF))).LE.ABS(DIFF(TRUQNT(ALF) - 1)))
+         THEN
        WRITE(1,4444) ALFA(ALF),
+         (TRUQNT(ALF) / REAL(MAXREP)) * 100.
        WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF))
        WRITE(1,4446)
        ELSE
+         WRITE(1,4444) ALFA(ALF),
+         ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
        WRITE(1,4449) RSHTBR(ALF,TRUQNT(ALF)-1)
        WRITE(1,4447)
621 END IF
401 CONTINUE
        ELSE
        END IF

```

FIND THE TRUE CONFIDENCE LEVEL OF THE SYSTEM REL. ESTIMATE
 *** ***** RHISTR ***** ***

```

IF(FLAG.EQ.1) THEN
DO 4400 ALF=1,MAXALF
TRUQNT(ALF) = 0
DO 5500 REPS=1, MAXREP
        DIFF(REPS) = RS - RHTSTR(ALF,REPS)
5500 CONTINUE
DO 6600 REPS=1, MAXREP
        IF(ABS(DIFF(REPS)).LE.EPS) THEN
                TRUQNT(ALF) = REPS
                WRITE(1,(' ' ' ',/'TRUE CONFIDENCE LIMIT IS: '' ,
+                 F8.4)')
+                 (TRUQNT(ALF) / REAL(MAXREP)) * 100.
                GO TO 6620
        ELSEIF(DIFF(REPS).LT.0.) THEN
                TRUQNT(ALF) = REPS
                GO TO 6610
        ELSE
        END IF
6600 CONTINUE
6610 IF(TRUQNT(ALF).EQ.0.) THEN
        WRITE(1,4443) ALFA(ALF)
        WRITE(1,(' ' ' ',/'THE SMALLEST'',
+         '' DIFFERENCE BETWEEN RS AND RHTSTR IS: '' ,
+         F9.5)') DIFF(MAXREP)
        ELSEIF(TRUQNT(ALF).EQ.1.) THEN

```

```

        WRITE(1,4442) ALFA(ALF)
        WRITE(1,(' ' ' ',/' ALL RHTSTR' ',
+         ' ' ARE GREATER THAN RS' '))
        ELSEIF(ABS(DIFF(TRUQNT(ALF))). LE. ABS(DIFF(TRUQNT(ALF) - 1)))
+         THEN
        WRITE(1,4444) ALFA(ALF),
+         (TRUQNT(ALF) / REAL(MAXREP)) * 100.
        WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF))
        WRITE(1,4446)
        ELSE
+         WRITE(1,4444) ALFA(ALF),
+         ((TRUQNT(ALF)-1) / REAL(MAXREP)) * 100.
        WRITE(1,4448) RHTSTR(ALF,TRUQNT(ALF)-1)
        WRITE(1,4447)
6620     END IF
4400 CONTINUE
        ELSE
        END IF

```

PRINT THE ARRAYS PERTINENT TO THE OUPUT OF EACH REPLICATION

```

        IF(PRNT.EQ.1) THEN
        I = 1
185  WRITE(1,REPSHD) ALFA(SELCTA), ALFA(SELCTA),
+ALFA(SELCTB),ALFA(SELCTB),ALFA(SELCTA),ALFA(SELCTA),ALFA(SELCTB),
+ALFA(SELCTB)
175  IF(I.GE.(MAXREP + 1)) THEN
        GOTO 180
        ELSE
+         IF( (I.EQ.71).OR.(I.EQ.211).OR.(I.EQ.351).OR.(I.EQ.491).OR.
+         (I.EQ.631).OR.(I.EQ.771).OR.(I.EQ.911).OR.(I.EQ.1051) ) THEN
                I = I + 70
                WRITE(1,(' '+' '))
                GOTO 185
        ELSE
+         WRITE(1,3336) I, INT(DEGFR(I)), CHISQ(1,I), QHTUPR(1,I),
+         CHISQ(2,I), QHTUPR(2,I)
        END IF
        IF(I + 70. LE. MAXREP) THEN
+         WRITE(1,3337) I+70,INT(DEGFR(I+70)),CHISQ(1,I+70),
+         QHTUPR(1,I+70),CHISQ(2,I+70),QHTUPR(2,I+70)
        ELSE
        END IF
        I = I + 1
        GOTO 175
180  END IF
        ELSE
        ENDIF
9999 WRITE(1,(' 'THE TOTAL NO OF REPS WAS: ',I8)') TOTREP
        WRITE(1,(' 'THE TOTAL NO OF EFFECTIVE REPS WAS: ',I8)') LOOP
        WRITE(1,(' 'THE TOTAL NO OF NO FAILURE RUNS WAS: ',I8)') ZFAILS
        WRITE(1,(' 'AVERAGE NO. OF COMPONENTS PER REPLICATION WITH ' ',
+ 'NO FAILURES: ',F5.2)') ZFPREP / MAXREP
        WRITE(1,(' 'THE TOTAL NO OF RUNS WITH FAILURES WAS: ',I8)') FAILS
0008 FORMAT (/ 3X,'C 1',5X,'C 2',

```

