DETECTION OF OUTLIERS

IN FAILURE DATA

by

DONALD ROBERT GALLUP

B.S., KANSAS STATE UNIVERSITY, 1979

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

Table of Contents

List of Figures

List of Tables

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING BEING SKEWED DIFFERENTLY FROM THE TOP OF THE PAGE TO THE BOTTOM.

THIS IS AS RECEIVED FROM THE CUSTOMER.

I. Introduction

In the safety analysis of nuclear power plants, one of the primary concerns is that of determining the failure rates of specified components. Frequently, the values for failure rates which are used are estimated from available failure data in the literature. Unfortunately, the available failure data for nuclear power plant components are usually sparse, and there are often large variations in the data for similar components. Thus, the estimated values for failure rates have large uncertainties.

One method which can be used to circumvent this problem with the data is to label data points which deviate greatly from the main body of data as "outliers." Once data points have been labeled as outliers, steps can be taken to reduce the effect which they have on the statistical interpretation of the remaining data. These steps can range from simply noting that outliers are present in order to reduce the confidence placed in the results, to weighting the data in order to reduce the effects of the outliers, or to rejecting the outliers before the data are interpreted.

The problem of outliers or "discordant data" has been around for decades. It is one of the most perplexing problems which occurs in the interpretation of data, because all elements of subjectivity must be removed, and because the question of what to do with discordant data once they are found has no concrete answer. Ferguson (1961) stated the problem as follows:

> "In a sample of moderate size taken from a certain population, it appears that one or two observations are surprisingly far away from the main group. The experimenter is tempted to throw away the apparently erroneous values, and not because he is certain that the values are spurious. On the contrary, he will undoubtedly admit that even if the population has a normal distribution there is a positive although extremely small probability that such values will occur in an experiment. It is rather because he feels that other explanations are more plausible, and

> that the loss in the accuracy of the experiment caused by throwing away good values is small compared to the loss caused by keeping even one bad value. The problem, then, is to introduce some degree of subjectivity into the rejection of outlying observations."

In this study, the specific problem of discordant data in failure rate data with small failure rates is addressed. The first section of the paper deals with the idealized case in which the failure rates themselves are known and are assumed to be distributed according to the gamma distribution. Several methods of outlier detection for the gamma distribution are presented, and their properties and comparisons between the methods are given.

The second section of the paper deals with the more practical case in which the data are in the form $(F_i,T_i,)$, where F_i and T_i are the number of failures is assumed to be distributed according to the Poisson distribution, and the failure rates of similar components are assumed to be distributed according to the gamma distribution. Two tests for detecting outliers in data distributed according to the compound model are given, and their properties and comparisons between the two methods are then presented.

In the third section of the paper, the problem of discordant values in time-to-failure data, i.e., data which are distributed according to the exponential distribution, is discussed. Two methods of detecting outliers in this type of data are presented, and their properties and comparisons between the two methods are given.

In the last section of the paper, an overview of the problem of outliers in failure rate data is given, and the "best" methods for outlier detection for the cases discussed are presented.

II. Outliers in a Gamma Distribution

II.1 Introduction

The gamma distribution,

$$g(x|\alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} , \quad x > 0 , \qquad (2.1)$$

is one of the prior distributions most frequently used with failure rate-data. The reason for this is twofold. First, the gamma-distribution can assume a wide variety of shapes, depending upon the selected values of its parameters. For a value of the parameter α less than 1, the gamma distribution assumes a $\frac{1}{x}$ -type shape. For α equal to 1, it becomes an exponential curve. And for α greater than 1, the gamma-distribution takes on a unimodal-type shape (See Fig. 2.1). Thus, the gamma-distribution can be seen, intuitively at least, as a distribution which can approximate almost any shape which failure rate data might assume. The second reason for the frequent use of the gamma distribution is the mathematical convenience which it affords by being the conjugate of the Poisson distribution. For a component which is in continuous operation, the probability of having F failures in time T can be modeled by the Poisson distribution,

$$f(F|\lambda,T) = \frac{(\lambda T)^{F}}{\Gamma(F+1)} e^{-\lambda T} , \qquad (2.2)$$

where λ is the component's failure rate. In the compound model, λ is not treated as a constant, but rather as a random variable. If the failure rate is assumed to be distributed according to the gamma-distribution then the number of failures F in time T is found to be

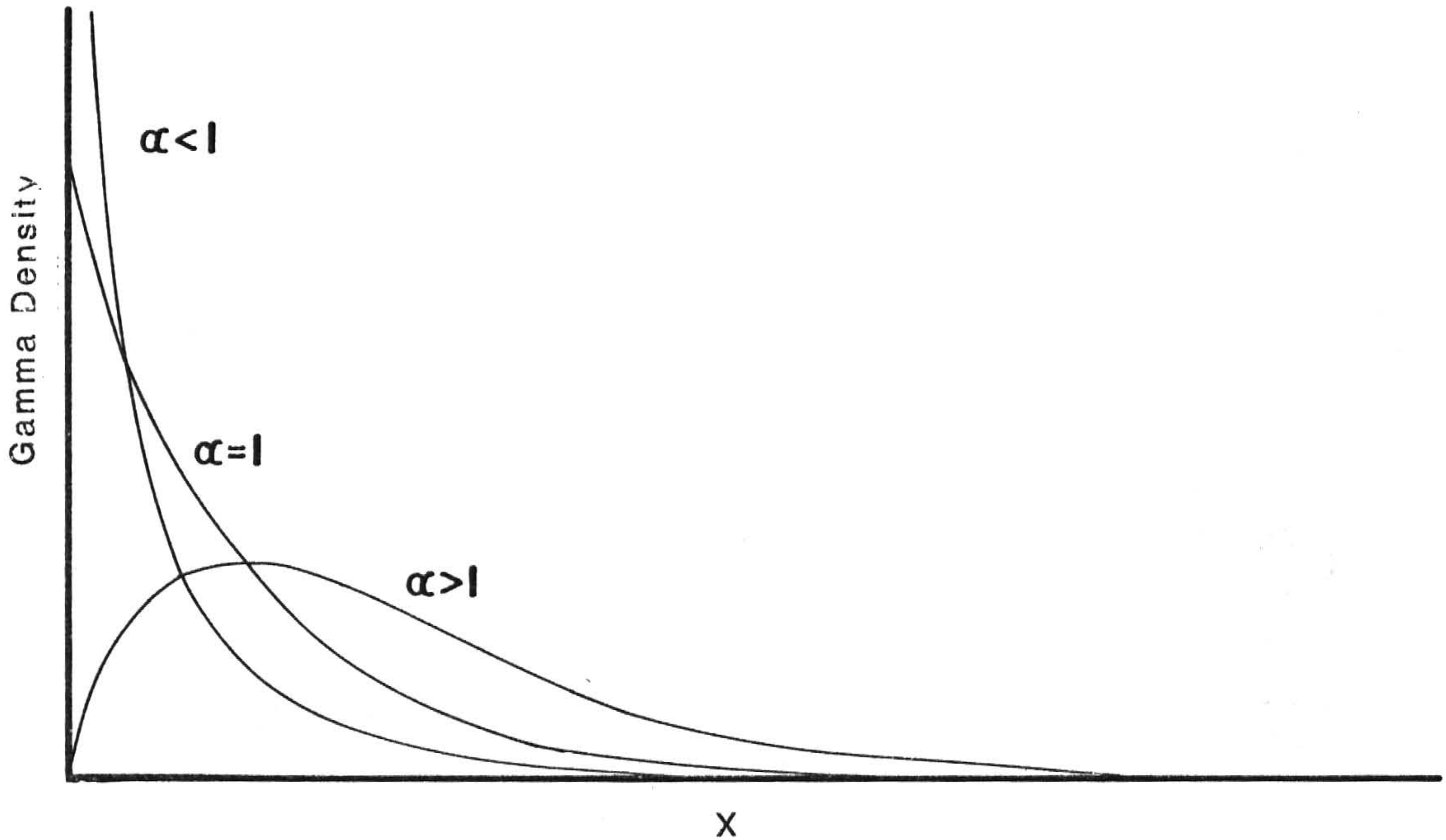


Figure 2.1: Variation of the shape of the gamma distribution with the alpha parameter.

$$h(F|T) = \int_0^\infty f(F|\lambda,T)g(\lambda|a,\beta)d\lambda = \frac{\Gamma(F+\alpha)}{\Gamma(\alpha)\Gamma(F+1)} \; \frac{T^F\beta^\alpha}{(T+\beta)^{F+\alpha}} ,$$

$$= 0, 1, 2 \ldots \quad . \qquad (2.3)$$

Thus, with the use of the gamma-distribution, complicated analytical techniques to find $h(F|T)$ can be avoided.

Under ideal conditions, i.e., laboratory conditions, the distribution of failure rate data for nuclear power plant components could be modeled by the gamma-distribution. Unfortunately no such data exist, and as a result, data from the literature which are plagued with wide variations must be used to approximate failure rates. It is appropriate, therefore, to begin a study of outliers or discordant data in the failure rate data, for nuclear power plant components with "semi-ideal" failure rate data, i.e., data in which the main body of data is known to come from a gamma distribution, and to which outliers have been added.

This chapter is divided into seven main sections. The first section deals with a method of detecting outliers in a gamma sample which is based on a methodology derived by Fisher(1929). The second section deals with a detection method in which the gamma sample is transformed into a normal sample before outlier detection procedures are applied. In the third section, the detection method involves the integration of the gamma-distribution to determine which data, if any, are too large. In the fourth and fifth sections, the problems of 'masking' and 'swamping' are discussed and a method for determining the number of outliers to be tested for is explained. The sixth section deals with power curves. And in the last section, the properties of the tests are presented.

II.2 Fisher's Method of Outlier Detection

Fisher (1929) developed a simple method for detecting a single outlier in a gamma-distribution, $g(\alpha,\beta)$, with known parameter α in 1929. His method has been expanded upon since that time, notably by Fieller(1976) who generalized the method to include the case of multiple outliers. Fisher's method of detecting outliers is simply a hypothesis test. The null hypothesis, H_o, is that all data come from a single gamma-distribution with a known constant α. The alternate hypothesis, H_1, is that a certain number of data points, $k < n$, come from a distribution(s) different from that of the main body of data.

The application of Fisher's method for detecting outliers is straight forward. It merely consists of calculating a test statistic, T_F, and comparing this statistic with the tabulated, critical values of the statistic, t_F. If T_F is greater than t_F, then the null hypothesis is rejected, and the data points under scrutiny are declared to be discordant.

The test statistic for upper outliers is

$$T_F = \frac{x_n + x_{n-1} + \ldots + x_{n-k+1}}{\sum_{i=1}^{n} x_i}, \qquad (2.4)$$

where $x_n \geq x_{n-1} \geq \ldots \geq x_1$ are the data and k is the number of outliers being tested for. In examples 2.1 and 2.2, T_F is calculated for the single and multiple outlier cases.

The critical values of the test statistic are found from the equation (Barnett and Lewis, 1978)

$$P(T_F \leq t_F) \leq \binom{n}{k} P\left[F_{2k\alpha,2(n-k)\alpha} < \frac{(n-k)t_F}{k(1-t_F)}\right], \tag{2.5}$$

where $P(T_F \leq t_F)$ is the probability that the calculated value of Fisher's test statistic, T_F, is less than its critical value, t_F, n is the number of data points, k is the number of outliers being tested for, and $F_{2k\alpha,2(n-k)\alpha}$ is the F-distribution with $2k\alpha$ and $2(n-k)\alpha$ degrees of freedom. There is no "right" or "wrong" way to select the value of $P(T_F \leq t_F)$)which is used in determining t_F. However, it is standard practice to use either 0.01 or 0.05. In doing so, the values of t_F at either the 99% or the 95% confidence level, respectively, are obtained.

Since tables of the critical values of the test statistic for multiple outliers were not found to exist, tables for these values were generated (see App. B and Tables 2.1 - 2.5). Comparing Table 2.1 to Table I in Barnett and Lewis (1978), the single outlier case, shows that the generated values are quite accurate. All generated values are within 0.5% and almost all are within 0.01% of the values given in Barnett and Lewis, and, since all generated values are greater than or equal to the already tabulated values, the generated values are seen to be slightly conservative.

Example 2.1:

The following data are known to come from a gamma-distribution with parameters $\alpha = 1.5$ and $\beta = 100$.

.00289, .00478, .00487, .00591, .00849, .0167, .0197, .0263, .0454, .973

NO. OF DATA PTS. (N)	ALPHA 0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	6.00
5	0.9603	0.8414	0.6838	0.5981	0.5440	0.5063	0.4783	0.4564	0.4387	0.3920
6	0.9278	0.7809	0.6162	0.5321	0.4803	0.4446	0.4184	0.3980	0.3817	0.3387
7	0.8919	0.7272	0.5612	0.4800	0.4307	0.3972	0.3726	0.3536	0.3384	0.2987
8	0.8557	0.6800	0.5157	0.4377	0.3910	0.3593	0.3362	0.3185	0.3043	0.2674
9	0.8206	0.6387	0.4775	0.4028	0.3584	0.3285	0.3067	0.2901	0.2768	0.2424
10	0.7874	0.6023	0.4450	0.3733	0.3311	0.3028	0.2823	0.2666	0.2541	0.2218
12	0.7273	0.5413	0.3924	0.3264	0.2880	0.2624	0.2440	0.2299	0.2187	0.1900
14	0.6753	0.4922	0.3517	0.2907	0.2554	0.2320	0.2153	0.2025	0.1924	0.1664
16	0.6303	0.4520	0.3193	0.2624	0.2298	0.2083	0.1929	0.1812	0.1719	0.1483
18	0.5913	0.4184	0.2927	0.2395	0.2092	0.1892	0.1750	0.1641	0.1556	0.1338
20	0.5571	0.3898	0.2705	0.2205	0.1921	0.1735	0.1602	0.1502	0.1422	0.1221
25	0.4879	0.3341	0.2281	0.1846	0.1601	0.1441	0.1328	0.1242	0.1174	0.1003
30	0.4352	0.2933	0.1980	0.1593	0.1377	0.1236	0.1137	0.1061	0.1003	0.0853
35	0.3930	0.2621	0.1753	0.1404	0.1210	0.1085	0.0996	0.0929	0.0876	0.0744
40	0.3593	0.2369	0.1575	0.1258	0.1082	0.0968	0.0887	0.0827	0.0779	0.0660
50	0.3080	0.2001	0.1315	0.1044	0.0895	0.0799	0.0731	0.0680	0.0640	0.0540
60	0.2704	0.1737	0.1132	0.0896	0.0765	0.0682	0.0623	0.0579	0.0544	0.0458
80	0.2189	0.1385	0.0892	0.0701	0.0597	0.0530	0.0483	0.0448	0.0421	0.0353
100	0.1849	0.1157	0.0739	0.0579	0.0491	0.0435	0.0396	0.0367	0.0344	0.0288

Table 2.1: Critical values of Fisher's test statistic for a single upper outlier in a gamma distribution at the 95% confidence level.

NO. OF DATA PTS. (N)	ALPHA 0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	6.00
6	0.9932	0.9423	0.8335	0.7606	0.7230	0.6743	0.6463	0.6241	0.6058	0.5563
7	0.9852	0.9108	0.7827	0.7042	0.6522	0.6150	0.5867	0.5645	0.5463	0.4976
8	0.9742	0.8788	0.7372	0.6557	0.6030	0.5657	0.5377	0.5158	0.4980	0.4506
9	0.9611	0.8475	0.6967	0.6137	0.5610	0.5242	0.4967	0.4753	0.4579	0.4121
10	0.9461	0.8175	0.6604	0.5770	0.5249	0.4887	0.4619	0.4410	0.4242	0.3759
12	0.9141	0.7625	0.5987	0.5162	0.4658	0.4313	0.4058	0.3862	0.3705	0.3293
14	0.8812	0.7139	0.5481	0.4678	0.4195	0.3866	0.3626	0.3442	0.3294	0.2911
16	0.8488	0.6712	0.5060	0.4283	0.3821	0.3509	0.3282	0.3109	0.2970	0.2612
18	0.8178	0.6335	0.4704	0.3954	0.3512	0.3216	0.3002	0.2838	0.2708	0.2372
20	0.7883	0.5998	0.4398	0.3675	0.3253	0.2971	0.2768	0.2613	0.2490	0.2174
25	0.7224	0.5307	0.3795	0.3135	0.2755	0.2504	0.2324	0.2187	0.2079	0.1803
30	0.6664	0.4769	0.3347	0.2742	0.2397	0.2171	0.2009	0.1886	0.1790	0.1545
35	0.6187	0.4338	0.3001	0.2442	0.2126	0.1920	0.1773	0.1662	0.1575	0.1353
40	0.5778	0.3984	0.2724	0.2205	0.1913	0.1724	0.1589	0.1487	0.1408	0.1206
50	0.5109	0.3437	0.2308	0.1853	0.1600	0.1436	0.1320	0.1233	0.1165	0.0993
60	0.4591	0.3032	0.2010	0.1604	0.1379	0.1235	0.1133	0.1056	0.0996	0.0846
80	0.3835	0.2468	0.1606	0.1271	0.1087	0.0969	0.0887	0.0825	0.0777	0.0656
100	0.3309	0.2092	0.1345	0.1058	0.0901	0.0802	0.0732	0.0679	0.0639	0.0537

Table 2.2: Critical values of Fisher's test statistic for two upper outliers in a gamma distribution at the 95% confidence level.

NO. OF DATA PTS. (N)	ALPHA 0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	6.00
9	0.9910	0.9347	0.8243	0.7521	0.7029	0.6670	0.6396	0.6178	0.5999	0.5513
10	0.9854	0.9142	0.7912	0.7149	0.6641	0.6275	0.5998	0.5778	0.5598	0.5115
12	0.9708	0.8730	0.7318	0.6509	0.5987	0.5617	0.5340	0.5123	0.4946	0.4477
14	0.9531	0.8333	0.6807	0.5979	0.5456	0.5091	0.4820	0.4608	0.4438	0.3987
16	0.9336	0.7962	0.6365	0.5533	0.5017	0.4661	0.4398	0.4194	0.4030	0.3599
18	0.9131	0.7617	0.5980	0.5154	0.4648	0.4302	0.4048	0.3851	0.3694	0.3283
20	0.8924	0.7300	0.5641	0.4827	0.4334	0.3999	0.3753	0.3564	0.3413	0.3021
25	0.8414	0.6611	0.4952	0.4176	0.3717	0.3408	0.3184	0.3012	0.2876	0.2524
30	0.7939	0.6045	0.4424	0.3691	0.3263	0.2979	0.2773	0.2616	0.2492	0.2174
35	0.7506	0.5574	0.4005	0.3313	0.2914	0.2650	0.2461	0.2317	0.2203	0.1912
40	0.7116	0.5176	0.3664	0.3011	0.2637	0.2392	0.2216	0.2082	0.1977	0.1710
50	0.6445	0.4541	0.3142	0.2555	0.2224	0.2007	0.1853	0.1737	0.1646	0.1414
60	0.5894	0.4054	0.2758	0.2226	0.1928	0.1735	0.1598	0.1495	0.1414	0.1209
80	0.5048	0.3357	0.2231	0.1781	0.1532	0.1372	0.1259	0.1174	0.1108	0.0942
100	0.4427	0.2678	0.1882	0.1491	0.1277	0.1140	0.1044	0.0971	0.0915	0.0774

Table 2.3: Critical values of Fisher's test statistic for three upper outliers in a gamma distribution at the 95% confidence level.

NO. OF DATA PTS. (N)	ALPHA 0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	6.00
12	0.9900	0.9320	0.8216	0.7497	0.7008	0.6652	0.6379	0.6162	0.5984	0.5501
14	0.9810	0.9019	0.7740	0.6965	0.6453	0.6086	0.5808	0.5589	0.5411	0.4932
16	0.9698	0.8720	0.7313	0.6505	0.5983	0.5614	0.5337	0.5119	0.4943	0.4474
18	0.9571	0.8428	0.6930	0.6105	0.5580	0.5214	0.4940	0.4727	0.4555	0.4098
20	0.9433	0.8150	0.6586	0.5754	0.5232	0.4870	0.4602	0.4394	0.4224	0.3784
25	0.9065	0.7517	0.5866	0.5040	0.4536	0.4192	0.3939	0.3745	0.3589	0.3183
30	0.8691	0.6970	0.5296	0.4454	0.4014	0.3689	0.3453	0.3272	0.3128	0.2754
35	0.8328	0.6499	0.4835	0.4062	0.3607	0.3301	0.3080	0.2911	0.2777	0.2431
40	0.7985	0.6089	0.4453	0.3712	0.3279	0.2991	0.2784	0.2626	0.2501	0.2179
50	0.7366	0.5416	0.3856	0.3175	0.2785	0.2527	0.2342	0.2202	0.2092	0.1811
60	0.6832	0.4887	0.3411	0.2783	0.2427	0.2194	0.2028	0.1902	0.1803	0.1553
80	0.5971	0.4104	0.2786	0.2245	0.1942	0.1746	0.1607	0.1503	0.1421	0.1214
100	0.5311	0.3553	0.2366	0.1890	0.1627	0.1457	0.1337	0.1247	0.1177	0.1000

Table 2.4: Critical values of Fisher's test statistic for four upper outliers in a gamma distribution at the 95% confidence level.

NO. OF CATA PTS. (N)	ALPHA 0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	6.CC
14	0.9923	0.9426	0.8411	0.7726	0.7253	0.6905	0.6637	0.6423	0.6247	0.5767
16	C.9862	0.9194	0.8017	0.7274	0.6774	0.6413	0.6137	0.5918	0.5740	0.5257
18	0.9785	0.8957	0.7654	0.6871	C.6357	0.5989	0.5710	0.5491	0.5313	0.4834
20	0.9697	0.8723	0.7320	0.6512	0.5990	0.5620	0.5342	0.5125	0.4948	0.4478
25	0.9438	0.8163	0.6602	0.5767	0.5244	0.4880	0.4611	0.4402	0.4233	0.3789
30	0.9152	0.7658	0.6016	0.5183	0.4673	0.4323	0.4066	0.3867	C.3708	0.3293
35	0.8859	0.7206	0.5532	0.4714	0.4221	0.3887	0.3642	0.3455	0.3305	0.2916
40	0.8570	0.6804	0.5125	0.4328	0.3855	0.3536	0.3304	0.3126	0.2985	0.2621
50	0.8022	0.6125	0.4478	0.3730	0.3294	0.3004	0.2795	0.2635	C.2509	0.2185
60	0.7527	0.5576	0.3986	0.3287	C.2885	0.2619	0.2428	0.2284	0.2170	0.1879
80	0.6691	0.4743	0.3285	0.2671	0.2323	0.2096	0.1935	0.1813	C.1717	0.1475
100	0.6025	0.4142	0.2807	0.2260	C.1954	0.1755	0.1615	0.1509	0.1427	0.1218

Table 2.5: Critical values of Fisher's test statistic for five upper outliers in a gamma distribution at the 95% confidence level.

Determine if the largest value should be labeled an outlier at the 95% confidence level.

From Eq. 2.4,

$$T_F = \frac{.973}{.00289 + .00478 + \ldots + .973} = .878.$$

From Table 2.1, the critical value of Fisher's test statistic for $\alpha = 1.5$, $n = 10$, and $k = 1$ is found to be t_F = .3733. Since $T_F > t_F$, i.e., .878 > .3733, the value .973 is seen to be discordant at the 95% confidence level.

Example 2.2:

The following data are known to come from a gamma-distribution with parameters $\alpha = 1.25$ and $\beta = 1000$.

.000152, .000324, .000360, .000592, .000696, .00156, .00179, .00219, .875, 1.37

Determine if the two largest values are discordant at the 95% confidence level.

From Eq. 2.4,

$$T_F = \frac{137 + .875}{.000152 + \ldots + 1.37} = .997.$$

From Table 2.2, the critical value of Fisher's test statistic for $\alpha = 1.25$, $n = 10$, and $k = 2$ is found by Bessel interpolation (Bajpai, 1974) to be $t_F = 0.612$. Thus, the values .875 and 1.37 are discordant at the 95% confidence level.

II.3 The Normal Conversion Method of Outlier Detection

The method of normal conversion consists essentially of converting a gamma sample into a normal sample, and then testing the normal sample for outliers.

If the random variable X is distributed according to the gamma-distribution $g(x|\alpha,\beta), \alpha>1$, it can be transformed into the normal distribution $N[(\alpha\beta)^{1/3}(1-\frac{1}{9\alpha}), \frac{1}{-}(\frac{\beta^2}{\alpha})^{1/3}]$ by taking the cube root of X (Kimber, 1979). Thus, if $w_i = x_i^{1/3}$ are data where x_i are distributed according to a gamma-distribution with the parameter α greater than 1, then w_i are distributed according to a normal distribution.

Example 2.3:

Convert the data from Example 2.1 into normally distributed data.

$$w_1 = x_1^{1/3} = .00289^{1/3} = 0.142$$
$$w_2 = x_2^{1/3} = .00478^{1/3} = 0.168$$
$$\vdots$$
$$w_{10} = x_{10}^{1/3} = 0.973^{1/3} = 0.991$$

Once the data have been transformed to normally distributed data, there are a number of methods available for detecting outliers. One of the more common methods, both because of its ease of application and intuitive appropriateness, is the use of Grubbs-type statistics.

Grubbs-type statistics, like the Fisher statistics, are based upon a hypothesis test. The null hypothesis, H_o, in this case is that all data come from a single normal distribution. The alternate hypothesis, H_1, is that a certain number of the data points, $k < n$, come from a distribution(s) distinct from the normal distribution which describes the main body of data.

The Grubbs test statistic used to test the null hypothesis is calulated as follows (Grubbs, 1950)

$$T_N = \frac{w_n + w_{n-1} + \ldots + w_{n-k+1} - k\bar{w}}{S_w}, \tag{2.5}$$

where $w_1 \leq \ldots \leq w_n$ are the normalized data, k is the number of outliers, and S_w is the standard deviation of the normalized data. Values for T_N are calculated for the single and multiple outlier cases in Examples 2.4 and 2.5, below.

The critical values of the test statistic are found from the equation (Barnett and Lewis, 1978)

$$P(T_N \geq t_N) \leq \binom{n}{k}\{t_{n-2} > [\frac{n(n-2)t_N^2}{k(n-k)(n-1)-nt_N^2}]^{1/2}\}, \tag{2.6}$$

where $P(T_N \geq t_N)$ is the probability that the Grubbs test statistic, T_N, is greater than or equal to its critical value t_n, t_{n-2} is Student's t-distribution with n-2 degrees of freedom, and t_N is the critical value of T_N for a given $P(T_N \geq t_N)$. As before, the value of $P(T_N \geq t_N)$, selected to determine the critical value of T_N, is customarily either 0.01 or 0.05, so that either the 99% or 95% confidence level, respectively, is used. Table 2.6 gives the critical values of T_N.

Example 2.4:

Using the data from Ex. 1, determine if the largest value is an outlier at the 95% confidence level according to the conversion to normal method.

The converted data are:

.142, .168, .170, .181, .204, .256, .270, .297, .357, .991.

$$\bar{w} = \frac{1}{10} \sum_{i=1}^{10} w_i = \frac{1}{10} (.142 + .168 + \ldots + .991) = .304$$

$$S_w = [\sum_{i=1}^{10} \frac{(w_i - w)^2}{n-1}]^{1/2} = \frac{1}{3}[(.142 - .304)^2 + \ldots$$

$$+ (.991 - .304)^2]^{1/2}$$

$$= .251$$

From Eq. 2.5,

$$T_N = \frac{.991 - .304}{.251} = 2.74$$

From Table 2.6, the critical value of the test statistic for n = 10 and k = 1 is found to be 2.18. Thus, .937 ($.991^3$) is labeled as an outlier.

Table 2.6: Critical values of the Grubbs test statistic, T, at the 95% confidence level. Taken from Barnett and Lewis (1974), pp. 298 and 304.

No. of Data Pts. (n)	Number of Outliers (K) 1	2	3	4
5	1.67	2.10		
6	1.82	2.41		
7	1.94	2.66	2.97	
8	2.03	2.87	3.29	
9	2.11	3.04	3.58	3.82
10	2.18	3.18	3.82	4.17
12	2.29	3.44	4.24	4.72
14	2.37	3.66	4.57	5.20
16	2.44	3.83	4.85	5.60
18	2.50	3.96	5.08	5.91
20	2.56	4.11	5.30	6.22
30	2.74	4.56	6.03	7.26
40	2.87	4.84	6.49	7.93
50	2.96	5.06	6.82	8.38
100	3.21	5.62	7.77	9.71

Example 2.5:

Using the data from Ex. 2.2, determine if the two largest values are discordant according to the conversion to normal method.

The converted data are:

.0534, .0687, .0711, .0840, .0886, .116, .121, .130, .956, 1.11.

$$\overline{w} = \frac{1}{10}(.0534 + \ldots + 1.11) = .280$$

$$S_w = \frac{1}{3}[(.0534 - .280)^2 + \ldots + (1.11 - 280)^2]^{1/2} = .399$$

From Eq. 2.3

$$T_n = \frac{1.11 + .956 - 2(.280)}{.399} = 3.77$$

From Table 2.6, the critical value of the test statistic for n=10 and k=2 is found to be 3.18. Thus, the two largest values are found to be discordant.

II.4 Integration Method of Outlier Detection

A third method for locating outliers in a gamma-distribution, which is not handled in the literature is a method involving the integration of the gamma-distribution in question. As was the case with the two previous methods, this method is a hypothesis test, where the null hypothesis, H_o, is that all data come from a single gamma-distribution. The alternate hypotheses, H_1, is that a few data points, $k < n$, are from a different distribution(s) than the main body of data.

The integration method is developed as follows. The probability that a point from the gamma-distribution $g(\alpha,\beta)$ is greater than a given value x_c is given by

$$p_i = P(X_i \geq x_c) = \int_{x_c}^{\infty} dG(\alpha,\beta) \qquad (2.7)$$

The probability that any one of n data points is greater than x_c (assuming independence among the data) is given by

$$P(\bigvee_{i=1}^{n} A_i) = 1-P(\bigwedge_{i=1}^{n} A_i') = 1-\prod_{i=1}^{n} (1-p_i), \qquad (2.8)$$

where A_i is the event $X_i > x_c$, A_i' is the complement of A_i, $\vee$ is the union symbol, and $\wedge$ is the intersection symbol. And if, according to the null hypothesis, all data come from the same distribution, then $p_1 = p_2 = \ldots = p_n \equiv p$, and thus, from Eq. 2.8

$$P(\bigvee_{i=1}^{n} A_i) = 1-(1-p)^n \qquad (2.9)$$

$$p = 1-[1-P(\bigvee_{i=1}^{n} p_i)]^{1/n} \qquad (2.10)$$

From Eq. 2.5,

$$\int_{x_c}^{\infty} dG(\alpha,\beta) = p \qquad (2.11)$$

and

$$x_c = G^{-1}(1-p;\alpha,\beta) \qquad (2.12)$$

As in the previous cases, the 95% and 99% confidence levels are of interest, and so p is given either the value $1-.95^{1/n}$ or $1-.99^{1/n}$, accordingly.

It is not necessary to solve Eq. 2.11 exactly for the detection of outliers. Rather, the data points $x_n, x_{n-1} \ldots$, where $x_n \geq x_{n-1} \geq \ldots$, can be used as approximations of x_c. These values are substituted into Eq. 2.11, beginning with x_n, until a value x_{n-k+1} is reached such that

$$\int_{x_{n-k+1}}^{\infty} dG(\alpha,\beta) < p.$$

The upper k values are then labeled as outliers.

Example 2.6:

Using the data from Example 2.1, determine if the largest value is to be labeled as an outlier, at the 95% confidence level.

From Eq. 2.12,

$$^{*}x_c = G^{-1}(.95^{1/10};\ 1.5,\ 100) = .0585.$$

Thus, the largest value is labeled as an outlier, because $.973 > .0585$.

Example 2.7:

Using the data from Ex. 2, determine if the two largest values are discordant at the 95% confidence level.

From Eq. 2.12,

$$^{*}x_c = G^{-1}(.95^{1/10};\ 1.25,\ 1000) = .00585$$

Thus, the two largest values are labelled as outliers, because $1.37 > .875 > .00585$

---

*The value of x_c was found using the function subprogram "PRCNT" (see Appendix C).

II.5 Estimation of the Values of the Parameters α and β

One problem which has been ignored so far is the determination of the values of the parameters α and β in the gamma-distribution. Until now, the values of α and β have been assumed to be known.

