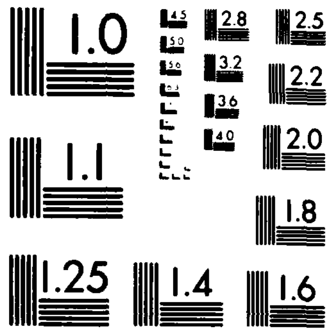
AD-A150 005 STATISTICS FOR THE MAXIMUM OF SEVERAL POSITIVE RANDOM VARIABLES WITH APPL..(U) ARMY ARMAMENT MUNITIONS AND CHEMICAL COMMAND ROCK ISLAND IL R.. G J SCHLENKER 1/1

UNCLASSIFIED DEC 84 AMSMC/RD/MR-5 F/G 12/1 NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AMSMC/RD/MR-5

12

STATISTICS FOR THE MAXIMUM OF SEVERAL POSITIVE RANDOM VARIABLES WITH APPLICATION TO NETWORKING

GEORGE J. SCHLENKER

DECEMBER 1984

Distribution Statement

Approved for public release; distribution unlimited.

US ARMY ARMAMENT, MUNITIONS AND CHEMICAL COMMAND
READINESS DIRECTORATE
ROCK ISLAND, ILLINOIS 61299-6000

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on input available at the time of preparation. Because the results may be subject to change, this document should not be construed to represent the official position of the US Army Armament, Munitions and Chemical Command unless so stated.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AMSMC/RD/MR-5	2. GOVT ACCESSION NO. AD-A150 005	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Statistics for the Maximum of Several Positive Random Variables with Application to Networking		5. TYPE OF REPORT & PERIOD COVERED Report - Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) George J. Schlenker		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Armament, Munitions and Chemical Command Readiness Directorate Rock Island, IL 61299-6000		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December 1984
		13. NUMBER OF PAGES 86
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Additions to/from the DISTRIBUTION LIST are invited and should be forwarded to the following address: Commander, US Army Armament, Munitions and Chemical Command, ATTN: AMSMC-RDA-S, Rock Island, IL 61299-6000
AUTOVON 793-5041/6370

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Operations Research	PERT/VERT
Activity Networks	Statistics
Project Management	Numerical Analysis
Industrial Operations	Distribution Theory

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report addresses a common problem in activity networks, popularly known as PERT networks. This is the problem of calculating statistics associated with the maximum of several parallel, random activity times. Generally, one desires the mean, standard deviation, and quantiles of the distribution of the maximum of a set of positive, continuous, and independent random variables. An accurate recursive numerical algorithm for calculating these statistics is presented here. This method is somewhat more efficient than more (cont'd)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 (cont'd):

straightforward numerical techniques.

The method developed and used here is not restricted by having all of the random variables belong to the same distribution or even by having the same functional form. Pertinent, general formulas are derived as well as some closed-form results for a special case. Numerical examples are presented to assess the method's computational error and to illustrate certain quantitative generalizations.

Altho not limited to activity network applications, these results and the enclosed computer programs can be used to assess the estimation errors associated with deterministic networking methods such as PERT and CPM.

AMSMC/RD/MR-5

STATISTICS FOR THE MAXIMUM OF SEVERAL POSITIVE RANDOM VARIABLES WITH APPLICATION TO NETWORKING

GEORGE J. SCHLENKER

December 1984

ACKNOWLEDGMENT

The author gratefully acknowledges the typing of this report by Mrs. Linda J. Lensly. Her assistance on this and other technical reports is certainly appreciated.

EXECUTIVE SUMMARY

This report is addressed to analysts with some background in probability, statistics, and stochastic networks. The literature on activity networks has identified shortcomings of deterministic methods such as the Program Evaluation and Review Technique (PERT). Much of the estimation error of PERT is due to the failure to adequately treat subnetworks of parallel activities. Statistics such as mean, standard deviation, and quantiles of completion time of these subnetworks can be accurately calculated by the methods of this report. Therefore, the errors in these statistics produced by deterministic methods can be evaluated for specific examples. However, in the author's opinion, in most instances deterministic networking methods should be abandoned in favor of accurate and comprehensive stochastic techniques such as the Venture Evaluation and Review Technique (VERT).

The numerical procedures developed here have considerable use apart from application to activity networks. The problem of finding descriptive statistics for the maximum of a set of positive continuous random variables is found in the areas of analysis of engineering tolerances and of reliability and maintainability.

Motivated by networking problems, various parametric analyses are performed. Goals are to determine the sensitivity of the subject statistics to network parameters and to draw pertinent inferences for activity networks. Some parameters of interest are the number of random variables (RV's) in the set, the relative size of the mean value of each of the RV's in the set, and the functional form of and shape parameters for the probability distributions of these RV's. Most of the examples use probability distributions having the domain zero to infinity. To meet the objection that the domain is always finite, an analysis is made of the effect of truncating the above distributions. A brief analysis is also done on the effect of network logic.

CONTENTS

AMSMC-RDA-S December 1984

MEMORANDUM REPORT

SUBJECT: Statistics for the Maximum of Several Positive Random Variables with Application to Networking

1. Reference:

References are designated by bracketed numbers and are included in the footnotes. The references are also listed.

2. Background

Deterministic* networking methods for estimating the time to complete a multi-activity project have existed for more than two decades. Methods of this sort such as PERT and CPM, while useful in some respects, can produce large errors in estimates of the mean value and upper quantiles of the probability distribution of project completion time. The error is due principally to the insistence that a deterministic critical path pass thru that activity, in a set of parallel activities, which has the largest mean completion time**. The errors of deterministic networking methods were identified very early. For example, see Grubbs (1962)[1].

---

* As opposed to Monte-Carlo methods.

** The deterministic approach replaces the random activity times by their mean values to find a critical path thru the network. Using the PERT assumption, the variance of the project completion time is just the sum of the activity variances along the critical path. To estimate quantiles of the distribution of project completion time, PERT makes the additional assumption that the distribution of this random variable is Normal.

[1] Grubbs, F. E. "Attempts to Validate Certain PERT Statistics or 'Picking on PERT'," Opns. Res., Vol. 10, pp. 912 - 915, 1962.

An analytic approximation was devised by Clark (1961)[2] to calculate the mean and standard deviation of project completion time. The method is based on Normally-distributed parallel activity times. These times are permitted to be correlated, if activities share a common node. Subsequent critiques of PERT, such as MacCrimmon and Ryavec (1964)[3] and Greer (1983)[4], attempted to quantify the approximate magnitude and sign of the estimation errors of completion time statistics. By necessity, these papers use quite simple examples, from which their inferences are drawn. Unfortunately, there is a great diversity in network structure, and no typical* network can be identified.

3. In at least one class of network problems, there are multiple -- typically, 2 to 20 -- independent, parallel activities throughout very large networks. This type of network [5] has been used by Department of Army organizations to plan for the reactivation of inactive ammunition production facilities. Motivation for the work reported here followed discussion with one of the authors of [5] (Moeller) about the estimation errors associated with deterministic

---

[2] Clark, C.E. "The Greatest of a Finite Set of Random Variables," Opns. Res., Vol. 9, No. 2, pp. 145 - 162, March - April 1961.

[3] MacCrimmon, K.R. and Ryavec, C.A. "An Analytical Study of the PERT Assumptions," Opns. Res., Vol. 12, No. 1, pp. 16 - 37, January - February 1964.

[4] Greer, W.R. Jr. "Why Doesn't PERT Work?," Resource Mgmt. Journal, pp. 27 - 31, Summer 1983.

* There are, of course, classes of problems which yield similar networks. However exceptions are manifest to generalizations such as "activity networks have few independent parallel activities."

[5] Matheiss, T.H., Moeller, G. and Kilar, J. "Improving Industrial Readiness with Venture Evaluation and Review Technique (VERT)," Interfaces, Vol. 12, No. 1, pp. 21 - 26, February 1982.

networking methods applied to this type of problem. It appears that the defects of deterministic networking methods, tho adequately reported*, have not been appreciated by all. The beginning of the present effort was an attempt to quantify the error of PERT completion time for a subnetwork of multiple parallel activities.

4. An approach to all stochastic networks which avoids the restrictive assumptions of PERT is called VERT, for Venture Evaluation and Review Technique. A recent expository paper by Moeller and Digman (1981)[6] describes the modeling technique and illustrates this with an example of planning in the electric power industry. The textbook on VERT [7] provides greater detail.

5. Objectives and Scope

The class of problem posed by parallel subnetworks of activities of random duration is the following. Given a set of k positive, continuous random variables, each having a possibly unique distribution, what are the values of statistics associated with the largest member of this set? The statistics of interest are the mean, standard deviation, and quantiles of the distribution of the largest value.** Objectives of this report are: (a) to develop a general procedure for obtaining this mean and standard deviation and (b) to derive some analytic results suitable for assessing the computational error of

---

* In fact some authors ([3] and [4]) have advocated patching up PERT via analytic corrections.

[6] Moeller, G.L. and Digman, L.A. "Operations Planning with VERT," Opns. Res., Vol 29, No. 4, pp. 676 - 697, July - August 1981.

[7] Lee, S.M., Digman, L.A., and Moeller, G.L. Network Analysis for Management Decisions, Kluwer-Nijhoff Pub., Hingham, MA, c. 1981.

** A method for obtaining these statistics for this problem has greater applicability than just to activity networks. Other applications lie in the areas of tolerance stackup (where parallel elements exist) and in reliability and maintainability analysis.

the numerical method. Additionally, considering the original problem context, certain numerical generalizations are desired which can be applied to activity networks.

6. Method

Prior to discussing methods, some problem nomenclature will be useful. The probability density function (p.d.f.) and its integral the cumulative distribution function (c.d.f.) for each of the random variables (RV's) in the set $\{x_i, 1 \leq i \leq k\}$ are denoted, respectively, by $f_i(x)$ and $F_i(x)$. These are the primary inputs or problem ingredients. It is assumed that the mean and variance of x_i -- $E[x_i]$ and $V[x_i]$ -- are readily available. The random variable of interest is denoted by z_k, where

$$z_k = \max_i(x_1, x_2, \ldots, x_i, \ldots, x_k),\ 0 \leq x_i < \infty \quad .$$

The p.d.f and c.d.f. of z_k are denoted by $g_k(z)$ and $G_k(z)$. Given that an expression for $g_k(z)$ can be derived, one can obtain an analytic expression for the jth origin moment $a_j(k)$ by

$$a_j(k) = \int_0^\infty z^j g_k(z)dz,\ 0 \leq j \quad .$$

From this result the mean and variance of z_k are, immediately,

$$E[z_k] = a_1(k)$$

$$V[z_k] = a_2(k) - a_1^2(k) \quad ,$$

with standard deviation of z_k equal to $\sqrt{V[z_k]}$. An expression for $g_k(z)$ is easily obtained for certain types of $F_i(x)$ distributions. Analytic results for examples of this sort are obtained by the above method. These results are derived in Annex A. Also included in this annex are probability arguments leading to general procedures for calculating $E[z_k]$ and $V[z_k]$, which can be easily implemented in a computer program. From a pragmatic point of view, it is immaterial whether numerical results are obtained by evaluating a closed-form expression or by following another numerical procedure, providing the latter is not computationally too expensive and yields a sufficiently small numerical error. A computer program was developed to implement the procedure derived in Annex A. The program listing is displayed in Annex B. The method presented here has the important practical advantage of being free of restrictions on the form of $F_i(x)$. For simplicity, numerical examples of this method were calculated for some familiar two-parameter distributions -- gamma and Weibull distributions. A parametric analysis is conducted which systematically examines the effects of the number (k) of RV's in the set and of the parameters of these probability distributions.

7. As indicated, the primary emphasis in this report is on the <u>maximum</u> of a set of k random variables. Ordinarily, the logic of an activity network corresponds to this problem. Passage to other activities downstream of a set of parallel activities is conditional upon completing all of the parallel activities. Occasionally, network logic permits passage when only k of n activities are complete. <u>If</u> the probability distribution of all parallel activities is the same, the latter problem reverts to a problem in order statistics. Guenther (1977)[9] presents an easily applied numerical technique for evaluating the c.d.f. of the kth ordered (in algebraic magnitude) statistic in a set of n. The method of [9] is exploited here for this special case. Details are presented at the end of Annex A.

8. <u>Numerical Results</u>

Following the derivation of equations in Annex A some numerical examples are considered. The random variables (x_i) for all examples are scaled so as to facilitate comparison between examples. The parameters in $F_i(x)$ are selected so as to make the largest over i of $E[x_i]$, $1 \leq i \leq k$, equal to unity. In fact, one may as well order the x_i in order of decreasing mean value. This scaling does not reduce the generality of our approach. (One can always convert between units.) It does, however, permit one to compare the value of

---

[9] Guenther, W.C. "An Easy Method for Obtaining Percentage Points of Order Statistics," <u>Technometrics</u>, Vol. 19, No. 3, pp. 319 - 321, August 1977.

$E[z_k]$ with the PERT-estimated value, which is always unity. Altho the method for calculating the statistics of z_k does not require the location parameter of $F_i(x)$ to be zero, all numerical results presented here assume that x_i is bounded from below by zero. It is recognized that this assumption may be unrealistic for certain applications.

9. The first examples treat the case in which $F_i(x)$ is exponential with rate parameter λ_i. An analytic solution exists for this case. This permits calculation of the computational errors in $E[z_k]$ and $V[z_k]$ associated with the numerical method. Results for the special case in which λ_i is unity for all i are shown in Table E.1. The c.d.f.'s of z_k, with k as a parameter, are shown in Figure E.1 for this case. Plots of $G_k(z)$ for all the examples have been made on Normal probability paper. Advantages to plotting in this manner are: (a) departures from a straight line (Normality) are evident and (b) the values of the c.d.f. in each tail can be accurately plotted and read. Results for other examples involving exponential $F_i(x)$ are shown in Tables E.2 and E.3. Figure E.2 illustrates the behavior of $G_k(z)$ as k increases, for the case in which $\lambda_1 = 1$ and $\lambda_i = 1.2\lambda_{i-1}$, $i > 1$. In this case each RV added to the set has a mean value that is 1/1.2 of its predecessor. Convergence of quantiles in the upper tail of $G_k(z)$ is evident. Additional observations concerning this and other examples are found in Annex A.

10. Accuracy of the numerical method is displayed for two examples in Tables E.4, E.5, and E.6. Accuracy is a function of the step size used in numerical integration and of the method of quadrature used. Two quadrature schemes are employed for comparison: rectangular and Simpson's rule [11]. The latter is preferred on the basis of computational efficiency, altho both schemes are easily implemented and yield satisfactory accuracy. Methods for obtaining $E[z_k]$ and $V[z_k]$ based on numerical evaluation of the integral

---

[11] Bennett, A.A., Milne, W.E. and Bateman, H. Numerical Integration of Differential Equations, Dover Pub., New York, NY, c. 1956.

$$\int_0^\infty z^j g_k(z)dz,\ j = 1,\ 2 \quad ,$$

are not recommended due to their relatively poor accuracy vis à vis the method of Annex A.

11. Some parametric analyses were performed using the number (k) of RV's in the set as a parameter and with $F_i(x)$ having two functional forms -- gamma and Weibull. The effect of k on $G_k(z)$ for a Weibull distribution with shape parameter 2 is shown in Figure E.3. The shape parameter (β) of these distributions influences both the degree of dispersion and the skewness. With the constraint that $E[x_i] = 1,\ 1 \leq i \leq k$, the shape parameter of the $F_i(x)$ distribution was changed systematically to determine the effect on $E[z_k]$. Results for the gamma distribution are shown in Table E.7. Comparable results for $F_i(x)$ Weibull are shown in Table E.8. The effects upon the coefficients of variation and of skewness of x_i due to changes in shape (β) are different in the gamma and Weibull distributions. These differential effects are illustrated in Table E.9. Even when the two values of β are chosen to give the same coefficient of variation of x_i, one should expect different values of $E[z_k]$ in the gamma and Weibull cases. In fact, such differences are observed. Stated differently, a difference exists between $E[z_k]$ when $F_i(x)$ is gamma versus $E[z_k]$ when $F_i(x)$ is Weibull with the same coefficient of variation. However, this difference is quite small* whenever the skewness of $F_i(x) \gtrsim 1$ and $k \lesssim 6$. Under these conditions the form of the c.d.f. of x_i is not important when calculating $E[z_k]$.

12. Another numerical study, applicable to activity networks, deals with

---

* A difference in these cases is about ±0.02 or less for $E[z_k] \cong 1.7$ and SD $z_k \cong 0.4$. To detect a difference of this magnitude via stochastic simulation would require a Monte-Carlo sample greater than 500 for the standard error of the estimate to be less than about 0.02.

the upper limit on the range of x_i. The previous types of distributions considered for $F_i(x)$ were defined on the semi-infinite domain $(0, \infty)$. This was done for analytic convenience. However, in practice some mechanism will act to truncate x_i from above. One may ask what the effect of truncating $F_i(x)$ at some large quantile is on the mean and standard deviation of z_k. It is clear that distributions of x_i which exhibit large positive skewness would be most sensitive to truncation for our problem. Therefore, an exponential c.d.f. was used. This distribution was truncated at the 0.99 and at the 0.999 quantile. The value of λ_i was adjusted so that $E[x_i]$ is the same (=1) for all cases. Results of these calculations are displayed in Table E.10. One observation of interest is that the mean of z_k is much less affected by truncation than the standard deviation of z_k. Truncation at even the 0.99 quantile does not have a remarkable effect on $E[z_k]$. For example, $E[z_6]$ is 2.45 without truncation; is 2.43 when truncation is at the 0.999 quantile; and is 2.36 when truncation occurs at the 0.99 quantile. However, the corresponding standard deviations of z_6 are, respectively, 1.22, 1.15, 0.97.

13. Conclusions

The deterministic approach to activity networks, which replaces random activity times by their mean values, can lead to large underestimates of the mean completion time for subnetworks of parallel activities. For specific subnetworks of parallel activities, the methods of this report can be used to estimate the PERT error in the mean, standard deviation, and quantile of the completion time. This approach is preferable to numerical generalizations, since activity networks are quite diverse. The PERT assumption regarding the Normality of the completion-time probability distribution is grossly wrong for (sub)networks having multiple parallel random activities, each of which is positively skewed. In those instances where network structure and logic are quite complex, it seems practical to use a stochastic networking technique such as VERT rather than attempt to "patchup" PERT.

14. The method derived to treat the problem of networks of parallel random activities is considerably more general than this application may suggest. The

method derived and used in this report is quite accurate for calculating the means, standard deviations, and quantiles of the maximum of a set of k positive, continuous random variables. In applications involving activity networks, the computational errors of the method are completely negligible.

15. Certain quantitative generalities have been induced from specific numerical examples. It is noted that when the mean of one of the exponential RV's (x_i) in a set of k is greater than the others, the standard deviation of z_k ($= \max_i(x_i)$) will eventually decrease with increasing k. An implication for activity networks is the following: As the number of parallel activities increases, the standard deviation of the completion time will actually decrease beyond a certain point. This contradicts one of the PERT assumptions. For positively skewed distributions of x_i -- such as gamma and Weibull $F_i(x)$, the mean of z_k for $k \lesssim 6$ is not very sensitive to the <u>form</u> of $F_i(x)$ provided the distribution parameters are chosen to yield the same first two statistical moments. This fact makes consideration of the precise form of the distribution of x_i somewhat academic for parallel activity networks.

16. Consider progressively adding RV's (x_i) to the set; i.e., increasing k, in such a manner that each x_i has a smaller mean than its predecessor. In this case the upper tail of the c.d.f. of the maximum, $G_k(z)$, is insensitive to k above a certain point. For example, for $E[x_i] = 1.2\ E[x_{i+1}]$, no appreciable change occurs in the 0.9 quantile for $k > 3$. The implication of this result for activity networks is that adding more activities to a parallel network may not change the low-risk forecasted completion time.