```

+5X,'C 3',5X,'C 4',5X,'C 5',5X,'C 6',5X,'C 7',5X,
+C 8',5X,'C 9',5X,'C 10',X,'C 11',4X,
+C 12',4X,'C 13',4X,'C 14',4X,'C 15')
0009 FORMAT(/1X,'REP NO',2X,'...',1X,'CHISQR(',F4.3,')',1X,
+'QHTUPR(',F4.3,')',1X,'CHISQR(',F4.3,')',1X,'QHTUPR(',F4.3,')',
+2X,'REP NO',2X,'DF',1X,'CHISQR(',F4.3,')',1X,
+'QHTUPR(',F4.3,')',1X,'CHISQR(',F4.3,')',1X,'QHTUPR(',F4.3,')')
0001 FORMAT (///'UNIFORM RANDOM DEVIATES ARE: ')
0002 FORMAT (///'BERNOULLI TRIALS ARE: ')
0003 FORMAT (///'TOTAL NO. OF FAILURES FOR EACH COMPONENT: ')
0004 FORMAT (///'ESTIMATED UNRELIABILITY FOR EACH COMPONENT: ')
0005 FORMAT (///'TOTAL NUMBER OF MISSION TESTS: ')
0006 FORMAT (///'ESTIMATED WEIGHTS FOR EACH COMPONENT: ')
0007 FORMAT (///'Q I FOR EACH COMPONENT: ')
1111 FORMAT (15F8.5)

2222 FORMAT (/1X,15(I4,4X))
3333 FORMAT (/1X,15(I4,4X))
3334 FORMAT (/15F8.5)
3335 FORMAT (/1X,15(I4,4X))
3336 FORMAT (T3,I4,T9,I3,T13,F11.5,T27,F8.5,T39,F11.5,T53,F8.5)
3337 FORMAT ('+',T67,I4,T73,I3,T77,F11.5,T91,F8.5,T103,F11.5,T117,F8.5)
4442 FORMAT ('',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+'LIMIT IS: ',T50,'00.000 ')
4443 FORMAT ('',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+'LIMIT IS: ',T50,'100.0000 ')
4444 FORMAT ('',///'THE RESULTING (1 - ',F4.3,') CONFIDENCE ',
+'LIMIT IS: ',T50,F8.4)
4445 FORMAT ('',/'THE RSHAT VALUE CLOSEST TO RS IS: ',T51,F8.5)
4446 FORMAT ('',/'(FIRST NEGATIVE DIFFERENCE)')
4447 FORMAT ('',/'(ELEMENT PRECEDING FIRST NEGATIVE DIFFERENCE)')
4448 FORMAT ('',/'THE RHTSTR VALUE CLOSEST TO RS IS: ',T51,F8.5)
4449 FORMAT ('',/'THE RSHTBR VALUE CLOSEST TO RSBRDG IS: ',T51,F8.5)
5555 FORMAT ('',///'THE ',I4,'(1-',F4.3,') QUANTILE IS: ',T49,F8.3)
5556 FORMAT ('',/'THE VALUE OF RSHAT FOR THAT QUANTILE IS: ',T51,F8.5)
5557 FORMAT ('',/'THE DIFFERENCE(RS - RSHAT) IS: ',T51,F8.5)
5656 FORMAT ('',/'THE VALUE OF RHTSTR FOR THAT QUANTILE IS: ',T51,F8.5)
5666 FORMAT ('',/'THE VALUE OF RSHTBR FOR THAT QUANTILE IS: ',T51,F8.5)
5657 FORMAT ('',/'THE DIFFERENCE(RS - RHTSTR) IS: ',T51,F8.5)
5667 FORMAT ('',/'THE DIFFERENCE(RS - RSHTBR) IS: ',T51,F8.5)
5755 FORMAT ('',///'SINCE THE NO. OF MISSION TESTS IS THE SAME FOR',
+' ALL COMPONENTS THE CLOSED FORM SERIES SYSTEM RELIABILITY ',
+' 'RHTSTR' IS COMPUTED')
6666 FORMAT ('+', '*****',
+'***** RUN INPUT SETTINGS *****',
+'*****')
6667 FORMAT ('',///'NUMBER OF REPLICATIONS: ',T50,I4)
6668 FORMAT ('',///'NUMBER OF COMPONENTS: ',T50,I4)
6669 FORMAT ('',///'SYSTEM RELIABILITY FUNCTION: ',T50,'SERIES')
6699 FORMAT ('',///'SYSTEM RELIABILITY FUNCTION: ',T50,'BRIDGE')
6670 FORMAT ('',///'MASTER UNRELIABILITY USED: ',T50,F8.5)
6671 FORMAT ('',///'INPUT WEIGHTS(A SUB I'S): ')
6674 FORMAT ('+',///'*****',
+'**R U N R E S U L T*****',
+'*****')
6675 FORMAT ('+',///'*****')

```