When Fisher's test statistic is being applied to the data, the following algorithm can be used to determine the value of α to be used in finding the critical value of the test statistic. First $\hat{\alpha}$ is calculated from the non-suspicious data as follows.

$$\bar{x}' = \frac{1}{n-k} \sum_{i=1}^{n-k} x_i \tag{2.13}$$

$$s'^2 = \frac{1}{n-k-1} \sum_{i=1}^{n-k} (x_i - \bar{x}') \tag{2.14}$$

$$\hat{\alpha}' = \bar{x}'^2 / s'^2 \tag{2.15}$$

where $x_1 \leq x_2 \leq \ldots \leq x_n$ are the suspected number of upper outliers. Then, the minimum outlier value, i.e., the minimum value at which a data point is labeled an outlier, is found using α' to determine the critical value of the test statistic.

$$x_c' = \frac{t_F' \sum_{i=1}^{n-k} x_i}{1-t_F'}\ , \tag{2.16}$$

where t_F' is the critical value of Fisher's test statistic at α'. Use the minimum outlier value to estimate, α is calculated as follows:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n-k} x_i + \frac{k}{n} x_c'\ , \tag{2.17}$$

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n-k}(x_i-\bar{x})^2 + \frac{k}{n-1}(x'_c - \bar{x})^2 , \qquad (2.18)$$

$$\hat{\alpha} = \bar{x}^2 \, s^2 \ . \qquad (2.19)$$

The above value of α is used to find the critical value of the test statistic; the procedure described in section II.2 is used to determine whether or not the upper k data points are discordant.

Example 2.8:

The following data come from a gamma-distribution with unknown parameters. .000313, .000560, .000852, .000862, .000898, .000971, .00107, .00198, .00223, .846

The largest value is suspected of being an outlier. Determine if it is labeled as an outlier at the 95% confidence level.

From Eq.'s 2.13, 2.14, and 2.15,

$$\bar{x}'^2 = \frac{1}{9} = (.000313 + \ldots + .00223) = .00108,$$

$$s'^2 = \frac{1}{8}\left[(.000313 \qquad) + \ldots + (.00223 \quad)\right] = 3.92\text{x}10^{-7}$$

$$\hat{\alpha}' = \bar{x}'^2 \, s'^2 = .00108^2 \ 3.92\text{x}10^{-7} = 2.99$$

From Table 2.1, the critical value of Fisher's test statistic for $\alpha = 3$, $n = 10$, and $k = 1$ is found to be $t'_F = .2823$.

From Eq. 2.6, the minimum outlier value for the 9 lower values is

$$x'_c = \frac{t'_F \sum_{i=1}^{n-k} x_i}{1-t'_F} = \frac{(.2823)(.009736)}{1-.2823} = .00384.$$

And from Eq.'s 2.17, 2.18, and 2.19,

$$\bar{x} = \frac{1}{10}(.000313 + \ldots + .00136 + .00384) = .00136$$

$$s^2 = \frac{1}{9}\left[(.000313-.00136)^2 + \ldots + (.00384-.00136)^2\right] = 1.11 \times 10^{-6},$$

$$\hat{\alpha} = .00136^2 \; 1.11\text{x}10^{-6} = 1.66 \; .$$

From Table 2.1, the critical value of Fisher's test statistic for $\alpha = 1.66$, $n = 10$, and $k = 1$ is found by Bessel interpolation to be $t_F = .3573$.

Using Eq. 2.3 to calculate Fisher's test statistic gives

$$T_F = \frac{.846}{.000313 + \ldots + .00223 + .846} = .989$$

Since $T_F > t_F$, i.e., $.989 > .3573$, the value .846 is labeled as an outlier at the 95% level.

When the integration method is being applied to locate outliers the algorithm used to estimate the values of α and β is very similar to the algorithm used for Fisher's method. First, the data points which are <u>not</u> suspected as being discordant are used to calculate α' and β':

$$\bar{x}' = \frac{1}{n-k}\sum_{i=1}^{n-k} x_i, \tag{2.20}$$

$$s'^2 = \frac{1}{n-k-1}\sum_{i=1}^{n-k} (x_i-\bar{x}')^2, \tag{2.21}$$

$$\hat{\alpha}' = \bar{x}'^2/s'^2 \tag{2.22}$$

$$\hat{\beta}' = \bar{x}'/s'^2 \tag{2.23}$$

These values of α' and β' are then used to determine the minimum outlier value for the n-k data points as follows:

$$\int_{o}^{x'_c} g(x;\alpha',\beta')\,dx = G(x'_c;\alpha,\beta'), \qquad (2.24)$$

$$G(x'_c;\alpha',\beta') = P, \qquad (2.25)$$

$$x'_c = G^{-1}(P;\alpha',\beta')\ , \qquad (2.26)$$

where P is either $0.95^{1/n}$ or $0.99^{1/n}$ depending on whether the 95% or 99% confidence level, respectively, is being used.

Then, using the n-k non-suspicious data points and x'_c, the values of α and β, which are used to test the upper k data points for discordancy, are estimated as follows:

$$x = {}_{n}\sum_{i=1}^{n-k} x_i + {}_{n}\, x'_c, \qquad (2.27)$$

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n-k}(x_i - \bar{x})^2 + \frac{k}{n-1}(x'_c - \bar{x})^2; \qquad (2.28)$$

$$\hat{\alpha} = \bar{x}^2/s^2, \qquad (2.29)$$

$$\hat{\beta} = \bar{x}/s^2. \qquad (2.30)$$

Once $\hat{\alpha}$ and $\hat{\beta}$ have been calculated, the procedure described in section II.4 is used to determine if the upper k data points are, in fact, discordant.

Example 2.9

The following data come from a gamma-distribution with unknown parameters.

.0000332, .000329, 000313, .000416, .000560, .000628, .000771, .000852, .000826, .000898, .000971, .00104, .00107, .00128, .00131, .00165, .00176, .00198, .00223, .365

Determine if the largest value is discordant at the 95% confidence level.

From Eq.'s 2.20 through 2.23,

$$\bar{x}' = \frac{1}{19} (.0000332 + \ldots + .00223) = .00101$$

$$s'^2 = \frac{1}{18}[(.0000332-.00101)^2 + \ldots + (.00223-00101)^2] = 3.43\text{x}10^{-7}$$

$$\hat{\alpha}' = \bar{x}'^2 / s'^2 = .00101^2 / 3.43\text{x}10^{-7} = 2.96,$$

From Eq. 2.26

$$^*x_c' = G^{-1}(.95^{1/20}; 2.96, 2936) = .00341$$

From Eq.'s 2.27, 2.28, 2.29, and 2.30

$$\bar{x} = \frac{1}{20} (.0000332 + \ldots + .00223 + .00341) = .00113$$

$$s^2 = \frac{1}{19}[(.0000332-.00113)^2 + \ldots + (.00341-.00113)^2]$$

$$= 6.14\ \text{x}10^7$$

$$\hat{\alpha} = \bar{x}^2 / s^2 = 2.07$$

$$\hat{\beta} = \bar{x} / s^2 = 1838$$

---

*The values of x_c were found using the function subprogram 'PRCNT" (see Appendix C).

Thus from Eq. 2.12

$$^{*}x_c = G^{-1}(.95^{1/20};2.07,\ 1838) = .00453.$$

Hence, the largest value is labeled an outlier, because $x_{20} > x_c$, i.e., $.365 > .00453$

II.6 Determination of the Number of Outliers

Another problem, which has not yet been addressed, yet which is of great importance in the detection of outliers, is the determination of the number of outliers to be tested for. Until now, the number of outliers in the data was assumed to be known.

One method for determining the number of outliers present is to do repetitive testing on a set of data, assuming for each test that one outlier may be present. That is, the data are tested first assuming the most extreme value may be discordant. If that value is not found to be discordant, then there are no outliers present, and the testing is complete. However, if the most extreme value is found to be an outlier, it is removed from the data, and the test is repeated assuming that the second most extreme value may be an outlier. If this value is found to belong to the distribution being tested, then the testing is complete, and there is one outlier in the data. However, if this value is found to be discordant, it is removed from the data, and the third most extreme value is tested. This process is continued until a non-discordant value is found. If k+1 repetitions are necessary to find a non-discordant value, then there are k outliers in the data.

---

*The values of x_c' were found using the function subprogram 'PRCNT' (see Appendix C).

At first glance, the repetition method for determining the number of outliers in a set of data seems quite good. However, there is one major pitfall in its application. The pitfall is the phenomenon known as masking, and is explained as follows. Suppose a set of data with two upper outliers is to be tested. And further suppose that the outlier detection technique to be applied is of the form $\frac{D}{S}$ where D is a measure of the separation of the most extreme value from the mean, and S is a measure of the spread of the data. Obviously, the greater the value of the test statistic $\frac{D}{S}$, the more likely it is that the data point under scrutiny will be detected as an outlier. In this case, however, the second most extreme value is also an outlier, and thus will cause the value of S to be larger and the value of $\frac{D}{S}$ to be smaller than anticipated. As a result, the most extreme value may not be detected as being an outlier, and thus two outliers would go undetected.

<u>Example 2.10:</u>

The following data come from a gamma-distribution with parameter $\alpha = 1.5$. .00289, .00478, .00487, .00591, .00849, .0167, .0197, .0263, .119, .121

Using repetitive testing and the Fisher test statistic, determine the number of outliers present in the data.

From Eq. 2.3,

$$T_F = \frac{.121}{.00289 + \ldots + .121} = 0.367$$

From Table 2.19, the critical value of the Fisher test statistic for $\alpha = 1.5$, $n = 10$, and $k = 1$ is $t_F = .3733$. Since $t_F > T_F$, no outlier is found.

However, if the repetitive method had not been used, and two outliers had been tested for, the following result would have been obtained.

From Eq. 2.1,

$$T_F = \frac{.119 + .121}{.00289 + \ldots + .121} = .728$$

From Table 2.2, the critical value of Fisher's test statistic for $\alpha = 1.5$, $n = 10$, and $k = 2$ is found to be $t_F = .5770$. Since $T_F > t_F,$ the two largest values are labeled as outliers, which is, in fact, the case!

Another pitfall arises in determining the number of outliers to be tested for when a non-discordant value is tested along with a discordant value. This pitfall is the phenomenon known as swamping, and is explained as follows. Suppose that a data set with only one outlier is to be tested for two outliers. And further suppose that, as in the previous case, a test statistic of the form $\frac{D}{S}$ is to be used. In this case, both D and S would be smaller than anticipated, and, depending on the discrepancy between the two tested values and the mean, both values will be labeled as either discordant or non-discordant.

<u>Example 2.11</u>

The following data come from a gamma-distribution with parameter $\alpha = 1.5$.

.00289, .00478, .00487, .00591, .00849, .0167, .0197, .0263, .0954, .121

Using Fisher's test statistic, determine if the two largest values are labeled as outliers at the 95% confidence level.

From Eq, 2.3,

$$T_F = \frac{.0454 + .121}{.2289 + \ldots + .212} = .650$$

From Table 2.2, the critical value of Fisher's test statistic for α =1.5, n = 10, and k = 2 is found to be $t_F = .5770$. Since $T_F > t_F$ the two largest values are labeled as outliers.

However, if repetitive testing had been used, the result would have been as follows.

From Eq. 2.3

$$T_F = \frac{.121}{.00282 + \ldots + .121} = .473$$

From Table 2.1, the critical value of Fisher's test statistic for α = 1.5, n = 10, and k = 1 is found to be $t_F = .3733$. Thus, .121 is found to be discordant. Repeating the test for the next largest value gives

$$T_F = \frac{.0954}{.00282 + \ldots + .0454} = .336$$

From Table 2.1, the critical value of Fisher's test statistic for α = 1.5, n = 9, and k = 1 is found to be $t_F = .4028$. Thus, since $T_F < t_F$, .0454 is not discordant, which is actually the case.

The wrong ways to determine the number of outliers to be tested for have already been shown. The right way is as follows. The data are first tested to locate the largest gap between two consecutive data points, the data being ordered such that $x_1 \leq x_2 \leq \ldots \leq x_n$. If this gap occurs between the two point x_{n-k} and x_{n-k+1} and $x_{n-k+1} > \bar{x}$, the k data points $x_{n-k+1}, x_{n-k+2}, \ldots, x_n$ are considered to be potential, upper outliers. These points are then tested to determine if they are, in fact, outliers or not. If they are not found to be discordant, then there are no outliers, and the testing is complete. If they are found to be discordant then they are removed from the data, and the test is repeated to determine whether or not there is another 'cluster' of outliers present.

Example 2.12:

The following data are known to come from a gamma-distribution with parameter $\alpha = 1.5$.

.00289, .00478, .00487, .00591, .00849, .0167, .0197, .119, .121, .837

Using Fisher's method, determine the number of discordant data points at the 95% confidence level.

The largest gap between consecutive data points occurs between .121 and .837. Thus, .837 is tested for discordancy.

From Eq. 2.3

$$T_F = \frac{.837}{.00289 + \ldots + .834} = .734$$

From Table 2.1, the critical value of Fisher's test statistic for = 1.5, n = 10, and k = 1 is found to be $t_F = .3733$. Thus, .837 is found to be discordant, and is removed from the data.

The largest gap between consecutive data points now occurs between .0197 and .119. Thus, .119 and .121 are tested for discordancy.

From Eq. 2.3,

$$T_F = \frac{.119 + \ldots + .121}{.00289 + \ldots + .121} = .791$$

From Table 2.2, the critical value of Fisher's test statistic for $\alpha = 1.5$, n = 9, and k = 2 is found to be $t_F = .614$. Thus, .119

and .121 are found to be discordant, and are removed from the data.

Now the largest gap between consecutive data points occurs between .00849 and .0167. Thus, .0167 and .0197 are tested for discordancy.

From Eq. 2.3,

$$T_F = \frac{.0167 + .0197}{.00289 + \ldots + .0197} = .575$$

From Table 2.2, the critical value of Fisher's test statistic for $\alpha = 1.5$, $n = 7$, and $k = 2$ is found to be $t_F = .7042$. Thus, .0167 and .0197 are not discordant, and the testing for discordant values is complete.

II.7 Power Curves

The properties and comparisons of the various tests which were described earlier in this chapter are determined by the use of power curves. The use of power curves for examining the properties of tests and making comparisons between tests is standard practice for hypothesis testing. The y-axis of a power curve shows the probability of the null hypotheses, H_o, being rejected and the x-axis is a measure of the variation between the null hypotheses and the alternate hypotheses, H_1. For our purposes, the y-axis of a power curve is the probability that a single outlier is detected, and the x-axis is a measure of the variation between the mean of the parent distribution and the mean of the distribution from which the outlier comes. The value of κ, the x-variable, is

$$\kappa = \frac{\mu_o}{\mu_p} ,$$

where μ_p is the mean of the parent distribution and μ_o is the mean of the outlier-generating distribution. (The variances of the two distributions are set equal to each other.) Note that $\kappa = 1$ means that all data points come from the same distribution. Thus, at $\kappa=1$, the probability of an outlier being detected will be less than or equal to 0.05 for the 95% confidence level.

The technique used to generate the power curves was developed by the author, and is an improvement over the previously used methods. In the past, power curves have been generated by generating sets of data from a parent distribution and generating outliers from a second distribution. The generated outliers were then tested for discordancy against the data sets from the parent distribution to obtain a power curve. Thus, both the data sets and the outliers had to be simulated. In the technique used in this paper, it was only necessary to generate data sets from the parent distribution. An analytical method was then used to derive a power curve (See App. A). Thus, only the data sets needed to be simulated. The remainder of the analysis was theoretical (See Fig. 2.2). Note that the power curves are saturation-type curves, i.e., asymptatically they approach the limiting value.

II.8 Properties and Comparisons of Methods

There are two properties which the tests have in common. First as the value of the gamma parameter α increases, the power of the tests increases i.e., saturation occurs at lower values of κ. And secondly except for the integration method, there is very little variation in the power of the tests as the value of the parameter β changes (See Fig. 2.3).

An interesting property of the normal conversion method is that as

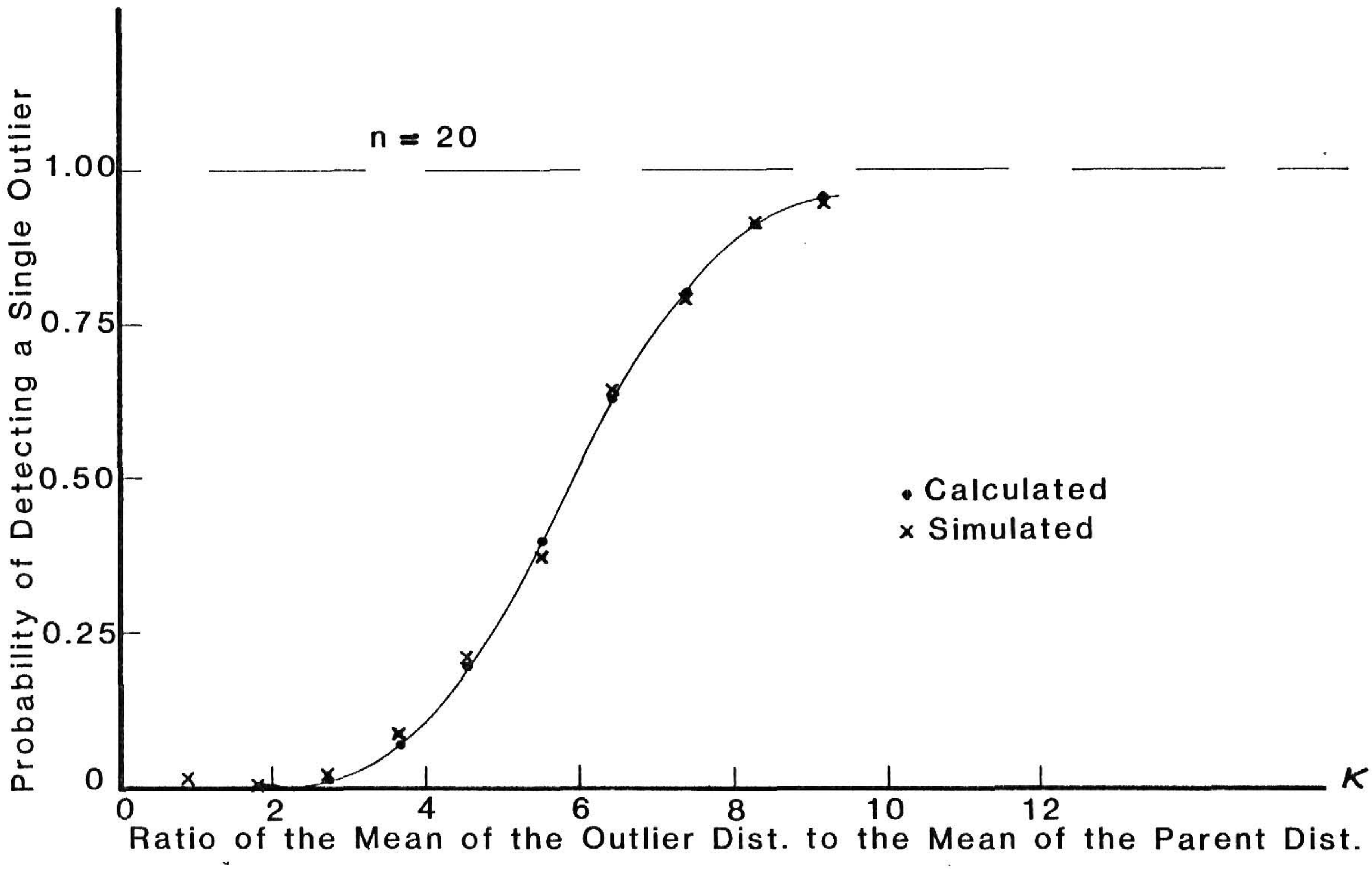


Figure 2.2: Power curves generated for Fisher's method with gamma parameters alpha = 1.25 and beta = 1.25×10^6 by the method of App. A and by simulation.

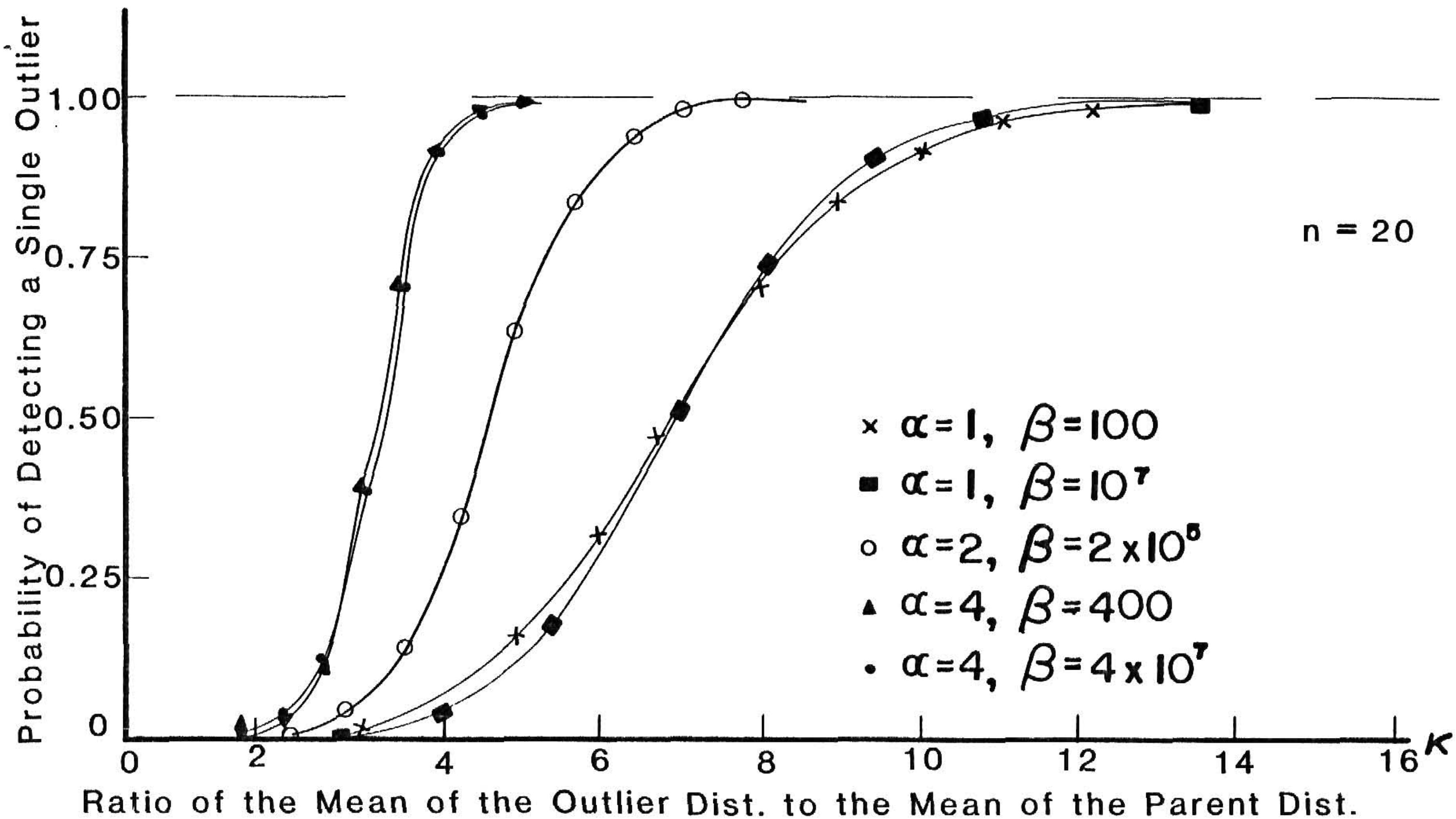


Figure 2.3: Variation in the power of Fisher's method of detecting a single outlier as the gamma parameters alpha and beta change.

the value of the parameter α increases, the power of the normal conversion method seems to approach the power of Fisher's method. This type of behavior is expected, since the conversion of data, which are distributed according to a gamma distribution to normally distributed data, gives a better approximation of normally distributed data as α increases (Kimber, 1978) (See Fig. 2.4).

The power of Fisher's method where the value of the parameter α is unknown is always less than the power of Fisher's method where α is known. Thus, when the value of α is unknown, Fisher's method is conservative in that suspicious data points are less likely to be labeled as outliers (See Fig. 2.5).

The integration method of detecting discordant data is not very powerful in relation to the other methods (See Fig. 2.5). However, its power increases as the parameter α increases and the parameter β decreases (See Fig. 2.6). (The integration method will eventually saturate at power equal to 1.0.)

The second most powerful method for all values of the gamma parameters α and β which were tested is Fisher's method (See Fig. 2.5). However, it is not always possible to apply Fisher's method while staying within its theoretical limitations, because the value of the parameter α is not always known.

The most powerful of all methods for all values of the gamma parameters considered is the normal conversion method (See Fig. 2.5). Not only is the normal conversion method the most powerful outlier detection method, it can always be applied, because it is not necessary to know the values of any parameters. Also, it is easily applied with the use of a hand calculator. Thus, the normal conversion method is the

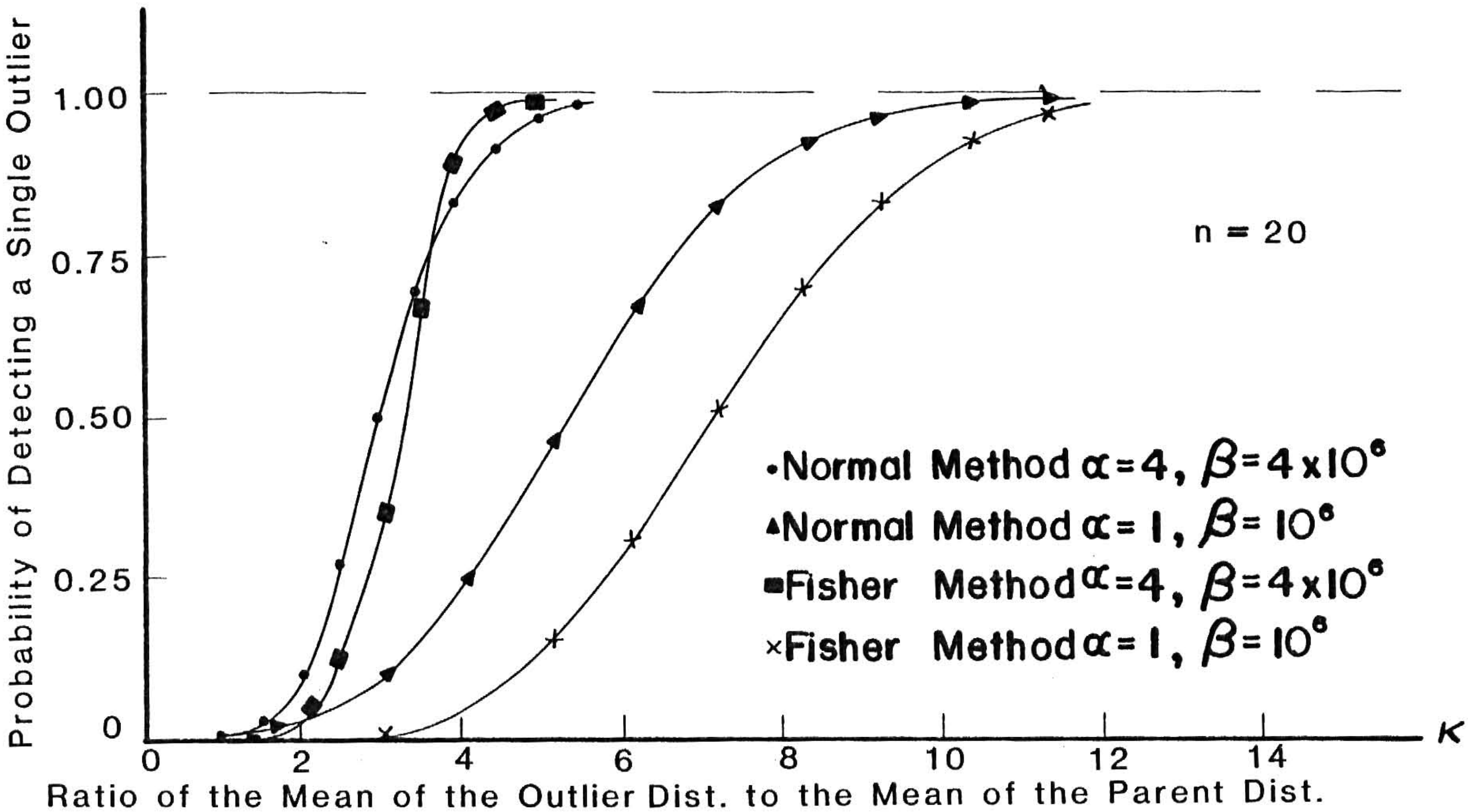


Figure 2.4: Relation between the powers of the normal conversion method and Fisher's method as the parameter alpha changes.

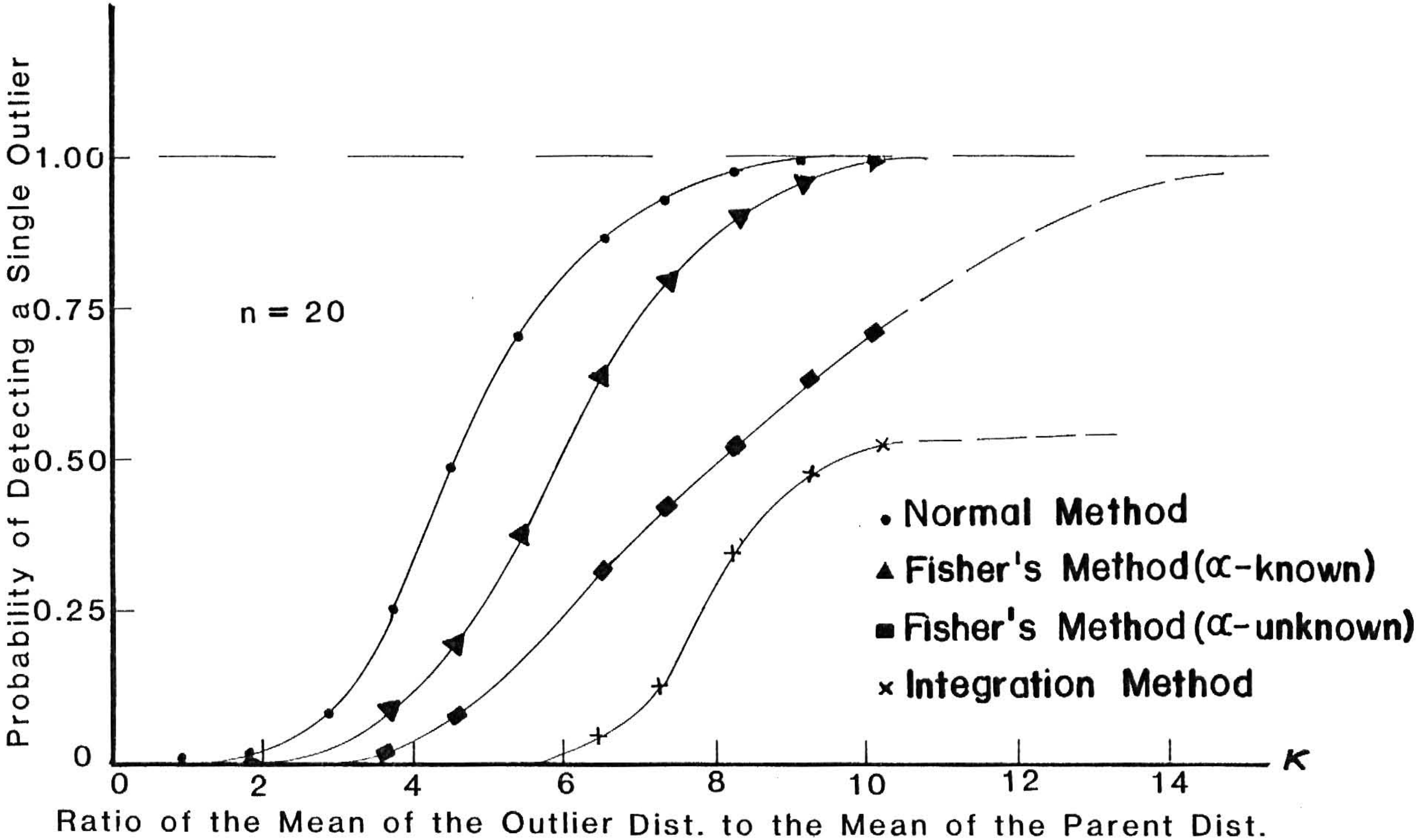


Figure 2.5: Relation between the powers of the normal conversion method, Fisher's method (alpha known), Fisher's method (alpha unknown), and the integration method for alpha = 1.25 and beta = 1.25×10^6.