17. The use of semi-infinite c.d.f.'s for x_i instead of truncated c.d.f.'s with the same mean <u>is justified</u> if: (a) the focus of interest is on $E[z_k]$ and (b) the truncation point exceeds the 0.99 quantile. In some cases network logic may require that only k of n ($k < n$) parallel activities need to be completed before passing this point in the network. Even tho n may be large, viz. > 6, the difference between mean completion times for the two cases n of n versus k of n can be remarkable. This fact suggests that particular attention be paid to characterizing network logic for parallel activities. Implementation of diverse logical forms is facilitated by using stochastic networking methods such as Venture Evaluation and Review Technique (VERT).

REFERENCES

1. Grubbs, F.E. "Attempts to Validate Certain PERT Statistics or 'Picking on PERT'," Opns. Res., Vol. 10, pp. 912 - 915, 1962.

2. Clark, C.E. "The Greatest of a Finite Set of Random Variables," Opns. Res., Vol. 9, No. 2, pp. 145 - 162, March - April 1961.

3. MacCrimmon, K.R. and Ryavec, C.A. "An Analytical Study of the PERT Assumptions," Opns. Res., Vol. 12, No. 1, pp. 16 - 37, January - February 1964.

4. Greer, W.R. Jr. "Why Doesn't PERT Work?", Resource Mgmt. Journal, pp. 27 - 31, Summer 1983.

5. Matheiss, T.H., Moeller, G. and Kilar, J. "Improving Industrial Readiness With Venture Evaluation and Review Technique (VERT)," Interfaces, Vol. 12, No. 1, pp. 21 - 26, February 1982.

6. Moeller, G.L. and Digman, L.A. "Operations Planning with VERT," Opns. Res., Vol. 29, No. 4, pp. 676 - 697, July - August 1981.

7. Lee, S.M., Digman, L.A. and Moeller, G.L. Network Analysis for Management Decisions, Kluwer-Nijhoff Pub., Hingham, MA, c. 1981.

8. Gumbel, E.J. Statistics of Extremes, Columbia Univ. Press, New York, NY, c. 1958.

9. Guenther, W.C. "An Easy Method for Obtaining Percentage Points of Order Statistics," Technometrics, Vol. 19, No. 3, pp. 319 - 321, August 1977.

10. Abramowitz, M. and Stegun, I. Handbook of Mathematical Functions, AMS 55, Nat. Bureau of Standards, August 1966.

11. Bennett, A.A., Milne, W.E. and Bateman, H. Numerical Integration of Differential Equations, Dover Pub., New York, NY, c. 1956.

DISTRIBUTION

No of Copies	
1	HQDA (DAMO-ZA) Wash Dc 20310
	Commander USAMC 5001 Eisenhower Av., Alexandria, VA 22333-0001
1	ATTN: AMCDE
1	AMCDMA-M
1	AMCPP-IM
	Director, US AMSAA Aberdeen PG, MD 21005-5066
1	ATTN: AMXSY-DL
1	AMXSY-R
	Commander, US Army Logistics Center Fort Lee, VA 23801
1	ATTN: ATCL-S
12	Defense Technical Information Center Cameron Station Alexandria, VA 22314
	Commander US AMCCOM (D) Dover, NJ 07801-5001
1	ATTN: AMSMC-OAS (D)
1	AMSMC-LCU-M (D)
1	AMSMC-LCS-E (D)
1	AMSMC-LCE-D (D)
1	AMSMC-RAA (D)
1	AMSMC-LCA-P (D)
1	AMCPM-ADG (D) Bldg 171 N
	Commander US AMCCOM (A) Aberdeen PG, MD 21005-5066
1	ATTN: AMSMC-CLY (A)
	Commander USA Communications and Electronics Command Ft. Monmouth, NJ 07703-5304
1	ATTN: AMSEL-PL-SA
	Commander USA MICOM Redstone Arsenal, AL 35898-5060
1	ATTN: AMSMI-DS

No of Copies	
	Commander
	USAAVSCOM
1	4300 Goodfellow Blvd
	St. Louis, MO 63120-1798
	ATTN: AMSAV-BA
	Commander
	Chemical Research & Dev Center
1	USAMCOM
	APG, MD 21010-5423
	Commander USA TACOM
	Warren, MI 48090
1	ATTN: AMSTA-V
	Commander USA TECOM
	Aberdeen PG, MD 21005-5055
1	ATTN: AMSTE-SY
	Commander
	USA Production Base Modernization Agency
	Dover, NJ 07801-5001
1	ATTN: Mr. S. Karlin
	Director US AMETA
	Rock Island, IL 61299-6000
1	ATTN: AMXOM-QA
	Director USA IBEA
	Rock Island, IL 61299-6000
1	ATTN: AMXIM-MT/Mr. S. McGlone
	Director
	US Army Inventory Research Office
	Room 800, Custom House
	2nd and Chestnut Streets
	Philadelphia, PA 19160
1	ATTN: AMXMC-IRO
	Commander US AMCCOM (R)
	Rock Island, IL 612-9-6000
1	ATTN: AMSMC-IRI (R)/Mr. J. Kilar
1	AMSMC-PP (R)/Mr. M. Craighead
1	AMSMC-OAA (R)/Mr. B. Witherspoon
1	AMSMC-OAA-S (R)/Mr. D. Eckman
1	AMSMC-OAA-S (R)/Mr. G. Moeller
1	AMSMC-OAA-L (R)/Mr. N. Trier
1	AMSMC-OAA-A (R)/Mr. B. White
1	AMSMC-OPP (R)/Mr. P. Woodhouse

DISTRIBUTION LIST (Cont'd)

No. of Copies	
1	AMSMC-OPI (R)/Mr. R. Heider
2	AMSMC-LEP-L (R)/Ms Sandi Paul
1	AMCPM-ADG-L (R)/Mr. J. Wasson
	Chief Benet Weapons Lab, LCWCL Armament R&D Center US AMCCOM Watervliet, NY 12189-5000
1	ATTN: AMSMC-LCB-TL
	Commander Belvoir R&D Center Fort Belvoir, VA 22060-5606
1	ATTN: STRBE-HA
	Commander UASOTEA 5600 Columbia Pike Falls Church, VA 22041
1	ATTN: CSTE-ZS
1	CSTE-TDD
	Commander USATRADOC Fort Montroe, VA 23651-5000
1	ATTN: ATCD-F
1	ATCD-ZS
1	ATCD-TA
	Commander USA Logistics Evaluation Agency New Cumberland Army Depot New Cumberland, PA 17070
1	ATTN: DAL-LEM/Tech Lib
	Commander USARDC Dover, NJ 07801-5001
1	ATTN: SMCAR-SCF

DISTRIBUTION LIST (Cont'd)

No. of Copies	
	Commander
	USA Aviation Center
	Fort Rucker, AL 36362
1	ATTN: ATZQ-MR (RAM Branch)
	Commander
	USATECOM
	APG, MD 21005-5055
1	ATTN: AMSTE-CM
1	AMSTE-TD
	Director
	USATRASANA
	White Sands Missile Range
	NM 88002-5502
1	ATTN: ATAA-SL
1	ATAA-TG
	Director
	USATORA
	White Sands Missile Range
	NM 88002-5502
1	ATTN: ATOR
	Commander
	USAMICOM
	Redstone Scientific Info Center
	Redstone Arsenal, AL 35898-5500
1	ATTN: Documents
	Commander
	Yuma Proving Ground
	Yuma, AZ 85365-9103
	ATTN: STEYP-MTE
	Director
	USA TRADOC Systems Analysis Activity
	White Sands Missile Range
	NM 88002-5502
1	ATTN: ATOR-T
	Project Manager
	Cannon Arty Wpns System
	Dover, NJ 07801-5001
1	ATTN: AMCPM-CAWS-S

DISTRIBUTION LIST (Cont'd)

No. of Copies	
	Commander USA Combined Arms Combat Development Activity Fort Leavenworth, KS 66027
1	ATTN: ATZL-CAM-M
	Superintendent Naval Postgraduate School Dept. of Mechanical Engineering Monterey, CA 93940
1	ATTN: A. E. Fuhs Code 1424 Library
	Commander Naval Surface Weapons Center Dahlgren, VA 22448
1	ATTN: Code DX-21 Tech Library
	Commander Naval Weapons Center China Lake, CA 93555
1	ATTN: Code 388, Info. Sci. Div.

ANNEX A

Mean and Standard Deviation for the Maximum of Several Positive Continuous Random Variables

This annex derives formulas for calculating the statistical moments of the largest random variable in a set of k positive random variables (x_i, $1 \leq i \leq k$). Application is made to an example in which x_i, $1 \leq i \leq k$, are exponential random variables (RV's) from distributions having different rate parameters. This example is sufficiently tractable to allow a closed-form solution for the statistical moments. These exact values are used to evaluate the accuracy of numerical procedures, which are suitable for a more general case. Specific examples are displayed, and some general inferences are made from them. The examples are chosen for their applicability to networks of parallel activities which must all be completed for passage thru the network. It is noted that this type of problem is a special case of the problem in which passage thru the network requires the completion of k of n, $k \leq n$, activities.

To start, consider two positive continuous random variables x_1 and x_2 having probability density functions (p.d.f.'s) $f_1(x)$ and $f_2(x)$, respectively. The associated cumulative distribution functions (c.d.f.'s) are denoted $F_1(x)$ and $F_2(x)$, with a domain of x: $(0 \leq x < \infty)$.

Define

$$y = \max (x_1, x_2) \tag{1}$$

with p.d.f. $f_y(y)$ and c.d.f. $F_y(y)$. The one's complement of F_i is denoted

$$\overline{F}_i = 1 - F_i \ , \ i = 1,2 \quad . \tag{2}$$

Invoking the definition of y in (1) and probability arguments, one can state that

$$f_y(y) = F_1(y)f_2(y) + F_2(y)f_1(y) \quad . \tag{3a}$$

Suppressing functional notation, this expression can be written as

$$dF_y = d(F_1F_2) \tag{3b}$$

or

$$F_y = F_1F_2 \quad . \tag{3c}$$

Using (2) with (3a),

$$f_y = f_1 + f_2 - (\overline{F}_1f_2 + \overline{F}_2f_1) \quad . \tag{4}$$

The last expression can be used to find a simple recursive equation for the expectation of y:

$$E[y] = \int_0^\infty yF_y(y)dy \quad . \tag{5}$$

From (4) and (5),

$$E[y] = \int_0^\infty yf_1(y)dy + \int_0^\infty yf_2(y)dy - \int_0^\infty y\overline{F}_1(y)f_2(y)dy - \int_0^\infty y\overline{F}_2(y)f_1(y)dy \quad . \tag{6a}$$

Recalling the definitions of f_1 and f_2,

$$E[y] = E[x_1] + E[x_2] - \int_0^\infty y\overline{F}_1f_2dy - \int_0^\infty y\overline{F}_2f_1dy \quad . \tag{6b}$$

Using integration by parts,

$$-\int_0^\infty y\overline{F}_1f_2dy = \int_0^\infty y\overline{F}_2f_1dy + \int_0^\infty \overline{F}_1F_2dy - E[x_1] \quad .$$

Combining this result with (6b) produces

$$E[y] = E[x_2] + \int_0^\infty \overline{F}_1F_2dy \quad . \tag{7a}$$

One can obtain an alternative expression by using symmetry arguments and by exchanging indices, or by the following argument. From (7a) using complements of $\overline{F}_1$ and F_2,

$$E[y] = E[x_2] + \int_0^\infty (\overline{F}_1 - \overline{F}_2 + F_1\overline{F}_2)dy$$

or

$$E[y] = E[x_1] + \int_0^\infty F_1\overline{F}_2dy \quad . \tag{7b}$$

The second moment of y with respect to the origin,

$$E[y^2] = \int_0^\infty y^2f_ydy \quad ,$$

can be obtained from (4):

$$E[y^2] = E[x_1^2] + E[x_2^2] - \int_0^\infty y^2\overline{F}_1 f_2 dy - \int_0^\infty y^2\overline{F}_2 f_1 dy \quad . \tag{8}$$

This expression is analogous to (6b) for the first statistical moment. Integrating the first integral in (8) by parts and combining terms gives

$$E[y^2] = E[x_2^2] + 2\int_0^\infty yF_2\overline{F}_1 dy \tag{9a}$$

or

$$E[y^2] = E[x_1^2] + 2\int_0^\infty yF_1\overline{F}_2 dy \quad . \tag{9b}$$

Example

For a specific example of the above theory, suppose that

$$F_1(x) = 1 - e^{-\lambda_1 x} \tag{E.1}$$

and

$$F_2(x) = 1 - e^{-\lambda_2 x} \quad . \tag{E.2}$$

From (7a), the expected value of y is

$$E[y] = \lambda_2^{-1} + \int_0^\infty e^{-\lambda_1 y}(1 - e^{-\lambda_2 y})dy$$

or

$$E[y] = \lambda_1^{-1} + \lambda_2^{-1} - (\lambda_1 + \lambda_2)^{-1} \quad . \tag{E.3}$$

We will return to this example for extension later.

The relation between distribution functions in (3) for the maximum of two RV's can be generalized to the max of k, as follows. Define a positive RV x_k with p.d.f. and c.d.f. denoted by f_k and F_k. Also, define the RV z_k:

$$z_k = \max_i(x_1, x_2, \ldots, x_i, \ldots, x_k) \quad , \tag{10}$$

with p.d.f. and c.d.f. denoted by g_k and G_k, respectively.
Note that

$$z_{k+1} = \max(z_k, x_{k+1}) \quad . \tag{11}$$

Then, (3) yields

$$g_{k+1}(z) = G_k(z)f_{k+1}(z) + F_{k+1}(z)g_k(z) \tag{12a}$$

and

$$G_{k+1}(z) = G_k(z)F_{k+1}(z) \tag{12b}$$

or

$$G_{k+1}(z) = \Pi_{j=1}^{k+1}F_j(z) \quad . \tag{12c}$$

Notice that both the p.d.f. and c.d.f. of z_k can be obtained recursively. Further, there is no requirement that all the F_j be identical, as in the case with order statistics.

In (12) G_k plays the role of F_1 and F_{k+1} plays the role of F_2 in (3a). A similar exchange of variables in (7a) produces the following relation for mean values of z_k

$$E[z_{k+1}] = E[x_{k+1}] + \int_0^\infty \overline{G}_k(y)F_{k+1}(y)dy \quad . \tag{13a}$$

If $\overline{F}_{k+1}$ has a simpler form than $\overline{G}_k$, the following analog of (7b) may be preferred

$$E[z_{k+1}] = E[z_k] + \int_0^\infty G_k(y)\overline{F}_{k+1}(y)dy \quad . \tag{13b}$$

A generalization of (9b) for the second origin moment is

$$E[z_{k+1}^2] = E[z_k^2] + 2\int_0^\infty yG_k(y)\overline{F}_{k+1}(y)dy, \; k \geq 1. \tag{14}$$

The variance of z_{k+1} can be obtained from the first and second origin moments via:

$$V[z_{k+1}^2] = E[z_{k+1}^2] - (E[z_{k+1}])^2 \quad . \tag{15}$$

Note that (13), (14), and (15) provide the basis of a numerical procedure for calculating the mean and variance of z_k recursively. Computational experience with this procedure indicates that somewhat better accuracy is obtained than is possible, at the same integration step size*, with a straightforward integration of G_k via

$$E[z_k] = \int_0^\infty \overline{G}_k(y)dy \tag{16}$$

and

$$E[z_k^2] = 2\int_0^\infty y\overline{G}(y)dy \quad . \tag{17}$$

These general results can be used to extend the example, started following (9). Suppose that a random variable x_3 also has an exponential distribution:

$$f_3(x) = \lambda_3 e^{-\lambda_3 x}$$

and

$$\overline{F}_3(x) = e^{-\lambda_3 x} \quad . \tag{E.4}$$

From (12), with $G_1 = F_1$,

$$g_2(z) = (1 - e^{-\lambda_1 z})\lambda_2 e^{-\lambda_2 z} + (1 - e^{-\lambda_2 z})\lambda_1 e^{-\lambda_1 z}$$

or

$$g_2(z) = \lambda_1 e^{-\lambda_1 z} + \lambda_2 e^{-\lambda_2 z} - (\lambda_1 + \lambda_2)e^{-(\lambda_1+\lambda_2)z} \quad . \tag{E.5}$$

Whence,

$$G_2(z) = 1 - e^{-\lambda_1 z} - e^{-\lambda_2 z} + e^{-(\lambda_1+\lambda_2)z} \quad . \tag{E.6}$$

Since $E[y]$ in (E.3) is the same as $E[z_2]$, in this notation, (13b) gives

$$E[z_3] = \lambda_1^{-1} + \lambda_2^{-1} - (\lambda_1 + \lambda_2)^{-1} + \int_0^\infty e^{-\lambda_3 y}(1 - e^{-\lambda_1 y} - e^{-\lambda_2 y} + e^{-(\lambda_1+\lambda_2)y})dy$$

---

* The same type of quadrature formula is also assumed. For computational efficiency Simpson's rule [11, p. 31] is recommended.