```

+ ' ESTIMATE ERRORS *****' ,
+ '*****')
6676 FORMAT ('+', '/', '*****' ,
+ ' TRUE CONFIDENCE LIMITS *****' ,
+ '*****')
END
```

APPENDIX I. SUBROUTINES

C	IMSL ROUTINE NAME	- USMNMX	USMN0010
C			USMN0020
C	-----		USMN0030
C			USMN0040
C	COMPUTER	- IBM/SINGLE	USMN0050
C			USMN0060
C	LATEST REVISION	- JANUARY 1, 1978	USMN0070
C			USMN0080
C	PURPOSE	- DETERMINATION OF THE MINIMUM AND MAXIMUM	USMN0090
C		VALUES OF A VECTOR	USMN0100
C			USMN0110
C	USAGE	- CALL USMNMX (X,N,INC,XMIN,XMAX)	USMN0120
C			USMN0130
C	ARGUMENTS	X - INPUT VECTOR OF LENGTH N FROM WHICH MINIMUM,	USMN0140
C		MAXIMUM VALUES ARE TO BE TAKEN.	USMN0150
C		N - LENGTH OF THE INPUT VECTOR X. (INPUT)	USMN0160
C		INC - DISPLACEMENT BETWEEN CONSECUTIVE VALUES OF X	USMN0170
C		TO BE CONSIDERED.	USMN0180
C		XMIN - OUTPUT SCALAR CONTAINING MINIMUM VALUE OF X.	USMN0190
C		XMAX - OUTPUT SCALAR CONTAINING MAXIMUM VALUE OF X.	USMN0200
C			USMN0210
C	PRECISION/HARDWARE	- SINGLE AND DOUBLE/H32.	USMN0220
C		- SINGLE/H36,H48,H60	USMN0230
C			USMN0240
C	REQD. IMSL ROUTINES	- NONE REQUIRED	USMN0250
C			USMN0260
C	NOTATION	- INFORMATION ON SPECIAL NOTATION AND	USMN0270
C		CONVENTIONS IS AVAILABLE IN THE MANUAL	USMN0280
C		INTRODUCTION OR THROUGH IMSL ROUTINE UHELP	USMN0290
C			USMN0300
C	COPYRIGHT	- 1978 BY IMSL, INC. ALL RIGHTS RESERVED.	USMN0310
C			USMN0320
C	WARRANTY	- IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN	USMN0330
C		APPLIED TO THIS CODE. NO OTHER WARRANTY,	USMN0340
C		EXPRESSED OR IMPLIED, IS APPLICABLE.	USMN0350
C			USMN0360
C	-----		USMN0370
C			USMN0380
C	SUBROUTINE USMNMX (X,N,INC,XMIN,XMAX)		USMN0390
C			USMN0400
C	DIMENSION	X(N)	USMN0410
C			USMN0420
		FIRST EXECUTABLE STATEMENT	USMN0430
	XMIN = X(1)		USMN0440
	XMAX = X(1)		USMN0450
	DO 10 I=1,N,INC		USMN0460
	IF (X(I) .GE. XMIN) GO TO 5		USMN0470
	XMIN = X(I)		USMN0480
	GO TO 10		USMN0490
5	IF (X(I) .GT. XMAX) XMAX = X(I)		USMN0500
10	CONTINUE		USMN0510
	RETURN		USMN0510

END	USMN0520
.....	SORT0010
A. IDENTIFICATION:	SORT0020