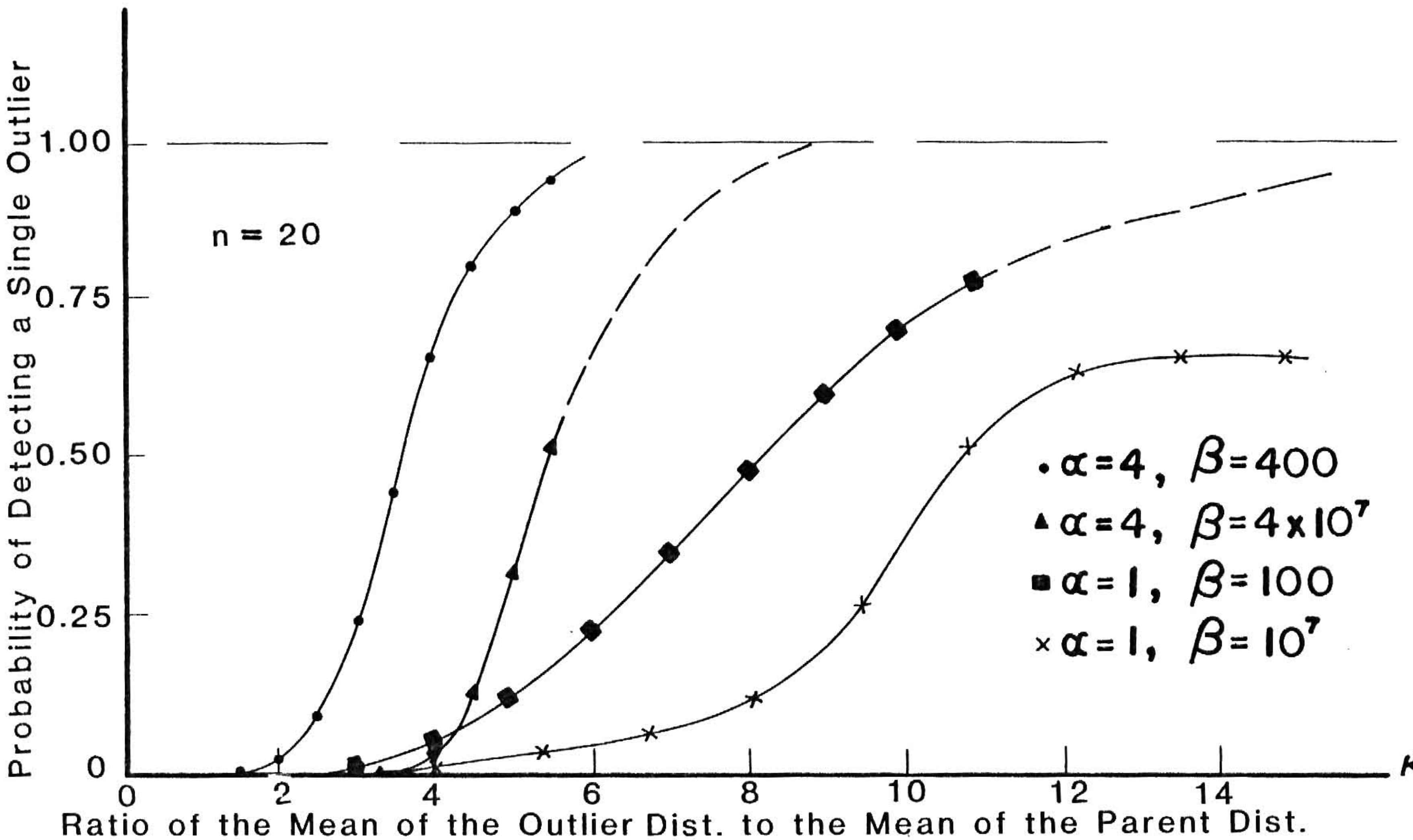


Figure 2.6: Variation in the power of the integration method as the gamma parameters alpha and beta change.

"best" method to detect a single outlier in a gamma distribution which has parameters $1 \leq \alpha \leq 4$ and $100 \leq \beta \leq 10^7$.

It should also be noted that although all of the power curves were generated at the 95% confidence level, the probability of outlier detection at $\kappa = 1$ is in every instance well below 0.01 (See Fig.'s 2.3 - 2.6). Thus, although the theoretical confidence level is 95%, type II errors, i.e., acceptance of the null hypothesis, H_o, when it should be rejected, cause the actual confidence level to be well above 99%. Also, since the actual confidence levels* of the various tests are not equal, it may not be completely "fair" to compare their powers.

*The probability of accepting H_o (outlier and data are from the same distribution) is less than the critical level of significance chosen a priori, e.g., a critical level of significance chosen a priori was 0.05 (5% chance of incorrectly rejecting H_o), while there was actually a much smaller chance than 5% of incorrectly rejecting H_o.

III. Outliers in (F_i, T_i) Data

III.1 Introduction

In the last chapter, outliers in a gamma distribution were discussed. This discussion was in connection with failure rate data for which the failure rates themselves, i.e., λ's, were known. Unfortunately, this is rarely the case. Normally, the failure rate data will be of the form (F_i, T_i), where F_i and T_i are the number of failures and testing time of component i, respectively. Thus, it is very important to be able to detect outliers in data of this form.

In this chapter, several methods will be presented which can be used to detect discordant values in (F_i, T_i) data. First, due to the complexity of the analysis which must be performed to detect outliers in this case, the theory behind the model used to describe (F_i, T_i) data will be presented.

III.2 Models for (F_i, T_i) Data

III.2.1 The Homogeneous Model: In the classical description of a component, the failure rate of a component, λ, is regarded as an unknown constant which does not change even if the component fails and is subsequently repaired. Thus, the number of failures F in time T is modeled by the Poisson distribution (sometimes called the likelihood distribution),

$$F(F|\lambda, T) = \frac{(\lambda T)^F}{\Gamma(F+1)} e^{-\lambda T}, \; F = 0,1,2,\ldots) \qquad (3.1)$$

where $\Gamma(\ldots)$ is the gamma function. A further assumption which is made is that similar components have the same failure rate, λ. Thus, λ can be estimated for groups of similar components rather than individual components.

For the model described above, the maximum likelihood estimator (MLE) of λ, $\hat{\lambda}$, is given by

$$\hat{\lambda} = \sum_{i=1}^{n} F_i / \sum_{i=1}^{n} T_i, \tag{3.2}$$

where n is the number of components in the group under consideration. And if the assumptions of the model hold, $\hat{\lambda}$ is the uniformly minimum variance unbiased estimator (UMVUE) of λ, i.e., the "best" unbiased estimator of λ.

Example 3.1:

Determine the average estimated failure rate for the homogeneous model for the following (F_i, T_i) data.

(0, 10,000), (0, 10,000), (0, 10,000), (0, 10,000), (0, 10,000),
(0, 10,000), (1, 10,000), (1, 10,000), (1, 10,000), (12, 10,000)

From Eq. 3.2,

$$\hat{\lambda} = \sum_{i=1}^{10} F_i \Big/ \sum_{i=1}^{10} T_i = \frac{0+0+\ ...\ +12}{10,000 +\ ...\ +10,000} = 1.5\text{x}10^{-4}.$$

There are two major weaknesses in this model. First, the assumption that similar components have the same failure rate is not necessarily valid. And second, if a group of components has an extremely small failure rate, as is the case for nuclear power plant components, and the test times, T_i, are not very long, it is possible that no failures would be observed. This would give $\hat{\lambda} = 0$, which is an unacceptable value.

III.2.2 The Compound Model: The compound model for (F_i, T_i) data is based upon the homogeneous model, but is more sophisticated. This sophistication comes about by the removal of one of the assumptions made in the homogeneous model. Specifically, the failure rate, λ, of a group of similar components is not assumed to be the same for all components, but rather is assumed to be distributed according to some distribution $g(\lambda)$. The distribution $g(\lambda)$ is called the "prior" distribution.

The equation which is the basis for this model is

$$h(F|T,g(\lambda)) = \int_0^\infty f(F|T,\lambda)\ g(\lambda)d\lambda, \qquad F = 0,1,2,\ldots \ . \qquad (3.3)$$

This discrete distribution is called the "marginal" distribution. Note that if similar components have the same failure rate, λ_o, then $g(\lambda)$ becomes a delta function, i.e., $g(\lambda) = \delta(\lambda-\lambda_o)$, and the compound model of Eq. 3.3 becomes the homogeneous model of Eq. 3.1.

As explained in Chapter 2, a good choice for the prior distribution, $g(\lambda)$, is the gamma distribution,

$$g(\lambda|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \ e^{-\beta\lambda}. \qquad (3.4)$$

Thus, substituting Eq.'s 3.1 and 3.4 into Eq. 3.3, the compound model becomes

$$h(F|T,\alpha,\beta) = \int_0^\infty \frac{(\lambda T)^F}{\Gamma(F+1)} e^{-\lambda T} \frac{\beta\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \ e^{-\lambda} \ d\lambda$$

$$= \frac{\Gamma(F+\alpha)}{\Gamma(\alpha)\Gamma(F+1)} \ \frac{T^F \beta^\alpha}{(T+\beta)^{F+\alpha}} \qquad (3.5)$$

This discrete distribution gives the probability of obtaining F Failures in test time T for a component randomly selected from a population whose distribution of failure rates is described by the gamma distribution, and whose conditional or likelihood distribution is the Poisson distribution.

For the compound model, the simplest, and one of the better methods for estimating the values for the parameters α and β in the gamma distibution is the prior matching moments method (PMMM). In the PMMM, estimates of the failure rate for each component in the class are assumed to be given by

$$\hat{\lambda}_i = F_i/T_i. \tag{3.6}$$

The mean and variance of these failure rate estimates are then equated to the mean and variance of the gamma prior distribution, as was done in Section II.5,

$$\bar{\hat{\lambda}} = \frac{1}{n} \sum_{i=1}^{n} \hat{\lambda}_i, \tag{3.7}$$

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\lambda_i - \bar{\hat{\lambda}})^2, \tag{3.8}$$

$$\hat{\alpha} = \bar{\hat{\lambda}}^2/S^2, \tag{3.9}$$

and

$$\hat{\beta} = \bar{\hat{\lambda}}/S^2. \tag{3.10}$$

Note that if at least one F_i is non-zero, then the PMMM estimator is non-zero, and thus gives mathematically acceptable values for α and β.

<u>Example 3.2</u>:

Determine the values of $\hat{\alpha}$ and $\hat{\beta}$ for the compound model by the PMMM from the data of Ex. 3.1

From Eq.'s 3.6 and 3.7,

$$\bar{\hat{\lambda}} = \frac{1}{10}\left(\frac{0}{10,000} + \ldots + \frac{12}{10,000}\right) = 1.5\text{x}10^{-4}.$$

From Eq. 3.8,

$$S^2 = \frac{1}{9}\left[(0-1.5\text{x}10^{-4})^2 + \ldots + (1.2\text{x}10^{-3} - 1.5\text{x}10^{-4})^2\right] =$$

$$1.38\text{x}10^{-7}.$$

From Eq. 3.9,

$$\hat{\alpha} = (1.5\text{x}10^{-4})^2/1.38\text{x}10^{-7} = 0.613.$$

From Eq. 3.10,

$$\hat{\beta} = 1.5\text{x}10^{-4}/1.38\text{x}10^{-7} = 1080.$$

Another method for estimating the values of the gamma parameters α and β for the compound model is the marginal matching moments method (MMMM). In this method, the moments of the data are paired with the moments of the marginal distribution. The values of $\hat{\alpha}$ and $\hat{\beta}$ obtained via the MMMM are (Shultis, et al, 1980)

$$\hat{\alpha} = \bar{\hat{\lambda}}^2/(S^2 - \bar{\hat{\lambda}}\,\frac{1}{n}\sum_{i=1}^{n} T_i^{-1}), \qquad (3.11)$$

$$\hat{\beta} = \bar{\hat{\lambda}}/(S^2 - \bar{\hat{\lambda}}\,\frac{1}{n}\sum_{i=1}^{n} T_i^{-1}), \qquad (3.12)$$

where $\bar{\hat{\lambda}}$ and S^2 are given by Eq.'s 3.7 and 3.8, respectively. Note that the values of $\hat{\alpha}$ and $\hat{\beta}$ can be negative (if S^2 is less than the other term of the denominator which is outside of the parameter space of α and β, and thus the MMMM can fail to give realistic values for α and β.

Example 3.3

Determine the values of α and β for the compound model by the MMMM from the data of Ex. 3.1.

From Ex. 3.2, $\bar{\hat{\lambda}} = 1.5\text{x}10^{-4}$ and $S^2 = 1.38 \text{ x } 10^{-7}$.

$$\frac{1}{n}\sum_{i=1}^{n} T_i^{-1} = \frac{1}{10}\left(\frac{1}{10,000} + \ldots + \frac{1}{10,000}\right) = 10^{-4}$$

From Eq. 3.11,

$$\hat{\alpha} = (1.5\text{x}10^{-4})^2 \;/\; (1.38\text{x}10^{-7} - (1.5\text{x}10^{-4})(10^{-4})) = 0.183$$

From Eq. 3.12,

$$\hat{\beta} = (1.5\text{x}10^{-4})/(1.38\text{x}10^{-7} - (1.5\text{x}10^{-4})(10^{-4})) = 1216$$

III.3 Outlier Detection in (F_i, T_i) Data

III.3.1 The Cumulative Marginal Method: The cumulative marginal method for detecting outliers in (F_i, T_i) data was developed by Shultis, et al, 1980, as follows. One may reasonably decide that (F^*, T^*) is an outlier in a sample of size n if the probability of observing F* failures in test time T* for at least one component is less than some preselected value λ, i.e.,

$$P\left(\bigvee_{i=1}^{n} F_i/T_i \geq F^*/T^*\right) \leq \gamma. \tag{3.13}$$

The value of γ should be selected as either 0.01 or 0.05 for the 99% or 95% confidence level, respectively.

If the data are independent, Eq. 3.13 can be rewritten

$$1-P\left(\bigwedge_{i=1}^{n} F_i/T_i < F^*/T^*\right) \leq \gamma, \tag{3.14}$$

$$\prod_{i=1}^{n} P(F_i/T_i < F^*/T^*) \geq 1 - \gamma \tag{3.15}$$

$$\prod_{i=1}^{n} P(F_i < F^*T_i/T^*) \geq 1 - \gamma. \tag{3.16}$$

In terms of the marginal distribution, Eq. 3.5, Eq. 3.16 becomes

$$\prod_{i=1}^{n} \left[\sum_{F=0}^{[F^*T_i/T^*]'} h(F|T_i,\alpha,\beta)\right] \geq 1 - \gamma, \tag{3.17}$$

where $[F^*T_i/T^*]'$ is the largest integer strictly less than F^*T_i/T^*. That is, for example, $[4.1]' = 4$, but $[4]' = 3$. In terms of the cumulative marginal distribution, Eq. 3.17 becomes

$$\prod_{i=1}^{n} H([F^*T_i/T^*]' \,|T_i,\alpha,\beta) \geq 1 - \gamma, \tag{3.18}$$

where the cumulative marginal distribution is

$$H(F|T,\alpha,\beta) = \sum_{i=0}^{F} h(i|T,\alpha,\beta). \tag{3.19}$$

From Ex. 3.2, $\alpha = 0.163$ and $\beta = 1080$. Thus, Eq. 3.18 becomes

$$\prod_{i=1}^{10} H([(12)(10^4)/(10^4)]^{\prime} | \, 10^4,.163,1080) \geq 1 - .05,$$

$$\prod_{i=1}^{10} H(11 | 10^4,.163,1080) \geq .95,$$

$$H(11 | 10^4,.163,1080)^{10} \geq .95.$$

From Eq. 3.5,

$$h(0 | 10^4,.163,1080) = \frac{\Gamma(.163)}{\Gamma(.163)\Gamma(1)} \frac{1080^{.163}}{(11,080)^{.163}} = .684,$$

$$h(11 | 10^4,.163,1080) = .101,$$

$$\vdots$$

$$h(11 | 10^4,.163,1080) = .00519.$$

From Eq. 3.19

$$H(11 | 10^4,.163.1080) = (.684 + .101 + \ldots + .00519) = .969.$$

$$H(11 | 10^4,.163,1980)^{10} = .969^{10} = 0.729$$

Thus, since $H(11)^{10} < 0.95$, the data point (12, 10,000) is <u>not</u> found to be discordant.

Example 3.5

Determine if the data point (12, 10,000) in the data of Ex. 3.1 is an outlier at the 95% confidence level according to the cumulative marginal method with the gamma parameters α and β estimated by the MMMM.

From Ex. 3.3, $\hat{\alpha} = 0.183$ and $\hat{\beta} = 1216$.

This problem is worked in exactly the same manner as Ex. 3.4. The result is $H(11)^{10} = 0.745$. Thus no outlier is found in the data.

A variation in the above methods can be made by changing the way in which the parameters α and β are estimated. They can be calculated without including the data points which are suspected of being outliers. However, by making this change, the number of type I errors will increase, and the number of type II errors will decrease, i.e., H_o will be rejected more often when it should be accepted, and H_o will be accepted less often when it should be rejected.

Example 3.6

Determine if the data point (12, 10,000) in the data of Ex. 3.1 is an outlier at the 95% confidence level according to the cumulative marginal method with the parameters α and β estimated by the PMMM with the suspected outlier omitted.

From Eq. 3.7,

$$\bar{\hat{\lambda}} = \frac{1}{9}\left(\frac{0}{10,000} + \ldots + \frac{1}{10,000}\right) = 3.33\text{x}10^{-5}.$$

From Eq. 3.8,

$$s^2 = \frac{1}{8}[(0-3.33\text{x}10^{-5})^2 + \ldots + (10^{-4}-3.33\text{x}10^{-5})^2] = 2.50\text{x}10^{-9}.$$

From Eq. 3.9,

$$\hat{\alpha} = (3.33\text{x}10^{-5})^2/2.5\text{x}10^{-9} = 0.444.$$

From Eq. 3.10,

$$\hat{\beta} = 3.33\text{x}10^{-5}/2.5\text{x}10^{-9} = 13,300.$$

From these values for α and β, and following the procedure of Ex. 3.4, the result is $H(11)^{10} = 0.9999$. Thus, (12, 10,000) is found to be an outlier at the 95% and 99% confidence levels.

Example 3.7

Determine if the data point (12, 10,000) in the data of Ex. 3.1 is an oulier at the 95% confidence level according to the cumulative marginal method with the parameters α and β estimated by the MMMM with the suspected outlier omitted.

From Ex. 3.6,

$$\overline{\hat{\lambda}} = 3.33\text{x}10^{-5} \text{ and } S^2 = 2.50\text{x}10^{-9}.$$

$$\frac{1}{n}\sum_{i=1}^{n} T_i^{-1} = \frac{1}{9}\left(\frac{1}{10,000}\right) + \ldots + \frac{1}{10,000}) = 10^{-4}$$

From Eq. 3.11

$$\hat{\alpha} = (3.33\text{x}10^{-5})^2 / (2.5\text{x}10^{-9} - (3.33\text{x}10^{-5})(10^{-4}) = -1.33$$

Since $\hat{\alpha} < 0$, it is out of the parameter range of α, and so this method cannot be used to determine if an outlier is present.

III.3.2 Binomial Method: Another method for detecting upper outliers in (F_i,T) data is the binomial method. Note that the binomial method only works if all T_i's are equal to one another. The theory behind the binimial method is as follows.

Assume that there are k suspected outliers in the data $F_1 \leq F_2 \leq \dots \leq F_n$ The probability of having F_{n-k} or fewer failures is given by

$$p = \sum_{i=0}^{Fn-k} h(i|T,\alpha,\beta) = H(F_{n-k}|T,\alpha,\beta), \qquad (3.20)$$

where h(...) is the marginal distribution.

The probability of having n-k or more data points with F_{n-k} or fewer failures is given by

$$P_{n-k} = \sum_{i=n-k}^{n} \binom{n}{i} p^i (1-p)^{n-i}. \qquad (3.21)$$

Thus, if $P_{n-k} \leq 0.05$ or 0.01 for the 95% or 99% confidence level, respectively, the k upper values are labeled as outliers.

Example 3.8

Determine if the data point (12, 10,000) in the data of Ex. 3.1 is an outlier at the 95% confidence level according to the binomial method with the parameters α and β estimated by the MMMM.

From Ex. 3.2, $\hat{\alpha} = 0.183$ and $\hat{\beta} = 1216$. Also, n-k = 9 and $F_{n-k} = 1$. From Eq. 3.20,

$$p = h(0|10^4,.183,1216) + h(1|10^4,.182,1216)$$

From Eq. 3.21,

$$P_{n-k} = 10(.775)^9(1-.775) + (.775)^{10} = .304.$$

Thus, since $P_{n-k} > .05$, the upper data point is not labeled an outlier.

The binomial method can also be used in conjunction with the homogeneous model. In this case, the probability of having F_{n-k} or fewer failures, where the data are $F_1 \leq F_2 \leq \ldots \leq F_n$, is given by

$$p = \sum_{i=0}^{F_{n-k}} f(i|\lambda,T), \tag{3.22}$$

where f(...) is the Poisson distribution (Eq. 3.1). The value obtained from Eq. 3.22 for p is then substituted into Eq. 3.21 to determine whether or not discordant data are present.

Example 3.9:

Determine if the data point (12, 10,000) in the data of Ex. 3.1 is an outlier at the 95% confidence level according to the binomial method and using the homogeneous model.

From Ex. 3.1, $\hat{\lambda} = 1.5 \times 10^{-4}$.

From Eq. 3.22,

$$p = f(0|1.5x10^{-4},10^4) + f(1|1.5 \times 10^{-4},10^4)$$

$$= 0.223 + 0.335 = 0.558$$

From Eq. 3.21,

$$P_{n-k} = 10(.558)^9(1-.558) + (.558)^{10} = 0.026$$

Thus, since $P_{n-k} < 0.05$, the upper data point is labeled an outlier at the 95% confidence level.

III.4 Properties and Comparisons of Tests

As in the previous chapter, the properties and comparisons of the various tests are determined by the use of power curves. However, due to the problems brought about by (F_i, T_i) data, the power curves were generated by simulation, and as a consequence the uncertainty in the power curves is greater than in Chapter 2.

The one property which all of the power curves have in common is that as the value of the gamma parameter β in the prior distribution increases, the power of the gests decreases, i.e., the power curve saturates at larger values of κ (See Fig. 3.1). The reason for this is that for $n = 20$, $T = 10{,}000$ hr, and an average failure rate significantly less that 10^{-4} hr^{-1}, data sets in which no failures are present are not uncommon, (See Table 3.1) and these data sets cannot be analyzed.

Neither the cumulative marginal method nor the binomial method using the compound model with the outlier included in the gamma parameter calculations appear in any of the power curves, because in no case did either of these methods detect an outlier. The reason for this is that in data sets of size 20, an outlier will distort the estimated values of the parameters of the prior distribution to such an extent that it is not possible to detect the outlier.

The best method for detecting an outlier for $1 \leq \alpha \leq 4$ and $10^4 \leq \beta \leq 4 \times 10^6$ (values for α and β outside these ranges were not studied in this work is the cumulative marginal method were the values of α and β are known (See Fig.'s 3.2 to 3.4). However, in general, the values of α and β are not known and thus, this method is not practical.

The cumulative marginal method with the gamma parameters calculated by the marginal matching moments method (MMMM) is a very poor method, and be used with caution. The reason for this is twofold. First, for small

DATA SET	DATA																	
10	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.
20	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
30	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
40	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
50	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.
60	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
70	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
80	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
90	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
100	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
110	0.	1.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
120	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
130	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
140	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
150	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
160	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.	0.	0.
170	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
180	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
190	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
200	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
210	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
220	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
230	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
240	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
250	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
260	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

Table 3.1: Number of failures, F_i, in 10,000 hrs. where the average failure rate is 10^{-6} hr^{-1}.

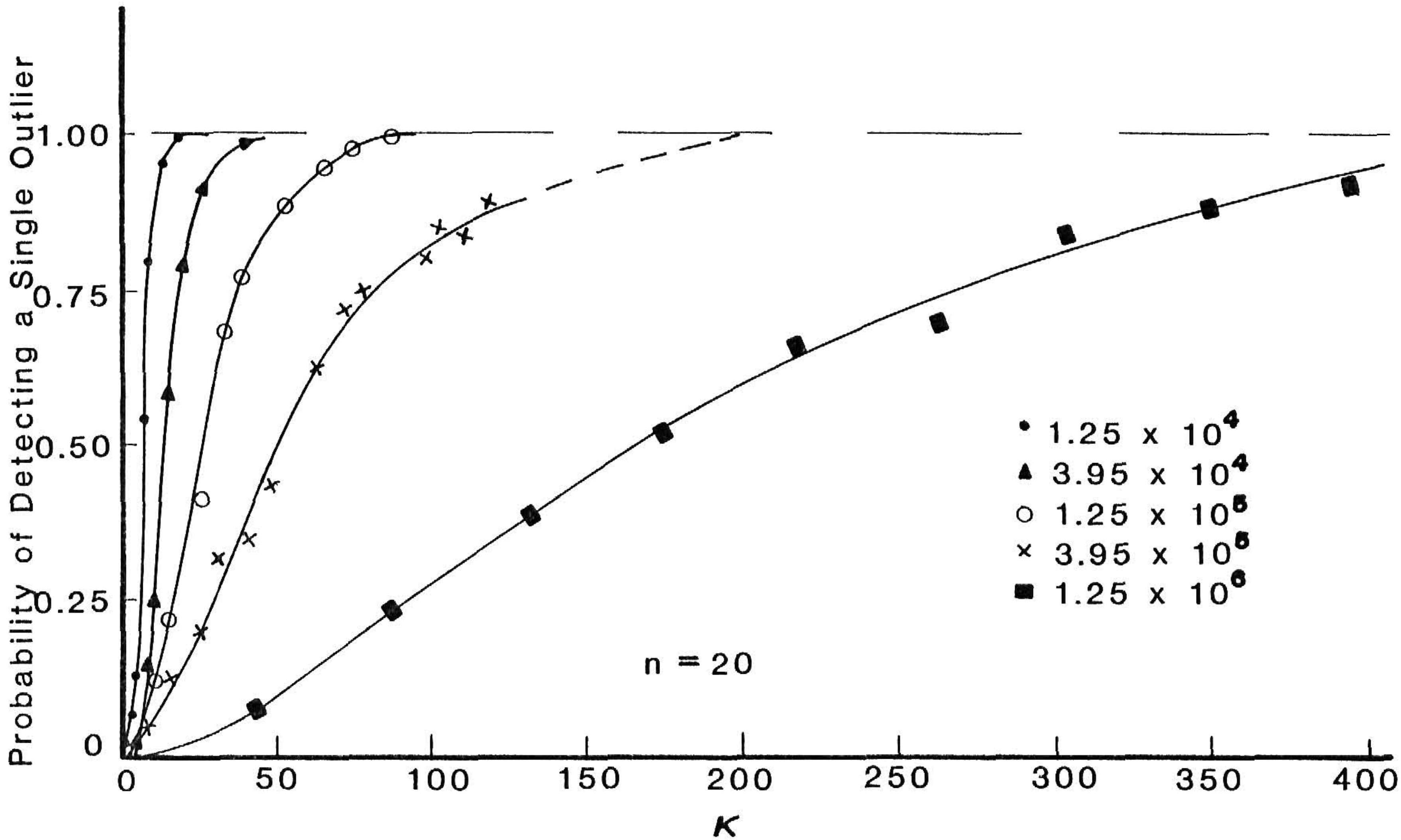


Figure 3.1: Varaition in the power of the cumulative marginal method for single outlier detection as the gamma parameter beta changes with alpha = 1.25.

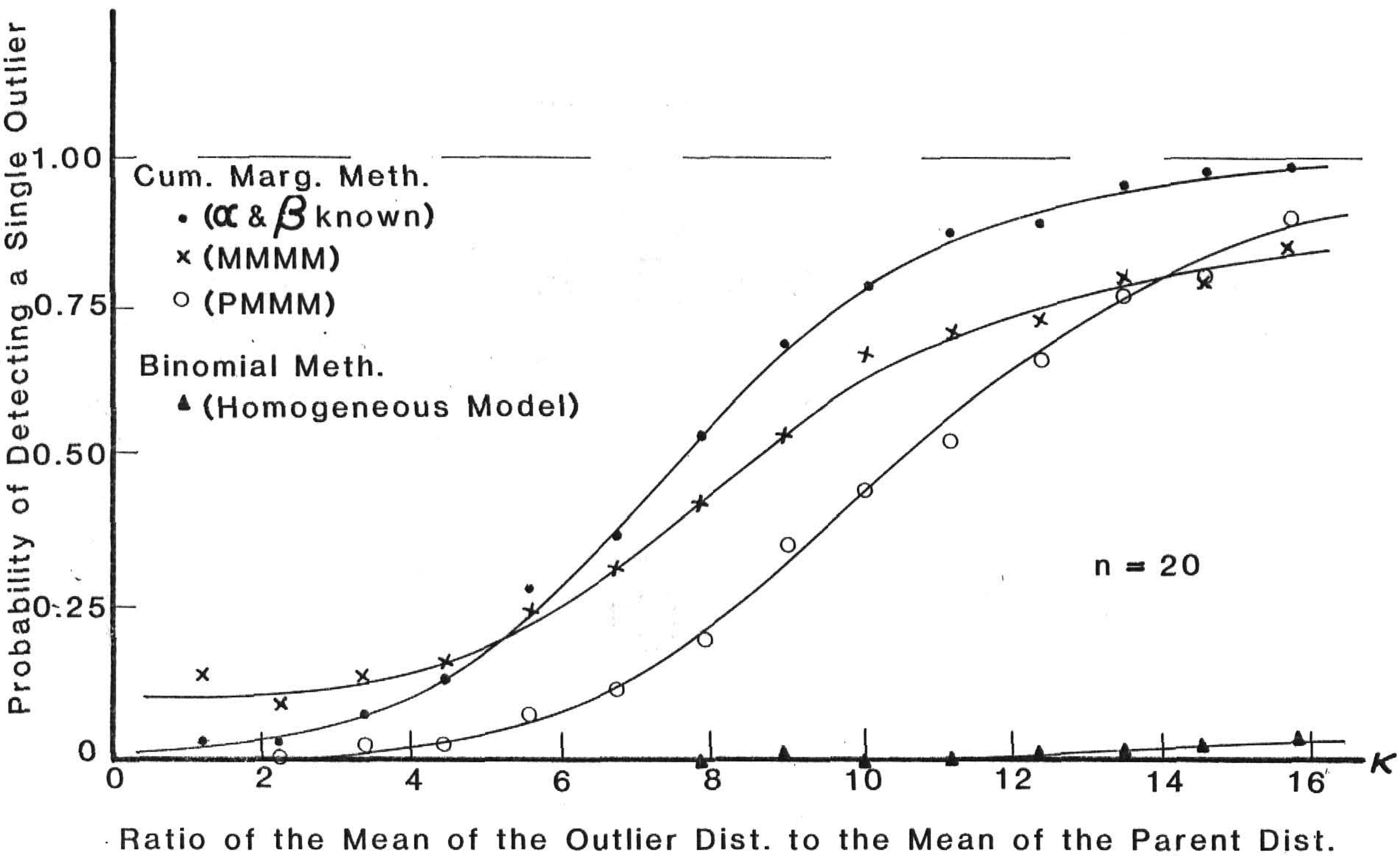
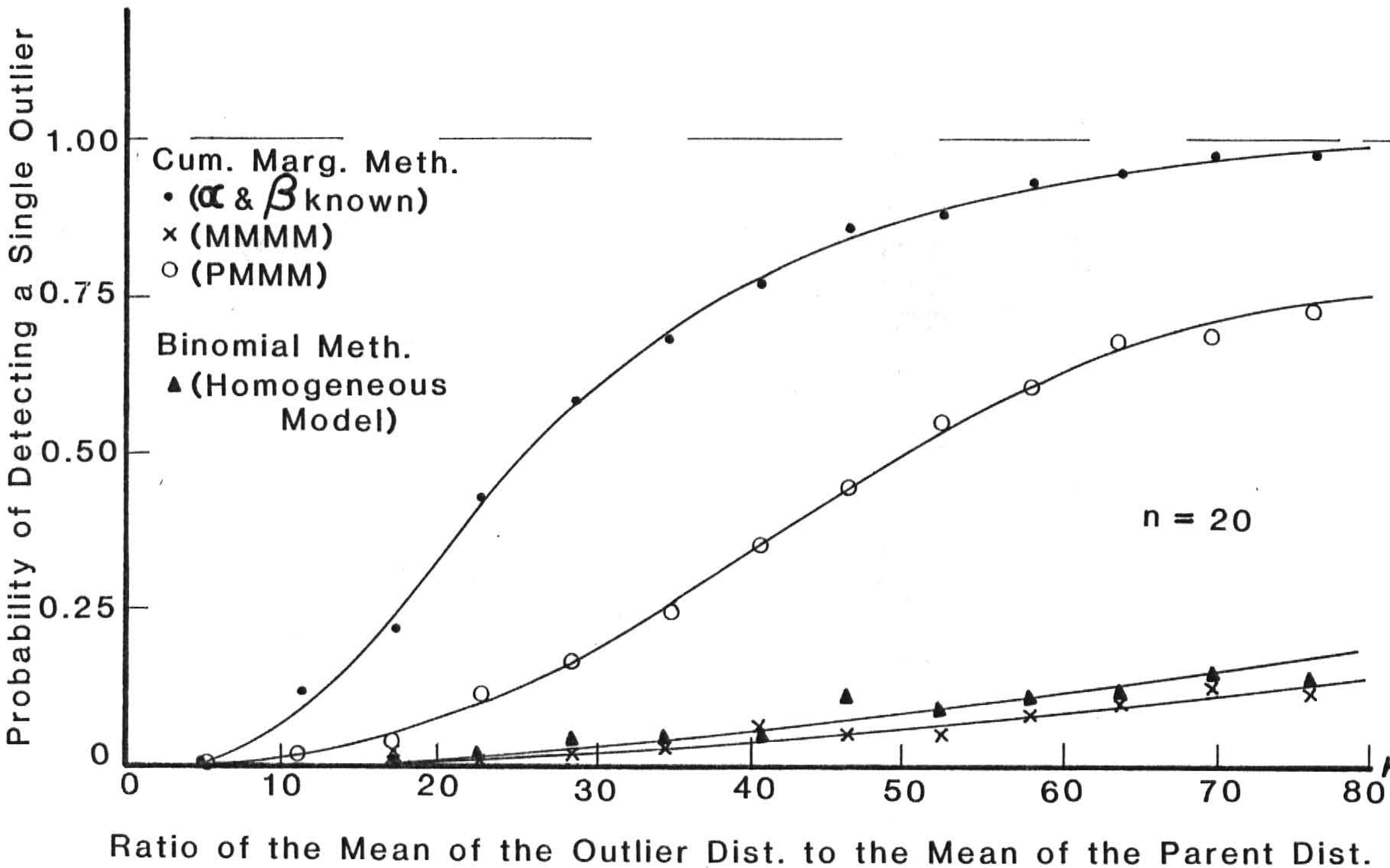


Figure 3.2: Relation between the powers of the cumulative marginal method with alpha and beta known, calculated by the MMMM, calculated by the PMMM, and the binomial method used in conjunction with the homogeneous model for alpha = 1.25 and beta = 1.25×10^4.