[11] Bennett, A.A., Milne, W.E., and Bateman, H. Numerical Integration of Differential Equations, Dover Pub., New York, NY, c. 1956.

or

$$E[z_3] = \lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1} - (\lambda_1 + \lambda_2)^{-1} - (\lambda_1 + \lambda_3)^{-1} - (\lambda_2 + \lambda_3)^{-1} + (\lambda_1 + \lambda_2 + \lambda_3)^{-1} \quad . \tag{E.7}$$

The p.d.f. of z_3, $g_3(z)$, is obtained from (12) using (E.4, E.5, E.6).

$$g_3(z) = (1 - e^{-\lambda_1 z} - e^{-\lambda_2 z} + e^{-(\lambda_1+\lambda_2)z})\lambda_3 e^{-\lambda_3 z} + (1 - e^{-\lambda_3 z})(\lambda_1 e^{-\lambda_1 z} + \lambda_2 e^{-\lambda_2 z} - (\lambda_1 + \lambda_2)e^{-(\lambda_1+\lambda_2)z}) \quad .$$

Simplifying,

$$g_3(z) = \lambda_1 e^{-\lambda_1 z} + \lambda_2 e^{-\lambda_2 z} + \lambda_3 e^{-\lambda_3 z} - (\lambda_1 + \lambda_2)e^{-(\lambda_1+\lambda_2)z} - (\lambda_1 + \lambda_3)e^{-(\lambda_1+\lambda_3)z} - (\lambda_2 + \lambda_3)e^{-(\lambda_2+\lambda_3)} + (\lambda_1 + \lambda_2 + \lambda_3) e^{-(\lambda_1+\lambda_2+\lambda_3)z} \quad ,$$

or

$$g_3(z) = \sum_{j=1}^{3} \lambda_j \exp - (\lambda_j z) - \sum_{k>j}^{3} \sum_{j=1}^{3} (\lambda_j + \lambda_k) \exp - (\lambda_j + \lambda_k)z + \sum_{j=1}^{3} \lambda_j \exp - (\sum_{j=1}^{3} \lambda_j)z \quad . \tag{E.8}$$

The expression for $G_3(z)$ is simplified if one uses the following notational convention. Let $E(r,z)$ denote

$$1 - e^{-rz} \quad .$$

Then, it is seen that each of the terms in (E.8) is an exponential density so that

$$G_3(z) = \sum_{j=1}^{3} E(\lambda_j, z) - \sum_{k>j}^{3} \sum_{j=1}^{3} E(\lambda_j + \lambda_k, z) + E(\sum_{j=1}^{3} \lambda_j, z) \quad . \tag{E.9}$$

Notice that when the x_i random variables in (10) are all exponential, the density g_k is composed of sums of exponential densities. This is seen in

(E.5) and (E.8). In the recursive relation for g_k, equation (12), one notes for the specific example being considered that

$$g_{k+1}(z) = g_k(z) + G_k(z)\lambda_{k+1}e^{-\lambda_{k+1}z} - e^{-\lambda_{k+1}z}g_k(z) \quad . \tag{E.10}$$

The last two terms in (E.10) are seen to contribute additional exponential density terms to those of $g_k(z)$. Specifically for k = 3, using (E.9),

$$\begin{aligned} g_4(z) = {} & g_3(z) + \lambda_4 e^{-\lambda_4 z}\sum_{j=1}^{3}E(\lambda_j,z) - \lambda_4 e^{-\lambda_4 z}\sum_{k>j}^{3}\sum_{j=1}^{3}E(\lambda_j + \lambda_k,z) + \\ & \lambda_4 e^{-\lambda_4 z}E(\sum_{j=1}^{3}\lambda_j,z) - e^{-\lambda_4 z}\sum_{j=1}^{3}\lambda_j \exp - (\lambda_j z) + \\ & e^{-\lambda_4 z}\sum_{k>j}^{3}\sum_{j=1}^{3}(\lambda_j + \lambda_k) \exp - (\lambda_j + \lambda_k)z - e^{-\lambda_4 z}\sum_{j=1}^{3}\lambda_j \exp - \\ & (\sum_{j=1}^{3}\lambda_j)z \quad . \end{aligned} \tag{E.11}$$

After some manipulation this expression simplifies to

$$\begin{aligned} g_4(z) = {} & \sum_{j=1}^{4}\lambda_j \exp - (\lambda_j z) - \sum_{k>j}^{4}\sum_{j=1}^{4}(\lambda_j + \lambda_k) \exp - (\lambda_j + \lambda_k)z + \\ & \sum_{k>j}^{4}\sum_{j>i}^{4}\sum_{i=1}^{4}(\lambda_i + \lambda_j + \lambda_k) \exp - (\lambda_i + \lambda_j + \lambda_k)z - \\ & \sum_{j=1}^{4}\lambda_j \exp - (\sum_{j=1}^{4}\lambda_j)z \quad . \end{aligned} \tag{E.12}$$

Using the notation of equation (E.9), the c.d.f. of z_4 can be written

$$\begin{aligned} G_4(z) = {} & \sum_{j=1}^{4}E(\lambda_j,z) - \sum_{k>j}^{4}\sum_{j=1}^{4}E(\lambda_j + \lambda_k,z) + \sum_{k>j}^{4}\sum_{j>i}^{4}\sum_{i=1}^{4}E(\lambda_i + \lambda_j + \lambda_k,z) - \\ & E(\sum_{j=1}^{4}\lambda_j,z) \quad . \end{aligned} \tag{E.13}$$

Since $G_k(z)$ is a sum of exponential probabilities, where the typical term $E(r,z)$ has an expected value of r^{-1}, one can immediately write

$$\begin{aligned} E[z_4] = {} & \sum_{j=1}^{4}\lambda_j^{-1} - \sum_{k>j}^{4}\sum_{j=1}^{4}(\lambda_j + \lambda_k)^{-1} + \sum_{k>j}^{4}\sum_{j>i}^{4}\sum_{i=1}^{4}(\lambda_i + \lambda_j + \lambda_k)^{-1} - \\ & (\sum_{j=1}^{4}\lambda_j)^{-1} \quad . \end{aligned} \tag{E.14}$$

The expected values of z_2, z_3, and of z_4 -- in, respectively, (E.3), (E.7), and (E.14) -- in this example all conform to a common formulation: Combine the rate parameters associated with the k exponential random variables by summing sets of terms, with sets alternating in sign. The first set is just the sum of all k inverse rates. The second set is negative. This second set is the sum of the inverses of the sum of unique pairs of parameters. The third set -- if it exists -- is the sum of the inverses of the sum of unique triples; i.e., combinations of k values of λ_i taken three at a time. The last set consists of a single term -- the inverse of the single combination of k values of λ_i taken k at a time. With this formulation the expectation of z_5 is immediately written as

$$E[z_5] = \sum_{i=1}^{5}\lambda_i^{-1} - \sum_{j>i}^{5}\sum_{i=1}^{5}(\lambda_i + \lambda_j)^{-1} + \sum_{k>j}^{5}\sum_{j>i}^{5}\sum_{i=1}^{5}(\lambda_i + \lambda_j + \lambda_k)^{-1} - \sum_{l>k}^{5}\sum_{k>j}^{5}\sum_{j>i}^{5}\sum_{i=1}^{5}(\lambda_i + \lambda_j + \lambda_k + \lambda_l)^{-1} + (\sum_{i=1}^{5}\lambda_i)^{-1} \quad . \tag{E.15}$$

The c.d.f. of z_5 is, by induction,

$$G_5(z) = \sum_{i=1}^{5}E(\lambda_i,z) - \sum_{j>i}^{5}\sum_{i=1}^{5}E(\lambda_i + \lambda_j,z) + \sum_{k>j}^{5}\sum_{j>i}^{5}\sum_{i=1}^{5}E(\lambda_i + \lambda_j + \lambda_k,z) - \sum_{l>k}^{5}\sum_{k>j}^{5}\sum_{j>i}^{5}\sum_{i=1}^{5}E(\lambda_i + \lambda_j + \lambda_k + \lambda_l,z) + E(\sum_{i=1}^{5}\lambda_i,z) \quad . \tag{E.16}$$

The second statistical moment of $G_k(z)$ with respect to the origin, $E[z_k^2]$, can be obtained directly from $G_k(z)$ by using the fact that the second origin moment of $E(r,z)$ is $2r^{-2}$. Then from $E[z_k^2]$ and $E[z_k]$, the variance of z_k is written, from (15), as

$$V[z_k] = E[z_k^2] - (E[z_k])^2 \quad .$$

Thus, from (E.5),

$$E[z_2^2] = 2\lambda_1^{-2} + 2\lambda_2^{-2} - 2(\lambda_1 + \lambda_2)^{-2} \quad . \tag{E.17}$$

From (E.3) and (E.17) and using (15),

$$V[z_2] = \lambda_1^{-2} + \lambda_2^{-2} - 3(\lambda_1 + \lambda_2)^{-2} \quad . \tag{E.18}$$

From (E.9),

$$E[z_3^2] = 2\sum_{j=1}^{3}\lambda_j^{-2} - 2\sum_{k>j}^{3}\sum_{j=1}^{3}(\lambda_j + \lambda_k)^{-2} + 2(\sum_{j=1}^{3}\lambda_j)^{-2} \quad . \tag{E.19}$$

From (E.13),

$$E[z_4^2] = 2\sum_{i=1}^{4}\lambda_i^{-2} - 2\sum_{j>i}^{4}\sum_{i=1}^{4}(\lambda_i + \lambda_j)^{-2} + 2\sum_{k>j}^{4}\sum_{j>i}^{4}\sum_{i=1}^{4}(\lambda_i + \lambda_j + \lambda_k)^{-2} -$$
$$2\left(\sum_{i=1}^{4}\lambda_i\right)^{-2} \quad . \tag{E.20}$$

And, from (E.16),

$$E[z_5^2] = 2\sum_{i=1}^{5}\lambda_i^{-2} - 2\sum_{j>i}^{5}\sum_{i=1}^{5}(\lambda_i + \lambda_j)^{-2} + 2\sum_{k>j}^{5}\sum_{j>i}^{5}\sum_{i=1}^{5}(\lambda_i + \lambda_j + \lambda_k)^{-2} -$$
$$2\sum_{l>k}^{5}\sum_{k>j}^{5}\sum_{j>i}^{5}\sum_{i=1}^{5}(\lambda_i + \lambda_j + \lambda_k + \lambda_l)^{-2} + 2\left(\sum_{i=1}^{5}\lambda_i\right)^{-2} \quad . \tag{E.21}$$

A great simplification of the sample results occurs when each of the RV's x_i has the same exponential distribution. Consider the case in which all $\lambda_i = 1$, for instance. Then,

$$E[x_i] = 1$$

and

$$E[z_k] = \sum_{j=1}^{k}(-1)^{j+1}\binom{k}{j}/j \quad . \tag{E.22}$$

For the second origin moment in this special case,

$$E[z_k^2] = 2\sum_{j=1}^{k}(-1)^{j+1}\binom{k}{j}/j^2 \quad . \tag{E.23}$$

Numerical values of these moments are found in Table E.1.

TABLE E.1

STATISTICAL MOMENTS OF THE DISTRIBUTION OF z_k, WHERE

$$z_k = \max_i(x_1, x_2, \ldots, x_i, \ldots, x_k)$$

WITH x_i A STANDARDIZED EXPONENTIAL R.V.

k	$E[z_k]$	$E[z_k^2]$	Std Dev z_k	Coef Var z_k
1	1.0000	2.0000	1.0000	1.0000
2	1.5000	3.5000	1.1180	0.7453
3	1.8333	4.7222	1.1667	0.6364
4	2.0833	5.7639	1.1932	0.5727
5	2.2833	6.6772	1.2098	0.5298
6	2.4500	7.4938	1.2212	0.4984
7	2.5929	8.2350	1.2296	0.4724

One observation of interest from Table E.1 is that the coefficient of variation of z_k decreases as k increases. In this example the mean increases more rapidly with k than does the standard deviation.

To demonstrate the effect of differences in the rate parameters λ_i, $(i = 1, \ldots, k)$, consider the following numerical example. Let $\lambda_1 = 1$, and for all succeeding values of λ_i, let $\lambda_i = 1.2\lambda_{i-1}$. The values of $E[z_k]$ and $E[z_k^2]$ must be calculated via the formulas preceding (E.22). Results are tabulated in Table E.2.

TABLE E.2

STATISTICAL MOMENTS OF THE DISTRIBUTION OF z_k, WHERE

$$z_k = \max_i(x_1, x_2, \ldots, x_i, \ldots, x_k)$$

WITH x_i EXPONENTIALLY DISTRIBUTED

WITH RATE PARAMETER λ_i: $\lambda_1 = 1$; $\lambda_i = 1.2\lambda_{i-1}$, $i > 1$

k	$E[x_k]$	Std Dev z_k	Coef Var z_k
1	1.0000	1.0000	1.0000
2	1.3788	1.0366	0.7518
3	1.5593	1.0182	0.6530
4	1.6514	0.9948	0.6024
5	1.6992	0.9763	0.5746
6	1.7237	0.9637	0.5591
7	1.7359	0.9560	0.5507

Note that $E[z_k]$ appears to be near an asymptote for $k = 7$. In this instance the coefficient of variation diminishes even more rapidly with k than it does in the case in which all $\lambda_i = 1$.

In contrast to the above examples, consider a case in which all the random variables (x_i) in the set, save one, have the same mean. The exceptional RV has a greater mean. A numerical example of this case is shown in Table E.3.

TABLE E.3

STATISTICAL MOMENTS OF THE MAXIMUM OF A SET
OF EXPONENTIAL RANDOM VARIABLES x_i, $1 \leq i \leq k$,
WHERE $E[x_1] = 1$ AND $E[x_i] = 0.79$, $i > 1$

k	$E[z_k]$	Std Dev z_k	Coef Var z_k
1	1.0000	1.0000	1.0000
2	1.3487	1.0197	0.7561
3	1.5855	1.0256	0.6469
4	1.7653	1.0277	0.5822
5	1.9105	1.0285	0.5383
6	2.0324	1.0286	0.5061
7	2.1375	1.0284	0.4811

The results in Table E.3 resemble those in Table E.1. In both cases the value of $E[z_k]$ increases with k, whereas the standard deviation of z_k increases more slowly with k. Thus, the coefficient of variation decreases with k, but not so rapidly as in Table E.2. It is noted that when the mean of one RV in the set is greater than the constant mean of all others, the variance of z_k will eventually decrease with k beyond a certain point. In the example above, with $E[x_i] = 0.79$, $i > 1$, the maximum variance occurs at $k = 6$.

Computational Errors

The results in Tables E.1, E.2, and E.3 are numerically exact to the number of significant digits displayed. These results were obtained from the closed-form solution equations using double-precision arithmetic.

When more general results are wanted, it is convenient to use a numerical procedure applicable to all distributions of positive, continuous RV's. The procedure displayed in Annex B uses equation (12) to obtain the c.d.f. of z_k. The mean and variance of z_k are obtained using (13), (14), and (15). Because the procedure involves numerical integration at each recursive step, a computational error is incurred. With rectangular integration an integration step size of 0.002 is judged a satisfactory compromise between accuracy and speed. With Simpson's rule, the step size can be relaxed to 0.005, yielding essentially five digit accuracy in $E[z_k]$ for $k < 7$. Computational errors with the first order procedure (step size 0.002) are displayed in Tables E.4, E.5, and E.6. These results are regarded as representative of the errors to be encountered in network applications.

TABLE E.4

ACCURACY OF A NUMERICAL METHOD FOR OBTAINING THE MEAN AND STD DEVIATION OF THE MAXIMUM OF A SET OF RV'S

Case A: All RV's (x_i) are exponentially distributed with rate parameter $\lambda_i = 1$, $1 \leq i \leq k$.

No. RV's k	Exact Values		Numerical Approx.	
	$E[z_k]$	Std Dev z_k	$E[z_k]$	Std Dev z_k
2	1.5000	1.1180	1.5000	1.1176
3	1.8333	1.1667	1.8332	1.1660
4	2.0833	1.1932	2.0832	1.1921
5	2.2833	1.2098	2.2832	1.2085
6	2.4500	1.2212	2.4498	1.2196

TABLE E.5

ACCURACY OF A NUMERICAL METHOD FOR OBTAINING THE MEAN AND STD DEVIATION OF THE MAXIMUM OF A SET OF RV'S

Case B: All RV's (x_i) are exponentially distributed with rate parameters λ_i:
$\lambda_1 = 1$, $\lambda_i = 1.2\lambda_{i-1}$, $i > 1$.

No. RV's k	Exact Values		Numerical Approx.	
	$E[z_k]$	Std Dev z_k	$E[z_k]$	Std Dev z_k
1	1.0000	1.0000	1.0000	1.0000
2	1.3788	1.0366	1.3788	1.0366
3	1.5593	1.0182	1.5593	1.0181
4	1.6514	0.9948	1.6514	0.9948
5	1.6992	0.9763	1.6992	0.9762
6	1.7237	0.9637	1.7237	0.9636

TABLE E.6

TYPICAL ERRORS OF THE NUMERICAL PROCEDURE FOR CALCULATING MEANS AND STANDARD DEVIATIONS OF THE MAXIMUM OF A SET OF POSITIVE RANDOM VARIABLES

No. RV's k	Errors for Case A		Errors for Case B	
	$E[z_k]$	Std Dev z_k	$E[z_k]$	Std Dev z_k
2	0.0000	-0.0004	0.0000	-0.0000
3	-0.0001	-0.0007	0.0000	-0.0001
4	-0.0001	-0.0011	0.0000	0.0000
5	-0.0001	-0.0013	0.0000	-0.0001
6	-0.0002	-0.0016	0.0000	-0.0001

It is noted that the computational errors of the mean value and standard deviation in these examples have a consistent sign, indicating underestimation. Since the numerical procedure used here is first-order, the errors in Table E.6 are proportional to integration step size.

Exact distribution functions of z_k, $(2 \leq k \leq 6)$, are shown in Figure E.1 for the case in which x_i, $1 \leq i \leq k$, are exponential with unity mean. It is noted that substantial positive skewness persists with increasing k up to $k = 6$. In Figure E.2 is displayed the c.d.f. of z_k for the case in which $\lambda_1 = 1$ and $\lambda_i = 1.2\lambda_{i-1}$, $2 \leq i \leq k$. Because each additional (ordered) RV has a smaller mean value than its predecessors, the upper tail of the distribution of z_k is approaching an asymptotic form as k grows indefinitely. From Figure E.2 it is apparent why the standard deviation of k decreases with increasing k. This example applies to those networks having parallel activities in which a

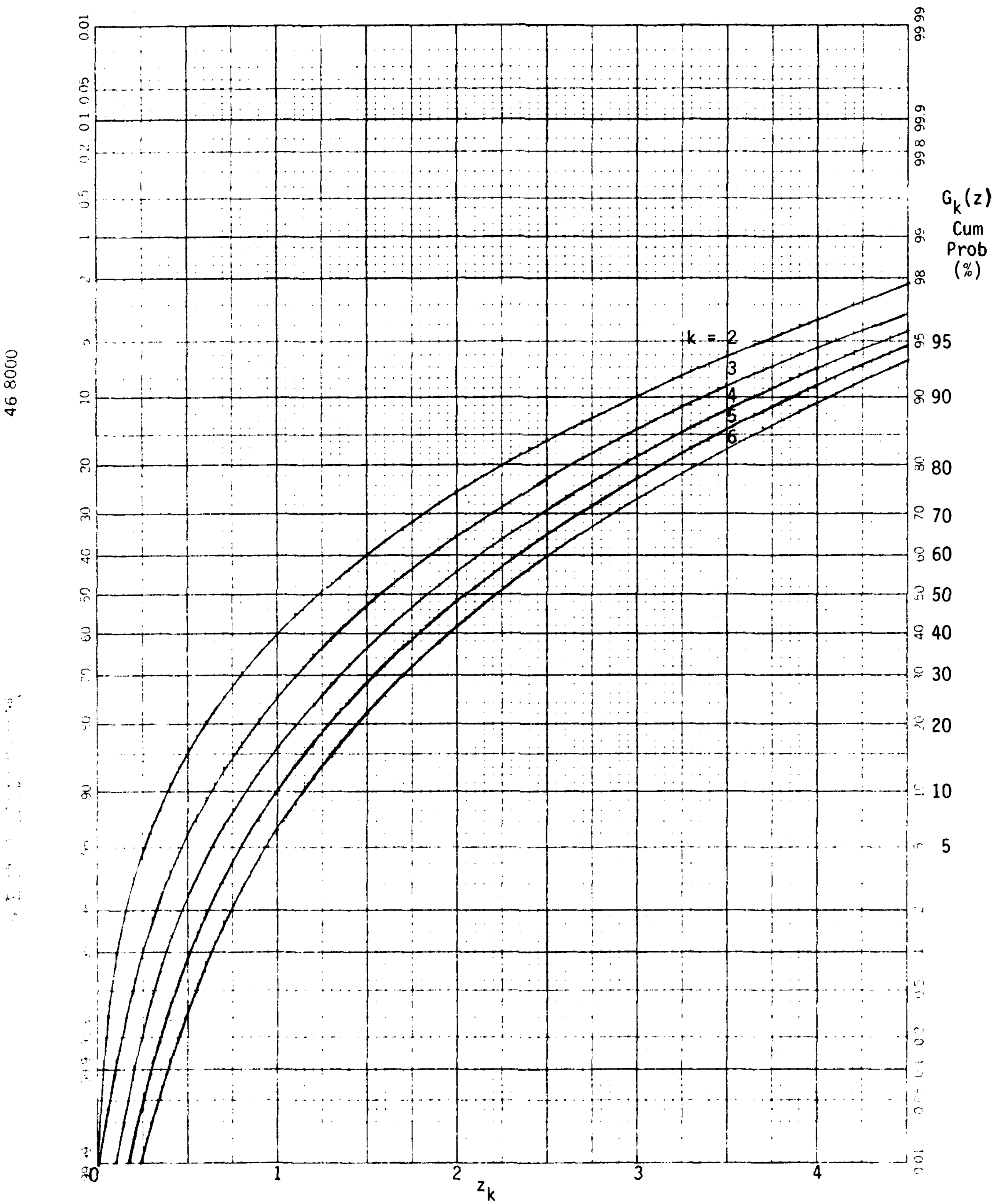


Figure E.1. Cumulative Distribution Functions for the Maximum of k Standardized Exponential Random Variables in a Set