TITLE: NUMERICAL SORT	SORT0030
ID: M1-NPG-SHRSORT (F-IV)	SORT0040
PROGRAMMER: R. BRUNELL	SORT0050
DATE: MARCH 1968	SORT0060
MODIFIED: DEC. 1973 BY L. NOLAN	SORT0070
	SORT0080
	SORT0090
B. PURPOSE:	SORT0100
TO SORT, IN ASCENDING ORDER, AN ARRAY OF SINGLE PRECISION REAL	SORT0110
NUMBERS BY THE METHOD OF SHELL, AND TO PRODUCE AN ARRAY OF INDEXES	SORT0120
SO USER CAN RE-ORDER OTHER CORRESPONDING INFORMATION ACCORDING TO	SORT0130
ASCENDING VALUES OF "A".	SORT0140
	SORT0150
C. USAGE:	SORT0160
1. CALLING STATEMENT:	SORT0170
CALL SHRSORT(A,KEY,N)	SORT0180
2. ARGUMENTS:	SORT0190
A - ARRAY OF NUMBERS TO BE SORTED. THIS ARRAY IS SORTED	SORT0200
(RE-ORDERED) BY "SHRSORT".	SORT0210
KEY - ARRAY, DIMENSIONED AT LEAST N IN CALLING PROGRAM, TO BE	SORT0220
FILLED BY USER WITH INTEGERS FROM 1 TO N. AFTER EXIT	SORT0230
FROM SHRSORT, KEY(1) WILL CONTAIN THE ORIGINAL INDEX OF	SORT0240
THE SMALLEST ELEMENT OF "A"; KEY(2) WILL CONTAIN THE	SORT0250
ORIGINAL INDEX OF THE NEXT-TO-SMALLEST ELEMENT OF "A";	SORT0260
ETC. KEY(N) WILL CONTAIN THE ORIGINAL INDEX OF THE	SORT0270
LARGEST ELEMENT OF "A".	SORT0280
N - NUMBER OF MEMBERS IN ARRAYS "A" AND "KEY".	SORT0290
	SORT0300
D. REFERENCES:	SORT0310
1. "ALGORITHM 201, SHELLSORT", BOOTHROYD, J., "COMMUNICATIONS OF	SORT0320
ACM", VOL 6, NO 8, AUGUST 1963, P. 445.	SORT0330
2. "CERTIFICATION OF ALGORITHM 201", BATTY, M. A., "COMMUNICATIONS	SORT0340
OF ACM", VOL 7, NO 6, JUNE 1964, P. 349.	SORT0350
	SORT0360

SUBROUTINE SHRSORT(A,KEY,N)	SORT0370
DIMENSION A(N),KEY(N)	SORT0380
M1=1	SORT0390
6 M1=M1*2	SORT0400
IF (M1 .LE. N) GO TO 6	SORT0410
M1=M1/2-1	SORT0420
MM=MAX0(M1/2,1)	SORT0430
GO TO 21	SORT0440
20 MM=MM/2	SORT0450
IF (MM .LE. 0) GO TO 100	SORT0460
21 K=N-MM	SORT0470
22 DO 1 J=1,K	SORT0480
II=J	SORT0490
11 IM=II+MM	SORT0500
IF (A(IM) .GE. A(II)) GO TO 1	SORT0510
TEMP=A(II)	SORT0520
IT=KEY(II)	SORT0530


```

A(II)=A(IM)
KEY(II)=KEY(IM)
A(IM)=TEMP
KEY(IM)=IT
II=II-MM
IF (II .GT. 0) GO TO 11
1 CONTINUE
GO TO 20
100 RETURN
END

```