<u>Figure 3.3</u>: Relation between the powers of the cumulative marginal method with alpha and beta known, calculated by the MMMM, calculated by the PMMM, and the binomial method used in conjunction with the homogeneous model for alpha = 1.25 and beta = 1.25×10^5.

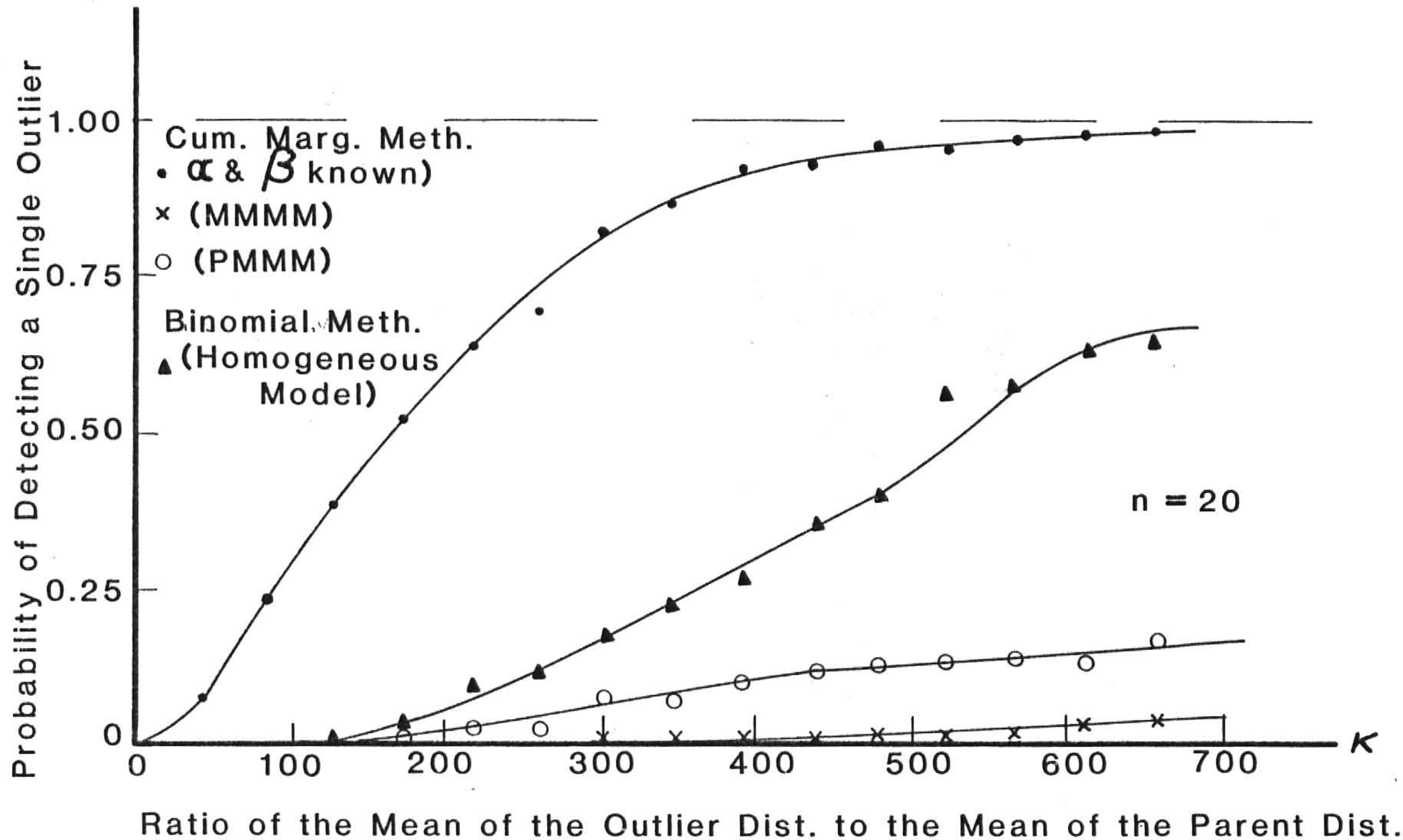


Figure 3.4: Relation between the powers of the cumulative marginal method with alpha and beta known, calculated by the MMMM, calculated by the PMMM, and the binomial method used in conjunction with the homogeneous model for alpha = 1.25 and beta = 1.25×10^6.

values of the gamma parameter β, the actual confidence level is less than the theoretical condidence level, e.g., at $\kappa = 1$ a good data point is labeled an outlier more often than it should be (See Fig. 3.2). And secondly, for larger values of β, the power of this method is lower than even the simple homogeneous model Fig.'s 3.3 and 3.4).

For smaller failure rates, i.e., $\hat{\lambda} < 5 \times 10^{-5} hr^{-1}$, and all T_i equal, the most powerful practical method of outlier detection is the binomial method used in conjunction with the homogeneous model (See Fig. 3.4). If all T_i are not equal, the most powerful method is the cumulative marginal method with the gamma parameters estimated using the prior matching moments method (PMMM) without using the potential outlier.

For larger failure rates, i.e., $\bar{\lambda} > 5 \times 10^{-5} hr^{-1}$, the most powerful practical method with the gamma parameters calculated using the PMMM, and without using the potential outlier (See Fig.'s 3.2 and 3.3).

Although the theoretical confidence level used in all calculations was 95%, except for the cumulative marginal method with $\hat{\alpha}$ and $\hat{\beta}$ calculated via the MMMM without using the potential outlier, the actual confidence level was always greater than 95%. This can be seen by noting that $\kappa = 1$, the powers of the various methods is below 0.05 (See Fig.'s 3.1 to 3.4). Also, since the actual confidence levels* of the various tests are not equal, it may not be completely fair to compare their powers.

---

*The probability of accepting H_o (outlier and data are from the same distribution) is less than the critical level of significance chosen a priori, e.g., a critical level of significance chosen, a priori, was 0.05 (5% chance of incorrectly rejecting H_o), while there was actually a much smaller chance than 5% of incorrectly rejecting H_o.

IV. Outliers in an Exponential Distribution

IV.1 Introduction

The exponential distribution

$$E(\lambda) = \lambda e^{-\lambda t}$$

is a distribution which is very important in the analysis of failure rate data for nuclear power plant components. The reason for this is that much of the data which are available in the literature are in the form of time-to-failure data. And since one of the basic premises of failure rate work is that failures occur according to an exponential distribution, time-to-failure data will be distributed according to an exponential distribution (Mann, etal, 1974). Since an exponential distribution is a special case of the gamma-distribution, the methods of detection of outliers in an exponential distribution are similar to those used for the detection of outliers in a gamma-distribution. In fact, Fisher's method of outlier detection which was discussed in Chapter 2 will be discussed further in this chapter.

As far as outlier detection is concerned, there is one major difference between time-to-failure data and failure rate data which are distributed according to a gamma-distribution. Failure rate data for nuclear power plant components, which are distributed according to a gamma-distribution, are assumed to contain no lower outliers, because the failure rates are so low. That is, data in the literature are of the form F Failures in time T, where F is normally less than 4 and often 0. Thus, it is impossible to detece lower outliers which might be present in these data, and so the assumption is made that no lower outliers are present. Time-to failure data, on the other hand, may contain data in which a component failed either abnormally early or abnormally late, and thus upper and lower outliers are to be expected.

This chapter is divided into four main sections. The first section deals with Fisher's method applied to the exponential case. Dixon-type statistics and their use in outlier detection are covered in the second section. In the third section the determination of the number of outliers to be tested for is discussed. And in the fourth section the properties of the two types of tests are covered.

IV.2 Fisher's Method of Outlier Detection

The foundation of Fisher's method for detecting outliers was discussed in Chapter 2. However, it was applied to upper outliers only. The application of Fisher's method to lower outliers is very similar to its application to upper outliers.

For the case of lower outliers, the Fisher test statistic is

$$T_F = \frac{x_1+x_2+\ldots+x_k}{\sum_{i=1}^{n} x_i}, \tag{4.1}$$

where $x_1, \leq x_2 \leq \ldots \leq x_n$ are the data, and k is the number of outliers being tested for.

The critical values of the test statistic are found from the following equation:

$$P(T_F \leq t_F) < \binom{n}{k} P[F_{2k\alpha,2(n-k)\alpha} < \frac{(n-k)t_F}{k(1-t_F)}], \tag{4.2}$$

where n is the number of data points, k is the number of outliers, and $F_{2k\alpha,2(n-k)\alpha}$ is the F distribution with $2k\alpha$ and $2(n-k)\alpha$ degrees of freedom. As before, $P(T_F \leq t_F)$ is given either the value 0.01 or 0.05 in order to obtain either the 99% or 95% confidence level, respectively.

Since tables of the critical values of the Fisher test statistic for lower outliers were not found to exist, Table 4.1 was generated with a computer program similar to the one used to generate the critical values of the test statistic for upper outliers (see Appendix B). Since the two programs are very similar, it can be assumed that the critical values which were generated for lower outliers are also quite accurate.

Example 4.1:

The following data come from an exponential distribution.

0.0549, 2.22, 17.4, 27.8, 39.7, 44.5, 63.9, 119, 127, 290.

Determine if the smallest value is labeled as an outlier at the 95% confidence level.

From Eq. 4.1,

$$T_F = \frac{0.0549}{0.0549+...+127+290} = 7.50\text{x}10^{-5}.$$

From Table 4.1, the critical value of Fisher's test statistic for n=10 and k=1 is found to be 5.568×10^{-4}. Thus, since $t_F > T_F$, i.e., $5.568 \times 10^{-4} > 7.50\text{x}10^{-5}$, the data point 0.0549 is labeled an outlier at the 95% confidence level.

Example 4.2:

The following data come from an exponential distribution.

9.84, 15.7, 1300, 2260, 2690, 3010, 5190, 5880, 8470, 9040, 9450, 9810, 14,800, 16,600, 21,000, 25,800

Determine if the two lower values are outliers at the 95% confidence level.

From Eq. 4.1

$$T_F = \frac{9.84+15.7}{9.84+15.7+...+25,800} = 1.89\text{x}10^{-4}$$

NO. OF DATA PTS. (N)	NUMBER OF OUTLIERS (K) 1	2	3	4	5	6
5	0.002509430					
6	0.001672250	0.018601832				
7	0.001194035	0.012816547				
8	0.000895258	0.009366702				
9	0.000696138	0.007144400	0.022621670			
10	0.000556794	0.005629040	0.017510286			
12	0.000379507	0.003753374	0.011383193	0.024342869		
14	0.000275179	0.002680305	0.007989488	0.016757755	0.029857087	
16	0.000208638	0.002009464	0.005915182	0.012236753	0.021471956	
18	0.000163613	0.001562288	0.004555290	0.009326496	0.016181849	0.025538801
20	0.000131735	0.001249348	0.003615721	0.007343488	0.012631246	0.019749088
25	0.000083413	0.000781552	0.002232367	0.004470921	0.007576622	0.011659573
30	0.000057518	0.000534646	0.001514134	0.003005319	0.005045072	0.007687078
35	0.000042046	0.000388638	0.001094050	0.002157950	0.003599015	0.005446583
40	0.000032071	0.000295195	0.000827314	0.001624311	0.002696096	0.004060003
50	0.000020418	0.000186846	0.000520445	0.001015350	0.001674343	0.002504470
60	0.000014130	0.000128806	0.000357328	0.000694227	0.001139937	0.001697704
80	0.000007914	0.000071795	0.000198176	0.000383055	0.000625715	0.000926945
100	0.000005052	0.000045699	0.000125768	0.000242361	0.000394680	0.000582869

Table 4.1: Critical values of Fisher's test statistic for lower outliers in an exponential distribution at the 95% confidence level.

From Table 4.1, the critical value of Fisher's test statistic for n=10 and k=1 is found to be 5.568×10^{-4}. Thus, since $t_f > T_F$, i.e., $5.568 \times 10^{-4} > 7.50 \times 10^{-5}$, the data point 0.049 is labeled an outlier at the 95% confidence level.

Example 4.2:

The following data come from an exponential distribution.

9.84, 15.7, 1300, 2260, 2690, 3010, 5190, 5880, 8470, 9040, 9450, 9810, 14,800, 16,600, 21,000, 25,800

Determine if the two lower values are outliers at the 95% confidence level.

From Eq. 4.1

$$T_F = \frac{9.84+15.7}{9.84+15.7+...+25,800} = 1.89 \times 10^{-4}$$

From Table 4.1, the critical value of Fisher's test statistic values are labeled as outliers at the 95% confidence level.

IV.3 Dixon's Method of Outlier Detection

Dixon (1950) developed a method for detecting outliers in normal samples, and Likes (1966) applied this method to exponential distributions. Dixon's method, like Fisher's, is a hypothesis test. The null hypothesis, H_o, is that all data come from the same exponential distribution. The alternate hypothesis, H_1, is that a certain number of data points, $k \leq n$, come from a distribution different from the distribution of the main body of data.

The application of Dixon's method is similar to that of Fisher's. It consists of calculating a test statistic, T_D, and then determining the probability of it being as large as it is.

The Dixon test statistic is

$$T_D = \frac{x_s - x_r}{x_q - x_p}, \tag{4.3}$$

where $x_1 \leq s_2 \leq \ldots \leq x_n$ are the data, $1 \leq p \leq r \leq s \leq n$, and $q-p > s-r$. For example, for a data set of size 10, q=10, s=3, r=1, and p=1.) For the upper case, Dixon's test statistic becomes

$$T_D = \frac{x_n - x_{n-k}}{x_n - x_{a+1}}, \tag{4.5}$$

where k is the number of upper outliers being tested for, and a is the suspected number of upper outliers. The use of a and b will be discussed in the next section.

The equation used to determine whether or not T_D is too large is

$$1 - F(t_D) = \frac{(n-p)!}{(n-q)!}(1-t_D)$$
$$X\{\sum_{i=1}^{q-s}\sum_{k=1}^{s-r}\frac{(-1)^{i+k}(q-r-i)![(n-s+k)t_D+(n-q+i)(1-t_D)]^{-1}}{(i-1)!(k-1)!(q-s-i)!(s-r-i)!(q-p-i)!(n-s+k)}$$
$$+\sum_{j=1}^{r-p}\sum_{k=1}^{s-r}\frac{(-1)^{q-s+j+k}(s-r+j-1)![(n-s+k)t_D+(n-r+j)(1-t_D)]^{-1}}{(j-1)!(k-1)!(r-p-j)!(s-r-k)!(q-r+j-1)!(n-s+k)}\}, \tag{4.7}$$

where F (t_D) is the probability that T_D is less than t_D, $P(T_D < t_D)$. T_D is substituted for t_D on the right hand side of Eq. 4.7, and the value obtained for $F(T_D)$ is the percent confidence which can be used in labeling the suspicious data as outliers. Thus, the critical value of $F(T_D)$ is either 0.95 or 0.99. (Note that if q=s, the first set of double sums is equal to zero, and if r=p, the second set of double sums is equal to zero.)

Example 4.3:

Using the data from Example 1, determine if the lower data point is an outlier at the 95% confidence level according to Dison's Method.

From Eq. 4.6,

$$T_D = \frac{2.22-.0549}{290-.0549} = .00747.$$

From Eq. 4.7, with n-b=q=10, p=1, k+1=s=2, and r=1,

$$1-F(.00747) = \frac{(10-1)!}{(10-10)!}(1-.00747)$$

$$X \{ \sum_{i=1}^{10-2} \sum_{k=1}^{2-1} \frac{(-1)^{i+k}(10-1-i)![(10-2+k)(.00747)}{(i-1)!(k-1)!(10-2-i)!(2-1-k)!}$$

$$\frac{+ (10-10+2)(1-.00747)]^{-1}}{(10-1-i)!(10-2+k)}$$

$$+ \sum_{j=1}^{1-1} \sum_{k=1}^{2-1} (\ldots)\}$$

$$= 9!(.99253) \sum_{i=1}^{8} \frac{(-1)^{i+1}[(.0672)+(.99253)i]^{-1}}{(i-1)!(8-i)!(9)}$$

$$= .835.$$

Thus,

$$F(.00747) = .165.$$

And thus, the value .0549 is not labeled as an outlier at the 95% confidence level, because F(.00747) = 0.165 < 0.95.

If Table 4.2 had been used, the critical value of Dixon's test statistic would have been found to be 0.140. And since T_D = .00747 < .0140, 0.0549 would not have been labeled an outlier.

Table 4.2. Critical values of Dixon's test statistic at the 95% confidence level for a single outlier with no suspected outliers on the opposite end of the spectrum. Taken from Barnett and Lewis (1974), pp. 293 and 296.

No. of Data Pts. (n)	$\frac{x_2-x_1}{x_n-x_1}$	$\frac{x_n-x_{n-1}}{x_n-x_1}$
5	0.429	0.830
6	0.316	0.728
7	0.246	0.746
8	0.198	0.717
9	0.165	0.694
10	0.140	0.675
12	0.106	0.644
14	0.085	0.620
16	0.070	0.601
18	0.059	0.586
20	0.051	0.573

Example 4.4:

The following data came from an exponential distribution.

1340, 2160, 4330, 4610, 8410, 11,500, 15,500, 17000,

31,000, 127,000

Determine if the largest data point is discordant at the 95% confidence level.

From Eq. 4.5,

$$T_D = \frac{127{,}000-31{,}000}{127{,}000-1340} = 0.764.$$

From Eq. 4.7, with q=n=10, p=q=1, s=n=10, and r=n-k=9,

$$1-F(.764) = \frac{(10-1)!}{(10-10)!} \quad (1-.764)$$

$$X\{ \sum_{i=1}^{10-10} \sum_{k=1}^{10-1} (\ldots)$$

$$+ \sum_{j=1}^{a-1} \sum_{k=1}^{10} \frac{(-1)^{10-10+j+k}(10-9+j-1)![(10-10+k)(.764)}{(j-1)!(k-1)!(9-1-j)!(10-9-k)!(10-9+j-1)}$$

$$\frac{+(10-9+j)(1-.764)]^{-1}}{(10-10+k)}\}$$

$$= 9!(.236) \sum_{j=1}^{8} \frac{(-1)^{j+1}[.764+(j+1)(.236)]^{-1}}{(j-1)!(8-j)!}$$

$$= (85{,}652)(1.6732\text{x}10^{-7}) = .0143.$$

Thus,

$$F(.764) = .9856.$$

And thus, since F(.764) > 0.95, the upper data point is discordant at the 95% confidence level. However, it is <u>not</u> discordant at the 99%

confidence level. And since $T_D = 0.764 \quad 0.675$, 127,000 would have been labeled an outlier had Table 4.2 been used.

IV. Determination of the Number of Outliers

Until now, no mention has been made of how to determine the number of outliers which should be tested for in the exponential case. Unfortunately, it is much more difficult in the exponential case than in the gamma case to make this determination, because both upper and lower outliers are of concern. However, due to the simplicity of the outlier tests themselves, this problem poses no major obstacle.

When the Fisher test statistic is used, the number of data points, which are to be tested for, is determined in a different way than is the gamma case. The largest gap between consecutive data points relative to the upper gap value is located, where $x_1 \quad x_2 \quad \ldots \quad x_n$. Then, depending upon whether the gap occurs above or below the mean value of the data, either upper or lower outliers, respectively, are tested for.

A difference between the gamma and exponential cases also exists in the criteria which are used to determine when the testing is complete. The first criterion, which must be met before testing is complete, is that both ends of the data spectrum must have negative tests for outliers. Even though there might be a larger relative gap between data in the upper region of the data than in the lower region, and no upper outlier is found, a lower outlier could still be present. This is because different distributions are used for the upper and lower regions to determine the critical values of T_F.

The second criterion which must be met is that lower outliers do not swamp upper outliers, and vise versa. Say, for example, that the upper region of a set of data is tested for outliers, and that some are found. Then the lower region is tested, and outliers are found. In this case, the upper region must be retested, because outliers there might have been swamped by the lower outliers during the first test.

Example 4.5:

The following data are known to come from an exponential distribution.

0.523, 35.6, 66.5, 105, 195, 197, 278, 282, 302, 1430

Determine the number of discordant values in this data set at the 95% confidence level.

The largest relative gap in the data occurs between 0.523 and 35.6. Thus, a single lower outlier is tested for. From Eq. 4.1,

$$T_F = \frac{0.523}{0.523+\ldots+1430} = .000181.$$

From Table 4.1, the critical value of Fisher's test statistic is found to be .0005568. Thus, 0.523 is found at present to be discordant, and is removed from the data.

The largest relative gap in the data now occurs between 302 and 1430. Thus, a single upper outlier is tested for. From Eq. 2.4,

$$T_F = \frac{1430}{35.6+\ldots+1430} = 0.495.$$

The critical value of Fisher's test statistic (Table 2.1) is found to be 0.4775. Thus, 1430 is found to be discordant, and will only be tested again for discordancy if more lower outliers are found.

Now, since an upper discordant value was found, the value 0.523 must be retested. From Eq. 4.1,

$$T_F = \frac{0.523}{0.523+\ldots+302} = .000684.$$

From Table 4.1, the critical value of Fisher's test statistic is found to be 0.0008953. Thus, 35.6 is not discordant.

The largest relative gap between consecutive data points in the upper end of the spectrum is between 197 and 278. Thus, 3 upper outliers are tested for. From Eq. 2.4,

$$T_F = \frac{278+282+302}{35.6+...+302} = .590.$$

From Table 2.3, the critical value of Fisher's test statistic is found to be greater than 0.824. Thus, 278, 282, and 302 are not outliers.

The testing is now complete, since no more upper or lower outliers are present. Thus, two discordant values, 0.523 and 1430, were found in the data set.

When the Dixon test statistic is used, the number of data points to be tested for being outliers is also determined by using the method of finding the largest relative gap. However, due to its greater flexibility, there are some added steps in testing for outliers.

The greater flexibility of Dixon's method comes from the fact that when the testing is being applied to find upper outliers, it can be taken into account that there might also be lower outliers present and vise versa. Thus, the effects of masking and swamping by outliers from the opposite end of the data spectrum can be reduced.

The method used to take into account that outliers might be present on the opposite end of the data spectrum is as follows. Suppose that suspicious data on the upper end of the spectrum are being tested. The test is first applied assuming that no lower outliers are present, i.e., "a" in Eq. 4 is given the value 0. If these data are discordant, they are labeled as outliers. However, if they are not discordant, the testing on them continues. First, using the gap method, the number of possible lower outliers is determined, and assigned the value a. Then, the value of a is put into Eq. 4, and the test is repeated. If the upper data are not found to be discordant this time, then they are not outliers, and the testing on them is complete, for the time being at least. If the upper data are found to be discordant when potential, lower outliers are present, then the potential, lower outliers must be tested for discordancy. This is done by using Eq. 4.6, and assuming that there are no upper outliers, i.e., b is given the value 0. If the lower data are found to be discordant by this method, then <u>both</u> the upper <u>and</u> the lower data are outliers. If the lower data are not found to be discordant, then it is tested again assuming that the upper data might be outliers, i.e., b is given the value of the number of suspected upper outliers. If, by this method, the lower data are found to be non-discordant, then <u>neither</u> the upper <u>nor</u> the lower data are discordant, and the testing is complete. If, on the other hand, the lower data are found to be non-discordant by this method, then it is not known whether both the upper and the lower data are discordant or both are non-discordant. Thus, to be conservative, they must be assumed to be non-discordant, are not labeled as outliers, and the testing is complete.

Since, when Dixon's method is used, masking from the opposite end of the data is taken into account, only one criterion need be met before the testing is complete. It is that both ends of the data must have undergone testing for outliers with the results being negative.

Example 4.6:

Using Eq. 4.6 to test for a lower outlier gives,

$$T_D = \frac{35.6-.523}{1430-.523} = .0245.$$

From Table 4.2, the critical value of Dixon's test statistic is found to be 0.140. Thus, 0.523 must be further tested with Eq. 4.6, setting b=1,

$$T_D = \frac{x_2-x_1}{x_9-x_1} = \frac{35.6-.523}{320-.523} = 0.110.$$

From Eq. 4.7,

$$1-F(T_D) = .192$$

and $$F(T_D) = 0.808.$$

Thus, since $F(T_D) < 0.95$, 0.523 is definitely not an outlier. Further, unless there is more than the one suspected upper outlier, there are no lower outliers.

Using Eq. 4.5 to test for a single upper outlier gives,

$$T_D = \frac{1430-302}{1430-.523} = 0.789.$$

From Table 4.2, the critical value of Fisher's test statistic is found to be 0.675. Thus, since $T_D > t_d$, i.e., 0.789 > 0.675, 1430 is labeled an outlier, and removed from the data.

The greatest relative gap between data in the upper end of the spectrum is now between 197 and 278. Since it is not plausible that there would be 3 outliers in a data set of size 9, testing for upper outliers will not be continued. And, since there is only the one upper outlier, there are no more lower outliers. Thus, according to Dis method, there is only one upper outlier in the data set of Ex. 4.5 at the 95% confidence level.

IV.5 Properties and Comparisons of Tests

The power of the tests for both upper and lower outliers show very little variation as the value of the failure rate, λ, varies. In fact, as the failure rate goes from 10^{-3} hr^{-1} to 10^{-7} hr^{-1}, the variation in the power of Fisher's test is so small that the calculated powers can be plotted on the same line (See Fig.'s 4.1 and 4.2).

When upper outliers are being tested for, there is very little variation in the power of the tests as the number of data points in the data set varies. In fact, as the number of data points n varies from 10 to 30, the variation is so small for Fisher's method that the calculated powers can be plotted on the same line (See Fig. 4.3).

However, when lower outliers are being tested for, there is a distinct variation in the power of the tests as the number of data points varies. As the value of n increases, the power of the tests decreases, i.e., saturation occurs at larger values of κ (See Fig. 4.4).

Fisher's method of outlier detection is better than Dixon's method in every case. For detecting upper outliers, the power of Fisher's method is, in general, 5% more powerful (See Fig. 4.5). For detecting lower outliers, Fisher's method is substantially more powerful than Dixon's method. However, neither method is very powerful (See Fig. 4.6).

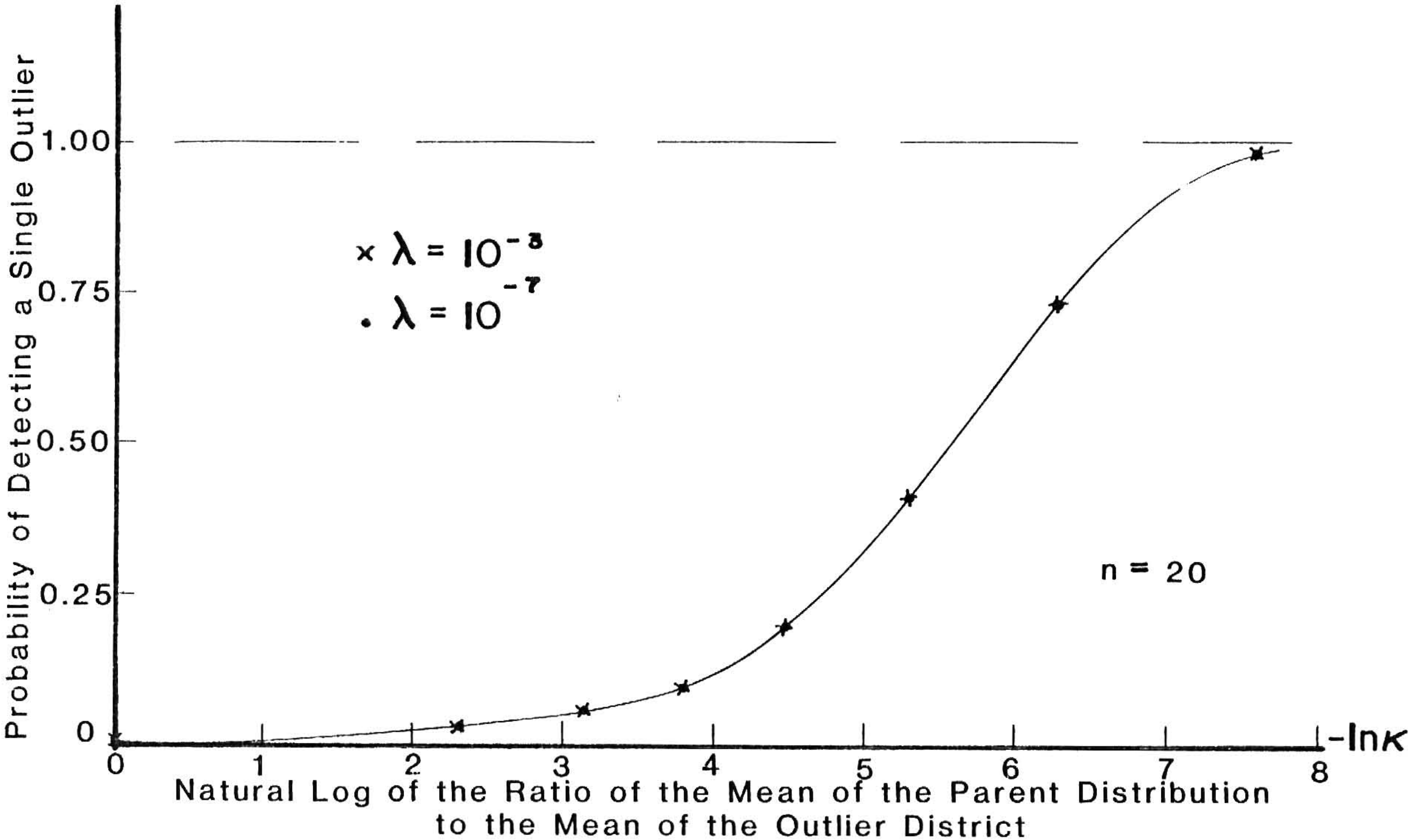


Figure 4.1: Lack of variation in the power of Fisher's method for detecting a single lower outlier as the failure rate changes.