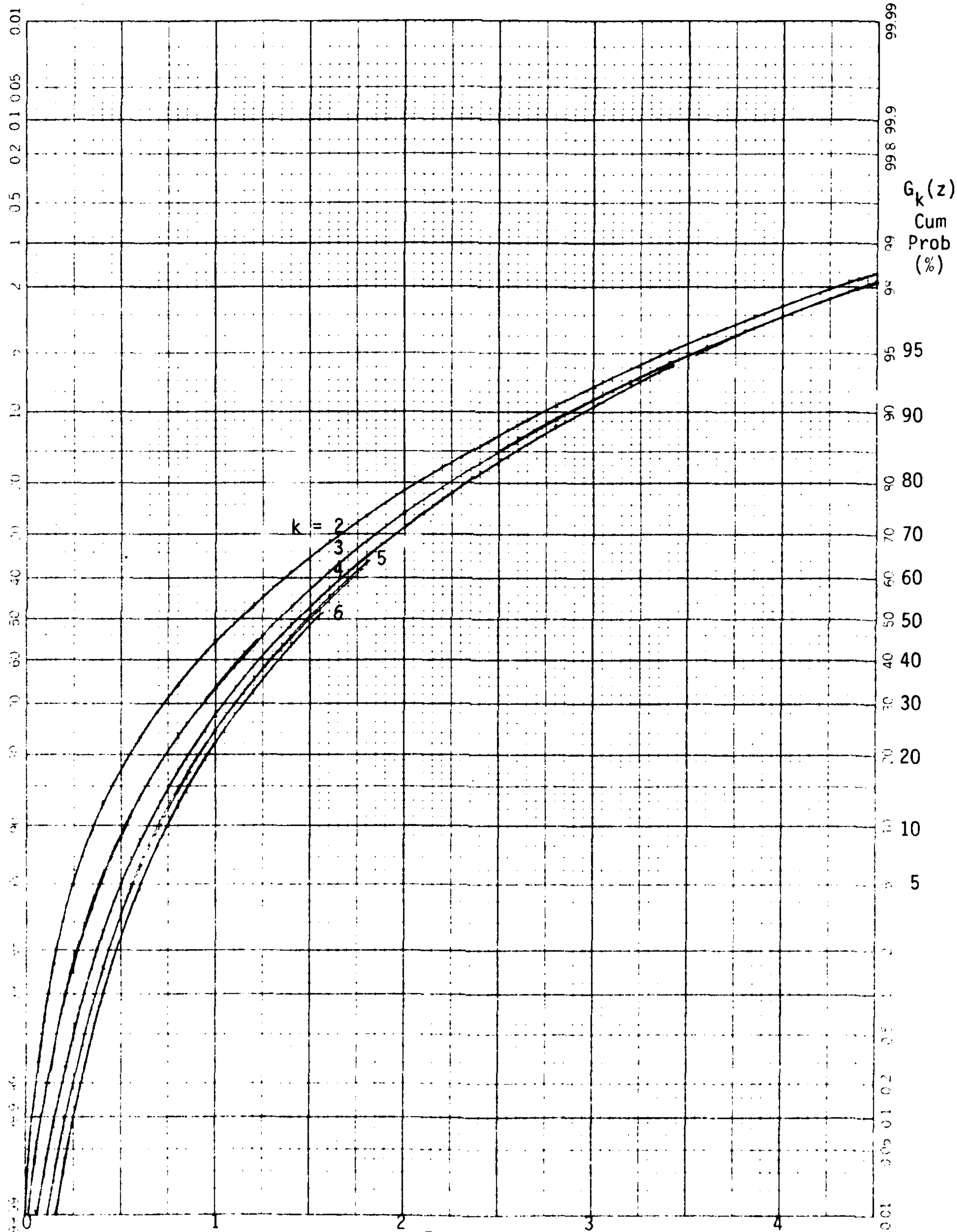


Figure E.2. CDFs for the Max of a Set of k Exponentially Distributed Random Variables with Rate Parameters (λ_i): $\lambda_1 = 1$ and $\lambda_i = 1.2\lambda_{i-1}$, $2 \leq i \leq k$

few activities have much greater mean completion times than the other activities. In this instance the mean value of the completion time for this portion of the network is insensitive to the addition of activities beyond a certain point.

Parametric Analyses

In a fully general situation each of the x_i random variables in the set may have a unique functional form. In such a setting few numerical generalizations about the distribution of max in set (z_k) can be made. However, one can be more restrictive with respect to assumptions with some beneficial consequences. Assume that the functional form of all the x_i is the same, for example, gamma or Weibull. Also assume that the mean value of all the RV's is the same. With these restrictions some interesting parametric analyses can be performed. One analysis of interest is the effect of distribution shape on the mean value of z_k. In the case of the gamma c.d.f., shape is affected by only one parameter -- β -- in the function:

$$F(x) = \int_0^x \beta\lambda(\lambda t)^{\beta-1}e^{-\lambda t}dt/\Gamma(\beta) \quad . \tag{18}$$

As β increases, with fixed mean, the distribution becomes less variable and less positively skewed. In this case the coefficient of variation is $1/\sqrt{\beta}$. One would expect that $E[z_k]$ would decrease with increasing β, since less probability density is associated with large values of x_i. As seen in Table E.7, this expectation is correct. Since our general numerical method was used here, results are presented to only four significant digits. Integration step size was chosen to yield four digits accuracy. This was checked against exact results for the exponential case ($\beta = 1$). A generalization from Table E.7 may be of interest. Over the range of gamma shape parameter shown, there is an approximate geometric decrease in $E[z_k] - 1$ with increasing β. The rate of decrease is observed to be greater for larger values of k. Clearly, in the case of zero variance as β approaches infinity, $E[z_k]$ approaches unity for all k.

In the case where $F(x)$ is Weibull, the single parameter β affects shape:

$$F(x) = 1 - \exp[-(\lambda x)^{\beta}] \quad . \tag{19}$$

Results for this case are shown in Table E.8. With increasing β the coefficient of variation uniformly decreases toward zero, as in the case where x is gamma. However, with increasing β in the Weibull case, the coefficient of skewness decreases thru positive values, becoming negative at $\beta = 3.6$. As β continues to increase, the coefficient of skewness asymptotically approaches -1.14. This behavior (among others) distinguishes the Weibull from the gamma c.d.f. Coefficients of variation and of skewness for the gamma and Weibull distributions are tabulated versus shape parameter in Table E.9. For large values of β, the Weibull is not really comparable with the gamma distribution for reasons given above. However, for values of β in which the Weibull is positively skewed and has the same coefficient of variation as a given gamma distribution, certain results are quite similar. For $\beta = 1$ both distributions are exponential and yield identical values of $E[z_k]$. Consider the non-trivial case in which the coefficient of variation is 1/2: β(gamma) = 4 and β(Weibull) = 2.102. In this case the form of the distribution has little effect upon $E[z_k]$. Results are nearly the same -- within ±0.02 -- for $2 \leq k \leq 6$.

The c.d.f. of z_k is displayed in Figure E.3 for several values of k, given that $F_i(x)$ is Weibull with shape parameter of 2. For this particular distribution the mean of x_i is located at the 54% percentile. Altho the skewness of this choice of $F_i(x)$ is rather small (0.63), the $G_k(z)$ functions are distinctly non-gaussian. (Note that the plots are made on Normal probability paper.) It is noted in passing that the assumption that z_k is Normal is generally grossly wrong.

TABLE E.7

PARAMETRIC ANALYSIS FOR THE MEAN VALUE OF THE MAXIMUM OF A SET OF GAMMA RV'S WITH SHAPE* AS A PARAMETER

$E[x_i] = 1,\ 1 \leq i \leq k$

No. RV's k	$E[z_k]$ for Gamma Shape Parameter:				
	1	2	3	4	5
2	1.500	1.375	1.312	1.273	1.246
3	1.833	1.606	1.498	1.433	1.387
4	2.083	1.774	1.631	1.544	1.486
5	2.283	1.904	1.733	1.630	1.561
6	2.450	2.012	1.816	1.700	1.622
7	2.593	2.102	1.886	1.759	1.673

*Parameter β: $F(x) = \int_0^x \beta\lambda(\lambda t)^{\beta-1} e^{-\lambda t} dt/\Gamma(\beta)$

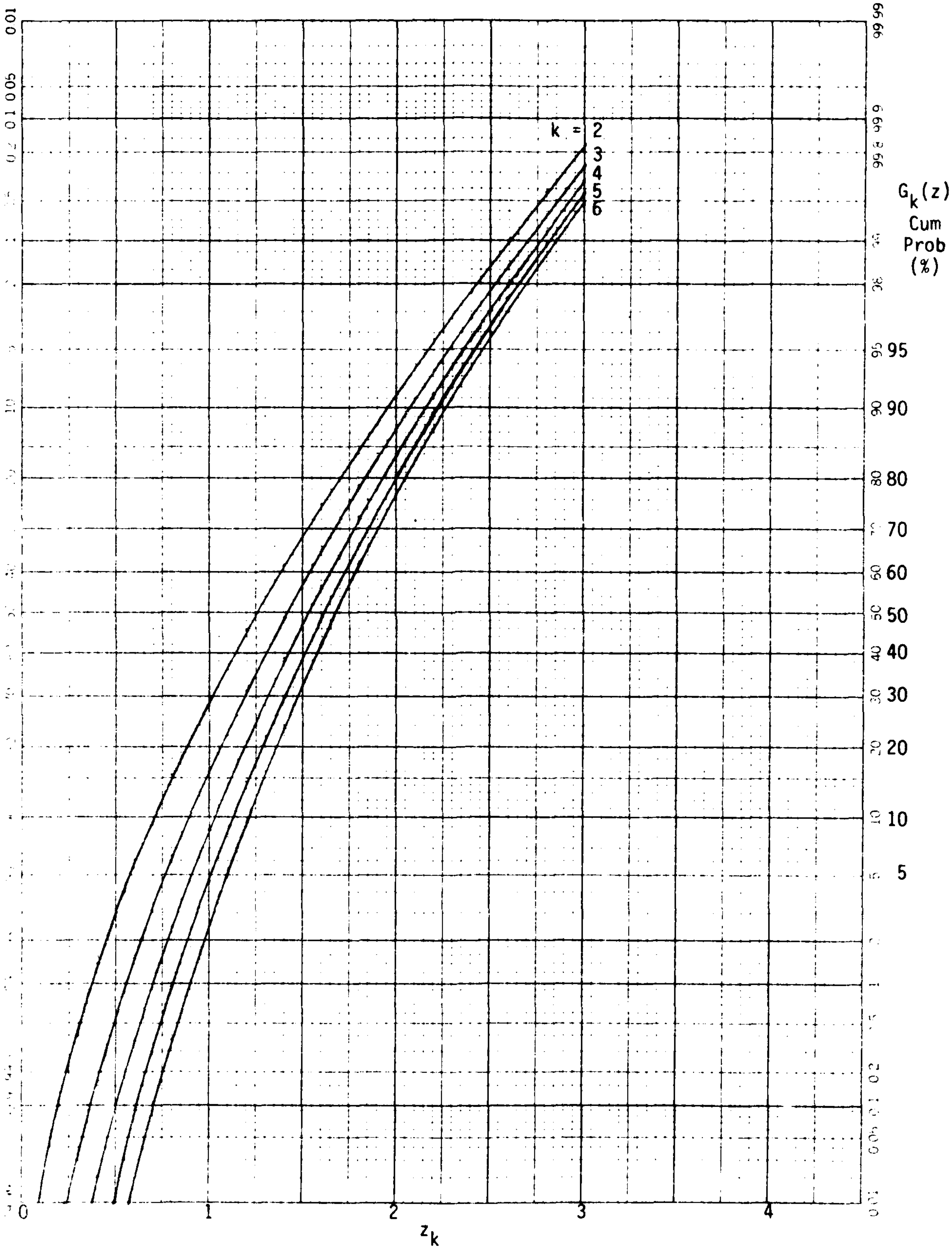


Figure E.3. Cumulative Distribution Functions for the Maximum of k Unity-Mean Weibull (2) Random Variables in a Set

TABLE E.8

PARAMETRIC ANALYSIS FOR THE MEAN VALUE OF THE MAXIMUM OF A SET OF WEIBULL RV'S WITH SHAPE* AS A PARAMETER

$E[x_i] = 1,\ 1 \leq i \leq k$

No. RV's k	$E[z_k]$ for Weibull Shape Parameter:				
	1	2	3	4	5
2	1.500	1.293	1.206	1.159	1.129
3	1.833	1.456	1.312	1.237	1.191
4	2.083	1.567	1.382	1.287	1.230
5	2.283	1.650	1.432	1.323	1.258
6	2.450	1.716	1.472	1.351	1.279
7	2.593	1.771	1.504	1.374	1.297

*Parameter β: $F(x) = 1 - \exp[-(\lambda x)^{\beta}]$

TABLE E.9A

COEFFICIENT OF VARIATION VERSUS SHAPE PARAMETER
FOR GAMMA AND WEIBULL RANDOM VARIABLES

Distribution Type	CV for Shape Parameter:				
	1	2	3	4	5
Gamma	1.0000	0.7071	0.5774	0.5000	0.4472
Weibull	1.0000	0.5227	0.3634	0.2805	0.2291

TABLE E.9B

COEFFICIENT OF SKEWNESS VERSUS SHAPE PARAMETER
FOR GAMMA AND WEIBULL RANDOM VARIABLES

Distribution Type	γ_1 for Shape Parameter:				
	1	2	3	4	5
Gamma	2.0000	1.4142	1.1547	1.0000	0.8944
Weibull	2.0000	0.6311	0.1681	-0.0869	-0.2540

Truncation Effects

It was stated that the intended application of the above analysis is to networking problems where activity times are, ultimately, bounded from above. Thus, the reader may be tempted to challenge the applicability of distributions of x which yield positive but unbounded values. All of the above methods hold for bounded distributions. Only the applications emphasis so far has been on unlimited distributions. This situation can be redressed by examining truncated versions of the distributions previously considered. Suppose that the RV x is defined on the finite domain: $0 \leq x \leq x_u$. If the non-truncated c.d.f. of x is denoted by $F(\beta,\lambda,x)$, the truncated form is given by

$$F'(\beta,\lambda',x) = F(\beta,\lambda',x)/F(\beta,\lambda',x_u) \quad ,$$

for $0 \leq x \leq x_u$. If the untruncated mean value is unity and one wishes the same value of the mean of the truncated RV, the rate parameter, λ, must be adjusted to λ' so that

$$\int_0^{x_u} \overline{F}'(\beta,\lambda',x)dx = 1 \quad .$$

This adjustment to preserve the mean facilitates a comparison between results for truncated and non-truncated c.d.f.'s. In Annex B we display the listing of the computer program which implements truncation for the 2-parameter Weibull family. In the case where the shape parameter $\beta = 1$, the exponential distribution is realized. The mean values of the maximum of a set of truncated exponential RV's were calculated for several values of x_u. The upper truncation point, x_u, was chosen to yield convenient values of $F(\beta,\lambda',x_u)$ such as 0.990 and 0.999, etc. Results are shown in Table E.10. These may be compared with the untruncated results in Table E.1. It is clear that truncation of the x_i values has a much greater effect on the standard deviation of z_k than on the mean of z_k. Specifically, when the upper 1% of the distribution is truncated, the standard deviation of z_7 is reduced to 78% of its untruncated value, whereas the mean of z_7 is reduced to 96% of its. As might be expected, very little difference exists between truncated and untruncated results when only 0.1% of the distribution is truncated.

TABLE E.10

EFFECT OF TRUNCATION OF THE EXPONENTIAL DISTRIBUTION OF x_i ON THE MEAN AND STD DEV OF $z_k = \max_i(x_1, \ldots, x_k)$, GIVEN $E[x_i] = 1$, $1 \leq i \leq k$

Upper Trunc. Point*:	4.8298		6.9559	
k	$E[z_k]$	$SD[z_k]$	$E[z_k]$	$SD[z_k]$
1	1.000	0.928	1.000	0.983
2	1.486	0.998	1.498	1.088
3	1.802	1.007	1.828	1.126
4	2.033	1.000	2.074	1.143
5	2.215	0.987	2.271	1.151
6	2.363	0.972	2.434	1.154
7	2.488	0.956	2.573	1.155

*The upper truncation point (x_t) is the value of x such that the c.d.f. is

$$F(x) = [1 - \exp(-\lambda x)]/q_t, \quad 0 \leq x \leq x_t ,$$

with

$$q_t = 1 - \exp(-\lambda x_t) .$$

The values of q_t for the above truncation points are, respectively, 0.990 and 0.999. Associated values of λ are, respectively, 0.95348 and 0.99309.

Probability Distribution of the kth Largest of n

The primary focus in this annex is on the statistics of the largest positive RV in a set of k RV's, each of which has a unique c.d.f. When all of the RV's in the set are from the same distribution, the problem addressed here is just that of the largest order statistic. Order statistics pertain to the kth ordered RV in an identically distributed set of n. An extensive literature exists relative to order statistics. Gumbel (1958)[8], for example, uses this theory to develop the statistics of extremes. As pointed out by Guenther (1977)[9], the c.d.f. of the kth order statistic from a set of n RV's with common distribution function F(x) is given by

$$G_{k,n}(z) = I_{F(z)}(k, n - k + 1) \quad , \tag{20}$$

where $I_x(a,b)$ is the beta distribution of x with parameters a and b. The beta distribution is expressed as

$$I_x(a,b) = \frac{1}{B(a,b)} \int_0^x u^{a-1}(1 - u)^{b-1} du \quad , \tag{21a}$$

with

$$B(a,b) = \frac{\Gamma(a)\,\Gamma(b)}{\Gamma(a + b)} \quad . \tag{21b}$$

Guenther indicates that it is computationally convenient to obtain $I_x(a,b)$ from an equivalent form of Fisher's F-distribution. A numerical method for evaluating the F-distribution is given on p. 944 of Abramowitz and Stegun (1966)[10].

---

[8] Gumbel, E.J. Statistics of Extremes, Columbia Univ. Press, New York, c. 1958.

[9] Guenther, W.C. "An Easy Method for Obtaining Percentage Points of Order Statistics," Technometrics, Vol. 19, No. 3, pp. 319 - 321, August 1977.

[10] Abramowitz, M. and Stegun, I. Handbook of Mathematical Functions, AMS 55, Nat. Bureau of Standards, August 1966.

Denote the upper tail probability of the F-distribution with argument y and with integer degrees of freedom parameters ν_1 and ν_2 by $Q(y,\nu_1,\nu_2)$. Then, the following relationship can be used relating $I_x(a,b)$ to $Q(y,\nu_1,\nu_2)$:

$$I_x(a,b) = Q\left(\frac{a(1-x)}{bx}, 2b, 2a\right) \quad . \tag{22}$$

Annex B includes a listing of the computer program TEST.K which implements equations (20), (21), and (22) to calculate the distribution of the kth order statistic for a set of n generally-distributed continuous random variables. The distribution function F(x) is calculated in a user-supplied function FUN.CDF. The form of this routine illustrated in Annex B calculates a (optionally) truncated Weibull distribution. The mean and standard deviation of the order statistics are evaluated using the method shown in equations (16) and (17) and employing Simpson's rule. Despite the ease of implementation, little applicability of order statistics is foreseen to networking. This is due to the fact that parallel activities seldom have the same c.d.f. When they (approximately) do, it is quite important to specify whether passage thru the network requires the completion of all of the activities or of a subset. To illustrate this point, consider Figure E.4. A comparison is made between $G_{6,7}(z)$ and $G_{7,7}(z)$ when F(x) is a standardized exponential distribution. It is seen that a remarkable difference exists between mean values of the completion times -- in a network context -- of two situations: (a) 6 activities of 7 must be complete for passage versus (b) all 7 activities must be complete for passage. This illustrates a suprisingly high sensitivity of project completion time to network logic.