```

SORT0540
SORT0550
SORT0560
SORT0570
SORT0580
SORT0590
SORT0600
SORT0610
SORT0620
SORT0630

```

```

SUBROUTINE RHTSRS(QHTUP,AAHTI,N,RRSHAT)
***// THIS ROUTINE CALCULATES THE VALUE OF THE SYSTEM RELIABILITY OF A
***// SERIES SYSTEM OF 'N' NO. OF COMPONENTS WHICH HAVE UNRELIABILITY
***// 'QHTUP'. THE FINAL SYSTEM RELIABILITY VALUE PASSED IS 'RRSHAT'
REAL*4 QHTUP, RRSHAT, AAHTI(N)
INTEGER I, N
RRSHAT = 1.
DO 100 I=1, N
    RRSHAT = RRSHAT * (1 -(AAHTI(I)* QHTUP))
100 CONTINUE
END

```

```

SUBROUTINE RRSRS(QIS,N,RRS)
***// THIS ROUTINE CALCULATES THE VALUE OF THE SYSTEM RELIABILITY OF A
***// SERIES SYSTEM OF 'COMP' NO. OF COMPONENTS WHICH HAVE UNRELIABILITY
***// 'QIS'. THE FINAL SYSTEM RELIABILITY VALUE PASSED IS 'RRS'
REAL*4 QIS(N), RRS
INTEGER I, N
RRS = 1.
DO 100 I=1, N
    RRS = RRS * (1 - QIS(I))
100 CONTINUE
END

```

```

***// THIS SUBROUTINE CALCULATES THE ESTIMATED RELIABILITY OF A
***// 5-COMPONENT BRIDGE STRUCTURE.(ONLY CARRIED OUT TO THE Q-CUBED TERM
SUBROUTINE RHTBRG(QHTUP,AHT,N,RRBRDG)
REAL*4 QHTUP, RRBRDG, AHT(N)
INTEGER N
* PRINT *, 'THE VALUES FOR AHT PASSED ARE: ',AHT
RRBRDG=1-(((QHTUP**2)*(AHT(1)*AHT(2)+AHT(4)*AHT(5)))-
C((QHTUP**3)*(AHT(1)*AHT(3)*AHT(5)+AHT(2)*AHT(3)*AHT(4)))+
C((QHTUP**4)*(AHT(1)*AHT(2)*AHT(3)*AHT(4)+AHT(1)*AHT(2)*AHT(3)*
CAHT(5)+AHT(1)*AHT(2)*AHT(4)*AHT(5)+AHT(1)*AHT(3)*AHT(4)*AHT(5)+
CAHT(2)*AHT(3)*AHT(4)*AHT(5)))-
C2*((QHTUP**5)*(AHT(1)*AHT(2)*AHT(3)*AHT(4)*AHT(5)))
* PRINT *, 'COMPUTED RRBRDG IS: ', RRBRDG
END

```

```

SUBROUTINE CPARE(FI,K,BFLAG)

```

```

INTEGER BFLAG,FI(K)
BFLAG=0
IF ((FI(1) .EQ. 0) .AND. (FI(3) .EQ. 0) .AND.
+ (FI(5) .EQ. 0)) THEN
    BFLAG=0
ELSE IF ((FI(1) .EQ. 0) .AND. (FI(4) .EQ. 0)) THEN
    BFLAG = 0
ELSE IF ((FI(2) .EQ. 0) .AND. (FI(5) .EQ. 0)) THEN
    BFLAG = 0
ELSE IF ((FI(2) .EQ. 0) .AND. (FI(3) .EQ. 0) .AND.
+ (FI(4) .EQ. 0)) THEN
    BFLAG = 0
ELSE IF (((FI(1) .EQ. 1) .AND. (FI(2) .EQ. 1)) .AND.
+ ((FI(2) .EQ. 0) .AND. (FI(4) .EQ. 0) .AND.
+ (FI(5) .EQ. 0))) THEN
    BFLAG = 1
ELSE IF (((FI(4) .EQ. 1) .AND. (FI(5) .EQ. 1)) .AND.
+ ((FI(1) .EQ. 0) .AND. (FI(2) .EQ. 0) .AND.
+ (FI(3) .EQ. 0))) THEN
    BFLAG = 1
ELSE IF ((FI(2) .EQ. 0) .AND. (FI(4) .EQ. 0)) .AND.
+ ((FI(1) .EQ. 1) .AND. (FI(3) .EQ. 1) .AND.
+ (FI(5) .EQ. 1)) THEN
    BFLAG = 1
ELSE IF (((FI(1) .EQ. 0) .AND. (FI(5) .EQ. 0)) .AND.
+ ((FI(2) .EQ. 1) .AND. (FI(3) .EQ. 1) .AND.
+ (FI(4) .EQ. 1))) THEN
    BFLAG = 1
ELSE
    BFLAG = 2
ENDIF