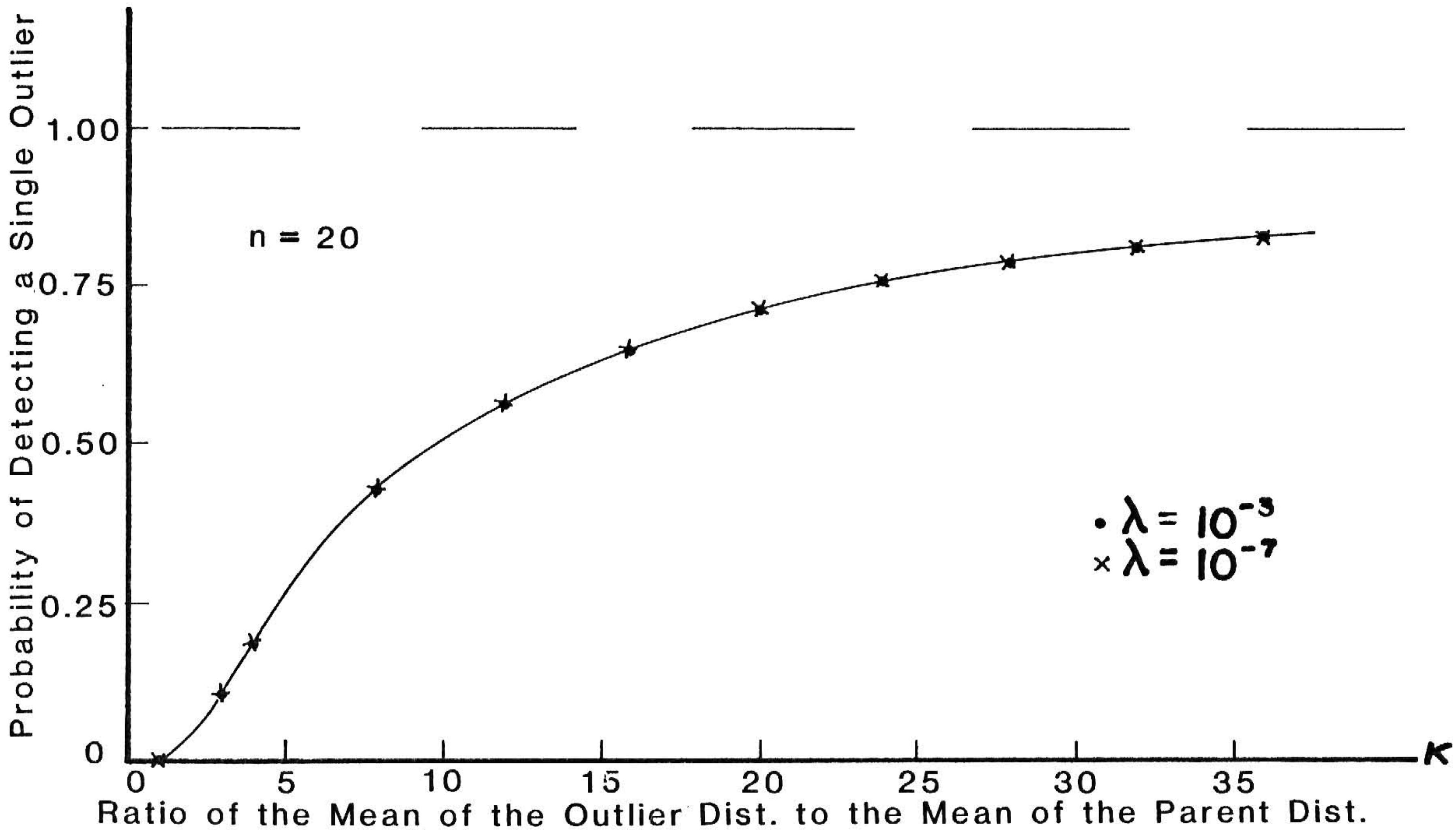


Figure 4.2: Lack of variation in the power of Fisher's method for detecting a single upper outlier as the failure rate changes.

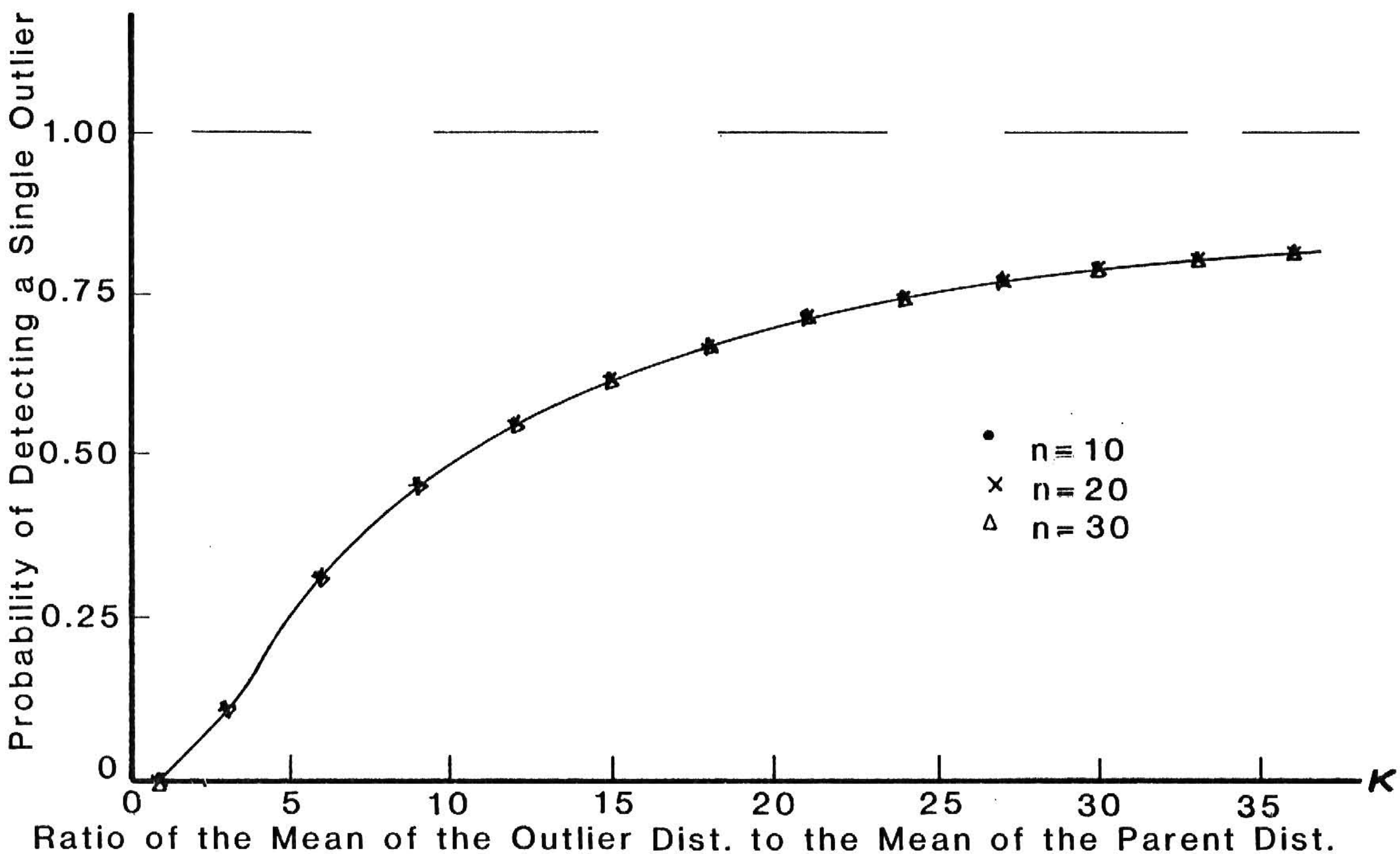


Figure 4.3: Lack of variation in the power of Fisher's method for detecting a single upper outlier as the number of data points in the data set changes (failure rate = 10^{-6} hr^{-1}).

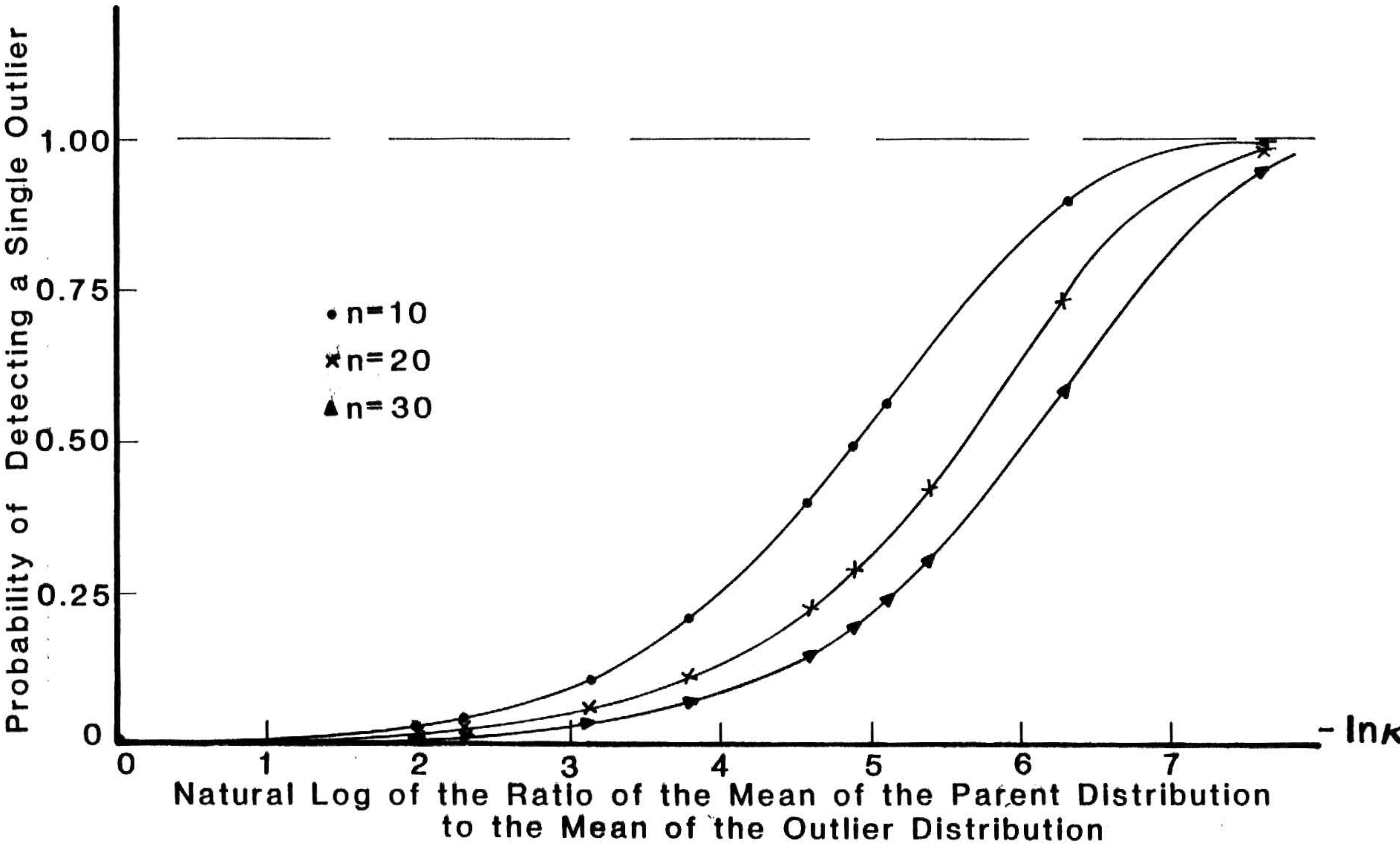


Figure 4.4: Variation in the power of Fisher's method for detecting a single lower outlier as the number of data points in the data set changes (failure rate = 10^{-6} hr^{-1}).

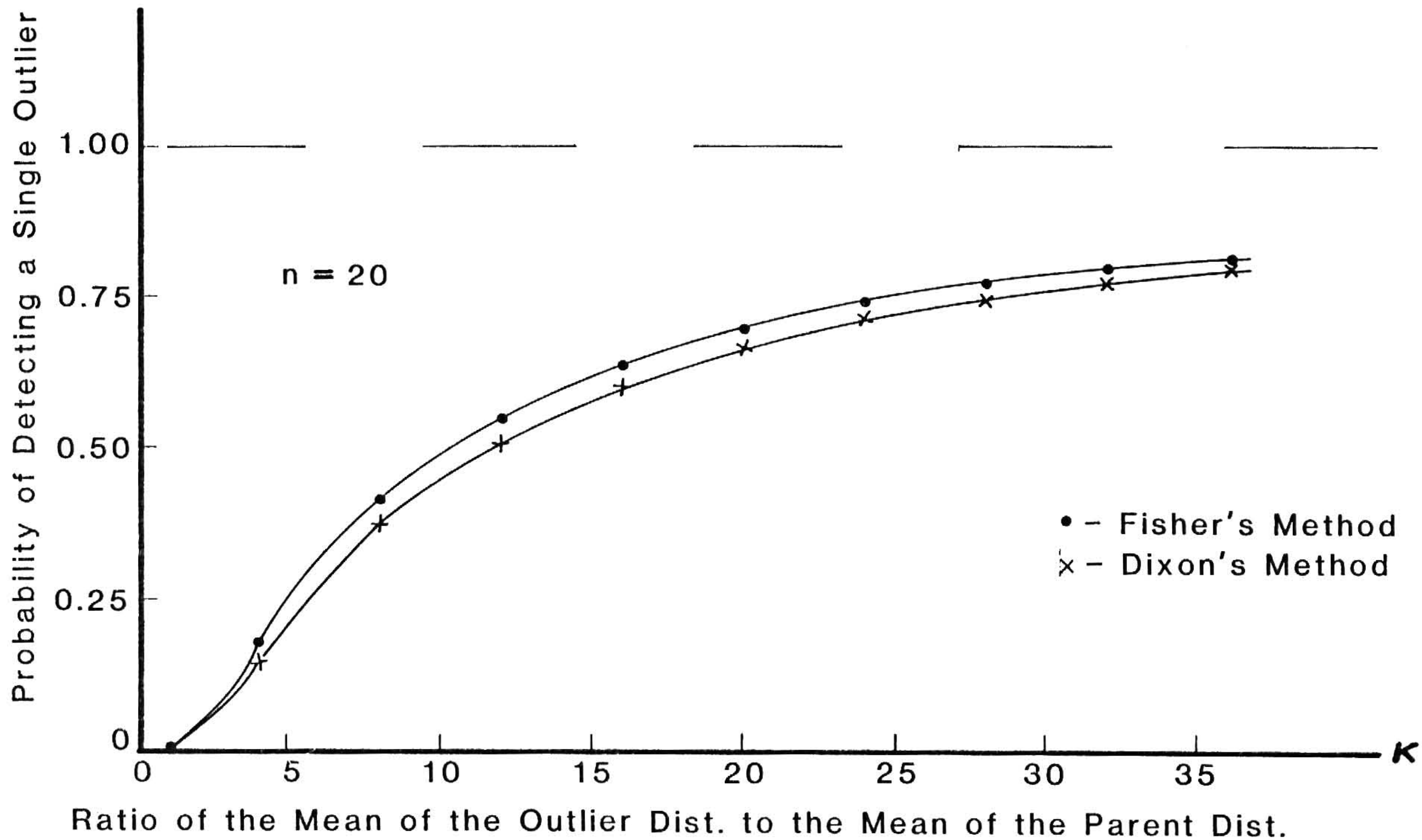


Figure 4.5: Comparison of the power of Fisher's method and Dixon's method for detecting a single upper outlier in an exponential distribution with failure rate = 10^{-6} hr^{-1}.

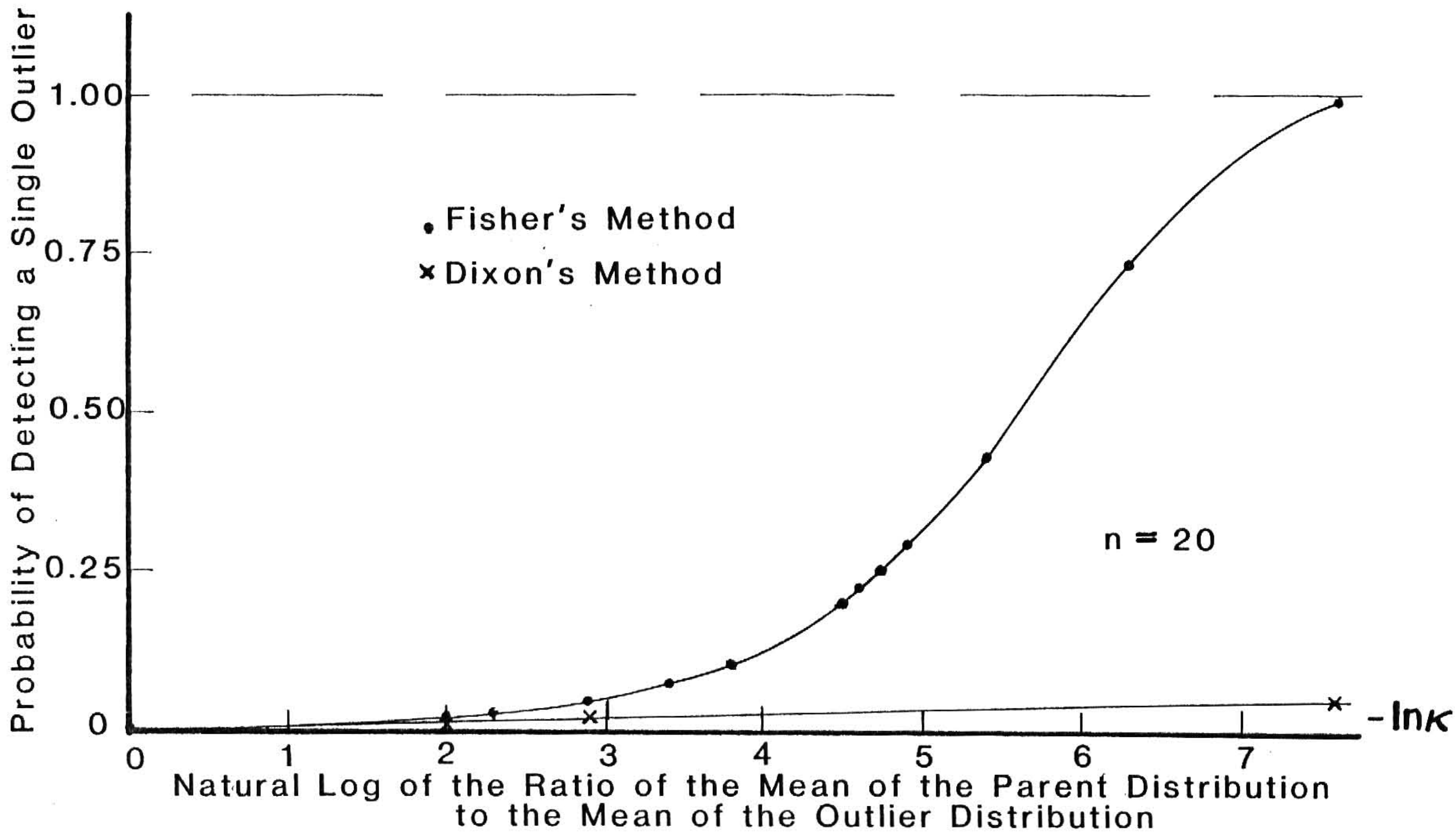


Figure 4.6: Comparison of the power of Fisher's method and Dixon's method for detecting a single lower outlier in an exponential distribution with failure rate = 10^{-6} hr^{-1}.

As in the case of outlier detection in gamma distributions, although the theoretical confidence level of all calculations carried out was 95%, the actual confidence level was greater than 99%. This is shown by the fact that the power curves at $\kappa = 1$ or $\ln\kappa = 0$ are below 0.01 (See Fig.'s 4.1 to 4.6). Also, since the actual confidence levels* of the various tests are not equal, it may not be completely "fair" to compare their powers.

---

*The probability of accepting H_o (outlier and data are from the same distribution) is less than the critical level of significance chosen a priori, e.g., a critical level of significance chosen, a priori, was 0.05 (5% chance of incorrectly rejecting H_o), while there was actually a much smaller chance than 5% of incorrectly rejecting H_o.

V. Conclusions

For the detection of a single outlier in a gamma distribution, the least desirable method is the integration method. The reason for this is twofold. First, the power of the integration method compared to other available methods is low. And second, the computations which are necessary to apply the integration method are extremely cumbersome. Fisher's method of outlier detection is much better than the integration method. However, when the value of the parameter is unknown, which often is the case for the distributions that model failure rate data, the power of Fisher's method is substantially reduced. Surprisingly, the best method for detecting outliers in a gamma distribution is the conversion to normal method. In every case investigated, the normal conversion method was the most powerful method for detecting a single outlier. The normal conversion method is also attractive, because it is simple, requiring only use of a hand calculator and a table of critical values of the Grubbs test statistic.

For the detection of a single outlier in (F_i, T_i) data, the methods in which the gamma parameters are calculated using the potential outlier fail completely. When there are only 20 data points per data set, an outlier will distort the values of the gamma parameters to such an extent that the outlier will not be detected. The cumulative marginal method with the gamma parameters calculated by the MMMM is a poor method, and caution should be exercised with its use. This is because for larger failure rates, its actual confidence level is less than its theoretical confidence level, and for smaller failure rates, its power is very low. For smaller failure rates, i.e., $\bar{\lambda} < 5 \times 10^{-5}\ hr^{-1}$, and with T = 10,000 hr and n = 20, the most powerful practical method of single outlier detection is the binomial method used in conjunction with the homogeneous failure model. For larger failure rates, i.e., $\bar{\lambda} > 5 \times 10^{-5}\ hr^{-1}$, the most powerful practical method is the cumulative marginal method with the gamma parameters calculated by the PMMM without using the potential outlier. This method is also the best method of detecting an outlier when $\bar{\lambda} < 5 \times 10^{-5}$ and all T_i are not the same.

For the detection of a single outlier in the exponential distribution, Fisher's method is the best. It is approximately 5% more powerful than Dixon's method for detecting an upper outlier, and much more powerful than Dixon's method for detecting a single lower outlier.

One property of all of the recommended methods of outlier detection in the above cases is that when the tests are performed for a theoretical confidence level of 95%, the actual confidence level is well above 99%. Thus, the theoretical confidence level is seen to be conservative. Also, since the actual confidence levels* of the various test are not equal, it may not be "completely" "fair" to compare their powers.

---

*The probability of accepting H_o (outlier and data are from the same distribution) is less than the critical level of significance chosen, a priori, e.g., a critical level of significance chosen, a priori, was 0.05 (5% chance of incorrectly rejecting H_o), while there was actually a much smaller chance than 5% of incorrectly rejecting H_o.

VI. ACKNOWLEDGEMENTS

The author wishes to thank Professors N. Dean Eckhoff, George Milliken, and Dallas Johnson for their advice and assistance throughout the course of this work. Special thanks also goes to Mrs. Mildred Buchman and Miss Sherry Berner for typing this manuscript and also to Miss Barb Wieliczka for drawing the figures. This research was supported by a subcontract with EG&G, who was under contract to the U.S. Nuclear Regulatory Commission.

VII. Bibliography

Anscombe, F. J. (1960), "Rejection of Outliers", Technometrics, Vol. 2, pp. 123-147.

Anscombe, F. J. and Tukey, J. W. (1963), "The examination and Analysis of Residuals", Technometrics, Vol. 5, pp. 141-160.

Bajpai, A. C., Mustoe, L. R., and Walker, D. (1974), Engineering Mathematics, John Wiley & Sons, New York.

Barnett, V., and Lewis, T. (1978), Outliers in Statistical Data, John Wiley & Sons, New York.

Basu, A. P. (1965), "On Some Tests of Hypotheses Relating to the Exponential Distribution when Some Outliers are Present", Journal of the American Statistical Societyy Vol. 60, pp. 548-559.

Bevington, P. R. (1969), Data Reduction and Error Analysis for the Sciences, McGraw-Hill Book Co., New York.

Dixon, W. J. (1950), "Analysis of Extreme Values", Annals of Mathematical Statistics, Vol. 21, pp. 488-506.

Dixon, W. J. (1953), "Processing Data for Outliers", Biometrics, Vol. 9, pp. 74-89.

Ferguson, T. S. (1961), "On the Rejection of Outliers", Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, pp. 253-287.

Fieller, N. R. J. (1976), Some Problems Related to the Rejection of Outlying Observations, Ph.D. Thesis, University of Sheffield.

Fisher, R. A. (1929), "Tests of Significance in Harmonic Analysis", Proceedings of the Royal Society A, Vol. 124, Vol. 125, pp. 54-59.

Grubbs, F. E. (1950), "Sample Criteria for Testing Outlying Observations", Annals of Mathematical Statistics, Vol. 21, pp. 27-58.

Grubbs, F. E. (1969), "Procedures for Detecting Outlying Observations in Samples", Technometrics, Vol. 11, pp. 1-21.

Hawkins, D. M. (1973), "Repeated Testing for Outliers", Statistica Neerlandica, Vol. 27, pp. 1-10.

Kimber, A. C. (1979), "Tests for a Single Outlier in a Gamma Sample with Unknown Shape and Scale Parameters", Applied Statistics, Vol. 28, No. 3, pp. 243-250.

Kreyszig, D. (1979), Statistische Methoden und ihre Anwendungen, Hubert & Co., Goettingen, W. Germany.

Lewis, T., and Fieller, N.R.M. (1979), "A Recursive Algorithm for Null Distributions for Outliers: I. Gamma Samples, Technometrics, Vol. 22.

Likes, J. (1966), "Distribution of Dixon's Statistics in the Case of an Exponential Population", Metrika, Vol. 11, pp. 46-54.

McMillan, R. G. (1971), "Tests for One of Two Outliers in Normal Samples with Unknown Variance", Technometrics, Vol. 13, pp. 87-100.

Mann, Nancy R., Schafer, Ray E., and Singpurwalla, Nozer D. (1974), Methods for Statistical Analysis of Reliability and Life Data, John Wiley and Sons, New York.

Mood, A. M., Graybill, F. A., and Boes, D. C. (1950), Introduction to the Theory of Statistics, McGraw-Hill Book Co., New York.

Prescott, P. (1978), "Examination of the Behaviour of Tests for Outliers When More Than One Outlier is Present", Applied Statistics, Vol. 27, No. 1, pp. 10-25.

Rosner, B. (1975), "On the Detection of Many Outliers", Technometrics, Vol. 17, pp. 221-227.

Shapiro, S. S. and Wilk, M. B. (1972), "An analysis of Variance Test for the Exponential Distribution", Technometrics, Vol. 14, pp. 355-370.

Shultis, J. K., Johnson, D. E., Milliken, G. A., and Eckhoff, N. D. (1980), Non-Conjugate Prior Distributions in Compound Failure Models, Prepared for U.S. Nuclear Regulatory Commission (Contract No. NRC-04-79-182).

Tietjen, G. L. and Moore, R. H. (1972), "Some Grubbs-type Statistics for the Detection of Several Outliers", Technometrics, Vol. 14, pp. 583-597.

Tiku, M. L. (1975), " A New Statistic for Testing Suspected Outliers", Communications in Statistics, Vol. 4, No. 8, pp. 737-752.

VIII. APPENDICES

Appendix A: Power Curve Generation

In the past, power curves for outlier detection techniques have been generated strictly via simulation. That is, sets of data were generated from a parent distribution, outliers were generated from a second distribution, and the outliers were then tested for discordancy against the data sets.

However, there is a second, and better method for generating a power curve for single outlier detection. It is developed as follows for the gamma family using Fisher's method of outlier detection (See Section II.2).

Let $g(\alpha,\beta)$ be the parent gamma distribution used to generate the data points x_{ij}, $i = 1, 2, \ldots , N$, $j = 1, 2, \ldots , n-1$, where N is the number of data sets and n is the number of data points per data set.

Let $g(\alpha_k,\beta_k)$, be the gamma distribution used to generate the outliers x_i', $i = 1, 2, \ldots , N$.

Let t_F, the critical value of Fisher's test statistic, be a known value.

And finally, let x_{ci} be calculated as follows,

$$x_{ci} = \frac{t_F \sum_{i=1}^{n-1} x_{ij}}{i-t_F} \quad . \qquad \text{(A.1)}$$

Thus, since

$$t_F = \frac{x_{ci}}{\sum_{i=1}^{n-1} x_{ij} + x_{ci}} \quad ,$$

x_{ci} is seen to be the critical value which is used to determine whether or not a data point is discordant or not.

Now, if one desires to know the probability that x_i' is detected as being an outlier, it is equivalent to wanting to know $P(x_i' \geq x_{ci})$.

When the values of x_{ci} are plotted, they give rise to a curve whose form is unknown, say $f_1(x)$. Letting

$$f(x) = \frac{f_1(x)}{N} \quad , \tag{A.2}$$

$f(x)$ is seen to be the probability distribution of x_{ci}. $F(x)$ is the cumulative distribution of x_{ci}, i.e.,

$$F(x) = \int_0^{\infty} f(x)dx \quad , \tag{A.3}$$

which can be thought of as the probability that any given value, say a, is greater than x_{ci}, i.e., $P(a \geq x_{ci})$.

In order to simplify matters, rather than having $f(x)$ be a continuous function of unknown form it can be a histogram with m intervals of width Δx, $f(x_1)$, where the x_1 occur in the middle of the intervals, and

$$f(x_\ell) = \frac{m_\ell}{N} \quad , \tag{A.4}$$

where m_ℓ is the number of x_{ci} which occur between $x_\ell - \Delta x$ and x_1. Thus,

$$F(x_b) = \sum_{\ell=1}^{b} f(x_\ell) \tag{A.5}$$

is approximately the average probability over the interval $(x_b - 1/2\ \Delta x,$ $x_b + 1/2\ \Delta x)$ that a given value is greater than x_{ci}, i.e.,

$$F(x_b) \simeq \frac{1}{\Delta x} \int_{x_b - \frac{1}{2}\Delta x}^{x_b + \frac{1}{2}\Delta x} F(x)dx \tag{A.6}$$

(see Fig. A.1).

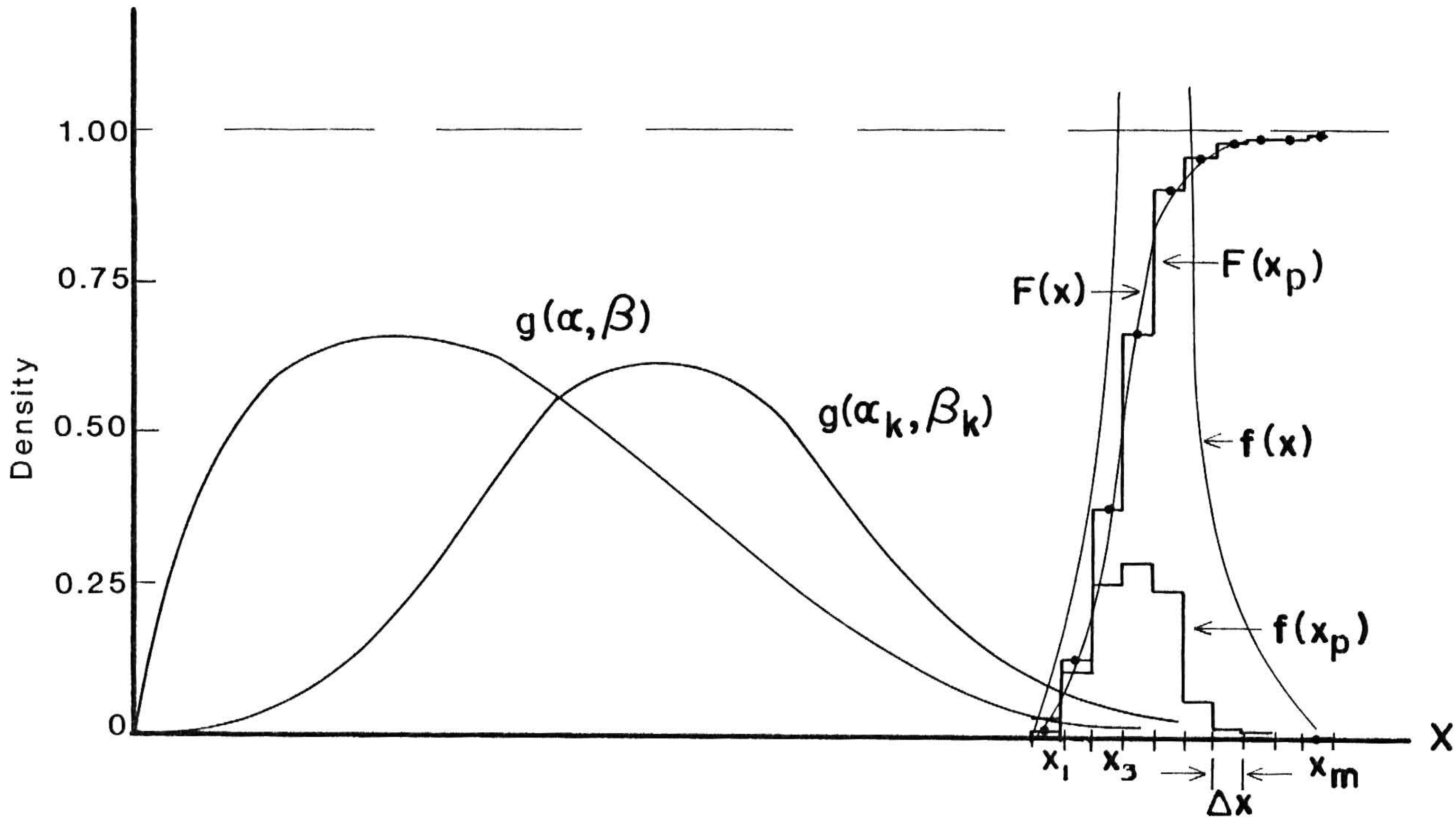


Figure A.1: Relation between the parent distribution, the outlier-generating distribution, the continuous distribution of x_{ci}, the continuous cumulative distribution of x_{ci}, the histogram of x_{ci}, and the cumulative of the histogram of x_{ci}.

The probability that a point x_i' generated by the function $g(\alpha_k, \beta_k)$ is detected as an outlier, $P(x_i' \geq x_{ci})$, is the probability that x_i' will have a particular value times the probability that that value is greater than x_{ci} summed over all possible values of x_i':

$$P\ (x_i' \geq x_{ci}) = \int_0^\infty F(x) g(\alpha_k, \beta_k) dx$$

$$\simeq \sum_{\ell=1}^{m} F(x_\ell) \int_{x_\ell - \frac{1}{2}\Delta x}^{x_\ell + \frac{1}{2}\Delta x_2} g(\alpha_k, \beta_k) dx + \int_{x_m + \frac{1}{2}\Delta x}^{\infty} g(\alpha_k, \beta_k) dx \qquad \text{(A.7)}$$

(see Fig. A.2).