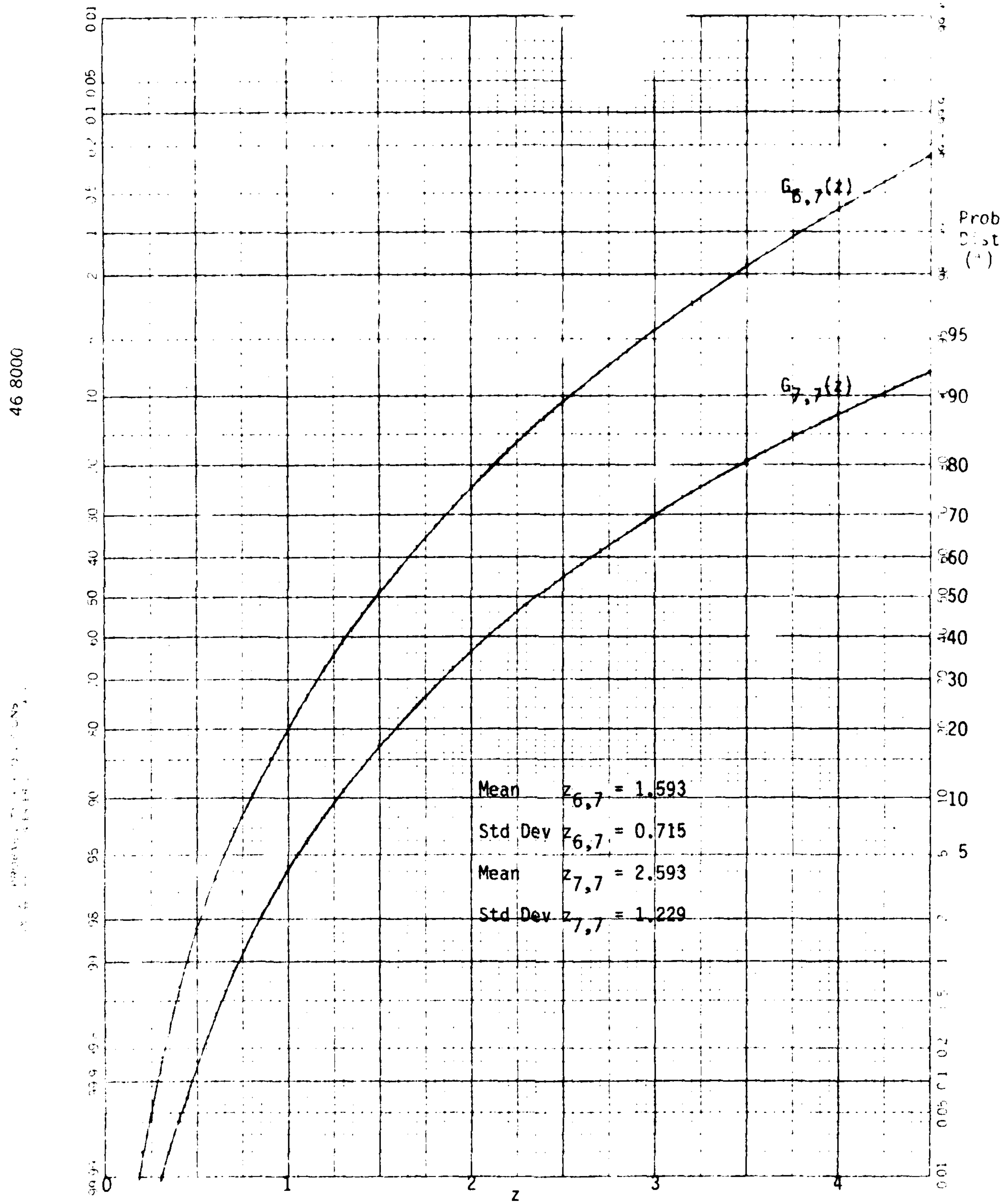


Figure E.4. Comparison of CDFs for the 6th Largest of 7 Versus the Largest of 7 Standardized Exponential RVs

ANNEX B

Computer Source Programs

Two MAIN computer programs are presented in this annex. The first (MAXG) can be used to calculate the mean, standard deviation, and probability distribution function of the largest of k positive, continuous random variables x_i, $1 \leq i \leq k$, each having its own c.d.f. The c.d.f. of x_i is calculated by the function FUN.CDF which accepts as arguments the shape and rate parameters and the value of x_i. As shown, FUN.CDF produces probability values for (optionally) truncated 2-parameter Weibull distributions. Comment code is also provided for calculating the gamma distribution. All supporting routines and functions are supplied in this listing.

The second MAIN program (TEST.K) calculates the distribution function, mean, and standard deviation for the order statistics of a general c.d.f. This general c.d.f. is calculated in the user-supplied function FUN.CDF. As shown FUN.CDF provides values for truncated Weibull distributions, just as this function does for MAXG. The fact that FUN.CDF is identical for these programs is, of course, not necessary. All utility routines are provided for TEST.K. Comment statements in these routines explain their purpose and define input and output arguments.

These programs are written in SIMSCRIPT 2.5 for the PR1ME 750 minicomputer. However, the code does not employ features unique to this computer. Cross reference lists are included with program statements to identify variable type, to tabulate the locations of each variable in the programs, and to facilitate the conversion of programs to another language. Both driver programs are interactive. Program input is read from the terminal and output is displayed at the terminal. No external files are used. Since the output may be lengthy, it is recommended that a COMO file be established to display or print it.

For convenience and without loss of generality, the random variables x_i are considered scaled in dimension so that the largest mean of the x_i is unity. Inputs to MAXG and TEST.K are provided in response to prompting messages sent to the terminal. Thus, MAXG requires as input the max number of random variables in the set, an indication of whether these are defined on a finite or semi-infinite domain (If finite, the upper truncation point must be specified.), the shape parameters of the distribution of the x_i, and the ratio of the mean x_i to the mean of x_1. At the user's option the values of $G_k(z)$ are printed out at intervals sufficient to permit a visually smooth point-to-point plot. Whether or not $G_k(z)$ is printed, the program provides the mean value and standard deviation of z_2, z_3, etc. up to the maximum set size specified.

In the program TEST.K, required inputs are: (a) the cumulative probability associated with the upper truncation point -- if truncation of F(x) is chosen -- (b) the number (n) of identically-distributed RV's in the set, (c) the index (k) of the order statistic wanted, (d) the shape parameter of F(x), and (e) the rate parameter of F(x). Output from this program is the distribution, $G_{k,n}(z)$, of the selected order statistic as well as its mean and standard deviation.

```
1  PREAMBLE ''MAYG
2  NORMALLY MODE IS REAL
3  DEFINE TRUNC.FLAG AS AN INTEGER VARIABLE
4  DEFINE CDF.P, UTP AS REAL VARIABLES
5  DEFINE COMPLETE.GAMMA AS A REAL FUNCTION GIVEN 1 ARGUMENT
6  DEFINE ERLANG.DIST AS A REAL FUNCTION GIVEN 2 ARGUMENTS
7  DEFINE FUN.CDF AS A REAL FUNCTION GIVEN 3 ARGUMENTS
8  DEFINE FUN.PDF AS A REAL FUNCTION GIVEN 4 ARGUMENTS
9  END ''PREAMBLE
```

C R O S S - R E F E R E N C E

NAME	TYPE			MODE	LINE NUMBERS OF REFERENCES
CDF.P	GLOBAL VARIABLE	ARR	4	DOUBLE	4
COMPLETE.GAMMA	ROUTINE			DOUBLE	5
ERLANG.DIST	ROUTINE			DOUBLE	6
FUN.CDF	ROUTINE			DOUBLE	7
FUN.PDF	ROUTINE			DOUBLE	8
TRUNC.FLAG	GLOBAL VARIABLE	ARR	5	INTEGER	3
UTP	GLOBAL VARIABLE	ARR	2	DOUBLE	4

```
MAIN **MAXG
**
**PROGRAM CALCULATES THE FIRST AND SECOND STATISTICAL MOMENTS OF THE
** PROBABILITY DISTRIBUTION OF Z(K), THE MAXIMUM OF A SET OF K GENERALLY
** DISTRIBUTED RANDOM VARIABLES, X(I), I=1, ..., K. THE USER MUST SPECIFY
**THE FUNCTIONAL FORM OF THE C.D.F. OF X(I) AND OF THE PARAMETERS, FOR
** ALL I.  THIS FUNCTION IS CALCULATED IN THE ROUTINE FUN.CDF.  THE ROUTINE
**FUN.PDF CALCULATES THE DIFFERENTIAL PROBABILITY FOR THE INTERVAL:
**  Z - DELZ TO Z.  PARAMETERS ARE CHOSEN SO THAT THE MAXIMUM OVER I OF
**THE AVG X(I) IS UNITY.
**
**
DEFINE ANSWER,DIST.TYPE AS TEXT VARIABLES
DEFINE FLAGCDF,I,J,K,L,M,MAX,MAXN,N AS INTEGER VARIABLES
DEFINE BTV,CDFV,PDFV,AEZV,LAV,VZV,XCDFV,UTPV AS REAL, 1-DIMENSIONAL ARRAYS
     LET MAXN=5000
     LET DIST.TYPE = "WEIBULL"
     RESERVE PDFV(*) AS MAXN
     RESERVE XCDFV(*) AS MAXN
     PRINT 1 LINE THUS
INPUT THE MAXIMUM NUMBER OF RANDOM VARIABLES IN THE SET.
     READ MAX
     LET MAX=MAX.F(2,MAX)
     LET MAX=MIN.F(20,MAX)
     RESERVE BTV(*),LAV(*),AEZV(*),VZV(*),UTPV(*) AS MAX
     PRINT 1 LINE WITH DIST.TYPE
     THUS
DO YOU WANT THE ********* DISTRIBUTION OF THE RV'S TRUNCATED?  (YES OR NO).
     READ ANSWER
     IF SUBSTR.F(ANSWER,1,1) = "Y"
          LET TRUNC.FLAG=1
          PRINT 1 LINE THUS
INPUT THE CUMULATIVE PROB ASSOCIATED WITH THE UPPER TRUNCATION POINT.
          READ CDF.P
          LET HOLD=-LOG.E.F(1.0-CDF.P)
     OTHERWISE
          LET TRUNC.FLAG=0
     ALWAYS
     FOR I=1 TO MAX DO
          PRINT 1 LINE WITH DIST.TYPE, I
          THUS
INPUT THE ******** SHAPE PARAMETER OF THE ** TH VARIABLE IN THE SET.
          READ BETA
          LET BTV(I)=MAX.F(1.0,BETA)
 **       LET LAV(I)=BTV(I) FOR NON-TRUNC GAMMA DIST MEAN=1
          LET LAV(I)=COMPLETE.GAMMA(1.0+1.0/BTV(I)) **FOR NON-TRUNC WEI MEAN=1
          IF TRUNC.FLAG=1
               LET BETA=BTV(I)
               LET LAMBDA=LAV(I)
               LET XTRUNC=HOLD**(1.0/BETA)/LAMBDA
               CALL WEI.TRUNC GIVEN BETA, LAMBDA, XTRUNC YIELDING ETX, SDTX
               LET LAV(I)=ETX*LAV(I)
               LET UTPV(I)=HOLD**(1.0/BETA)/LAV(I)
          ALWAYS
     LOOP **OVER (I) RV'S
     LET P1=BTV(1)
     LET P2=LAV(1)
```

```
        LET UTF=UTFV(1)
        FOR I=2 TO MAX DO
            PRINT 1 LINE WITH I
            THUS
INPUT THE RATIO OF THE FIRST (LARGEST) MEAN TO THE ** TH MEAN VALUE.
            READ RATIO
            LET LAV(I)=LAV(I)*RATIO
            IF TRUNC.FLAG=1
                LET UTFV(I)=UTFV(I)/RATIO
            ALWAYS
        LOOP **OVER (I) RV'S IN THE SET
        SKIP 1 LINE
        PRINT 1 LINE THUS
DO YOU WANT TO DISPLAY THE C.D.F.'S OF THE MAX IN SET?  (YES OR NO).
        READ ANSWER
        IF SUBSTR.F(ANSWER,1,1) = "Y"
            LET FLAGCDF=1
        OTHERWISE
            LET FLAGCDF=?
        ALWAYS
        IF TRUNC.FLAG=1
            LET ANSWER = "TRUNCATED"
        OTHERWISE
            LET ANSWER = "NON-TRUNC"
        ALWAYS
        SKIP 2 LINES
        PRINT 6 LINES WITH ANSWER,DIST.TYPE
        THUS
PARAMETERS OF PROBABILITY DISTRIBUTIONS OF THE RANDOM VARIABLES IN THE SET

DISTRIBUTION TYPE: *********  *********

NUMBER   SHAPE PARM   RATE PARM      MEAN     STD DEV    COEF VAR  TRUNC PT
---------------------------------------------------------------------------
        FOR I=1 TO MAX DO
  **        LET MEAN=BTV(I)/LAV(I)          FOR GAMMA RV'S
  **        LET SD=SQRT.F(BTV(I))/LAV(I)  FOR GAMMA RV'S
            IF TRUNC.FLAG=1
                LET BETA=BTV(I)
                LET LAMBDA=LAV(I)
                LET XTRUNC=UTFV(I)
                CALL WEI.TRUNC GIVEN BETA, LAMBDA, XTRUNC YIELDING MEAN, SDTX
                LET VAR=SDTX**2+MEAN**2
            OTHERWISE
                LET MEAN=COMPLETE.GAMMA(1.0+1.0/BTV(I))/LAV(I)
                LET VAR=COMPLETE.GAMMA(1.0+2.0/BTV(I))/LAV(I)**2
            ALWAYS
            IF I=1
                LET VZV(1)=VAR
            ALWAYS
            LET VAR=VAR-MEAN**2
            LET SD=SQRT.F(VAR)
            LET CV=SD/MEAN
            IF TRUNC.FLAG=1
                PRINT 1 LINE WITH I,BTV(I),LAV(I),MEAN,SD,CV,UTPV(I)
                THUS
   *      **.***      **.****     *.****     *.****     *.****     *.****
```

```
          OTHERWISE
              PRINT 1 LINE WITH I,BTV(I),LAV(I),MEAN,SD,CV
              THUS
*      **.***     **.****     *.*****     *.****     *.****
          ALWAYS
      LOOP ''OVER ALL RV'S
      PRINT 2 LINES THUS
------------------------------------------------------------------------
      LET AEZV(1)=1.0
''    LET VZV(1)=BTV(1)/LAV(1)**2+AEZV(1)**2 FOR GAMMA DIST
      LET DELZ=0.005
      LET Z=DELZ
''
''INITIALIZE THE P.D.F. AND C.D.F. OF Z(K) FOR K = 1.
''
      LET PDFV(1)=FUN.CDF(P1,P2,DELZ)
      LET XCDFV(1)=PDFV(1)
      FOR N=2 TO MAXN DO
          LET Z=N*DELZ
          LET CDF=FUN.CDF(P1,P2,Z)
          LET XCDFV(N)=CDF
          LET PDFV(N)=XCDFV(N)-XCDFV(N-1)
      LOOP ''OVER N
''
''PRINT HEADINGS FOR C.D.F. OF Z(2).
''
      SKIP 2 LINES
      IF FLAGCDF=1
          PRINT 6 LINES
          THUS
PROBABILITY DISTRIBUTION OF THE MAXIMUM OF TWO POSITIVE RANDOM VARIABLES
------------------------------------------------------------------------
ARGU-      PROB        PROB
MENT       DIST        DENS
------------------------------------------------------------------------
      ALWAYS
      LET P1=BTV(2)
      LET P2=LAV(2)
      LET UTP=UTFV(2)
      LET ASUM=0.
      LET VSUM=0.0
      LET XSUM=1.0
      LET XXSUM=0.0
      FOR N=1 TO MAXN DO
          LET Z=DELZ*N
          IF MOD.F(N,2)=0
              LET COEF=2.0
          OTHERWISE
              LET COEF=4.0
          ALWAYS
''
''GET INTEGRAL TO OBTAIN ALTERNATE ESTIMATE OF EXPECTATION OF Z(2).
''
          LET CDFX=FUN.CDF(P1,P2,Z)
          LET HOLD=XCDFV(N)*(1.0-CDFX)
```

```
            ADD COEF*HOLD TO ASUM
            ADD COEF*Z*HOLD TO VSUM
    ''
    ''UPDATE P.D.F. AND C.D.F. OF Z(2).
    ''
            LET PDFV(N)=XCDFV(1)*FUN.PDF(F1,P2,Z,DELZ)+CDFX*PDFV(N)
            LET XCDFV(N)=XCDFV(N)*CDFX
            LET UPPER=1.0-XCDFV(N)
            ADD COEF*UPPER TO XSUM
            ADD COEF*Z*UPPER TO XXSUM
            IF MOD.F(N,20)=0 ''PRINT THE C.D.F.
                IF FLAGCDF=1
                    PRINT 1 LINE WITH Z,XCDFV(N),PDFV(N)/DELZ
                    THUS
**.***    *.*****     ***********
                ALWAYS
            ALWAYS
            IF XCDFV(N) > 0.99998
                GO TO LO
            OTHERWISE
        LOOP ''OVER (N) TIME STEPS
    'LO'PRINT 2 LINES THUS
--------- ---------------------------------------------------------------
        LET AEZV(2)=AEZV(1)+ASUM*DELZ/3.0
        LET VZV(2)=VZV(1)+2.0*VSUM*DELZ/3.0
        LET ASD=SQRT.F(VZV(2)-AEZV(2)**2)
        LET XSUM=XSUM*DELZ/3.0
        LET XXSUM=2.0*DELZ/3.0*XXSUM-XSUM**2
        PRINT 4 LINES WITH AEZV(2),ASD,XSUM,SQRT.F(XXSUM)
        THUS
MEAN OF Z(2) ___   **.**** CALCULATED BY RECURSIVE EXPECTATION WITH XCDF
STD DEV Z(2) ___   **.**** CALCULATED BY RECURSIVE EXPECTATION WITH XCDF
MEAN OF Z(2) ___   **.**** CALCULATED BY INTEGRATION OF EXACT CDF
STD DEV Z(2) ___   **.**** CALCULATED BY INTEGRATION OF EXACT CDF
    SKIP 2 LINES
    ''
    ''RESULTS FOR THE MAX OF K RANDOM VARIABLES, Z(K), WITH K > 2.
    ''
    FOR K=3 TO MAX DO
    ''
    ''PRINT HEADINGS FOR THE C.D.F. OF Z(K).
    ''
        SKIP 2 LINES
        IF FLAGCDF=1
            PRINT 6 LINES WITH K
            THUS
PROBABILITY DISTRIBUTION OF THE MAXIMUM OF ** POSITIVE RANDOM VARIABLES

ARGU-      PROB       PROB -----------------------------------------------
MENT       DIST       DENS

-----------ALWAYS-------------------------------------------------------
        LET P1=BTV(K)
        LET P2=LAV(K)
        LET UTP=UTPV(K)
```

```
      LET ASUM=0.
      LET VSUM=0.0
      LET XSUM=1.0
      LET XXSUM=0.
      FOR N=1 TO MAXN DO
         LET Z=DELZ*N
         IF MOD.F(N,2)=0
            LET COEF=2.0
         OTHERWISE
            LET COEF=4.0
         ALWAYS
         LET CDFX=FUN.CDF(P1,P2,Z)
         LET HOLD=XCDFV(N)*(1.0-CDFX)
         ADD COEF*HOLD TO ASUM
         ADD COEF*Z*HOLD TO VSUM
         LET PDFV(N)=XCDFV(N)*FUN.PDF(P1,P2,Z,DELZ)+CDFX*PDFV(N)
         LET XCDFV(N)=XCDFV(N)+CDFX
         LET UPPER=1.0-XCDFV(N)
         ADD COEF*UPPER TO XSUM
         ADD COEF*Z*UPPER TO XXSUM
         IF MOD.F(N,20)=0 AND FLAGCDF=1 **PRINT EXACT VALUES
            PRINT 1 LINE WITH Z,XCDFV(N),PDFV(N)/DELZ
            THUS
**.***    *.*****    ***********
         ALWAYS
         IF XCDFV(N) > 0.99998
            GO TO L2
         OTHERWISE
      LOOP **OVER (N) TIME STEPS
   *L2*IF FLAGCDF=1
         PRINT 2 LINES THUS
------------------------------------------------------------------------
      ALWAYS
      LET AEZV(K)=AEZV(K-1)+ASUM*DELZ/3.0
      LET VZV(K)=VZV(K-1)+2.0*VSUM*DELZ/3.0
      LET ASD=SQRT.F(VZV(K)-AEZV(K)**2)
      LET XSUM=XSUM*DELZ/3.0
      LET XXSUM=2.0*DELZ/3.0*XXSUM-XSUM**2
      PRINT 4 LINES WITH K,AEZV(K),K,ASD,K,XSUM,K,SQRT.F(XXSUM)
      THUS
MEAN OF Z(**) ___   **.**** CALCULATED BY RECURSIVE EXPECTATION WITH XCDF
STD DEV Z(**) ___   **.**** CALCULATED BY RECURSIVE EXPECTATION WITH XCDF
MEAN OF Z(**) ___   **.**** CALCULATED BY INTEGRATION OF EXACT CDF
STD DEV Z(**) ___   **.**** CALCULATED BY INTEGRATION OF EXACT CDF
   LOOP **OVER (K) RV'S
      SKIP 2 LINES
      STOP
   END **MAXG
```

```
MAIN ROUTINE                              CACI SIMSCRIPT II.5 FOR PRIME SYSTEMS, RELEASE 2.1            PAGE    7
OPTIONS = SEQUENCE,ID,SUBCHK,XREF,NOEXPLIST,TRACE?                 1? DEC 1984 1?:49:11

                              C R O S S - R E F E R E N C E

 NAME                      TYPE                                  MODE      LINE NUMBERS OF REFERENCES

AEZV                     RECURSIVE VARIABLE   WORD  15  (1-D) DOUBLE     15     24*    1?8    171*   173    176    225*
                                                                         227    23?
ANSWER                   RECURSIVE VARIABLE   WORD   1        TEXT       13     27     28     66     67     73     75
                                                                         78
ASD                      RECURSIVE VARIABLE   WORD  80        DOUBLE     173    176    227    23?
ASUM                     RECURSIVE VARIABLE   WORD  66        DOUBLE     134    150*   171    194    207*   225
BETA                     RECURSIVE VARIABLE   WORD  33        DOUBLE     39     40     44     46     47     49     84
                                                                         87
BTV                      RECURSIVE VARIABLE   WORD  12  (1-D) DOUBLE     15     24*    40     42     44     52     84
                                                                         90     91     1?0    1?3    131    191
CDF                      RECURSIVE VARIABLE   WORD  63        DOUBLE     119    120
CDF.P                    GLOBAL VARIABLE      ARR    4        DOUBLE     31     32
CDFV                     RECURSIVE VARIABLE   WORD  13  (1-D) DOUBLE     15
CDFX                     RECURSIVE VARIABLE   WORD  76        DOUBLE     148    149    155    156    205    206    209
                                                                         210
COEF                     RECURSIVE VARIABLE   WORD  74        DOUBLE     141    143    150    151    158    159    201
                                                                         2?3    207    2?8    212    213
COMPLETE.GAMMA           ROUTINE                              DOUBLE     42     90     91
CV                       RECURSIVE VARIABLE   WORD  57        DOUBLE     98     100    103
DELZ                     RECURSIVE VARIABLE   WORD  59        DOUBLE     110    111    115    118    139    155    162
                                                                         171    172    174    175    199    209    215
                                                                         225    226    228    229
DIST.TYPE                RECURSIVE VARIABLE   WORD   2        TEXT       13     17     25     37     78
ETX                      RECURSIVE VARIABLE   WORD  39        DOUBLE     47     48
FLAGCDF                  RECURSIVE VARIABLE   WORD   3        INTEGER    14     68     70     127    161    187    214
                                                                         222
FUN.CDF                  ROUTINE                              DOUBLE     115    119    148    205
FUN.PDF                  ROUTINE                              DOUBLE     155    2?9
HOLD                     RECURSIVE VARIABLE   WORD  31        DOUBLE     32     46     49     149    150    151    206
                                                                         2?7    208
I                        RECURSIVE VARIABLE   WORD   4        INTEGER    14     36*    37     4?     42*    44     45
                                                                         48*    49*    55*    56     59*    61*    80*
                                                                         84     85     86     90*    91*    93     10?*
                                                                         103*
J                        RECURSIVE VARIABLE   WORD   5        INTEGER    14
K                        RECURSIVE VARIABLE   WORD   6        INTEGER    14     182*   188    191    192    193    225*
                                                                         226*   227*   23?*
L                        RECURSIVE VARIABLE   WORD   7        INTEGER    14
L?                       LABEL                                           167    170
L2                       LABEL                                           219    222
LAMBDA                   RECURSIVE VARIABLE   WORD  35        DOUBLE     45     46     47     85     87
LAV                      RECURSIVE VARIABLE   WORD  16  (1-D) DOUBLE     15     24*    42     45     48*    49     53
                                                                         59*    85     90     91     100    103    132
                                                                         192
LOG.E.F                  ROUTINE                              DOUBLE     32
M                        RECURSIVE VARIABLE   WORD   8        INTEGER    14
MAX                      RECURSIVE VARIABLE   WORD   9        INTEGER    14     21     22*    23*    24*    36     55
                                                                         80     182
MAX.F                    ROUTINE                              INTEGER    22     40
MAXN                     RECURSIVE VARIABLE   WORD  10        INTEGER    14     16     18     19     117    138    198
MEAN                     RECURSIVE VARIABLE   WORD  51        DOUBLE     87     88     9?     96     98     1??    103
MIN.F                    ROUTINE                              INTEGER    23
MOD.F                    ROUTINE                              INTEGER    140    160    200    214
```

```
N             RECURSIVE VARIABLE   WORD  11         INTEGER   14   117*  118   120   121*  139*  139
                                                              140  149   155*  156*  157   160   162*
                                                              166  198*  199   200   206   209*  210*
                                                              211  214   215*  218
P1            RECURSIVE VARIABLE   WORD  43         DOUBLE    52   115   119   131   148   155   191
                                                              205  205
P2            RECURSIVE VARIABLE   WORD  45         DOUBLE    53   115   119   132   148   155   192
                                                              205  205
PDFV          RECURSIVE VARIABLE   WORD  14  (1-D)  DOUBLE    15   18*   115   116   121   155*  162
                                                              209* 215
RATIO         RECURSIVE VARIABLE   WORD  47         DOUBLE    58   59    61
SD            RECURSIVE VARIABLE   WORD  55         DOUBLE    97   98    100   103
SDTX          RECURSIVE VARIABLE   WORD  41         DOUBLE    47   87    86
SQRT.F        ROUTINE                               DOUBLE    97   173   176   227   230
SUBSTR.F      ROUTINE                               TEXT      28   67
TRUNC.FLAG    GLOBAL VARIABLE      ARR    6         INTEGER   29   34    43    60    72    83    99
UIB.W         IMPLIED SUBSCRIPT    SYS    4         INTEGER   64   77    126   178   186   233
UPPER         RECURSIVE VARIABLE   WORD  78         DOUBLE    157  158   159   211   212   213
UTP           GLOBAL VARIABLE      ARR    2         DOUBLE    54   133   193
UTPV          RECURSIVE VARIABLE   WORD  15  (1-D)  DOUBLE    15   24*   49    54    61*   86    100
                                                              133  193
VAR           RECURSIVE VARIABLE   WORD  53         DOUBLE    88   91    94    96*   97
VSUM          RECURSIVE VARIABLE   WORD  68         DOUBLE    135  151*  172   195   208*  226
VZV           RECURSIVE VARIABLE   WORD  17  (1-D)  DOUBLE    15   24*   94    172*  173   226*  227
WEI.TRUNC     ROUTINE                               INTEGER   47   87
XCDFV         RECURSIVE VARIABLE   WORD  18  (1-D)  DOUBLE    15   19*   116   120   121*  149   155
                                                              156* 157   162   166   206   209   210*
                                                              211  215   218
XSUM          RECURSIVE VARIABLE   WORD  70         DOUBLE    136  158*  174*  175   176   196   212*
                                                              228* 229   230
XTRUNC        RECURSIVE VARIABLE   WORD  37         DOUBLE    46   47    86    87
XXSUM         RECURSIVE VARIABLE   WORD  72         DOUBLE    137  159*  175*  176   197   213*  229*
                                                              230
Z             RECURSIVE VARIABLE   WORD  61         DOUBLE    111  118   119   139   148   151   155
                                                              159  162   199   205   208   209   213
                                                              215
```

```
 1  FUNCTION FUN.CDF (P1, P2, T)
 2  ''
 3  ''FUNCTION CALCULATES THE C.D.F. OF A GENERAL, POSITIVE RANDOM VARIABLE.
 4  ''THE PARAMETERS OF THE DISTRIBUTION ARE P1 AND P2.  THE RV ARGUMENT IS T.
 5  ''
 6  ''THE VARIABLES TRUNC.FLAG, CDF.P, AND UTP ARE GLOBAL.
 7  ''
 8      DEFINE N AS AN INTEGER VARIABLE
 9  ''  LET N=MAX.F(1,TRUNC.F(P1))
10      LET ARG=P2*T
11  ''  RETURN WITH ERLANG.DIST( ARG, N)
12      IF TRUNC.FLAG=1
13          IF T < UTP
14              RETURN WITH (1.0-EXP.F(-(ARG)**P1))/CDF.P
15          OTHERWISE
16          RETURN WITH 1.0
17      OTHERWISE
18      RETURN WITH 1.0-EXP.F(-(ARG)**P1)
19  END ''FUN.CDF
```