RETURN
END

```

```

***// THIS SUBROUTINE CALCULATES THE "TRUE" RELIABILITY OF A 5-COMPONENT
***// BRIDGE STRUCTURE
SUBROUTINE RBRIDG(QI,N,RRSS)
REAL*4 QI(N), RRSS
INTEGER N
IF(N.NE.5) THEN
    WRITE(1,('WARNING: BRIDGE STRUCTURE ONLY USES ',
+ 'THE FIRST 5 COMPONENTS'))
ELSE
    END IF
    RRSS=(1-QI(1))*(1-QI(4))+(1-QI(2))*(1-QI(5))+(1-QI(1))*(1-QI(3))*
C(1-QI(5))+(1-QI(2))*(1-QI(3))*(1-QI(4))-(1-QI(1))*(1-QI(2))*
C(1-QI(3))*(1-QI(4))-(1-QI(1))*(1-QI(2))*(1-QI(3))*(1-QI(5))-
C(1-QI(1))*(1-QI(2))*(1-QI(4))*(1-QI(5))-(1-QI(1))*(1-QI(3))*
C(1-QI(4))*(1-QI(5))-(1-QI(2))*(1-QI(3))*(1-QI(4))*(1-QI(5))+
C2*(1-QI(1))*(1-QI(2))*(1-QI(3))*(1-QI(4))*(1-QI(5))
END

```

LIST OF REFERENCES

1. Mann, Nancy R. and Grubbs, Frank E., *Approximate Optimum Confidence Bounds for System Reliability Based on Component Test Data*, Technometrics 16, pp. 335-47, 1974.
2. Myhre, J., Rosenfeld, A., and Saunders, S., *Determining Confidence Bounds for Highly Reliable Coherent Systems Based on a Paucity of Component Failures*, Naval Research Logistics Quarterly, 25, June 1978.
3. Easterling, Robert G., *Approximate Confidence Limits for System Reliability*, Journal of the American Statistical Association 67, March 1972.
4. Winterbottom, Alan, *Lower Confidence Limits for Series System Reliability from Binomial Subsystem Data*, Journal of the American Statistical Association 69, September 1974.
5. Lee, Hyeon-Soo, *Approximate Interval Estimation Methods for the Reliability of Systems Using Component Data With Exponential and Weibull Distributions*, M.S. Thesis, Naval Postgraduate School, Monterey, California, September 1989.
6. Lomnicki, Z.A., *Two Terminal Series-Parallel Networks*, Advances in Applied Probability 4, pp.109-150,1973.
7. *Confidence Limits For Attributes Data*, LMSC-803324, Lockheed Missiles and Space Company.
8. Bellini, E.F., *Approximate Interval Estimation Methods for the Reliability of Systems Using Discrete Component Data*, M.S. Thesis, Naval Postgraduate School, Monterey, California, September 1990.

BIBLIOGRAPHY

Bain, L.J. and Engelhardt, M., *Introduction to Probability and Mathematical Statistics*, Duxbury Press, Boston, 1987.

Mann, N.R., Schafer, R.E., and Signpurwalla, N.D., *Methods for Statistical Analysis of Reliability and Life Data*, John Wiley and Sons, Inc., New York, pp. 487-524, 1975.

Lloyd, D. and Lipow, M., *Reliability Management, Methods, and Mathematics*, 2nd Edition, American Society for Quality Control, 1984.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code-52 Naval Postgraduate School Monterey, CA 93943-5002	2
3. Professor W. Max Woods Naval Postgraduate School, Code OR-Wo Monterey, CA 93943-5000	4
4. Professor Lyn R. Whitaker Naval Postgraduate School, Code OR-Wh Monterey, CA 93943-5000	1
5. Chief of Naval Operations (OP-81) Department of the Navy Washington, DC 20350	1
6. Base Library FL 4887 Luke Air Force Base, AZ 85309	2
7. LT Valerie A. Covington, USN P.O. Box 62 Litchfield Park, AZ 85340	3