By using some scheme to determine values for α_k and β_k, a power curve showing the power of Fisher's method of outlier detection can be generated. One possible scheme is as follows:

1) Set $\mu_o = \mu = \alpha/\beta$
2) Set $\sigma^2 = \alpha/\beta^2$
3) Set $\kappa = 1$
4) $\alpha_k = \mu^2/\sigma^2$
5) $\beta_k = \mu/\sigma^2$
6) Find $P(x_i' \geq x_{ci})$
7) $\kappa = \kappa + 1$
8) $\mu = \kappa\mu_o$
9) If $P(x_i' \geq x_{ci}) \leq 0.99$, go to step 4, otherwise stop.

The above development was done for Fisher's method. However, it can easily be extended to other methods. The only thing which must be changed is the manner in which the x_{ci} are calculated. Also, the development can be extended to a family other than the gamma family simply changing the form of g(...).

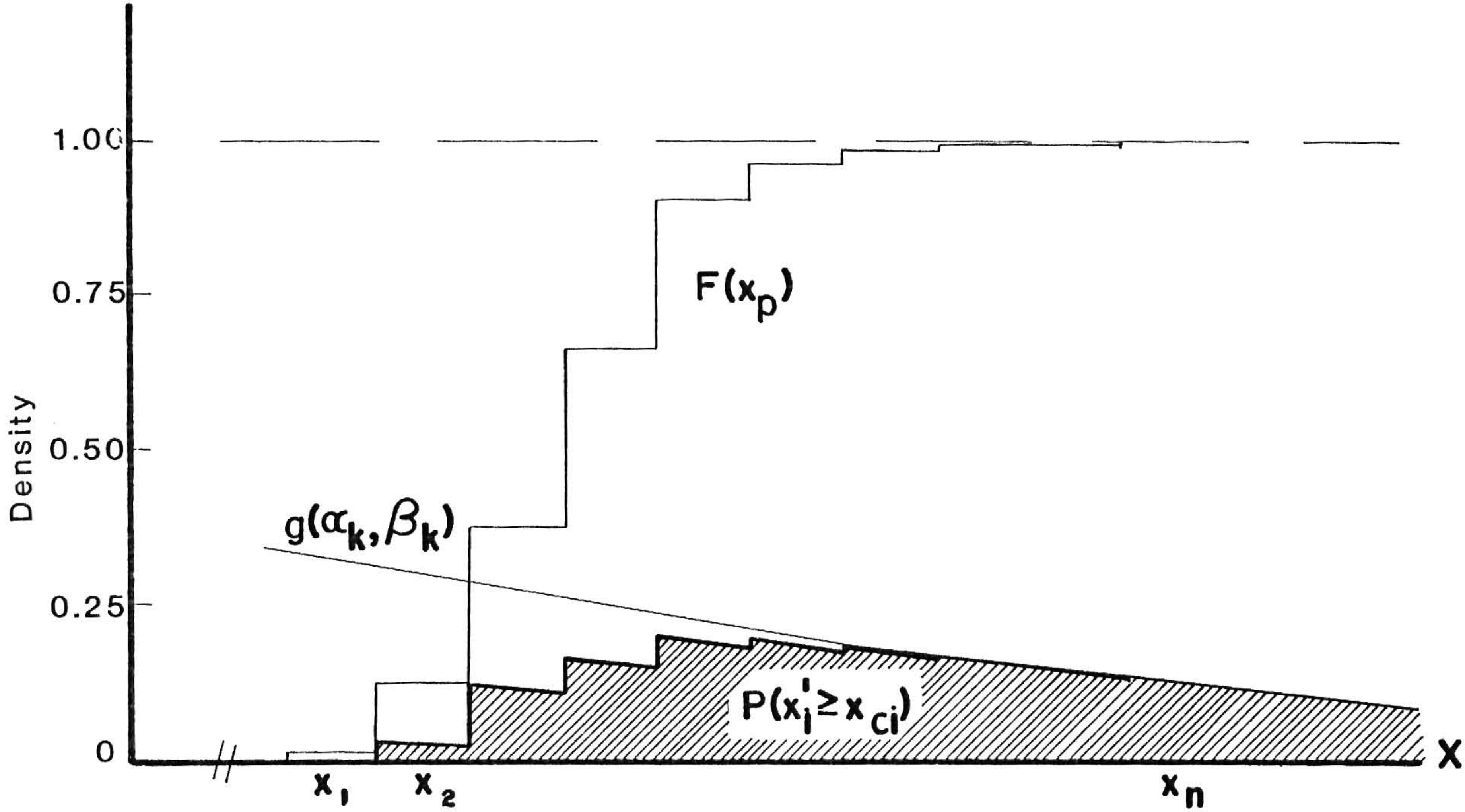


Figure A.2: Graphical illustration of the calculation of $P(x'_i \geq x_{ci})$.

Appendix B: Critical Values of Fisher's Test Statistic

The following two programs are for calculating critical values of Fisher's test statistic, t_F. The theory behind these programs is as follows.

The critical values of Fisher's test statistic are found from Eq.'s 2.4 and 4.2, i.e.,

$$P(T_F \le t_F) \le \binom{n}{k} P[F_{2k\alpha,2(n-k)\alpha} > \frac{(n-k)t_F}{k(1-t_F)}], \qquad (B.1)$$

where $P(T_F \le t_F)$ is either 05. or .01 for upper outliers for either the 95% or 99% confidence level, respectively, and $P(T_F \le t_F)$ is either .95 or .99 for lower outliers for either the 95% or 99% confidence level, respectively. Thus, if an x can be found which satisfies

$$\int_0^x F_{2k\alpha,2(n-k)\alpha}(x)dx = P(T_F \le t_F)/\binom{n}{k}, \qquad (B.2)$$

then

$$x = \frac{(n-k)t_F}{k(1-t_F)}, \qquad (B.3)$$

$$t_F = \frac{xk}{xk+n-k}. \qquad (B.4)$$

The following programs find the values of x which satisfy Eq. B.2 by an iterative method, and then the critical value of Fisher's test statistic is found via Eq. B.4.

The following is a program for generating the critical values of Fisher's test statistic for upper outliers in a gamma distribution.

In the driving program:

N = number of data points in a data set

K = number of outliers in a data set

C = constant in the F-distribution

A = lower limit of integration

B = upper limit of integration

X(I,J) = values of x in Eq. B.2

T(I,J) = critical values of Fisher's test statistic

The subroutine 'FDIST' is for calculating values of the F-distribution.

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 INT(20,10),NA,KA,NKA
      DIMENSION T(10,20,10),X(20,10),NR(20),ALPH(10),INMIN(10)
      COMMON /CONST/ C,NA,KA,N,K
      EXTERNAL FDIST
C
C *** READ IN VALUES OF N (5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 25,
C *** 30, 35, 40, 50, 60, 80, 100)
C
      READ(5,8) (NR(I),I=1,19)
    8 FORMAT (19I3)
C
C *** SET THE PERCENT OF THE DISCORDANCY TEST
C
C *** USE 'DO 3 IPCT=1,2' FOR BOTH 95% AND 99% CONFIDENCE LEVELS
      IPCT=1
      IF(IPCT.EQ.1)PCT=0.05
      IF(IPCT.EQ.2)PCT=0.01
      DO 3 JOT=1,10
      ALPHA=0.5*DFLOAT(JOT)
      IF(JOT.EQ.9)ALPHA=6.0D0
      IF(JOT.EQ.10)ALPHA=0.25D00
      ALPH(JOT)=ALPHA
      DO 1 J=1,6
      K=7-J
C
C *** ALLOW ONLY 33% OF DATA TO BE OUTLIERS
C
      IF(K.EQ.1)I1=1
      IF(K.EQ.2)I1=2
      IF(K.EQ.3)I1=5
      IF(K.EQ.4)I1=7
      IF(K.EQ.5)I1=8
      IF(K.EQ.6)I1=10
      IF(K.EQ.7)I1=11
      IF(K.GE.8)I1=12
      NOI=20-I1
      INMIN(K)=I1
      DO 1 M=1,NOI
      I=20-M
      N=NR(I)
C
C *** CALCULATE DESIRED PRECISION AND MAXIMUM VALUE OF INTEGRAL
C
      FN=DLGAMA(DFLOAT(N)+1.D0)
      FK=DLGAMA(DFLOAT(K)+1.D0)
      FNK=DLGAMA(DFLOAT(N-K)+1.D0)
      FAC=DEXP(FN-FK-FNK)
      PREC=0.001D00/FAC
      D=PCT/FAC
      TINTMX=1.0D0-D
C
C *** AVOID INTEGRAL EQUALING 1 (DUE TO COMPUTER ACCURACY)
C
```

```
      IF(D.LE.1.D-11)X(I,K)=1.240DC
      IF(D.LE.1.D-11)GO TO 1
C
C *** CALCULATE CONSTANT IN F DISTRIBUTION
C
      NA=DFLOAT(N)*ALPHA
      KA=DFLCAT(K)*ALPHA
      NKA=DFLOAT(N-K)*ALPHA
      GNA=DLGAMA(NA)
      GKA=DLGAMA(KA)
      GNKA=DLGAMA(NKA)
      GAM=DEXP(GNA-GKA-GNKA)
      C=GAM*(DFLOAT(K)/DFLOAT(N-K))**KA
      TOTINT=0.0D0
C
C *** DETERMINE FIRST INTERVAL OF INTEGRATICN
C
      A=0.0D0
      IF(K.EQ.6.AND.I.EQ.19)B=1.24CD0
      IF(K.LE.5.AND.I.EC.19)B=0.80DC*X(I,K+1)
      IF(I.LE.18)B=0.80D0*X(I+1,K)
      IF(KA.GT.1.0D0)GO TO 5
C
C *** AVOID SINGULARITY IN F-DISTRIBUTION
C
      A=1.0D0
      B=10000.0D0
      CALL CNC7(FDIST,A,B,PREC,APRCX,IERR)
      TOTINT=1.0D0-APROX
      A=1.0D0
      B=2.0D0
    5 CALL CNC7(FDIST,A,B,PREC,PARINT,IERR)
      TOTINT=TOTINT+PARINT
      IF(TOTINT.GT.TINTMX)GO TO 10
C
C *** DETERMINE IF FURTHER INTEGRATION IS NEEDED
C
      ERR=DABS((B-A)/B)
C
C *** DETERMINE 2ND, 3RD,... INTERVALS OF INTEGRATION
C
      A=B
      FDISTA=FDIST(A)
      B=A+(TINTMX-TOTINT)/FDISTA
      IF(ERR.LE.0.005)GO TO 4
      GO TO 5
   10 TOTINT=TCTINT-PARINT
      B=B-0.20D0*(B-A)
      GO TO 5
    4 FDISTB=FDIST(B)
      X(I,K)=A+(TINTMX-TOTINT)/FDISTB
      T(JOT,I,K)=X(I,K)*DFLOAT(K)/(X(I,K)*DFLCAT(K)+DFLOAT(N-K))
    1 CONTINUE
    3 CONTINUE
C
C *** OUTPUT RESULTS
C
      DO 15 K=1,6
      WRITE(6,16) K
```

```
      INM=INMIN(K)
      WRITE(6,17) ALPH(10),(ALPH(I),I=1,9)
      WRITE(6,18) (NR(I),T(10,I,K),(T(JOT,I,K),JOT=1,9),I=INM,19)
   15 CONTINUE
C
C *** FORMATS
C
   16 FORMAT('1',51X,I1,' OUTLIER(S)',/,/,/,/,/)
   17 FORMAT(' ',3X,'NO. OF   *',42X,'ALPHA',/,'  DATA PTS. *',/,5X,
     1  '(N)      *',F8.2,9F9.2,/,2X,10('-'),'*',93('-'))
   18 FORMAT(' ',4X,19(I3,4X,'*',10(3X,F6.4),/,5X))
      WRITE(6,9)
    9 FORMAT('1')
      STOP
      END


C *** CALCULATE VALUE OF F-DISTRIBUTION
C
      REAL FUNCTION FDIST*8(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 NA,KA,NKA
      COMMON /CONST/ C,NA,KA,N,K
      IF(X.EQ.0.0.AND.KA.LT.1.)GO TO 2
      IF(X.EQ.0.0.AND.KA.EQ.1.)GO TO 3
      IF(X.EQ.0.0)GO TO 1
      A=(KA-1.)*DLOG(X)
      B=NA*DLOG(1.+X*DFLOAT(K)/DFLOAT(N-K))
      ABC=A-B+DLOG(C)
      IF(ABC.LE.-150.)GO TO 1
      IF(ABC.GE.172.577)GO TO 2
      FDIST=DEXP(ABC)
      RETURN
    1 FDIST=0.
      RETURN
    2 FDIST=8.9E 74
      RETURN
    3 FDIST=(NA-1.)*DFLOAT(K)/DFLOAT(N-K)
      RETURN
      END
```

```
C****
C****  ..............................................................
C****
C****  SUBROUTINE QNC7 -- DOUBLE PRECISION IBM 370 VERSION
C****
C****  PURPOSE
C****     QNC7 INTEGRATES REAL FUNCTIONS OF ONE VARIABLE OVER FINITE
C****     INTERVALS, USING AN ADAPTIVE 7-POINT NEWTON-COTES ALGORITHM.
C****     QNC7 IS EFFICIENT OVER A WIDE RANGE OF ACCURACIES, BUT QNC3
C****     MAY BE MORE EFFICIENT ON DIFFICULT LOW ACCURACY PROBLEMS,
C****     AND GAUS8 MAY BE MORE EFFICIENT ON HIGH ACCURACY PROBLEMS
C****     (ERR LESS THAN 1.D-09, SAT) OR ON PROBLEMS INVOLVING VERY
C****  SMOOTH FUNCTIONS.
C****
C****  USAGE
C****     CALL QNC7(FUN,A,B,ERR,ANS,IERR)
C****
C****     FUN  - NAME OF EXTERNAL FUNCTION TO BE INTEGRATED.  THIS NAME
C****            MUST BE IN AN EXTERNAL STATEMENT IN THE CALLING PROGRAM
C****            FUN MUST BE A FUNCTION OF ONE REAL ARGUMENT (THE
C****            VARIABLE OF INTEGRATION).
C****     A    - LOWER LIMIT OF INTEGRAL.
C****     B    - UPPER LIMIT OF INTEGRAL (MAY BE LESS THAN A).
C****     ERR  - USER-SUPPLIED ERROR PARAMETER.  ANS WILL NORMALLY HAVE
C****            NO MORE ERROR THAN ERR TIMES THE INTEGRAL OF THE
C****           ABSOLUTE VALUE OF FUN(X).
C****     ANS  - COMPUTED VALUE OF INTEGRAL.
C****     IERR - ERROR PARAMETER SET BY QNC7:
C****            IERR = 1 IS NORMAL.
C****            IERR = 2 MEANS ANS IS PROBABLY INSUFFICIENTLY ACCURATE.
C****
C****  SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C****     THE EXTERNAL FUNCTION FUN(X) MUST BE SUPPLIED BY THE USER.
C****
C****  METHOD
C****     AN ADAPTIVE 7-POINT NEWTON-COTES ALGOR2THM WITH INTERVAL
C****     BISECTION, COMBINED RELATIVE/ABSOLUTE ERROR CONTROL,
C****     ESTIMATION OF TOTAL QUADRATURE ERROR, AND COMPUTED MAXIMUM
C****     REFINEMENT LEVEL WHEN A IS CLOSE TO B.
C****
C****  REMARKS
C****     THIS ROUTINE HAS BEEN ADAPTED FOR DOUBLE PRECISION USE ON THE
C****     IBM 370/158 BY JEFFREY C. RYMAN, DEPT. OF NUCLEAR ENGINEERING,
C****     KANSAS STATE UNIVERSITY, MANHATTAN, KANSAS 66506.  THIS IS AN
C****     ADAPTATION OF THE SUBROUTINE QNC7 BY RANDALL E. JONES, FROM
C****     THE SANDIA MATHEMATICAL PROGRAM LIBRARY, MATHEMATICAL
C****     COMPUTING SERVICES DIVISION 5422, SANDIA LABORATORIES,
C****     P.O. BOX 5800, ALBUQUERQUE, N.M. 87115, CDC 6600 VERSION 4.7,
C****     18 JULY 1972.  A COPY OF THIS ROUTINE MAY BE FOUND ON PAGE 60
C****     OF:  ADAMS, K.G., BIGGS, F., AND RENKEN, J.H., 'DINT:  A
C****     COMPUTER PROGRAM WHICH PREPARES MULTIGROUP COHERENT-INCOHERENT
C****     CROSS SECTIONS FOR PHOTON TRANSPORT CALCULATIONS', AEC REPORT
```

```
C****     SC-RR-72 0684, DEC. 1972.  THE ORIGINAL ROUTINE WAS BY DAVID
C****     L. KAHANER, LASL.
C****
C****     IT SHOULD BE NOTED THAT THE DEFAULT AND MAXIMUM ALLOWABLE
C****     REFINEMENT LEVELS HAVE BEEN LEFT THE SAME AS IN THE CDC
C****     6600 VERSION.
C****
C**** ..............................................................
C****
      SUBROUTINE QNC7(FUN,/A/,/B/,/ERR/,/ANS/,/IERR/)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AA(40),HH(40),LR(40),VL(40),Q7R(40),F(13),F1(40),F2(40),
     1F3(40),F4(40),F5(40),F6(40),F7(40)
C**** MISCELLANEOUS PARAMETERS.
      DATA SQ2/1.41421356237309500/
      DATA LMN,NLMX,KMX,KML,NBITS/2,40,5000,7,48/
C**** INITIALIZE.
      ANS=0.D0
      IERR=1
      IF(A.EQ.B) RETURN
      W1=41.D0/140.D0
      W2=216.D0/140.D0
      W3=27.D0/140.D0
      W4=272.D0/140.D0
      LMX=NLMX
      IF(B.EQ.0.D0) GO TO 3
      IF(DSIGN(1.D0,B)*A.LE.0.D0) GO TO 3
      C=DABS(1.D0-A/B)
      IF(C.GT.0.1D0) GO TO 3
      NIB=-IDINT(DLOG(C)/DLOG(2.D0))
      LMX=MIN0(NLMX,NBITS-NIB-6)
      LMX=MAX0(LMX,LMN)
    3 TOL=DMAX1(ERR,2.D0**(5-NBITS))/2.D0
      IF(ERR.LT.0.D0) TOL=0.5D-6
      EPS=TOL
      HH(1)=(B-A)/12.D0
      AA(1)=A
      LR(1)=1
      DO 4 I=1,13,2
    4 F(I)=FUN(A+DFLOAT(I-1)*HH(1))
      K=7
      L=1
      AREA=0.D0
      Q7=0.D0
      EF=128.D0/255.D0
      CE=0.D0
C**** COMPUTE REFINED ESTIMATES, ESTIMATE THE ERROR, ETC.
    5 DO 6 I=2,12,2
    6 F(I)=FUN(AA(L)+DFLOAT(I-1)*HH(L))
      K=K+6
      Q7L=HH(L)*((W1*(F(1)+F(7))+W2*(F(2)+F(6)))+(W3*(F(3)+F(5))+W4*F(4
     1)))
      Q7R(L)=HH(L)*((W1*(F(7)+F(13))+W2*(F(8)+F(12)))+(W3*(F(9)+F(11))+W
     14*F(10)))
      AREA=AREA+(DABS(Q7L)+DABS(Q7R(L))-DABS(Q7))
      IF(L.LT.LMN) GO TO 11
      Q13=Q7L+Q7R(L)
      IF (DABS(Q7-Q13).EQ.0.0D0.OR.EF.EQ.0.0D0) GOTO 676
      IS=SGN(EF)
```

```
      TMP1=DLOG(DABS(Q7-Q13))+DLOG(CABS(EF))
      EE=DXP(TMP1)*IS
      GOTO 679
 678  EE=C.CDC
 679  IF (EPS.NE.0.0D0.AND.AREA.NE.0.0D0) GOTO 551
      TMP1=0.CD0
      GOTO 552
551   IS=SGN(AREA)*SGN(EPS)
      TMP1=DXP(DLOG(DABS(EPS))+DLOG(DABS(AREA)))*IS
552   IF (Q13.NE.0.0D0) GOTO 555
      TMP2=0.CD0
      GOTO 558
555   IS=SGN(TOL)
      TMP2=DXP(DLOG(DABS(TOL))+DLCG(DABS(Q13)))*IS
558   AE=DMAX1(TMP1,TMP2)
      IF(EE-AE) 8,8,10
    7 CE=CE+(C7-Q13)
      GO TC 9
 8    IF (Q7-Q13.EQ.0.0D0) GOTC 9
      IS1=SGN(C7-Q13)
      TMP=DXP(DLOG(DABS(Q7-Q13))-DLCG(255.0D0))
      CE=CE+IS1*TMP
    9 IF(LR(L)) 12,12,14
C**** CONSIDER THE LEFT HALF OF NEXT LEVEL.
   10 IF(K.GT.KMX) LMX=KML
      IF(L.GE.LMX) GC TO 7
   11 L=L+1
      EPS=EPS/2.D0
      IF(L.LE.15) EF=EF/SQ2
      HH(L)=HH(L-1)/2.D0
      LR(L)=-1
      AA(L)=AA(L-1)
      Q7=Q7L
      F1(L)=F(7)
      F2(L)=F(8)
      F3(L)=F(9)
      F4(L)=F(10)
      F5(L)=F(11)
      F6(L)=F(12)
      F7(L)=F(13)
      F(13)=F(7)
      F(11)=F(6)
      F(9)=F(5)
      F(7)=F(4)
      F(5)=F(3)
      F(3)=F(2)
      GO TO 5
C**** PROCEED TO RIGHT HALF AT THIS LEVEL.
   12 VL(L)=C13
   13 Q7=Q7R(L-1)
      LR(L)=1
      AA(L)=AA(L)+12.D0*HH(L)
      F(1)=F1(L)
      F(3)=F2(L)
      F(5)=F3(L)
      F(7)=F4(L)
      F(9)=F5(L)
      F(11)=F6(L)
      F(13)=F7(L)
```

```
      GO TO 5
C**** RETURN ONE LEVEL.
   14 VR=C13
   15 IF(L.LE.1) GO TO 18
      IF(L.LE.15) EF=EF*SQ2
      EPS=EPS*2.D0
      L=L-1
      IF(LR(L).GT.0) GO TO 17
      VL(L)=VL(L+1)+VR
      GO TO 13
   17 VR=VL(L+1)+VR
      GO TO 15
C**** EXIT.
   18 ANS=VR
      IF (TOL.EQ.0.0D0.OR.AREA.EQ.0.0D0) GOTO 789
      IS=SGN(TOL)*SGN(AREA)
      TMP1=DLOG(DABS(TOL))+DLOG(DABS(AREA))
      TMP1=DXP(TMP1)
      GOTO 790
 789  TMP1=0.0D0
 790  IF(DABS(CE).LE.2.D0*TMP1)  RETURN
      IERR=2
      RETURN
      END
C************************************************************************
C*    DXP FUNCTION
C************************************************************************
C*
C*  THIS FUNCTION ERROR CHECKS THEN EVALATES DEXP(X).
C*
C*
C*
      REAL FUNCTION DXP*8(X)
      REAL*8 DEXP,X,DABS
      IF (DABS(X).GT.170.0D0) GOTO 10
      DXP=DEXP(X)
      RETURN
   10 IF (X) 20,20,30
   20 DXP=0.0D0
      RETURN
   30 DXP=1.0D70
      RETURN
      END
C************************************************************************
C*
C*                              SGN
C*
C************************************************************************
C
C
C   THIS FUNCTION RETURNS A NEGATIVE 1.0D0 FOR NEGATIVE X, OR A POSITIVE
C   1.0D0 FOR VALUES OF X GREATER THAN OR EQUAL TO 0.0D0.
C
C
      REAL FUNCTION SGN*8 (X)
      REAL*8 X
      IF (X) 100,200,200
 100  SGN=-1.D0
      RETURN
 200  SGN=1.D0
      RETURN
      END
```

The following is a program for generating the critical values of Fisher's test statistic for lower outliers in an exponential distribution.

In the driving program;

N = number of data points in a data set

K = number of outliers in a data set

C = constant in the F-distribution

A = lower limit of integration

B = upper limit of integration

X(I,J) = values of x in Eq. B.2

T(I,J) = critical values of Fisher's test statistic

The subprogram 'FDIST' is for calculating values of the F-distribution.

```
      IMPLICIT REAL*8 (A-H,C-Z)
      REAL*8 INT(20,10),NA,KA,NKA
      DIMENSION T(20,10),NR(20)
      COMMON /CONST/ C,NA,KA,N,K
      EXTERNAL FDIST
C
C *** READ IN VALUES OF N (5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 25,
C *** 30, 35, 40, 50, 60, 80, 100)
C
      READ(5,8) (NR(I),I=1,19)
    8 FORMAT (19I3)
C
C *** SET THE PERCENT OF THE DISCORDANCY TEST
C
C *** USE 'DO 3 IPCT=1,2' FOR BOTH 95% AND 99% CONFIDENCE LEVELS
      IPCT=1
      IF(IPCT.EQ.1)PCT=0.05000
      IF(IPCT.EQ.2)PCT=0.01000
      ALPHA=1.000
      DO 1 J=1,6
      K=7-J
C
C *** ALLOW ONLY 33% OF DATA TO BE OUTLIERS
C
      IF(K.EQ.1)I1=1
      IF(K.EQ.2)I1=2
      IF(K.EQ.3)I1=5
      IF(K.EQ.4)I1=7
      IF(K.EQ.5)I1=8
      IF(K.EQ.6)I1=10
      IF(K.EQ.7)I1=11
      IF(K.GE.8)I1=12
      NOI=20-I1
      DO 1 M=1,NOI
      I=20-M
      N=NR(I)
C
C *** CALCULATE DESIRED PRECISION AND MAXIMUM VALUE OF INTEGRAL
C
      FN=DLGAMA(DFLOAT(N)+1.D0)
      FK=DLGAMA(DFLOAT(K)+1.D0)
      FNK=DLGAMA(DFLOAT(N-K)+1.D0)
      FAC=DEXP(FN-FK-FNK)
      PREC=0.001D00/FAC
      TINTMX=PCT/FAC
C
C *** CALCULATE CONSTANT IN F DISTRIBUTION
C
      NA=DFLOAT(N)*ALPHA
      KA=DFLOAT(K)*ALPHA
      NKA=DFLOAT(N-K)*ALPHA
      GNA=DLGAMA(NA)
      GKA=DLGAMA(KA)
      GNKA=DLGAMA(NKA)
      GAM=DEXP(GNA-GKA-GNKA)
```

```
      C=GAM*(DFLOAT(K)/DFLOAT(N-K))**KA
      TOTINT=0.0D0
C
C *** DETERMINE FIRST INTERVAL OF INTEGRATION
C
      A=0.0D0
      B=0.25D0
      IF(KA.GT.1.0D0)GO TO 5
      B=(TINTMX-TOTINT)/FDIST(A)
    5 CALL GNC7(FDIST,A,B,PREC,PARINT,IERR)
      IF(A.GT.B.AND.PARINT.GT.0.0D0)PARINT=0.0D0-PARINT
      TOTINT=TOTINT+PARINT
C
C *** DETERMINE IF FURTHER INTEGRATION IS NEEDED
C
      ERR=DABS((B-A)/B)
C
C *** DETERMINE 2ND, 3RD,... INTERVALS OF INTEGRATION
C
      A=B
      FDISTA=FDIST(A)
      B=A+(TINTMX-TOTINT)/FDISTA
      IF(ERR.LE.0.005D0)GO TO 4
      GO TO 5
    4 X=A+(TINTMX-TOTINT)/FDIST(A)
      IF(A.GT.B)X=A+(TINTMX-TOTINT)/FDIST(B)
      T(I,K)=X*DFLOAT(K)/(X*DFLOAT(K)+DFLOAT(N-K))
    1 CONTINUE
C
C *** OUTPUT
C
      WRITE(6,11) ALPHA
   11 FORMAT('1',51X,'ALPHA=',F5.2,/,/,/,/,/)
      WRITE(6,6) (I,I=1,6)
    6 FORMAT(' ',3X,'NO. OF',2X,'*',23X,'NUMBER OF OUTLIERS (K)',/,2X,
     1  'DATA PTS.',1X,'*',/,5X,'(N)',4X,'*',6X,6(I2,12X),/,2X,10('-'),
     2  '*',87('-'))
      WRITE(6,7) (NR(I),(T(I,J1),J1=1,6),I=1,19)
    7 FORMAT(' ',4X,19(I3,4X,'*',2X,6(F11.9,3X),/,5X))
    3 CONTINUE
      WRITE(6,9)
    9 FORMAT('1')
      STOP
      END
```

```
      REAL FUNCTION FDIST*8(X)
      IMPLICIT REAL*8 (A-H,C-Z)
      REAL*8 NA,KA,NKA
      COMMON /CONST/ C,NA,KA,N,K
      IF(X.EQ.0.0D0.AND.KA.LT.1.0D0)GO TO 2
      IF(X.EQ.0.0D0.AND.KA.EQ.1.0D0)GO TO 3
      IF(X.EQ.0.0D0)GO TO 1
      A=(KA-1.0D0)*DLOG(X)
      B=NA*DLOG(1.0D0+X*DFLOAT(K)/DFLOAT(N-K))
      ABC=A-B+DLOG(C)
      IF(ABC.LE.-150.)GO TO 1
      IF(ABC.GE.172.577)GO TO 2
      FDIST=DEXP(ABC)
      RETURN
    1 FDIST=0.0D0
      RETURN
    2 FDIST=8.9D 74
      RETURN
    3 FDIST=(NA-1.0D0)*DFLOAT(K)/DFLOAT(N-K)
      RETURN
      END
```

```
C
C
C****************************************************************************
C*     DXP FUNCTION
C****************************************************************************
C*
C*  THIS FUNCTION ERROR CHECKS THEN EVALATES DEXP(X).
C*
C*
C*
      REAL FUNCTION DXP*8(X)
      REAL*8 DEXP,X,DABS
      IF (DABS(X).GT.170.0D0) GOTO 10
      DXP=DEXP(X)
      RETURN
   10 IF (X) 20,20,30
   20 DXP=0.0D0
      RETURN
   30 DXP=1.0D70
      RETURN
      END
```

```
C
C
C
C
C****
C****  ..............................................................
C****
C****  SUBROUTINE QNC7 -- DOUBLE PRECISION IBM 370 VERSION
C****
C****  PURPOSE
C****      QNC7 INTEGRATES REAL FUNCTIONS OF ONE VARIABLE OVER FINITE
C****      INTERVALS, USING AN ADAPTIVE 7-POINT NEWTON-COTES ALGORITHM.
C****      QNC7 IS EFFICIENT OVER A WIDE RANGE OF ACCURACIES, BUT QNC3
C****      MAY BE MORE EFFICIENT ON DIFFICULT LOW ACCURACY PROBLEMS,
C****      AND GAUS8 MAY BE MORE EFFICIENT ON HIGH ACCURACY PROBLEMS
C****      (ERR LESS THAN 1.D-08, SAT) OR ON PROBLEMS INVOLVING VERY
C****  SMOOTH FUNCTIONS.
C****
C****  USAGE
C****      CALL QNC7(FUN,A,B,ERR,ANS,IERR)
C****
C****      FUN  - NAME OF EXTERNAL FUNCTION TO BE INTEGRATED.  THIS NAME
C****             MUST BE IN AN EXTERNAL STATEMENT IN THE CALLING PROGRAM.
C****             FUN MUST BE A FUNCTION OF ONE REAL ARGUMENT (THE
C****             VARIABLE OF INTEGRATION).
C****      A    - LOWER LIMIT OF INTEGRAL.
C****      B    - UPPER LIMIT OF INTEGRAL (MAY BE LESS THAN A).
C****      ERR  - USER-SUPPLIED ERROR PARAMETER.  ANS WILL NORMALLY HAVE
C****             NO MORE ERROR THAN ERR TIMES THE INTEGRAL OF THE
C****            ABSOLUTE VALUE OF FUN(X).
C****      ANS  - COMPUTED VALUE OF INTEGRAL.
C****      IERR - ERROR PARAMETER SET BY QNC7:
C****             IERR = 1 IS NORMAL.
C****             IERR = 2 MEANS ANS IS PROBABLY INSUFFICIENTLY ACCURATE.
C****
C****  SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C****      THE EXTERNAL FUNCTION FUN(X) MUST BE SUPPLIED BY THE USER.
C****
C****  METHOD
C****      AN ADAPTIVE 7-POINT NEWTON-COTES ALGOR2THM WITH INTERVAL
C****      BISECTION, COMBINED RELATIVE/ABSOLUTE ERROR CONTROL,
C****      ESTIMATION OF TOTAL QUADRATURE ERROR, AND COMPUTED MAXIMUM
C****      REFINEMENT LEVEL WHEN A IS CLOSE TO B.
C****
C****  REMARKS
C****      THIS ROUTINE HAS BEEN ADAPTED FOR DOUBLE PRECISION USE ON THE
C****      IBM 370/158 BY JEFFREY C. RYMAN, DEPT. OF NUCLEAR ENGINEERING,
C****      KANSAS STATE UNIVERSITY, MANHATTAN, KANSAS 66506.  THIS IS AN
C****      ADAPTATION OF THE SUBROUTINE QNC7 BY RANDALL E. JONES, FROM
C****      THE SANDIA MATHEMATICAL PROGRAM LIBRARY, MATHEMATICAL
C****      COMPUTING SERVICES DIVISION 5422, SANDIA LABORATORIES,
C****      P.O. BOX 5800, ALBUQUERQUE, N.M. 87115, CDC 6600 VERSION 4.7,
C****      18 JULY 1972.  A COPY OF THIS ROUTINE MAY BE FOUND ON PAGE 60
C****      OF:  ADAMS, K.G., BIGGS, F., AND RENKEN, J.H., 'QINT:  A
C****      COMPUTER PROGRAM WHICH PREPARES MULTIGROUP COHERENT-INCOHERENT
C****      CROSS SECTIONS FOR PHOTON TRANSPORT CALCULATIONS', AEC REPORT
```