C R O S S - R E F E R E N C E

NAME	TYPE			MODE	LINE NUMBERS OF REFERENCES		
ARG	RECURSIVE VARIABLE	WORD	2	DOUBLE	10	14	18
CDF.P	GLOBAL VARIABLE	ARR	4	DOUBLE	14		
EXP.F	ROUTINE			DOUBLE	14	18	
FUN.CDF	ROUTINE			DOUBLE	1		
N	RECURSIVE VARIABLE	WORD	1	INTEGER	8		
P1	ARGUMENT	NO.	1	DOUBLE	1	14	18
P2	ARGUMENT	NO.	2	DOUBLE	1	10	
T	ARGUMENT	NO.	3	DOUBLE	1	10	13
TRUNC.FLAG	GLOBAL VARIABLE	ARR	6	INTEGER	12		
UTP	GLOBAL VARIABLE	ARR	2	DOUBLE	13		

```
1  FUNCTION FUN.PDF (P1, P2, T, DT)
2  ''
3  ''FUNCTION CALCULATES THE DIFFERENTIAL PROBABILITY IN THE INTERVAL: (T-DT,T).
4  ''
5      IF T < DT
6          RETURN WITH 0..
7      OTHERWISE
8      RETURN WITH FUN.CDF(P1,P2,T) - FUN.CDF(P1,P2,T-DT)
9  END ''FUN.PDF
```

C R O S S - R E F E R E N C E

NAME	TYPE			MODE	LINE NUMBERS OF REFERENCES		
DT	ARGUMENT	NO.	4	DOUBLE	1	5	8
FUN.CDF	ROUTINE			DOUBLE	8*		
FUN.PDF	ROUTINE			DOUBLE	1		
P1	ARGUMENT	NO.	1	DOUBLE	1	8*	
P2	ARGUMENT	NO.	2	DOUBLE	1	8*	
T	ARGUMENT	NO.	3	DOUBLE	1	5	8*

```
OPTIONS = SEQUENCE,TO,SUBCHK,XREF,ACEXPLIST,TERSE

 1  FUNCTION ERLANG.DIST(X,N)
 2  ''
 3  ''FUNCTION COMPUTES THE CUMULATIVE PROBABILITY FOR AN ERLANG DISTRIBUTION
 4  ''WITH STANDARDIZED ARGUMENT X AND SHAPE PARAMETER N.
 5  ''
 6  DEFINE I AND N AS INTEGER VARIABLES
 7      LET EXPX=EXP.F(-X)
 8      IF N=1
 9          RETURN WITH 1.0-EXPX
10      OTHERWISE
11      LET FACTORIAL=1.0
12      LET ARG=1.0
13      LET SUM=1.0
14  FOR I=1 TO N-1 DO
15      LET FACTORIAL=I*FACTORIAL
16      LET ARG=X*ARG
17      ADD ARG/FACTORIAL TO SUM
18  LOOP ''OVER I
19      RETURN WITH 1.0-EXPX*SUM
20  END ''ERLANG.DIST
```

C R O S S - R E F E R E N C E

NAME	TYPE			MODE	LINE NUMBERS OF REFERENCES			
ARG	RECURSIVE VARIABLE	WORD	6	DOUBLE	12	16*	17	
ERLANG.DIST	ROUTINE			DOUBLE	1			
EXP.F	ROUTINE			DOUBLE	7			
EXPX	RECURSIVE VARIABLE	WORD	2	DOUBLE	7	9	19	
FACTORIAL	RECURSIVE VARIABLE	WORD	4	DOUBLE	11	15*	17	
I	RECURSIVE VARIABLE	WORD	1	INTEGER	6	14*	15	
N	ARGUMENT	NO.	2	INTEGER	1	6	8	14
SUM	RECURSIVE VARIABLE	WORD	8	DOUBLE	13	17*	19	
X	ARGUMENT	NO.	1	DOUBLE	1	7	16	

```
FUNCTION COMPLETE.GAMMA(XX)
''
'''PROGRAM CALCULATES THE COMPLETE GAMMA FUNCTION WITH SINGLE REAL ARGUMENT XX.
'''METHOD: THE RECURSION RELATION AND POLYNOMIAL APPROXIMATION IS TAKEN
'''FROM:  C. HASTINGS, JR, "APPROXIMATIONS FOR DIGITAL COMPUTERS,"
'''PRINCETON UNIV. PRESS, 1955.
''
        IF XX > 57.0
            GO TO L130
        OTHERWISE
'L5'    LET X = XX
        LET ERR = 0.000001
        LET GAMMA = 1.0
        IF X LE 2.0
            GO TO L50
        OTHERWISE
            GO TO L15
'L10'   IF X LE 2.0
            GO TO L110
        OTHERWISE
'L15'   SUBTRACT 1.0 FROM X
        LET GAMMA = GAMMA * X
        GO TO L10
'L50'   IF X = 1.0
            GO TO L120
        OTHERWISE
        IF X > 1.0
            GO TO L110
        OTHERWISE
'L60'   IF X > ERR
            GO TO L80
        OTHERWISE
        LET Y = REAL.F(TRUNC.F(X))-X
        IF ABS.F(Y) LE ERR
            GO TO L130
        OTHERWISE
        IF Y+ERR GE 1.0
            GO TO L130
        OTHERWISE
'L70'   IF X > 1.0
            GO TO L110
        OTHERWISE
'L80'   LET GAMMA = GAMMA / X
        ADD 1.0 TO X
        GO TO L70
'L110'  LET Y = X - 1.0
        LET GY = 1.0+Y*(-0.57710166+Y*(0.98585399+Y*(-0.87642182+Y*(0.83282120
        +Y*(-0.56847290+Y*(0.25482049+Y*(-0.05149930)))))))
        LET GAMMA = GAMMA * GY
'L120'  RETURN WITH GAMMA
'L130'  PRINT 1 LINE WITH XX THUS
ERROR IN COMPLETE.GAMMA.  ARGUMENT = ***.*****
        STOP
END''OF COMPLETE.GAMMA
```

```
ROUTINE COMPLETE.GAMMA                 CACI SIMSCRIPT II.5 FOR PRIME SYSTEMS, RELEASE 2.1          PAGE   13
OPTIONS = SEQUENCE,TE,SUBCHK,XREF,NOEXFLIST,TRACE                    1  DEC 1984 1 :4 :11

                         C R O S S - R E F E R E N C E

NAME                     TYPE                                 MODE      LINE NUMBERS OF REFERENCES

ABS.F                    ROUTINE                              INTEGER    34
COMPLETE.GAMMA           ROUTINE                              DOUBLE      1
ERR                      RECURSIVE VARIABLE   WORD     3      DOUBLE     12    3     34    37
GAMMA                    RECURSIVE VARIABLE   WORD     5      DOUBLE     13    22*   43*   49*   50
GY                       RECURSIVE VARIABLE   WORD     9      DOUBLE     47    49
L10                      LABEL                                           18    23
L110                     LABEL                                           19    28    41    46
L120                     LABEL                                           25    50
L130                     LABEL                                            9    35    38    51
L15                      LABEL                                           17    21
L5                       LABEL                                           15    24
L6                       LABEL                                           11
L60                      LABEL                                           30
L70                      LABEL                                           40    45
L8                       LABEL                                           31    43
REAL.F                   ROUTINE                              INTEGER    33
TRUNC.F                  ROUTINE                              INTEGER    33
X                        RECURSIVE VARIABLE   WORD     1      DOUBLE     11    14    18    21*   22    24    27
                                                                         30    33*   4     43    44*   46
XX                       ARGUMENT             NO.      1      DOUBLE      1     8    11    51
Y                        RECURSIVE VARIABLE   WORD     7      DOUBLE     33    34    37    46    47*   48*
```

```
ROUTINE WEI.TRUNC GIVEN BETA, LAMBDA, XMAX YIELDING ETX, SDTX
''
'' PROGRAM CALCULATES THE TRUNCATED MEAN AND STD DEV OF A WEIBULL DIST OF
'' THE FORM: 1 - EXP( -(LAMBDA*X)**BETA ).
''
''   F: 3.317, GRADSHTEYN AND RYZHIK (1965), TABLE OF INTEGRALS,SERIES,PRODUCTS.
''
'' INPUT:
'' BETA   ____ SHAPE PARAMETER
'' LAMBDA ____ RATE PARAMETER
'' XMAX   ____ UPPER TRUNCATION POINT
''
'' OUTPUT:
'' ETX  _____ EXPECTED VALUE OF TRUNCATED RANDOM VARIABLE
'' SDTX _____ STANDARD DEVIATION OF THE TRUNCATED RV
''
DEFINE I,J,K,M,N AS INTEGER VARIABLES
DEFINE SUMV AS A REAL, 1-DIMENSIONAL ARRAY
RESERVE SUMV(*) AS 2 ''LOCALLY
     LET N=50
     LET XBAR=COMPLETE.GAMMA(1.0+1.0/BETA)/LAMBDA
     LET LIM=XBAR/10.0**7
     IF XMAX LE 0.0
          PRINT 1 LINE THUS
INPUT ERROR TO ROUTINE WEI.TRUNC.  XMAX LE 0.
          STOP
     OTHERWISE
     LET CDF=1.0-EXP.F(-(LAMBDA*XMAX)**BETA)
     IF CDF LE 0.
          PRINT 1 LINE THUS
INPUT ERROR TO ROUTINE WEI.TRUNC.  CDF(XMAX) LE 0.
          STOP
     OTHERWISE
     LET U=(LAMBDA*XMAX)**BETA
     LET ELIM=LIM*EXP.F(U)
  FOR M=1 TO 2 ''STATISTICAL MOMENTS'' DO
     LET P=M/BETA+1.0
     LET TERM=1.0/P*U**P
     LET DENOM=P
     LET SUMV(M)=TERM
     FOR K=1 TO N DO
          LET DENOM=DENOM*(P+K)
          LET TERM=U**(P+K)/DENOM
          ADD TERM TO SUMV(M)
          IF K > 12 AND TERM < ELIM
               GO TO L1
          OTHERWISE
     LOOP ''OVER K
 'L1'LOOP ''OVER (M) MOMENTS
     LET FACT=EXP.F(-U)
     LET ETX=FACT*SUMV(1)/CDF/LAMBDA
     LET VTX=FACT*SUMV(2)/CDF/LAMBDA**2-ETX**2
     IF VTX LE 0.0
          PRINT 1 LINE WITH VTX
          THUS
TROUBLE IN WEI.TRUNC.  VTX = ...........
          STOP
```

```
58        OTHERWISE
59        LET SDTX = ...(VTY)
60        RELEASE SUMV(*)
61        RETURN
62        END
```