```
C****     SC-RR-72 0684, DEC. 1972.  THE ORIGINAL ROUTINE WAS BY DAVID
C****     L. KAHANER, LASL.
C****
C****     IT SHOULD BE NOTED THAT THE DEFAULT AND MAXIMUM ALLOWABLE
C****     REFINEMENT LEVELS HAVE BEEN LEFT THE SAME AS IN THE CDC
C****     6600 VERSION.
C****
C**** ..............................................................
C****
      SUBROUTINE QNC7(FUN,/A/,/B/,/ERR/,/ANS/,/IERR/)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AA(40),HH(40),LR(40),VL(40),Q7R(40),F(13),F1(40),F2(40),
     1F3(40),F4(40),F5(40),F6(40),F7(40)
C**** MISCELLANEOUS PARAMETERS.
      DATA SQ2/1.4142135623730950D/
      DATA LMN,NLMX,KMX,KML,NBITS/2,40,5000,7,48/
C**** INITIALIZE.
      ANS=0.D0
      IERR=1
      IF(A.EQ.B) RETURN
      W1=41.D0/140.D0
      W2=216.D0/140.D0
      W3=27.D0/140.D0
      W4=272.D0/140.D0
      LMX=NLMX
      IF(B.EQ.0.D0) GO TO 3
      IF(DSIGN(1.D0,B)*A.LE.0.D0) GO TO 3
      C=DABS(1.D0-A/B)
      IF(C.GT.0.1D0) GO TO 3
      NIB=-IDINT(DLOG(C)/DLOG(2.D0))
      LMX=MIN0(NLMX,NBITS-NIB-6)
      LMX=MAX0(LMX,LMN)
    3 TOL=DMAX1(ERR,2.D0**(5-NBITS))/2.D0
      IF(ERR.LT.0.D0) TOL=0.5D-6
      EPS=TOL
      HH(1)=(B-A)/12.D0
      AA(1)=A
      LR(1)=1
      DO 4 I=1,13,2
    4 F(I)=FUN(A+DFLOAT(I-1)*HH(1))
      K=7
      L=1
      AREA=0.D0
      Q7=0.D0
      EF=128.D0/255.D0
      CE=0.D0
C**** COMPUTE REFINED ESTIMATES, ESTIMATE THE ERROR, ETC.
    5 DO 6 I=2,12,2
    6 F(I)=FUN(AA(L)+DFLOAT(I-1)*HH(L))
      K=K+6
      Q7L=HH(L)*((W1*(F(1)+F(7))+W2*(F(2)+F(6)))+(W3*(F(3)+F(5))+W4*F(4)
     1))
      Q7R(L)=HH(L)*((W1*(F(7)+F(13))+W2*(F(8)+F(12)))+(W3*(F(9)+F(11))+W
     14*F(10)))
      AREA=AREA+(DABS(Q7L)+DABS(Q7R(L))-DABS(Q7))
      IF(L.LT.LMN) GO TO 11
      Q13=Q7L+Q7R(L)
      IF (DABS(Q7-Q13).EQ.0.0D0.OR.EF.EQ.0.0D0) GOTO 678
      IS=SGN(EF)
```

```
      TMP1=DLOG(DABS(Q7-Q13))+DLOG(DABS(EF))
      EE=DXP(TMP1)*IS
      GOTO 679
 678  EE=0.0D0
 679  IF (EPS.NE.0.0D0.AND.AREA.NE.0.0D0) GOTO 551
      TMP1=0.0D0
      GOTO 552
551   IS=SGN(AREA)*SGN(EPS)
      TMP1=DXP(DLOG(DABS(EPS))+DLOG(DABS(AREA)))*IS
552   IF (Q13.NE.0.0D0) GOTO 555
      TMP2=0.0D0
      GOTO 558
555   IS=SGN(TOL)
      TMP2=DXP(DLOG(DABS(TOL))+DLOG(DABS(Q13)))*IS
558   AE=DMAX1(TMP1,TMP2)
      IF(EE-AE) 8,8,10
    7 CE=CE+(Q7-Q13)
      GO TO 9
 8    IF (Q7-Q13.EQ.0.0D0) GOTO 9
      IS1=SGN(Q7-Q13)
      TMP=DXP(DLOG(DABS(Q7-Q13))-DLOG(255.0D0))
      CE=CE+IS1*TMP
    9 IF(LR(L)) 12,12,14
C**** CONSIDER THE LEFT HALF OF NEXT LEVEL.
   10 IF(K.GT.KMX) LMX=KML
      IF(L.GE.LMX) GO TO 7
   11 L=L+1
      EPS=EPS/2.D0
      IF(L.LE.15) EF=EF/SQ2
      HH(L)=HH(L-1)/2.D0
      LR(L)=-1
      AA(L)=AA(L-1)
      Q7=Q7L
      F1(L)=F(7)
      F2(L)=F(8)
      F3(L)=F(9)
      F4(L)=F(10)
      F5(L)=F(11)
      F6(L)=F(12)
      F7(L)=F(13)
      F(13)=F(7)
      F(11)=F(6)
      F(9)=F(5)
      F(7)=F(4)
      F(5)=F(3)
      F(3)=F(2)
      GO TO 5
C**** PROCEED TO RIGHT HALF AT THIS LEVEL.
   12 VL(L)=Q13
   13 Q7=Q7R(L-1)
      LR(L)=1
      AA(L)=AA(L)+12.D0*HH(L)
      F(1)=F1(L)
      F(3)=F2(L)
      F(5)=F3(L)
      F(7)=F4(L)
      F(9)=F5(L)
      F(11)=F6(L)
      F(13)=F7(L)
```

```
      GO TO 5
C**** RETURN ONE LEVEL.
   14 VR=Q13
   15 IF(L.LE.1) GO TO 18
      IF(L.LE.15) EF=EF*SQ2
      EPS=EPS*2.D0
      L=L-1
      IF(LR(L).GT.0) GO TO 17
      VL(L)=VL(L+1)+VR
      GO TO 13
   17 VR=VL(L+1)+VR
      GO TO 15
C**** EXIT.
   18 ANS=VR
      IF (TOL.EQ.0.0D0.OR.AREA.EQ.0.0D0) GOTO 789
      IS=SGN(TOL)*SGN(AREA)
      TMP1=DLOG(DABS(TOL))+DLOG(DABS(AREA))
      TMP1=DXP(TMP1)
      GOTO 790
 789  TMP1=0.0D0
 790  IF(DABS(CE).LE.2.D0*TMP1)  RETURN
      IERR=2
      RETURN
      END



C
C
C
C***************************************************************************
C*
C*                              SGN
C*
C***************************************************************************
C
C
C   THIS FUNCTION RETURNS A NEGATIVE 1.0D0 FOR NEGATIVE X, OR A POSITIVE
C   1.0D0 FOR VALUES OF X GREATER THAN OR EQUAL TO 0.0D0.
C
C
      REAL FUNCTION SGN*8 (X)
      REAL*8 X
      IF (X) 100,200,200
 100  SGN=-1.0D0
      RETURN
 200  SGN=1.0D0
      RETURN
      END
```

Appendix C: Data Generation

The following three programs are for generating random data. The theory behind them is as follows.

Let y be distributed according to the uniform distribution, $f(y) = I_{(0,1)}(y)$*, and let x be distributed according to a known function, say $g(x|\theta)$. Then a point y_i corresponds to a point x_i in the following manner:

$$\int_0^{y_i} f(y)dy = \int_0^{x_i} g(x|\theta)dx, \tag{C.1}$$

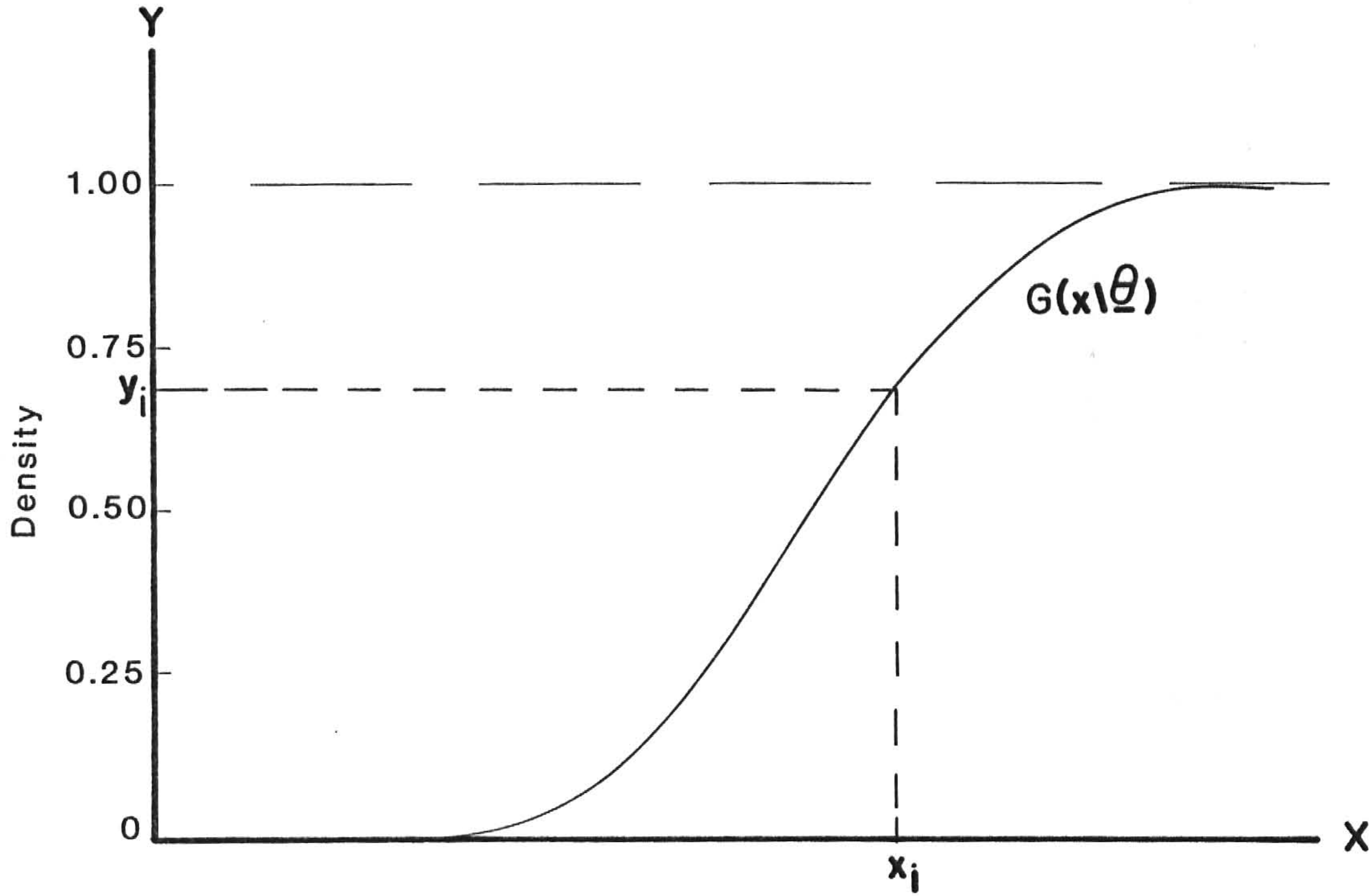
$$F(y_i) = y_i = G(x_i|\theta), \tag{C.2}$$

$$x_i = G^{-1}(y_i|\theta), \tag{C.3}$$

where G^{-1} (...) is the inverse of the cumulative distribution of x (See Fig. C.1).

---

*That is $I_{(0,1)}(y)$ is unity from 0 to 1 and zero at all other points. The point y is generated by a random generator program, such as RANDU.



<u>Figure C.1</u>: Graphical illustration of the generation of random data which is distributed according to the function g(...).

The following is a program for generating data which is distributed according to the gamma distribution.

In the driving program:

NDATA = number of data points per data set

NSETS = number of data sets generated

A = alpha parameter

B = beta parameter

DATA(J,I) = random data

The subroutine 'RANDU' is for generating random numbers between 0 and 1, and comes from IBM Manual C20-8011.

```
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMCN /PRIR/ PARM(4),RNG(2),IPRIR
      COMMON /RTPASS/ XLI,XRI,EPS,IEND,IERR
      DIMENSION DATA(1000,50)
      XLI=0.0D0
      XRI=1.D10
      EPS=1.D-4
      IEND=250
      IPRIR=1
      PARM(3)=0.0D0
      PARM(4)=0.0D0
      NDATA=20
      NSETS=250
      IX=3742968
      DO 1 I=1,43
      CALL RANDU(IX,IY,YFL)
    1 IX=IY
      DO 2 IA=1,5
      IF(IA.EQ.1)A=1.0D0
      IF(IA.EQ.2)A=1.25D0
      IF(IA.EQ.3)A=1.5D0
      IF(IA.EQ.4)A=2.0D0
      IF(IA.EQ.5)A=4.0D0
      PARM(1)=A
      DO 2 IB=1,5
      IF(IB.EQ.1)B=A/1.D-2
      IF(IB.EQ.2)B=A/1.D-4
      IF(IB.EQ.3)B=A/1.D-5
      IF(IB.EQ.4)B=A/1.D-6
      IF(IB.EQ.5)B=A/1.D-7
      PARM(2)=B
      DO 3 L=1,NSETS
      DO 3 I=1,NDATA
      CALL RANDU(IX,IY,YFL)
      IX=IY
    3 DATA(L,I)=PRCNT(YFL)
      WRITE(9,4)((DATA(L,I),I=1,NDATA),L=1,NSETS)
    4 FORMAT(100(100D12.4/))
    2 CCNTINUE
      STOP
      END
C
C
C
      SUBROUTINE RANDU(IX,IY,YFL)
      REAL*8 YFL
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
```

```
      REAL FUNCTION PRCNT*8 (X)
C**************************************************************************
C*                              PRCNT                                    *
C*                                                                       *
C**************************************************************************
C*   THIS FUNCTION RETURNS THE CORRESPONDING LAMBDA VALUE AT THE         *
C*   X*(100) PERCENTILE FOR THE PRIOR DISTRIBUTION. THE FUNCTION USES    *
C*   THE ORTMI ROUTINE TO FIND THE ROOT OF F(LAMBDA)=CUM(LAMBDA)-X=0.    *
C*   /SUBPRM/ COMMON BLOCK IS USED TO PASS PARAMETERS TO THE FUNCTION    *
C*   PRFCT. PARAMETERS IN COMMON BLOCK /RTPASS/ ARE SET BY THE CALLING   *
C*   ROUTINE AND ARE EXPLAINED IN THE ORTMI DOCUMENTATION.               *
C*                                                                       *
C*                                                                       *
C*   WRITTEN BY: S. D. HANSEN          5/80.                             *
C*                                                                       *
C**************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/RTPASS/  XLI,XRI,EPS,IEND,IERR
      COMMON/SUBPRM/ ALPHA
      EXTERNAL PRFCT
      ALPHA=X
      CALL DRTMI (ANS,F,PRFCT,XLI,XRI,EPS,IEND,IERR)
      IF (IERR.NE.0) WRITE(9,100) IERR
 100  FORMAT(' ERROR IN PRCNT ROUTINE, IERR= ',I4)
      PRCNT=ANS
      RETURN
      END
C*
C*
C*
C*
      REAL FUNCTION PRFCT*8(P)
C**************************************************************************
C*                                                                       *
C*   THIS FUNCTION EVALUATES F(LAMBDA)=CUM(LAMBDA)-ALPHA FOR THE         *
C*   PRIOR DISTRIBUTION.                                                 *
C*                                                                       *
C**************************************************************************
      REAL*8 PRCUM,ALPHA,P
      COMMON/SUBPRM/ ALPHA
      PRFCT=PRCUM(P)-ALPHA
C&&   WRITE (9,*) P,PRFCT
      RETURN
      END
```

```
      REAL FUNCTION PRIOR*8(X)
C***********************************************************************
C*                                                                     *
C* THIS SUBROUTINE EVALUATES THE PRIOR DENSITY FUNCTION FOR THE FAIL-  *
C* URE RATE CASES. DISTRIBUTION PARAMETER VALUES AND FUNCTION DEFINING *
C* INDEX ARE PASSED TO SUBROUTINE THROUGH THE COMMON BLOCK /PRIR/      *
C*                                                                     *
C* SUBROUTINES NEEDED: DXP                                             *
C*                                                                     *
C* SUBROUTINE ARGUMENTS:                                               *
C*    X      = FAILURE RATE VALUE AT WHICH DENSITY FUNCTION IS DESIRED *
C*                                                                     *
C* COMMON BLOCK /PRIR/ VARIABLES:                                      *
C*    PARM  = DISTRIBUTION PARAMETER VECTORS: PARM(1)=ALPHA, PARM(2)=  *
C*            BETA (=TAU FOR GAMMA), PARM(3)=MINIMUM LAMBDA OR SHIFT   *
C*            PARAMETER, PARM(4)=MAXIMUM LAMBDA PARAMETER (USED ONLY   *
C*            FOR THE BETA DISTRIBUTION)                               *
C*    IPRIR = DISTRIBUTION INDEX; =1 GAMMA, =2 LOGNORMAL, =3 WEIBULL,  *
C*            =4 LOGBETA                                               *
C*                                                                     *
C* WRITTEN BY J.K. SHULTIS 3/79.  MODIFIED 6/79, 7/79.                 *
C*                                                                     *
C***********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/PRIR/PARM(4),RNG(2),IPRIR
      REAL*8 X,DLGAMA,DEXP,DLOG,DABS,DXP
      PRIOR=0.0D0
      IF(X.LE.PARM(3)) RETURN
C*****
C***** GAMMA DENSITY FUNCTION:  (LAMBDA-THETA)=GAMMA(ALPHA,TAU)
C***** PARAMETERS: ALPHA>0,  TAU>0,  THETA>0;   THETA<LAMBDA<INFINITY
      GO TO (10,20,30,40),IPRIR
   10 ARG=X-PARM(3)
      AA=PARM(1)
      BB=(AA-1.0D0)*DLOG(ARG)+AA*DLOG(PARM(2))-PARM(2)*ARG
     1-DLGAMA(AA)
      PRIOR=DXP(BB)
      RETURN
C*****
C***** LOGNORMAL DENSITY FUNCTION: LOG(LAMBDA-THETA)=NORMAL(ALPHA,BETA)
C***** PARAMETERS: -INF<ALPHA<INF, BETA>0, THETA>0;  LAMBDA>0
   20 ARG=DLOG(X-PARM(3))
      B2=PARM(2)**2
      BB=-ARG-0.5D0*DLOG(6.2831853071795 9*B2)-0.5D0*((ARG-PARM(1))**2)/B
     12
      PRIOR=DXP(BB)
      RETURN
C*****
C***** WEIBULL DENSITY FUNCTION: (LAMBDA-THETA)=WEIBULL(ALPHA,BETA)
C***** PARAMETERS: ALPHA,BETA,THETA>0;  LAMBDA>THETA>0
   30 BB=PARM(2)
      ARG=(X-PARM(3))/PARM(1)
      AA=DLOG(BB)-DLOG(PARM(1))+(BB-1.0D0)*DLOG(ARG)-ARG**BB
      PRIOR=DXP(AA)
      RETURN
```

```
C*****
C***** FOUR PARAMETER LOGBETA:  LOG(LAMBDA)=BETA(ALPHA,BETA,A,B)
C***** PARAMETERS: ALPHA,BETA>0;  -INF<A<LOG(LAMBDA)<B<INF
   40 A=PARM(3)
      B=PARM(4)
      DL=DLOG(X)
      IF((DL.GE.B).OR.(DL.LE.A)) RETURN
      AA=PARM(1)
      BB=PARM(2)
      ARG=(DLGAMA(AA+BB)-DLGAMA(BB)-DLGAMA(AA)) - DL +
     1(-(AA+BB-1.0D0)*DLOG(B-A) + (AA-1.0D0)*DLOG(DL-A)+(BB-1.D0)*DLOG(B
     2-DL))
      PRIOR=DXP(ARG)
      RETURN
      END
```

```
C        SUBROUTINE DRTMI
C
C        PURPOSE
C           TO SOLVE GENERAL NONLINEAR EQUATIONS OF THE FORM FCT(X)=0
C           BY MEANS OF MUELLER-S ITERATION METHOD.
C
C        USAGE
C           CALL DRTMI (X,F,FCT,XLI,XRI,EPS,IEND,IER)
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT.
C
C        DESCRIPTION OF PARAMETERS
C           X      - RESULTANT ROOT OF EQUATION FCT(X)=0.
C           F      - RESULTANT FUNCTION VALUE AT ROOT X.
C           FCT    - NAME OF THE EXTERNAL FUNCTION SUBPROGRAM USED.
C           XLI    - INPUT VALUE WHICH SPECIFIES THE INITIAL LEFT BOUND
C                    OF THE ROOT X.
C           XRI    - INPUT VALUE WHICH SPECIFIES THE INITIAL RIGHT BOUND
C                    OF THE ROOT X.
C           EPS    - INPUT VALUE WHICH SPECIFIES THE UPPER BOUND OF THE
C                    ERROR OF RESULT X.
C           IEND   - MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED.
C           IER    - RESULTANT ERROR PARAMETER CODED AS FOLLOWS
C                     IER=0 - NO ERROR,
C                     IER=1 - NO CONVERGENCE AFTER IEND ITERATION STEPS
C                             FOLLOWED BY IEND SUCCESSIVE STEPS OF
C                             BISECTION,
C                     IER=2 - BASIC ASSUMPTION FCT(XLI)*FCT(XRI) LESS
C                             THAN OR EQUAL TO ZERO IS NOT SATISFIED.
C
C        REMARKS
C           THE PROCEDURE ASSUMES THAT FUNCTION VALUES AT INITIAL
C           BOUNDS XLI AND XRI HAVE NOT THE SAME SIGN. IF THIS BASIC
C           ASSUMPTION IS NOT SATISFIED BY INPUT VALUES XLI AND XRI, THE
C           PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER=2.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           THE EXTERNAL FUNCTION SUBPROGRAM FCT(X) MUST BE FURNISHED
C           BY THE USER.
C
```

```
C        METHOD
C           SOLUTION OF EQUATION FCT(X)=0 IS DONE BY MEANS OF MUELLER-S
C           ITERATION METHOD OF SUCCESSIVE BISECTIONS AND INVERSE
C           PARABOLIC INTERPOLATION, WHICH STARTS AT THE INITIAL BOUNDS
C           XLI AND XRI. CONVERGENCE IS QUADRATIC IF THE DERIVATIVE OF
C           FCT(X) AT ROOT X IS NOT EQUAL TO ZERO. ONE ITERATION STEP
C           REQUIRES TWO EVALUATIONS OF FCT(X). FOR TEST ON SATISFACTORY
C           ACCURACY SEE FORMULAE (3,4) OF MATHEMATICAL DESCRIPTION.
C           FOR REFERENCE, SEE G. K. KRISTIANSEN, ZERO OF ARBITRARY
C           FUNCTION, BIT, VOL. 3 (1963), PP.205-206.
C
C     ..................................................................
C
      SUBROUTINE DRTMI(X,F,FCT,XLI,XRI,EPS,IEND,IER)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C
C     PREPARE ITERATION
      IER=0
      XL=XLI
      XR=XRI
      X=XL
      TOL=X
      F=FCT(TOL)
      IF(F)1,16,1
    1 FL=F
      X=XR
      TOL=X
      F=FCT(TOL)
      IF(F)2,16,2
    2 FR=F
      IF(DSIGN(1.0D0,FL)+DSIGN(1.0D0,FR))25,3,25
C
C     BASIC ASSUMPTION FL*FR LESS THAN 0 IS SATISFIED.
C     GENERATE TOLERANCE FOR FUNCTION VALUES.
    3 I=0
      TOLF=100.*EPS
C
C
C     START ITERATION LOOP
    4 I=I+1
C
C     START BISECTION LOOP
      DO 13 K=1,IEND
      X=.5*(XL+XR)
      TOL=X
      F=FCT(TOL)
      IF(F)5,16,5
    5 IF(DSIGN(1.0D0,F)+DSIGN(1.0D0,FR))7,6,7
C
C     INTERCHANGE XL AND XR IN ORDER TO GET THE SAME SIGN IN F AND FR
    6 TOL=XL
      XL=XR
      XR=TOL
      TOL=FL
      FL=FR
```

```
      FR=TOL
    7 TOL=F-FL
      A=F*TOL
      A=A+A
      IF(A-FR*(FR-FL))8,9,9
    8 IF(I-IEND)17,17,9
    9 XR=X
      FR=F
C
C     TEST ON SATISFACTORY ACCURACY IN BISECTION LOOP
      TOL=EPS
      A=DABS(XR)
      IF(A-1.)11,11,10
   10 TOL=TOL*A
   11 IF(DABS(XR-XL)-TOL)12,12,13
   12 IF(DABS(FR-FL)-TOLF)14,14,13
   13 CONTINUE
C     END OF BISECTION LOOP
C
C     NO CONVERGENCE AFTER IEND ITERATION STEPS FOLLOWED BY IEND
C     SUCCESSIVE STEPS OF BISECTION OR STEADILY INCREASING FUNCTION
C     VALUES AT RIGHT BOUNDS. ERROR RETURN.
      IER=1
   14 IF(DABS(FR)-DABS(FL))16,16,15
   15 X=XL
      F=FL
   16 RETURN
C
C     COMPUTATION OF ITERATED X-VALUE BY INVERSE PARABOLIC INTERPOLATION
   17 A=FR-F
      DX=(X-XL)*FL*(1.+F*(A-TOL)/(A*(FR-FL)))/TOL
      XM=X
      FM=F
      X=XL-DX
      TOL=X
      F=FCT(TOL)
      IF(F)18,16,18
C
C     TEST ON SATISFACTORY ACCURACY IN ITERATION LOOP
   18 TOL=EPS
      A=DABS(X)
      IF(A-1.)20,20,19
   19 TOL=TOL*A
   20 IF(DABS(DX)-TOL)21,21,22
   21 IF(DABS(F)-TOLF)16,16,22
C
C     PREPARATION OF NEXT BISECTION LOOP
   22 IF(DSIGN(1.D00,F)+DSIGN(1.D00,FL))24,23,24
   23 XR=X
      FR=F
      GO TO 4
   24 XL=X
      FL=F
      XR=XM
      FR=FM
      GO TO 4
C     END OF ITERATION LOOP
```

```
C       ERROR RETURN IN CASE OF WRONG INPUT DATA
   25 IER=2
      RETURN
      END



      REAL FUNCTION PRCUM*8(X)
C***********************************************************************
C*                                                                     *
C* THIS SUBROUTINE EVALUATES THE PRIOR CUMULATIVE FUNCTION FOR THE     *
C* FAILURE RATE CASES. DISTRIBUTION PARAMETER VALUES AND FUNCTION DE-  *
C* FINITION INDEX ARE PASSED TO SUBROUTINE THROUGH THE COMMON BLOCK    *
C* /PRIR/.                                                             *
C*                                                                     *
C*   SUBROUTINES NEEDED:   GAMIC, MDBETA                               *
C*                                                                     *
C* SUBROUTINE ARGUMENTS:                                               *
C*    X       = FAILURE RATE VALUE AT WHICH DENSITY FUNCTION IS DESIRED *
C*                                                                     *
C* COMMON BLOCK /PRIR/ VARIABLES:                                      *
C*    PARM  = DISTRIBUTION PARAMETER VECTORS: PARM(1)=ALPHA, PARM(2)=  *
C*            BETA (=TAU FOR GAMMA), PARM(3)=MINIMUM LAMBDA PARAMETER  *
C*            OR SHIFT PARAMETER, PARM(4)=MAXIMUM LAMBDA PARAMETER     *
C*            (USED ONLY FOR THE BETA DISTRIBUTION)                    *
C*    IPRIR = DISTRIBUTION INDEX; =1 GAMMA, =2 LOGNORMAL, =3 WEIBULL,  *
C*            =4 LOGBETA                                               *
C*                                                                     *
C* WRITTEN BY J.K. SHULTIS, 3/79.  MODIFIED 6/79, 7/79.                *
C*                                                                     *
C***********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 XXX,AAA,BBB,PROB
      COMMON/PRIR/PARM(4),RNG(2),IPRIR
      REAL*8 X, YY(1), DLGAMA, DEXP, DLOG, DABS
      PRCUM=0.0D0
      GO TO (10,20,30,40),IPRIR
C*****
C***** GAMMA CUMULATIVE DISTRIBUTION
   10 IF(X.LE.PARM(3)) RETURN
      ARG=X-PARM(3)
      ALPHA=PARM(1)
      TAU=PARM(2)*ARG
      CALL GAMIC(TAU,ALPHA,1.D-7,1,YY,NZ)
      IF (NZ.NE.0) WRITE(9,11)
   11 FORMAT(' PROBABLE INACCURATE RESULT FOR THE GAMMA CUMULATIVE DISTR
     1IBUTION')
      PRCUM=YY(1)
      RETURN
C*****
C***** CUMULATIVE OF THE LOGNORMAL DISTRIBUTION
   20 IF(X.LE.PARM(3)) RETURN
      XL=(-DLOG(X-PARM(3))+PARM(1))/PARM(2)
      XLA=DABS(XL)/1.414213562373095
      A=DERF(XLA)
      PRCUM=0.5D0*(1.0D0-DSIGN(A,XL))
      RETURN
```

```
C*****
C***** WEIBULL CUMULATIVE DISTRIBUTICN
   30 IF(X.LE.PARM(3)) RETURN
      PRCUM=1.0D0
      ARG=((X-PARM(3))/PARM(1))**PARM(2)
      IF(ARG.GT.36.0D0) RETURN
      PRCUM=1.0D0-DEXP(-ARG)
      RETURN
C*****
C***** CUMULATIVE OF THE LOGBETA DISTRIBUTION
   40 B=PARM(4)
      A=PARM(3)
      DL=DLOG(X)
      IF((B-DL)*(DL-A)) 41,41,42
   42 XXX=(DL-A)/(B-A)
      AAA=PARM(1)
      BBB=PARM(2)
      CALL MDBETA(XXX,AAA,BBB,PROB,IER)
      PRCUM=PRCB
      RETURN
   41 IF(DL.LE.A) RETURN
      IF(DL.GE.B) PRCUM=1.D0
      RETURN
      END
      SUBROUTINE MDBETA(X, P, C, PROB, IER)
C
C***************************************************************************
C*
C*  FUNCTION:       EVALUATE THE INCOMPLETE BETA DISTRIBUTION FUNCTION
C*
C*  PARAMETERS:
C*  X     - VALUE TO WHICH FUNCTION IS TO BE INTERGRATED. X MUST BE IN TH
C*          RANGE (0,1) INCLUSIVE.
C*  P     - INPUT (1ST) PARAMETER (MUST BE GREATER THAN 0)
C*  Q     - INPUT (2ND) PARAMETER (MUST BE GREATER THAN 0)
C*  PROB  - OUTPUT PROBABILITY THAT A RANDCM VAPIABLE FROM A BETA DISTRIE
C*            HAVING PARAMETERS P AND Q WILL BE LESS THAN OR EQUAL TC 0.
C*  IER   - ERROR PARAMETER.
C*            IER = 0 INDICATES A NORMAL EXIT
C*            IER = 1 INDICATES THAT X IS NOT IN THE RANGE (0,1) INCLUSIV
C*            IER = 2 INDICATES THAT P AND/OR Q IS LESS THAN OR EQUAL TO
C*
C* CCDE BASED ON SIMILAR CODE BY N. BOSTEN AND E.BATTISTE AS MODIFIED BY
C* M. PIKE AND J. HOO.
C*
C***************************************************************************
      DOUBLE PRECISION PS, PX, Y, P1, DP, INFSUM, CNT, WH, XB,
     * DQ, C, EPS, EPS1, ALEPS, FINSUM, PQ, DA, DLGAMA
C DOUBLE PRECISION FUNCTION DLGAMA
C MACHINE PRECISION
      DATA EPS/1.D-6/
C SMALLEST POSITIVE NUMBER REPRESENTABLE
      DATA EPS1/1.D-78/
C NATURAL LOG OF EPS1
      DATA ALEPS/-179.6016D0/
C CHECK RANGES OF THE ARGUMENTS
      Y = X
      IF ((X.LE.1.0) .AND. (X.GE.0.0)) GC TO 10
```