C R O S S - R E F E R E N C E

NAME	TYPE		MODE	LINE NUMBERS OF REFERENCES						
BETA	ARGUMENT	NO. 1	DOUBLE	1	21	28	34	37		
CDF	RECURSIVE VARIABLE	WORD 22	DOUBLE	26	29	51	52			
COMPLETE.GAMMA	ROUTINE		DOUBLE	21						
DENOM	RECURSIVE VARIABLE	WORD 32	DOUBLE	39	42*	43				
ELTM	RECURSIVE VARIABLE	WORD 26	DOUBLE	35	45					
ETX	ARGUMENT	NO. 4	DOUBLE	1	51	52				
EXP.F	ROUTINE		DOUBLE	28	35	50				
FACT	RECURSIVE VARIABLE	WORD 34	DOUBLE	50	51	52				
I	RECURSIVE VARIABLE	WORD 1	INTEGER	17						
J	RECURSIVE VARIABLE	WORD 2	INTEGER	17						
K	RECURSIVE VARIABLE	WORD 3	INTEGER	17	41*	42	43	45		
L1	LABEL			46	49					
LAMBDA	ARGUMENT	NO. 2	DOUBLE	1	21	28	34	51	52	
LIM	RECURSIVE VARIABLE	WORD 8	DOUBLE	22	35					
M	RECURSIVE VARIABLE	WORD 4	INTEGER	17	36*	37	40	44*		
N	RECURSIVE VARIABLE	WORD 5	INTEGER	17	29	41				
P	RECURSIVE VARIABLE	WORD 28	DOUBLE	37	38*	39	42	43		
SDTX	ARGUMENT	NO. 5	DOUBLE	1	59					
SQRT.F	ROUTINE		DOUBLE	59						
SUMV	RECURSIVE VARIABLE	WORD 6 (1-D)	DOUBLE	18	19*	40	44*	51	52	60*
TERM	RECURSIVE VARIABLE	WORD 30	DOUBLE	38	40	43	44	45		
U	RECURSIVE VARIABLE	WORD 24	DOUBLE	34	35	38	43	50		
VTY	RECURSIVE VARIABLE	WORD 36	DOUBLE	52	53	54	59			
WET.TRUNC	ROUTINE		INTEGER	1						
XBAR	RECURSIVE VARIABLE	WORD 7	DOUBLE	21	22					
XMAX	ARGUMENT	NO. 3	DOUBLE	1	23	28	34			

```
 1  PREAMBLE ''TO TEST KOFN.DIST FUNCTION
 2  NORMALLY MODE IS REAL
 3  DEFINE DIST.TYPE AS A TEXT VARIABLE
 4  DEFINE CDF.P, UTP AS REAL VARIABLES
 5  DEFINE TRUNC.FLAG AS AN INTEGER VARIABLE
 6  DEFINE COMPLETE.GAMMA AS A REAL FUNCTION WITH 1 ARGUMENT
 7  DEFINE XNORM AS A REAL FUNCTION WITH 1 ARGUMENT
 8  DEFINE ERLANG.DIST AS A REAL FUNCTION WITH 2 ARGUMENTS
 9  DEFINE F.DIST AS A REAL FUNCTION WITH 3 ARGUMENTS
10  DEFINE F.DIST.INV AS A REAL FUNCTION WITH 3 ARGUMENTS
11  DEFINE FUN.CDF AS A REAL FUNCTION WITH 3 ARGUMENTS
12  DEFINE KOFN.DIST AS A REAL FUNCTION WITH 5 ARGUMENTS
13  END ''PREAMBLE
```

C R O S S - R E F E R E N C E

NAME	TYPE			MODE	LINE NUMBERS OF REFERENCES
CDF.P	GLOBAL VARIABLE	ARR	5	DOUBLE	4
COMPLETE.GAMMA	ROUTINE			DOUBLE	6
DIST.TYPE	GLOBAL VARIABLE	ARR	2	TEXT	3
ERLANG.DIST	ROUTINE			DOUBLE	8
F.DIST	ROUTINE			DOUBLE	9
F.DIST.INV	ROUTINE			DOUBLE	10
FUN.CDF	ROUTINE			DOUBLE	11
KOFN.DIST	ROUTINE			DOUBLE	12
TRUNC.FLAG	GLOBAL VARIABLE	ARR	7	INTEGER	5
UTP	GLOBAL VARIABLE	ARR	3	DOUBLE	4
XNORM	ROUTINE			DOUBLE	7

```
MAIN **TEST.K
**
**DRIVER TO TEST THE FUNCTION KOFN.DIST.  THIS REPRESENTS THE PROBABILITY
**DISTRIBUTION FUNCTION FOR THE K TH LARGEST IN A SAMPLE OF N RANDOM VARIABLES.
**
DEFINE I,J,K,M,N AS INTEGER VARIABLES
DEFINE ANSWER AS A TEXT VARIABLE
DEFINE FV, FHOLDV, SUMV AS REAL, 1-DIMENSIONAL ARRAYS
RESERVE FV(*), FHOLDV(*), SUMV(*) AS 4
     LET DIST.TYPE = "WEIBULL"
     SKIP 1 LINE
     PRINT 1 LINE WITH DIST.TYPE
     THUS
DO YOU WANT TO TRUNCATE THE ********* DISTRIBUTION?  (YES OR NO).
     READ ANSWER
     IF SUBSTR.F(ANSWER,1,1) = "Y"
          LET TRUNC.FLAG=1
          PRINT 1 LINE THUS
INPUT THE PROBABILITY ASSOCIATED WITH THE UPPER TRUNCATION POINT.
          READ CDF.P
          LET HOLD=-LOG.E.F(1.0-CDF.P)
     OTHERWISE
          LET TRUNC.FLAG=0
     ALWAYS
     LET M=3000
'L1'SKIP 1 LINE
     PRINT 1 LINE THUS
INPUT THE NUMBER OF RANDOM VARIABLES IN THE SET.  ZERO STOPS.
     READ N
     IF N LE 0
          STOP
     OTHERWISE
     PRINT 1 LINE THUS
INPUT THE VALUE OF K FOR THE K TH LARGEST OF THE RV'S.
     READ K
     PRINT 1 LINE THUS
INPUT THE VALUE OF THE WEIBULL SHAPE PARAMETER.
     READ P1
     PRINT 1 LINE THUS
INPUT THE RATE PARAMETER OF THE WEIBULL DIST OF EACH OF THE RV'S.
     READ P2
**
**PRINT HEADINGS.
**
     IF TRUNC.FLAG=1
          LET ANSWER = "TRUNCATED"
          LET UTP=HOLD**(1.0/P1)/P2
          LET XTRUNC=UTP
          CALL WEI.TRUNC GIVEN P1, P2, XTRUNC YIELDING MEAN, STDV
     OTHERWISE
          LET ANSWER = "NON-TRUNC"
          LET MEAN=COMPLETE.GAMMA(1.0+1.0/P1)/P2
          LET STDV=SQRT.F(COMPLETE.GAMMA(1.0+2.0/P1)/P2**2-MEAN**2)
     ALWAYS
     SKIP 2 LINES
     PRINT 7 LINES WITH K,N,ANSWER,DIST.TYPE,P1,P2,MEAN,STDV
     THUS
```

```
MAIN ROUTINE                                CACI SIMSCRIPT II.5 FOR PRIME SYSTEMS, RELEASE 2.1          PAGE     7
OPTIONS = SEQUENCE.ID.SUBCHK.XREF.NOEXPLIST.TRACE?                                   ?5 DEC 1984 ?9:20:23

DISTRIBUTION FUNCTION FOR THE ** TH LARGEST OF ** ********* ********* R V'S

SHAPE PARAM = **.****  RATE PARM = **.****  MEAN = **.****  STD DEV = **.****

ARGU-      COPL      CDF FOR
MENT       PROB      MAX IN SET
--------------------------------------------------------------------------------
        LET DELZ=?.??
        LET FV(1)=1.?
        LET FV(2)=?.0
        LET FV(3)=1.0
        LET FV(4)=0.?
        FOR J=1 TO 4 DO
            LET SUMV(J)=FV(J)
        LOOP ''OVER J
        FOR I=1 TO M DO
            LET Z=I*DELZ
            IF MOD.F(I,2)=0
                LET COEF=2.0
            OTHERWISE
                LET COEF=4.0
            ALWAYS
            LET GN=(FUN.CDF(P1,P2,Z))**N
            LET GN.COMPL=1.0-GN
            LET FV(1)=GN.COMPL
            LET FV(2)=Z*GN.COMPL
            LET P=KOFN.DIST(K,N,P1,P2,Z)
            LET Q=1.0-P
            LET FV(3)=Q
            LET FV(4)=Z*Q
            FOR J=1 TO 4, ADD COEF*FV(J) TO SUMV(J)
            IF MOD.F(I,20)=0
                PRINT 1 LINE WITH Z,P,GN
                THUS
**.***     *.****    *.****
            ALWAYS
            IF GN > 0.9999 AND COEF =2.0
                FOR J=1 TO 4, SUBTRACT FV(J) FROM SUMV(J)
                GO TO L1
            OTHERWISE
        LOOP ''OVER I
    'L1'PRINT 2 LINES THUS
--------------------------------------------------------------------------------
        IF TRUNC.FLAG=1
            PRINT 1 LINE WITH XTRUNC
            THUS
UPPER TRUNCATION POINT OF CDF _______ **.****
        ALWAYS
        LET EZ=SUMV(1)*DELZ/3.0
        LET SDZ=SQRT.F(2.0*SUMV(2)*DELZ/3.0-EZ**2)
        LET EX=SUMV(3)*DELZ/3.?
        LET SDX=SQRT.F(2.0*SUMV(4)*DELZ/3.0-EX**2)
        PRINT 4 LINES WITH EZ,SDZ,K,N,EX,K,N,SDX
        THUS
MEAN VALUE OF MAXIMUM RV IN THE SET _ **.****
STANDARD DEV OF MAXIMUM  IN THE SET _ **.****
```

MAIN ROUTINE CACI SIMSCRIPT II.5 FOR PRIME SYSTEMS, RELEASE 2.1 PAGE 4
OPTIONS = SEQUENCE.ID,SUBCHK,XREF,NOEXPLIST,TRACE3 05 DEC 1984 09:20:23

```
MEAN OF ** TH OF ** RV*S IN THE SET _ **.****
STD DEV ** TH OF ** RV*S IN THE SET _ **.****
  96      GO TO L0
  97  END **TEST.K
```

C R O S S - R E F E R E N C E

NAME	TYPE				MODE	LINE NUMBERS OF REFERENCES
ANSWER	RECURSIVE VARIABLE	WORD	6		TEXT	7 14 15 40 45 51
CDF.P	GLOBAL VARIABLE	ARR	5		DOUBLE	18 19
COEF	RECURSIVE VARIABLE	WORD	40		DOUBLE	63 65 75 80
COMPLETE.GAMMA	ROUTINE				DOUBLE	46 47
DELZ	RECURSIVE VARIABLE	WORD	36		DOUBLE	52 61 90 91 92 93
DIST.TYPE	GLOBAL VARIABLE	ARR	2		TEXT	10 12 5
EX	RECURSIVE VARIABLE	WORD	54		DOUBLE	92 93 94
EZ	RECURSIVE VARIABLE	WORD	50		DOUBLE	90 91 94
FHOLDV	RECURSIVE VARIABLE	WORD	8	(1-D)	DOUBLE	8 9*
FUM.CDF	ROUTINE				DOUBLE	67
FV	RECURSIVE VARIABLE	WORD	7	(1-D)	DOUBLE	8 9* 53 54 55 56 58
						69 70 73 74 75 81
GN	RECURSIVE VARIABLE	WORD	42		DOUBLE	67 68 77 80
GN.COMPL	RECURSIVE VARIABLE	WORD	44		DOUBLE	68 69 70
HOLD	RECURSIVE VARIABLE	WORD	22		DOUBLE	19 41
I	RECURSIVE VARIABLE	WORD	1		INTEGER	6 60* 61 62 76
J	RECURSIVE VARIABLE	WORD	2		INTEGER	6 57* 58* 75* 81*
K	RECURSIVE VARIABLE	WORD	3		INTEGER	6 31 50 71 94*
KOFN.DIST	ROUTINE				DOUBLE	71
L0	LABEL					24 96
L1	LABEL					62 85
LOG.E.F	ROUTINE				DOUBLE	19
M	RECURSIVE VARIABLE	WORD	4		INTEGER	6 23 60
MEAN	RECURSIVE VARIABLE	WORD	31		DOUBLE	43 46 47 50
MOD.F	ROUTINE				INTEGER	62 76
N	RECURSIVE VARIABLE	WORD	5		INTEGER	6 26 27 50 67 71 94*
P	RECURSIVE VARIABLE	WORD	46		DOUBLE	71 72 77
P1	RECURSIVE VARIABLE	WORD	25		DOUBLE	33 41 43 46 47 5 67
						71
P2	RECURSIVE VARIABLE	WORD	27		DOUBLE	35 41 43 46 47 50 67
						71
Q	RECURSIVE VARIABLE	WORD	48		DOUBLE	72 73 74
SDX	RECURSIVE VARIABLE	WORD	56		DOUBLE	93 94
SDZ	RECURSIVE VARIABLE	WORD	52		DOUBLE	91 94
SQRT.F	ROUTINE				DOUBLE	47 91 93
STDV	RECURSIVE VARIABLE	WORD	33		DOUBLE	43 47 50
SUBSTR.F	ROUTINE				TEXT	15
SUMV	RECURSIVE VARIABLE	WORD	9	(1-D)	DOUBLE	8 9* 58 75* 81* 90 91
						92 93
TRUNC.FLAG	GLOBAL VARIABLE	ARR	7		INTEGER	16 21 39 86
UTP.W	IMPLIED SUBSCRIPT	SYS	4		INTEGER	11 24 49
UTP	GLOBAL VARIABLE	ARR	3		DOUBLE	41 42
WEI.TRUNC	ROUTINE				INTEGER	43
XTRUNC	RECURSIVE VARIABLE	WORD	29		DOUBLE	42 43 87
Z	RECURSIVE VARIABLE	WORD	38		DOUBLE	61 67 70 71 74 77

```
 1  FUNCTION KOFN.DIST (K, N, P1, P2, Z)
 2  ''
 3  ''FUNCTION CALCULATES THE CUMULATIVE PROBABILITY THAT THE K TH LARGEST
 4  ''OF N CONTINUOUS RANDOM VARIABLES, EACH WITH CDF F(Z), IS LESS THAN Z.
 5  ''
 6  ''THIS FUNCTION CALLS THE GENERAL CDF FUNCTION FUN.CDF WITH PARAMETERS
 7  ''P1 AND P2 AND WITH ARGUMENT Z.  THIS FUNCTION ALSO CALLS FOR THE CUM
 8  ''PROB FROM FISHER'S F-DISTRIBUTION--F.DIST(XF,NU1,NU2).
 9  ''
10  DEFINE K,N,NU1,NU2 AS INTEGER VARIABLES
11      LET NU1=2*(N-K+1)
12      LET NU2=2*K
13      LET A=K
14      LET B=N-K+1
15      LET X=FUN.CDF(P1,P2,Z)
16      IF X LE 0.0
17          RETURN WITH 0.0
18      OTHERWISE
19      LET XF=A*(1.0-X)/B/X
20      RETURN WITH 1.0-F.DIST(XF,NU1,NU2)
21  END ''KOFN.DIST
```

C R O S S - R E F E R E N C E

NAME	TYPE			MODE	LINE NUMBERS OF REFERENCES					
A	RECURSIVE VARIABLE	WORD	3	DOUBLE	13	19				
B	RECURSIVE VARIABLE	WORD	5	DOUBLE	14	19				
F.DIST	ROUTINE			DOUBLE	20					
FUN.CDF	ROUTINE			DOUBLE	15					
K	ARGUMENT	NO.	1	INTEGER	1	10	11	12	13	14
KOFN.DIST	ROUTINE			DOUBLE	1					
N	ARGUMENT	NO.	2	INTEGER	1	10	11	14		
NU1	RECURSIVE VARIABLE	WORD	1	INTEGER	10	11	20			
NU2	RECURSIVE VARIABLE	WORD	2	INTEGER	10	12	20			
P1	ARGUMENT	NO.	3	DOUBLE	1	15				
P2	ARGUMENT	NO.	4	DOUBLE	1	15				
X	RECURSIVE VARIABLE	WORD	7	DOUBLE	15	16	19*			
XF	RECURSIVE VARIABLE	WORD	6	DOUBLE	19	20				
Z	ARGUMENT	NO.	5	DOUBLE	1	15				

```
FUNCTION F.DIST(X,NU1,NU2)
''
''ROUTINE CALCULATES THE CUMULATIVE PROBABILITY OF FISHER'S F-DISTRIBUTION
''GIVEN ARGUMENTS X, AND THE INTEGER DEGREES OF FREEDOM NU1 (FOR THE
''NUMERATOR) AND NU2 (FOR THE DENOMINATOR).
''REFERENCE: ABRAMOWITZ AND STEGUN, AMS 55, P. 946, AUG 1966.
''
     DEFINE K,KL,N,NU1, AND NU2 AS INTEGER VARIABLES
     LET T = REAL.F(NU2)/(REAL.F(NU2)+REAL.F(NU1)*X)
     IF 2*DIV.F(NU1,2) < NU1
          GO TO L2
     OTHERWISE ''NU1 IS EVEN
     LET T1=1.0-T
     LET N=NU2
     LET A=0.0
     LET B=T**(REAL.F(NU2)/2.0)
     LET KL=DIV.F(NU1-2,2)
     GO TO L4
'L2' IF 2*DIV.F(NU2,2) < NU2
          GO TO L8 ''NU1 AND NU2 ARE ODD
     OTHERWISE ''NU2 IS EVEN
     LET T1=T
     LET N=NU1
     LET A=1.0
     LET B=-(1.0-T)**(REAL.F(NU1)/2.0)
     LET KL=DIV.F(NU2-2,2)
'L4' LET SUM=1.0
     LET TRM=1.0
     IF KL LE 0
          GO TO L7
     OTHERWISE
     FOR K = 1 TO KL DO
          LET TRM=TRM*REAL.F(N+2*K-2)/REAL.F(2*K)*T1
          ADD TRM TO SUM
     LOOP ''OVER K
'L7' RETURN WITH 1.0-(A+B*SUM)
'L8' LET ARG1=SQRT.F(REAL.F(NU1)*X)
     LET ARG2=SQRT.F(REAL.F(NU2))
     LET THETA=ARCTAN.F(ARG1,ARG2)
     LET SINT=SIN.F(THETA)
     LET COST=COS.F(THETA)
     IF NU2 > 1
          GO TO L10
     OTHERWISE
     LET A=2.0*THETA/PI.C
     GO TO L14
'L10'LET SUM=COST
     LET TRM=COST
     LET KL=DIV.F(NU2,2)-1
     IF KL LE 0
          GO TO L13
     OTHERWISE
     FOR K = 1 TO KL DO
          LET TRM=TRM*REAL.F(2*K)/REAL.F(2*K+1)*COST**2
          ADD TRM TO SUM
     LOOP ''OVER K
'L13'LET A=2.0/PI.C*(THETA+SINT*SUM)
```

```
'L14'IF NU1 > 1
          GO TO L16
     OTHERWISE
     LET B=0.0
     GO TO L25
'L16'LET SUM=1.
     LET TRM=1.
     LET KL=DIV.F(NU1,2)-1
     IF KL LE 0
          GO TO L19
     OTHERWISE
     FOR K=1 TO KL DO
          LET TRM=TRM*REAL.F(NU2+2*K-1)/REAL.F(2*K+1)*SINT**2
          ADD TRM TO SUM
     LOOP ''OVER K
'L19'IF NU2 GE 2
          GO TO L21
     OTHERWISE
     LET C=1.0
     GO TO L24
'L21'LET C=2.0
     LET KL=DIV.F(NU2,2)-1
     IF KL LE 0
          GO TO L24
     OTHERWISE
     FOR K=1 TO KL DO
          LET C=C*2.0*REAL.F(DIV.F(NU2,2)+1-K)/REAL.F(NU2-2*K)
     LOOP ''OVER K
'L24'LET B=2.0/PI.C*C*SINT*COST**NU2*SUM
'L25'RETURN WITH A-B
END ''OF FUNCTION F.DIST
```

```
                         C R O S S - R E F E R E N C E

  NAME                       TYPE                                 MODE      LINE NUMBERS OF REFERENCES

A                            RECURSIVE VARIABLE    WORD   6       DOUBLE     15      24      76      45      57      87
ARCTAN.F                     ROUTINE                              DOUBLE     39
ARG1                         RECURSIVE VARIABLE    WORD  16       DOUBLE     37      39
ARG2                         RECURSIVE VARIABLE    WORD  18       DOUBLE     38      39
B                            RECURSIVE VARIABLE    WORD  10       DOUBLE     16      25      36      61      86      87
C                            RECURSIVE VARIABLE    WORD  26       DOUBLE     76      78      84*     86
COS.F                        ROUTINE                              DOUBLE     41
COST                         RECURSIVE VARIABLE    WORD  24       DOUBLE     41      47      48      54      86
DIV.F                        ROUTINE                              INTEGER    10      17      19      26      49      65      79
                                                                             84
F.DIST                       ROUTINE                              DOUBLE      1
K                            RECURSIVE VARIABLE    WORD   1       INTEGER     8      32*     33*     53*     54*     69*     70*
                                                                             83*     84*
KL                           RECURSIVE VARIABLE    WORD   2       INTEGER     8      17      26      29      32      49      50
                                                                             53      65      66      69      79      80      83
L10                          LABEL                                           43      47
L13                          LABEL                                           51      57
L14                          LABEL                                           46      58
L16                          LABEL                                           59      63
L19                          LABEL                                           67      73
L2                           LABEL                                           11      19
L21                          LABEL                                           74      78
L24                          LABEL                                           77      81      86
L25                          LABEL                                           62      87
L4                           LABEL                                           18      27
L7                           LABEL                                           30      36
L8                           LABEL                                           20      37
N                            RECURSIVE VARIABLE    WORD   3       INTEGER     8      14      23      33
NU1                          ARGUMENT              NO.    2       INTEGER     1       8       9      10*     17      23      25
                                                                             37      58      65
NU2                          ARGUMENT              NO.    3       INTEGER     1       8       9*     14      16      19*     26
                                                                             38      42      49      70      73      79      84*
                                                                             66
PI.C                         PERMANENT ATTRIBUTE   SYS   33       DOUBLE     45      57      86
REAL.F                       ROUTINE                              INTEGER     9*     16      25      33*     37      38      54*
                                                                             70*     84*
SIN.F                        ROUTINE                              DOUBLE     40
SINT                         RECURSIVE VARIABLE    WORD  22       DOUBLE     40      57      70      86
SQRT.F                       ROUTINE                              DOUBLE     37      38
SUM                          RECURSIVE VARIABLE    WORD  12       DOUBLE     27      34*     36      47      55*     57      63
                                                                             71*     86
T                            RECURSIVE VARIABLE    WORD   4       DOUBLE      9      13      16      22      25
T1                           RECURSIVE VARIABLE    WORD   6       DOUBLE     13      22      33
THETA                        RECURSIVE VARIABLE    WORD  20       DOUBLE     39      40      41      45      57
TRM                          RECURSIVE VARIABLE    WORD  14       DOUBLE     28      33*     34      48      54*     55      64
                                                                             70*     71
X                            ARGUMENT              NO.    1       DOUBLE      1       9      37
```

```
FUNCTION F.DIST.INV(P,NU1,NU2)
''
''ROUTINE CALCULATES THE QUANTILE OF FISHER'S F-DISTRIBUTION, GIVEN THE
'' CUMULATIVE (LOWER TAIL) PROBABILITY (P) AND THE INTEGER DEGREES OF
''FREEDOM OF THE NUMERATOR (NU1) AND DEGREES OF FREEDOM OF THE
''DENOMINATOR (NU2).
''REFERENCE:  ABRAMOWITZ AND STEGUN, AMS 55, P. 947, AUG 1966.
''
     DEFINE NU1 AND NU2 AS INTEGER VARIABLES
     LET Q=1.0-P ''FOR THE UPPER TAIL PROBABILITY
     LET ERR=0.0001
     IF NU1=2
          GO TO L3
     OTHERWISE
     IF NU1 LE 0 OR NU2 LE 0
          PRINT 1 LINE WITH NU1 AND NU2 THUS
INPUT ERROR IN F.DIST.INV.  NU1 = ***  NU2 = ***
          STOP
     OTHERWISE
     IF NU1 > 2
          GO TO L4
     OTHERWISE
     IF NU2=1
          RETURN WITH (TAN.F(P*PI.C/2.0))**2
     OTHERWISE
     IF NU2=2
          GO TO L5
     OTHERWISE
'L3'LET X=Q**(2.0/REAL.F(NU2)) ''NU1 >2; NU2=2
     LET FI=REAL.F(NU2)/2.0*(1.0-X)/X
     IF NU1=2
          RETURN WITH FI
     OTHERWISE
          GO TO L7
'L4'IF NU2 < 2
          GO TO L3
     OTHERWISE
     IF NU2 > 2
          GO TO L6
     OTHERWISE ''NU2=2
'L5'LET X=1.0-P**(2.0/REAL.F(NU1))
     LET FI=2.0/REAL.F(NU1)*(1.0-X)/X
     IF NU2=2
          RETURN WITH FI
     OTHERWISE
          GO TO L7
'L6'LET XP=XNORM(P) ''NU1,NU2>2
     LET X=(XP**2-3.0)/6.0
     LET H=2.0/(1.0/(REAL.F(NU2)-1.0)+1.0/(REAL.F(NU1)-1.0))
     LET W=XP*SQRT.F(H+X)/H-(1.0/(REAL.F(NU1)-1.0)-1.0/(REAL.F(NU2)-1.0))*
      (X+5.0/6.0-2.0/(3.0*H))
     LET FI=EXP.F(2.0*W)
'L7'LET X1=FI
     LET Y1=F.DIST(X1,NU1,NU2)
     LET X2=1.01*X1
'L8'LET Y2=F.DIST(X2,NU1,NU2)
     IF ABS.F(Y1-Y2) LE ERR
```

```
ROUTINE F.DIST.INV                    CACI SIMSCRIPT II.5 FOR PRIME SYSTEMS, RELEASE 2.1          PAGE   1?
OPTIONS = SEQUENCE.ID.SUECHK.XREF.NOEXPLIST.TRACE?                              ?? DEC 1984 ?9:2?:?3

   57          RETURN WITH X?
   58       OTHERWISE
   59       LET X3=X2+(X?-X1)/(Y2-Y1)*(P-Y?)
   60       LET X1=X2
   61       LET X2=X3
   62       LET Y1=Y2
   63       GO TO L8
   64   END **OF F.DIST.INV

                   C R O S S - R E F E R E N C E

  NAME                  TYPE                              MODE      LINE NUMBERS OF REFERENCES

ABS.F                 ROUTINE                             INTEGER    56
ERR                   RECURSIVE VARIABLE    WORD    3     DOUBLE     11     56
EXP.F                 ROUTINE                             DOUBLE     51
F.DIST                ROUTINE                             DOUBLE     53     55
F.DIST.INV            ROUTINE                             DOUBLE      1
FI                    RECURSIVE VARIABLE    WORD   18     DOUBLE     29     31     41     43     51     52
H                     RECURSIVE VARIABLE    WORD   22     DOUBLE     48     49*    50
L3                    LABEL                                          13     28     35
L4                    LABEL                                          20     34
L5                    LABEL                                          26     40
L6                    LABEL                                          38     46
L7                    LABEL                                          33     45     52
L8                    LABEL                                          55     63
NU1                   ARGUMENT              NO.     2     INTEGER     1      9     12     15     16     19     30
                                                                     40     41     48     49     53     55
NU2                   ARGUMENT              NO.     3     INTEGER     1      9     15     16     22     25     28
                                                                     29     34     37     42     48     49     53
                                                                     55
P                     ARGUMENT              NO.     1     DOUBLE      1     10     23     43     46     58
PI.C                  PERMANENT ATTRIBUTE   SYS    33     DOUBLE     23
Q                     RECURSIVE VARIABLE    WORD    1     DOUBLE     10     28
REAL.F                ROUTINE                             INTEGER    28     29     40     41     48*    49*
SQRT.F                ROUTINE                             DOUBLE     49
TAN.F                 ROUTINE                             DOUBLE     23
W                     RECURSIVE VARIABLE    WORD   24     DOUBLE     48     51
X                     RECURSIVE VARIABLE    WORD   16     DOUBLE     28     29*    4?     41*    47     49     5?
X1                    RECURSIVE VARIABLE    WORD   26     DOUBLE     52     53     54     59     60
X2                    RECURSIVE VARIABLE    WORD   3?     DOUBLE     54     55     57     59*    60     61
X3                    RECURSIVE VARIABLE    WORD   34     DOUBLE     59     61
XNORM                 ROUTINE                             DOUBLE     46
XP                    RECURSIVE VARIABLE    WORD   2?     DOUBLE     46     47     49
Y1                    RECURSIVE VARIABLE    WORD   2?     DOUBLE     53     56     59     62
Y2                    RECURSIVE VARIABLE    WORD   32     DOUBLE     55     56     59*    62
```

```
OPTIONS = SEQUENCE.ID.SUBCHK.XREF.NOEXPLIST.TRACE3              15 DEC 1984 09:20:23

  1  ''ROUTINE.XNORM
  2  ROUTINE XNORM(CUM.PROB)
  3  ''
  4  ''ROUTINE FOR THE STANDARD NORMAL PROBABILITY INVERSE FUNCTION.  ROUTINE
  5  ''RETURNS THE QUANTILE OF THE NORMAL C.D.F. GIVEN THE CUMULATIVE PROBABILITY
  6  ''(CUM.PROB).  REFERENCE:  AMS 55, "HANDBOOK OF MATHEMATICAL FUNCTIONS",
  7  ''NAT. BUREAU OF STANDARDS, (P. 933).
  8  ''
  9  IF CUM.PROB LE 0.5
 10       LET P=CUM.PROB
 11       LET SIGN=-1.0
 12  OTHERWISE
 13       LET P=1.0-CUM.PROB
 14       LET SIGN=1.0
 15  ALWAYS
 16  LET T=SQRT.F(LOG.E.F(1.0/P/P))
 17  LET Z=T-(2.515517+T*(0.802853+T*0.010328))/(1.0+T*(1.432788+T*(0.189269+
 18       T*0.001308)))
 19  RETURN WITH Z*SIGN
 20  END ''OF XNORM
```

C R O S S - R E F E R E N C E

NAME	TYPE			MODE	LINE NUMBERS OF REFERENCES			
CUM.PROB	ARGUMENT	NO.	1	DOUBLE	2	9	10	13
LOG.E.F	ROUTINE			DOUBLE	16			
P	RECURSIVE VARIABLE	WORD	1	DOUBLE	10	13	16*	
SIGN	RECURSIVE VARIABLE	WORD	3	DOUBLE	11	14	19	
SQRT.F	ROUTINE			DOUBLE	16			
T	RECURSIVE VARIABLE	WORD	5	DOUBLE	16	17*	18	
XNORM	ROUTINE			DOUBLE	2			
Z	RECURSIVE VARIABLE	WORD	7	DOUBLE	17	19		

```
 1  FUNCTION ERLANG.DIST(X,N)
 2  ''
 3  ''FUNCTION COMPUTES THE CUMULATIVE PROBABILITY FOR AN ERLANG DISTRIBUTION
 4  ''WITH STANDARDIZED ARGUMENT X AND SHAPE PARAMETER N.
 5  ''
 6  DEFINE I AND N AS INTEGER VARIABLES
 7       LET EXPX=EXP.F(-X)
 8       IF N=1
 9            RETURN WITH 1.0-EXPX
10       OTHERWISE
11       LET FACTORIAL=1.0
12       LET ARG=1.0
13       LET SUM=1.0
14  FOR  I=1 TO N-1 DO
15       LET FACTORIAL=I*FACTORIAL
16       LET ARG=X*ARG
17       ADD ARG/FACTORIAL TO SUM
18  LOOP ''OVER I
19       RETURN WITH 1.0-EXPX*SUM
20  END ''ERLANG.DIST
```

C R O S S - R E F E R E N C E

NAME	TYPE			MODE	LINE NUMBERS OF REFERENCES			
ARG	RECURSIVE VARIABLE	WORD	6	DOUBLE	12	16*	17	
ERLANG.DIST	ROUTINE			DOUBLE	1			
EXP.F	ROUTINE			DOUBLE	7			
EXPX	RECURSIVE VARIABLE	WORD	2	DOUBLE	7	9	19	
FACTORIAL	RECURSIVE VARIABLE	WORD	4	DOUBLE	11	15*	17	
I	RECURSIVE VARIABLE	WORD	1	INTEGER	6	14*	15	
N	ARGUMENT	NO.	2	INTEGER	1	6	8	14
SUM	RECURSIVE VARIABLE	WORD	8	DOUBLE	13	17*	19	
X	ARGUMENT	NO.	1	DOUBLE	1	7	16	

```
 1  FUNCTION FUN.CDF (P1, P2, T)
 2  ''
 3  ''FUNCTION CALCULATES THE C.D.F. OF A GENERAL, POSITIVE RANDOM VARIABLE.
 4  ''THE PARAMETERS OF THE DISTRIBUTION ARE P1 AND P2.  THE RV ARGUMENT IS T.
 5  ''
 6  ''THE VARIABLES TRUNC.FLAG, CDF.P, AND UTP ARE GLOBAL.
 7  ''
 8      DEFINE N AS AN INTEGER VARIABLE
 9  ''  LET N=MAX.F(1,TRUNC.F(P1))
10      LET ARG=P2*T
11  ''  RETURN WITH ERLANG.DIST( ARG, N)
12      IF TRUNC.FLAG=1
13          IF T < UTP
14              RETURN WITH (1.0-EXP.F(-(ARG)**P1))/CDF.P
15          OTHERWISE
16          RETURN WITH 1.0
17      OTHERWISE
18      RETURN WITH 1.0-EXP.F(-(ARG)**P1)
19  END ''FUN.CDF
```

C R O S S - R E F E R E N C E

NAME	TYPE			MODE	LINE NUMBERS OF REFERENCES		
ARG	RECURSIVE VARIABLE	WORD	2	DOUBLE	10	14	18
CDF.P	GLOBAL VARIABLE	ARR	5	DOUBLE	14		
EXP.F	ROUTINE			DOUBLE	14	18	
FUN.CDF	ROUTINE			DOUBLE	1		
N	RECURSIVE VARIABLE	WORD	1	INTEGER	8		
P1	ARGUMENT	NO.	1	DOUBLE	1	14	18
P2	ARGUMENT	NO.	2	DOUBLE	1	10	
T	ARGUMENT	NO.	3	DOUBLE	1	10	13
TRUNC.FLAG	GLOBAL VARIABLE	ARR	7	INTEGER	12		
UTP	GLOBAL VARIABLE	ARR	3	DOUBLE	13		

```
    FUNCTION COMPLETE.GAMMA(XX)
    ''
    ''PROGRAM CALCULATES THE COMPLETE GAMMA FUNCTION WITH SINGLE REAL ARGUMENT XX.
    ''METHOD: THE RECURSION RELATION AND POLYNOMIAL APPROXIMATION IS TAKEN
    ''FROM C. HASTINGS, JR., "APPROXIMATIONS FOR DIGITAL COMPUTERS,"
    '' PRINCETON UNIV. PRESS 1955.
    ''
              IF XX ... 57.0
                  GO TO L130
              OTHERWISE
    'L0'      LET X = XX
              LET ERR = 0.000001
              LET GAMMA = 1.0
              IF X LE 2.0
                  GO TO L50
              OTHERWISE
                  GO TO L15
    'L10'     IF X LE 2.0
                  GO TO L110
              OTHERWISE
    'L15'     SUBTRACT 1.0 FROM X
              LET GAMMA = GAMMA * X
              GO TO L10
    'L50'     IF X = 1.0
                  GO TO L120
              OTHERWISE
              IF X > 1.0
                  GO TO L110
              OTHERWISE
    'L60'     IF X > ERR
                  GO TO L80
              OTHERWISE
              LET Y = REAL.F(TRUNC.F(X))-X
              IF ABS.F(Y) LE ERR
                  GO TO L130
              OTHERWISE
              IF Y+ERR GE 1.0
                  GO TO L130
              OTHERWISE
    'L70'     IF X > 1.0
                  GO TO L110
              OTHERWISE
    'L80'     LET GAMMA = GAMMA / X
              ADD 1.0 TO X
              GO TO L70
    'L110'    LET Y = X - 1.0
              LET GY = 1.0+Y*(-0.57710166+Y*(0.98585399+Y*(-0.87642182+Y*(0.83282120
              +Y*(-0.56847290+Y*(0.25482049+Y*(-0.05149930)))))))
              LET GAMMA = GAMMA * GY
    'L120'    RETURN WITH GAMMA
    'L130'    PRINT 1 LINE WITH XX THUS
ERROR IN COMPLETE.GAMMA.  ARGUMENT = ***.*****
              STOP
    END''OF COMPLETE.GAMMA
```

```
ROUTINE COMPLETE.GAMMA                    CACI SIMSCRIPT II.5 FOR PRIME SYSTEMS, RELEASE 2.1              PAGE   15
OPTIONS - SEQUENCE.ID.SUBCHK.XREF.NOEXPLIST.TRACE                                 DEC 1984  09:20:23

                         C R O S S - R E F E R E N C E

  NAME                       TYPE                              MODE      LINE NUMBERS OF REFERENCES

ABS.F                      ROUTINE                             INTEGER   34
COMPLETE.GAMMA             ROUTINE                             DOUBLE     1
ERR                        RECURSIVE VARIABLE   WORD   3       DOUBLE    12     32     34     37
GAMMA                      RECURSIVE VARIABLE   WORD   5       DOUBLE    13     22*    43*    49*    50
GY                         RECURSIVE VARIABLE   WORD   9       DOUBLE    47     49
L10                        LABEL                                         18     23
L110                       LABEL                                         19     28     41     46
L120                       LABEL                                         25     50
L130                       LABEL                                          9     35     38     51
L15                        LABEL                                         17     21
L50                        LABEL                                         15     24
L6                         LABEL                                         11
L60                        LABEL                                         30
L70                        LABEL                                         40     45
L80                        LABEL                                         31     43
REAL.F                     ROUTINE                             INTEGER   33
TRUNC.F                    ROUTINE                             INTEGER   33
X                          RECURSIVE VARIABLE   WORD   1       DOUBLE    11     14     18     21*    22     24     27
                                                                         30     33*    40     43     44*    46
XX                         ARGUMENT             NO.    1       DOUBLE     1      8     11     51
Y                          RECURSIVE VARIABLE   WORD   7       DOUBLE    33     34     37     46     47*    48*
```

END

FILMED

3-85

DTIC