```
      IER = 1
      GO TO 140
   10 IF ((P.GT.0.0) .AND. (Q.GT.0.0)) GO TO 20
      IER = 2
      GO TO 140
   20 IER =0
      IF (X.GT.0.5) GO TO 30
      INT = 0
      GO TO 40
C SWITCH ARGUMENTS FOR MORE EFFICIENT USE OF THE POWER
C SERIES
   30 INT = 1
      TEMP = P
      P = Q
      Q = TEMP
      Y = 1.D0 - Y
   40 IF (X.NE.0. .AND. X.NE.1.) GO TO 60
C SPECIAL CASE - X IS 0. OR 1.
   50 PROB = 0.
      GO TO 130
   60 IB = Q
      TEMP = IB
      PS = Q - FLOAT(IB)
      IF (Q.EQ.TEMP) PS = 1.D0
      DP = P
      DQ = Q
      PX = DP*DLOG(Y)
      PQ = DLGAMA(DP+DQ)
      P1 = DLGAMA(DP)
      C = DLGAMA(DQ)
      D4 = DLOG(DP)
C DLGAMA IS A FUNCTION WHICH CALCULATES THE DOUBLE
C PRECISION LOG GAMMA FUNCTION
      XB = PX + DLGAMA(PS+DP) - DLGAMA(PS) - D4 - P1
C SCALING
      IB = XB/ALEPS
      INFSUM = 0.D0
C FIRST TERM OF A DECREASING SERIES WILL UNDERFLOW
      IF (IB.NE.0) GO TO 90
      INFSUM = DEXP(XB)
      CNT = INFSUM*DP
C CNT WILL EQUAL DEXP(TEMP)*(1.D0-PS)I*P*Y**I/FACTORIAL(I)
      WH = 0.0D0
   80 WH = WH + 1.D0
      CNT = CNT*(WH-PS)*Y/WH
      IF (CNT.LT.EPS1/EPS) GOTO 90
      XB = CNT/(DP+WH)
      INFSUM = INFSUM + XB
      IF (XB/EPS.GT.INFSUM) GO TO 80
C DLGAMA IS A FUNCTION WHICH CALCULATES THE DOUBLE
C PRECISION LOG GAMMA FUNCTION
   90 FINSUM = 0.D0
      IF (DQ.LE.1.D0) GO TO 120
```

```
      XB = PX+ DQ*DLOG(1.D0-Y) + PQ - P1 - DLOG(DQ) - C
C SCALING
      IB = XB/ALEPS
      IF (IB.LT.0) IB = 0
      C = 1.D0/(1.D0-Y)
      CNT = DEXP(XB-FLOAT(IB)*ALEPS)
      PS = DQ
      WH = DQ
  100 WH =WH -1.D0
      IF (WH.LE.0.0D0) GO TO 120
      PX = (PS*C)/ (DP+WH)
      IF (PX.GT.1.0D0) GO TO 105
      IF (CNT/EPS.LE.FINSUM.OR.CNT.LE.EPS1/PX) GO TO 120
  105 CNT =CNT*PX
      IF (CNT.LE.1.D0) GO TO 110
C RESCALE
      IB = IB - 1
      CNT = CNT*EPS1
  110 PS =WH
      IF (IB.EQ.0) FINSUM = FINSUM + CNT
      GO TO 100
  120 PROB =FINSUM + INFSUM
  130 IF (INT.EQ.0) GO TO 140
      PROB = 1.0 - PROB
      TEMP = P
      P = Q
      Q = TEMP
  140 RETURN
      END


      SUBROUTINE GAMIC(X,ALPHA,REL,N,Y,NZ)
C***********************************************************************
C
C     WRITTEN BY D.E. AMOS AND S.L. DANIEL, NOVEMBER, 1974.
C     EXTRACTED FROM THE SANDIA LABORATORY'S MATHEMATICAL
C     LIBRARY AND ADAPTED FOR THE IBM 370 SYSTEM BY J. K. SHULTIS
C     JULY, 1979.
C
C     REFERENCE SC-DR-72 0303
C
C     ABSTRACT
C         GAMIC COMPUTES AN N MEMBER SEQUENCE OF INCOMPLETE GAMMA
C         FUNCTIONS NORMALIZED SO THAT AT X=INFINITY, THE INCOMPLETE
C         GAMMA FUNCTION HAS THE VALUE 1. THE SEQUENCE IS DENOTED BY
C
C         Y(K)=INCGAMMA(ALPHA+K-1,X)/GAMMA(ALPHA+K-1),   K=1,2,...,N
C
C         AND IS COMPUTED TO A RELATIVE ERROR REL OR BETTER WHERE ALPHA
C         .GT.0. IF ALPHA+N-1.GE.X, THE LAST MEMBER IS COMPUTED BY THE
```

```
C          CONFLUENT HYPERGECMETRIC SERIES WITH THE CTHER MEMBERS
C          COMPUTED BY BACKWARD RECURSION CN A TWO-TERM FORMULA,
C
C            Y(K-1)=Y(K)+DEXP((ALPHA+K-1)*DLCC(X)-X-DLGAMA(ALPHA+K)).
C
C          IF ALPHA+N-1.LT.X, AN INTEGER M IS ADDED SC THAT
C          ALPHA+N-1+M.GE.X AND THE FIRST PROCEDURE IS APPLIED. SPECIAL
C          PROCEDURES APPLY FCR ALPHA.EQ.1 CR AN UNDERFLOW CCCURS OR
C          X EXCEEDS A CRITICAL VALUE, APTEST, WHERE ALL MEMBERS ARE 1.
C          TO THE WORD LENGTH OF THE CDC 6600. GAMIC USES DLGAMA.
C
C      DESCRIPTION OF ARGUMENTS
C
C          INPUT
C            X        - ARGUMENT, X.GE.0.0
C            ALPHA    - PARAMETER, ALPHA.GT.0.0
C            REL      - RELATIVE ERROR TOLERANCE, REL=1.E-S FOR S
C                         SIGNIFICANT DIGITS
C            N        - NUMBER CF GAMMA FUNCTIONS IN THE SEQUENCE
C                         BEGINNING AT PARAMETER ALPHA, N.GE.1
C
C          OUTPUT
C            Y        - A VECTCR CCNTAINING AN N MEMBER SEQUENCE
C                         Y(K)= INCGAMMA(ALPHA+K-1,X)/GAMMA(ALPHA+K-1),
C                         K=1,....,N TC A RELATIVE ERROR REL.
C            NZ       - UNDERFLCW FLAG
C                         NZ.EQ.0, A NORMAL RETURN
C                         NZ.NE.0, UNDERFLCW, Y(K)=0.0, K=N-NZ+1,N RETURNED
C
C      ERRCR CONDITIONS
C          IMPROPER INPUT PARAMETERS - A FATAL ERROR
C          UNDERFLOW - A NON-FATAL ERROR.
C
C***********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 X ALPHA,REL,Y,DLCG,DEXP,DFLOAT
      DIMENSICN Y(1)
      DIMENSICN AA(6),BB(6),CC(5)
C
      DATA   AA              / 1.18399941922176D+00, 3.30888136276886lD+02,
     1 1.04930832947926D+04, 3.78420325596 9C80+04, 1.57586618187374D+02,
     2 1.30569632410551D+03/
      DATA   BB              / 1.02652821626751D+00, 9.29753107520368D+03,
     1 6.53848923630220D+06, 2.89543295992889D+08, 8.16836456953161D+03,
     2 4.12237656364399D+06/
      DATA   CC              / 4.30952856710482D+05, 8.27988256743362D+09,
     1 2.41944468684445D+12, 4.21722873236008D+05, 7.56593802747116D+09/
      DATA SLOG/-160.0D0/
      DATA   CON14           / 3.32361913019165D+1/
      DATA   SCALE           /1.D-60/
      DATA   ASCL            /1.D-16/
C
      IF(REL.LE.0.) GO TO 91
      IF(N.LT.1) GO TO 92
      IF(ALPHA.LE.0.0) GO TO 93
      NZ=0
      IF(X) 94,10,20
   10 DO 11 I=1,N
   11 Y(I)=0.
      RETURN
```

```
C
C     IF X.GE.XLIM(ALPHA+N-1), THEN Y(K)=1., K=1,2,...,N
C
   20 NN=N
      RX=1./X
      NBAR=0
      APN=ALPHA+DFLOAT(N)-1.
      APTEST=APN
      AM1=ALPHA-1.
      IF(APN.LE.1.) GO TO 22
   21 IF(X.LE.APTEST) GO TO 40
      IF(APTEST.GT.200.) GO TO 25
      S1=((AA(1)*APTEST+AA(2))*APTEST+AA(3))*APTEST+AA(4)
      S2=(APTEST+AA(5))*APTEST+AA(6)
      GO TO 226
   25 IF(APTEST.GT.10000.) GO TO 36
      S1=((BB(1)*APTEST+BB(2))*APTEST+BB(3))*APTEST+BB(4)
      S2=(APTEST+BB(5))*APTEST+BB(6)
  226 XLIM=S1/S2
C
      IF(X.GE.XLIM) GO TO 26
      GO TO 35
   36 S1=((APTEST+CC(1))*APTEST+CC(2))*APTEST+CC(3)
      S2=(APTEST+CC(4))*APTEST+CC(5)
      GO TO 226
   26 DO 27 I=1,N
   27 Y(I)=1.
      RETURN
C
   22 IF(ALPHA.NE.1.) GO TO 32
      IF(X.GT.CCN14) GO TO 26
      IF(X.LT.0.1) GO TO 40
      Y(NN)=1.-DEXP(-X)
      RETURN
   32 NBAR=1
      APTEST=APN+1.
      GO TO 21
   35 NBAR=X-APTEST+5.+DFLOAT(NBAR)
   40 ABAR=APN+DFLOAT(NBAR)
      XLOG=DLOG(X)
      A1=1.
      SUM=1.
      ABK=ABAR+1.
 80   A1 = A1*X/ABK
      SUM = SUM + A1
      IF(A1.LT.REL) GO TO 100
      ABK=ABK+1.
      GO TO 80
  100 YY=SUM*SCALE
      D=ABAR
      IF(NBAR.EQ.0) GO TO 110
  105 CONTINUE
      DO 106 K=1,NBAR
      XOD=X/D
      IF(XOD.LT.ASCL) GO TO 106
      YY=XOD*YY+SCALE
  106 D=D-1.
C
      IF(NZ.NE.0) GO TO 114
 110  E= -X+D*XLOG-DLGAMA(D+1.)
```

```
  114 IF(E.GE.SLOG) GO TO 120
      Y(NN)=0.
      NZ=NZ+1
      NN=NN-1
      IF(NN.EQ.0) RETURN
      NBAR=1
      APN=APN-1.
      E=E+DLOG(D*RX)
      GO TO 105
 120  EXE = DEXP(E)
      Y(NN)=(EXE/SCALE)*YY
      NM1=NN-1
      IF(NM1.EQ.0) RETURN
      F=EXE*APN*RX
      KK = NN
      AK=DFLOAT(NN)+AM1
      DO 125 K=1,NM1
      Y(KK-1)=Y(KK)+F
      KK=KK-1
      AK=AK-1.
      F=F*AK*RX
  125 CONTINUE
      RETURN
C
   91 WRITE(9,191)
  191 FORMAT('IN GAMIC, IMPROPER INPUT FOR REL.')
      RETURN
   92 WRITE(9,192)
  192 FORMAT('IN GAMIC, IMPROPER INPUT FOR N.')
      RETURN
   93 WRITE(9,193)
  193 FORMAT('IN GAMIC, IMPROPER INPUT FOR ALPHA.')
      RETURN
   94 WRITE(9,194)
  194 FORMAT('IN GAMIC, IMPROPER INPUT FOR X.')
      RETURN
      END
C************************************************************************
C*     DXP FUNCTION
C************************************************************************
C*
C*   THIS FUNCTION ERROR CHECKS THEN EVALATES DEXP(X).
C*
C*
C*
      REAL FUNCTION DXP*8(X)
      REAL*8 DEXP,X,DABS
      IF (DABS(X).GT.170.D00) GOTO 10
      DXP=DEXP(X)
      RETURN
   10 IF (X) 20,20,30
   20 DXP=0.D00
      RETURN
   30 DXP=1.D070
      RETURN
      END
```

The following is a program for generating the number of failures, F_i, which occur in time T for (F_i,T) data which is distributed according to the compound model of section III.2.2.

In the driving program:

NDATA = number of data points per data set

NSETS = number of data sets generated

A = Alpha parameter

B = beta parameter

DATA(J,I) = random data

The subroutine 'RANDU' is for generating random numbers between 0 and 1, and comes from IBM Manual C20-8011.

The subroutine 'RF' is for converting a random number between 0 and 1 into a number of failures F_i.

```
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON A,B,IX,IY
      DIMENSION F(250,20)
      NDATA=20
      NSETS=250
      T=10000.0D0
      IX=647298693
      DO 1 I=1,43
      CALL RANDU(IX,IY,YFL)
    1 IX=IY
      DO 2 IA=1,5
      IF(IA.EQ.1)A=1.0D0
      IF(IA.EQ.2)A=1.25D00
      IF(IA.EQ.3)A=1.50D0
      IF(IA.EQ.4)A=2.0D0
      IF(IA.EQ.5)A=4.0D0
      DO 2 IB=1,5
      B=A*10.0D0**(3.50D0+DFLOAT(IB)*0.50D0)
      DO 3 L=1,NSETS
      DO 3 I=1,NDATA
    3 F(L,I)=RF(T)
      WRITE(9,11) ((F(L,I),I=1,NDATA),L=1,NSETS)
    2 CONTINUE
   11 FORMAT(100(10F3.0/))
      STOP
      END



      REAL FUNCTION RF*8(T)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON A,B,IX,IY
      DIMENSION XC(201)
      XC(1)=0.0D0
      J=2
      DO 1 I=1,200
      F=DFLOAT(I-1)
      P=DLGAMA(F+A)+F*DLOG(T)+A*DLOG(B)-DLGAMA(A)-DLGAMA(F+1.0D0)
     1       -(F+A)*DLOG(T+B)
      XC(I+1)=XC(I)+DEXP(P)
      IF(XC(I+1).GE.0.9999D0)GO TO 2
    1 J=J+1
    2 CALL RANDU(IX,IY,YFL)
      IX=IY
      IF(YFL.GT.0.9999D0)GO TO 4
      IF(YFL.GT.XC(J))GO TO 4
      DO 3 I=2,J
      IF(YFL.LE.XC(I))RF=DFLOAT(I-2)
      IF(YFL.LE.XC(I))GO TO 5
    3 CONTINUE
      GO TO 5
    4 RF=DFLOAT(J-1)
    5 RETURN
      END
```

```
      SUBROUTINE RANDU(IX,IY,YFL)
      REAL*8 YFL
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
```

The following is a program for generating data which is distributed according to the exponential distribution.

In the driving program:

NDATA = number of data points per data set

NSETS = number of data sets generated

BETA = beta parameter

DATA(J,I) = random data

The subroutine 'RANDU' is for generating random numbers between 0 and 1, and comes from IBM Manual C20-8011.

```
      IMPLICIT REAL*8 (A-H,C-Z)
      DIMENSION DATA(250,20)
      NDATA=20
      NSETS=250
      IX=527693423
      DO 1 I=1,43
      CALL RANDU(IX,IY,YFL)
    1 IX=IY
      DO 2 IB=1,5
      BETA=0.10D0**(IB+2)
      DO 3 L=1,NSETS
      DO 3 I=1,NDATA
      CALL RANDU(IX,IY,YFL)
      IX=IY
    3 DATA(L,I)=-DLOG(1.0D0-YFL)/BETA
      WRITE(9,4) ((DATA(L,I),I=1,NDATA),L=1,NSETS)
    4 FORMAT(10(10D12.4/))
    2 CONTINUE
      STOP
      END
```

```
      SUBROUTINE RANDU(IX,IY,YFL)
      REAL*8 YFL
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
```

Appendix D: Power Curve Programs

The following 4 programs were used to generate power curves for single outlier detection for outliers in gamma distributions, (F_i, T_i) data, and exponential distributions.

The following is a program for generating power curves for several methods of detecting a single outlier in a gamma distribution by the method described in App. A and by simulation.

In the driving program:

NDATA = number of data points per data set less one

NSETS = number of data sets analyzed

X(J,I) = x_{ci} of App. A for the various methods

FCUM(J,K) = $F(x_b)$ of App. A for the various methods

PRCUM(J,L) = points in the power curve for the method of App. A

PRSIM(J,L) = points in the power curve for the simulation method

The subroutine 'ORDER' is for ordering the critical points at which an outlier is detected (the values of x_{ci} of App. A)

The subroutine 'RANDU' is for generating a random number between 0 and 1, and comes from IBM Manual C20-8011.

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SNGL
      COMMON /PRIR/ PARM(4),RNG(2),IPRIR
      COMMON /RTPASS/ XLI,XRI,EPS,IEND,IERR
      COMMON /ORD/ X(5,1000),IND(5,1000),XO(5,1000)
      COMMON/FUNSUB/ ALPH(21),BET(21),L
      DIMENSION DATA(1000,50),A2(1000),A3(1000),NINT(5),T95(15),T99(15),
     1     B3(1000),T(15),AL(15),XC(5,302),FC(5,302),PROB(5,25),A1(1000)
     2     ,B1(1000),ICH(5),RLAM(21),PRSIM(5,1000),XSIM(1000)
C
C *** INITIALIZE FUNCTION SUBPROGRAMS 'PRCNT' AND 'PRCUM'
C
      XLI=0.0D0
      XRI=1.0D0
      EPS=1.D-4
      IENC=250
      IPRIR=1
      PARM(3)=0.0D0
      PARM(4)=0.0D0
C
C *** INITIALIZE SUBPROGRAM 'GAMINT'
C
      ERR=1.D-6
C
C *** INITIALIZE MAIN PROGRAM
C
      NCATA=19
      NSETS=263
C *** READ IN CRITICAL VALUES OF FISCHER'S TEST STATISTIC
      READ(5,*) (T95(J),J=1,13)
      READ(5,*) (T99(J),J=1,13)
C *** READ IN VALUES OF ALPHA FOR WHICH THE CRITICAL VALUES OF THE
C *** TEST STATISTIC ARE KNOWN (.25, .5, 1, 1.25, 1.5, 2, 2.5, 3, 3.5,
C *** 4, 6, 8, 18)
      READ(5,*) (AL(J),J=1,13)
C
C *** INITIALIZE SUBPROGRAM 'RANDU'
C
      IX=6481739
      DO 43 J=1,47
      CALL RANDU(IX,IY,YFL)
   43 IX=IY
C
C *** SET ALPHA AND BETA VALUES AND READ AND OUTPUT DATA
C
      DO 100 IA=1,5
      IF(IA.EQ.1)ALPHA=1.0D0
      IF(IA.EQ.2)ALPHA=1.25D0
      IF(IA.EQ.3)ALPHA=1.5D0
      IF(IA.EQ.4)ALPHA=2.0D0
      IF(IA.EQ.5)ALPHA=4.0D0
      DO 100 IB=1,5
      IF(IB.EQ.1)BETA=ALPHA/1.D-2
      IF(IB.EQ.2)BETA=ALPHA/1.D-4
      IF(IB.EQ.3)BETA=ALPHA/1.D-5
      IF(IB.EQ.4)BETA=ALPHA/1.D-6
      IF(IB.EQ.5)BETA=ALPHA/1.D-7
      PARM(1)=ALPHA
      PARM(2)=BETA
```

```
      READ(9,50)((DATA(L,I),I=1,NDATA),L=1,NSETS)
      READ(9,50) DUMMY
   50 FORMAT(100(100D12.4/))
      IF(NSETS.NE.263)WRITE(6,*)NSETS
      WRITE(6,67)
      WRITE(6,23) (L,(DATA(L,I),I=1,NDATA),L=15,NSETS,15)
      IF(NSETS.NE.263)WRITE(6,*)NSETS
C *** CHANGE FORMAT 23 TO ACCOMODATE 'NDATA'
   23 FORMAT(' ',4X,100(I3,2X,'*',10D10.3,/,10X,'*',9D10.3,/,10X,'*',/,5
     1X))
   67 FORMAT('1',4X,'DATA',1X,'*',/,5X,'SET',2X,'*',51X,'DATA',/,
     1      4X,6('-'),'*',100('-'))
C
C *** SET THE CRITICAL VALUES OF THE TEST STATISTICS FOR THE 95% AND
C *** 99% CONFIDENCE LEVELS
C
C *** USE 'DO 100 IT=1,2' FOR BOTH 95% AND 99% CONFIDENCE LEVELS
      IT=1
      IF(IT.EQ.2)GO TO 37
      DO 38 IT1=1,13
   38 T(IT1)=T95(IT1)
      TN=2.18D00
      CUM=0.95**(1./DFLOAT(NDATA+1))
      GO TO 39
   37 DO 40 IT1=1,13
   40 T(IT1)=T99(IT1)
      TN=3.18D00
      CUM=0.99**(1./DFLOAT(NDATA+1))
   39 DO 2 J3=1,12
      IF(ALPHA.GE.AL(J3).AND.ALPHA.LE.AL(J3+1))TFA=T(J3)+(T(J3+1)-T(J3))
     1      *(ALPHA-AL(J3))/(AL(J3+1)-AL(J3))
      IF(ALPHA.GE.AL(J3).AND.ALPHA.LE.AL(J3+1))JX=J3
    2 CONTINUE
      IF(TFA.GE.1.D00)WRITE(6,*)TFA,JX,ALPHA,T(JX),T(JX+1),AL(JX),
     1      AL(JX+1)
C
C *** MAKE PRELIMINARY CALCULATIONS
C
      DO 18 L=1,NSETS
      SUM1=0.D00
      DO 1 I=1,NDATA
    1 SUM1=SUM1+DATA(L,I)
      AVG=SUM1/DFLOAT(NDATA)
      VAR=0.D00
      DO 97 I=1,NDATA
   97 VAR=VAR+(DATA(L,I)-AVG)**2
      VAR=VAR/DFLOAT(NDATA-1)
      A1(L)=AVG**2/VAR
      B1(L)=AVG/VAR
      IF(VAR.EQ.0.D00)WRITE(6,*)VAR,AVG,L,NDATA,NSETS,(DATA(L,I),I=1,19)
      IF(NDATA.EQ.1)WRITE(6,*)NDATA
C
C *** FIND THE MINIMUM OUTLIER USING FISCHER'S METHOD WITH KNOWN ALPHA
C
      X(1,L)=TFA*SUM1/(1.-TFA)
C
C *** FIND MINIMUM OUTLIER USING FISCHER'S METHOD WITH ALPHA SET
C *** EQUAL TO 1
C
      X(2,L)=T(3)*SUM1/(1.-T(3))
```

```
      IF(T(3).GE.1.0D0)WRITE(6,*)T(3)
C
C *** FIND MINIMUM OUTLIER USING FISCHER'S METHOD AND EXCLUDING
C *** THE OUTLIER
C
      A=A1(L)
      IF(A.LT.0.25D0)A=0.25D0
      IF(A.GT.18.0D0)A=18.0D0
      DO 7 J1=1,12
      IF(A.GE.AL(J1).AND.A.LE.AL(J1+1))TF=T(J1)+(T(J1+1)-T(J1))
     1       *(A-AL(J1))/(AL(J1+1)-AL(J1))
    7 CONTINUE
      TF1=TF
      X3=TF*SUM1/(1.0D0-TF)
      AVG=(SUM1+X3)/DFLOAT(NDATA+1)
      VAR=(X3-AVG)**2
      DO 5 I=1,NDATA
    5 VAR=VAR+(DATA(L,I)-AVG)**2
      VAR=VAR/DFLOAT(NDATA)
      A2(L)=AVG**2/VAR
      IF(A2(L).LT.0.25D0)A2(L)=0.25D0
      IF(A2(L).GT.18.0D0)A2(L)=18.0D0
      DO 6 I=1,12
      IF(A2(L).GE.AL(I).AND.A2(L).LE.AL(I+1))TF=T(I)+(T(I+1)-T(I))
     1       *(A2(L)-AL(I))/(AL(I+1)-AL(I))
    6 CONTINUE
      X(3,L)=TF*SUM1/(1.0D0-TF)
  101 FORMAT(' ',4X,'FULL ACCURACY NOT REACHED FOR CONFIDENCE LEVEL',
     1  I2,' DATA SET',I5,' METHOD',I2)
C
C *** FIND MINIMUM OUTLIER USING THE INTEGRATION METHOD
C
    8 AVG1=SUM1/DFLOAT(NDATA)
      VAR1=0.0D0
      DO 9 I=1,NDATA
    9 VAR1=VAR1+(DATA(L,I)-AVG1)**2
      VAR1=VAR1/DFLOAT(NDATA-1)
      PARM(1)=AVG1**2/VAR1
      PARM(2)=AVG1/VAR1
      X5=PRCNT(CUM)
      IF(IERR.EQ.2)WRITE(6,*)L,PARM(1),PARM(2),X5
      AVG=(SUM1+X5)/DFLOAT(NDATA+1)
      VAR=(X5-AVG)**2
      DO 10 I=1,NDATA
   10 VAR=VAR+(DATA(L,I)-AVG)**2
      VAR=VAR/DFLOAT(NDATA)
      PARM(1)=AVG**2/VAR
      PARM(2)=AVG/VAR
      A3(L)=AVG**2/VAR
      B3(L)=AVG/VAR
      X(5,L)=PRCNT(CUM)
      IF(IERR.EQ.2)WRITE(6,*)L,PARM(1),PARM(2),X5
C
C *** FIND MINIMUM OUTLIER BY CONVERTING THE DATA TO NORMALLY
C *** DISTRIBUTED DATA AND USING DIXON'S METHOD
C
      TNM=DSQRT(DFLOAT(NDATA))
      IF(TN.GT.TNM)GO TO 98
      DO 12 I=1,NDATA
   12 DATA(L,I)=DATA(L,I)**(1.0D0/3.0D0)
```

```
      W=X(1,L)**(1.0D0/3.0D0)
      IF(IT.EQ.2)W=X(4,L)**(1.0D0/3.0D0)
      WSUM1=0.0D0
      DO 13 I=1,NDATA
   13 WSUM1=WSUM1+DATA(L,I)
      DO 16 J=1,250
      WAVG=(WSUM1+W)/DFLOAT(NDATA+1)
      WDSQ=(W-WAVG)**2
      DO 14 I=1,NDATA
   14 WDSQ=WDSQ+(DATA(L,I)-WAVG)**2
      WVAR=WDSQ/DFLOAT(NDATA)
      WN=WAVG+TN*DSQRT(WVAR)
      WD=DABS(WN-W)/WN
      IF(WD.LT.0.0002)GO TO 15
   16 W=WN
      IM=4
      WRITE(6,101) IT,L,IM
   15 X(4,L)=WN**3
      IF(IT.EQ.2)GO TO 18
      DO 51 I=1,NDATA
   51 DATA(L,I)=DATA(L,I)**3
      GO TO 18
   98 X(4,L)=1.D75
   18 CONTINUE
C
C *** CALCULATE THE TRUE MINIMUM OUTLIER VALUE
C
      PARM(1)=ALPHA
      PARM(2)=BETA
      X5=PRCNT(CUM)
C
C *** OUTPUT CALCULATED ALPHA AND BETA VALUES
C
      WRITE(6,64)
      WRITE(6,65)
      WRITE(6,66) (L,A1(L),B1(L),A2(L),A3(L),B3(L),L=15,NSETS,15)
   64 FORMAT('1',4X,'A1 = ALPHA CALCULATED FROM THE DATA',/,5X,'B1 = BET
     1A CALCULATED FROM THE DATA',/,5X,'A2 = ALPHA CALCULATED WITH THE O
     2UTLIER USING FISCHER''S METHOD (MIN = 0.25, MAX = 18)',/,5X,'A3 =
     3ALPHA CALCULATED WITH THE OUTLIER USING THE INTEGRATION METHOD',/,
     45X,'B3 = BETA CALCULATED WITH THE OUTLIER USING THE INTEGRATION ME
     5THOD')
   65 FORMAT('-',31X,'DATA',1X,'*',4X,'A1',4X,'*',5X,'B1',5X,'*',4X,
     1  'A2',4X,'*',4X,'A3',4X,'*',5X,'B3',/,31X,6('-'),'*',10('-'),
     2  '*',12('-'),'*',2(10('-'),'*'),12('-'))
   66 FORMAT(' ',31X,100(I3,2X,'*',F9.4,1X,'*',D11.4,1X,'*',2(F9.4,
     1  1X,'*'),D11.4,/,32X))
C
C *** ORDER AND OUTPUT MINIMUM OUTLIER VALUES
C
      WRITE(6,32)
      WRITE(6,60)
      WRITE(6,62) (L,(X(J,L),J=1,5),L=15,NSETS,15)
      WRITE(6,61) X5
   60 FORMAT('-',29X,'*',12X,'FISCHER''S METHOD',2(13X,'*'),/,25X,
     1  'DATA',1X,2('*',4X,'ALPHA',4X),'*',1X,'ALPHA CALC.',1X,'*',4X,
     2  'NORMAL',3X,'*',1X,'INTEGRATION',/,25X,'SET',2X,'*',4X,'KNOWN',
     3  4X,'*',5X,'= 1',5X,'*',1X,'W/ OUTLIER',2X,2('*',4X,
     4  'METHOD',3X),/,24X,6('-'),5('*',13('-')))
   62 FORMAT(' ',24X,100(I3,2X,5('*',2X,D9.3,2X),/,25X))
```

```
   74 NFL=NFL+NF
   94 FC(J,I+1)=FC(J,I)+DFLOAT(NF)/DFLOAT(NSETS)
      GO TO 71
   93 ICH(J)=1
   71 CONTINUE
      DO 117 J=1,5
      NIN2=NINT(J)+2
      DO 117 I=NIN2,300
      FC(J,I)=0.0D0
  117 XC(J,I)=0.0D0
C *** OUTPUT F-CUM
      DO 75 J=1,5
      IF(ICH(J).EQ.1)GO TO 75
      WRITE(6,87)
      IF(J.EQ.1)WRITE(6,82)
      IF(J.EQ.2)WRITE(6,83)
      IF(J.EQ.3)WRITE(6,84)
      IF(J.EQ.4)WRITE(6,85)
      IF(J.EQ.5)WRITE(6,86)
      WRITE(6,76)
      DO 103 I=1,50
      N1=I
      N2=I+50
      N3=I+100
      WRITE(6,77) N1,XC(J,I),XC(J,I+1),FC(J,I+1),N2,XC(J,I+50),
     1  XC(J,I+51),FC(J,I+51),N3,XC(J,I+100),XC(J,I+101),FC(J,I+101)
  103 CONTINUE
   87 FORMAT('1',4X,'CUMULATIVE PROBABILITY THAT A GIVEN VALUE IS DETECT
     1ED AS AN OUTLIER USING')
   82 FORMAT(' ',4X,'FISCHER''S METHOD WITH ALPHA KNOWN')
   83 FORMAT(' ',4X,'FISCHER''S METHOD WITH ALPHA = 1')
   84 FORMAT(' ',4X,'FISCHER''S METHOD AND CALCULATING ALPHA')
   85 FORMAT(' ',4X,'THE CONVERSION TO NORMAL METHOD')
   86 FORMAT(' ',4X,'THE INTEGRATION METHOD')
   76 FORMAT('-',2(5X,'*',3X,'LOWER',3X,'*',3X,'UPPER',3X,'*',11X,
     1  '*'),5X,'*',3X,'LOWER',3X,'*',3X,'UPPER',3X,'*',/,1X,2(1X,'INT',
     2  1X,'*',2(3X,'LIMIT',3X,'*'),3X,'F-CUM',3X,'*'),1X,'INT',1X,
     3  2('*',3X,'LIMIT',3X),'*',3X,'F-CUM',/,1X,2(5('-'),'*',11('-'),
     4  '*',11('-'),'*',11('-'),'*'),5('-'),'*',11('-'),'*',11('-'),'*',
     5  11('-'))
   77 FORMAT(' ',2(I4,1X,'*',D10.3,1X,'*',D10.3,1X,'*',F10.3,1X,'*'),
     1  I4,1X,'*',D10.3,1X,'*',D10.3,1X,'*',D10.3)
      IF(NINT(J).LE.150)GO TO 75
      WRITE(6,87)
      IF(J.EQ.1)WRITE(6,82)
      IF(J.EQ.2)WRITE(6,83)
      IF(J.EQ.3)WRITE(6,84)
      IF(J.EQ.4)WRITE(6,85)
      IF(J.EQ.5)WRITE(6,86)
      WRITE(6,76)
      DO 104 I=151,200
      N1=I
      N2=I+50
      N3=I+100
      WRITE(6,77) N1,XC(J,I),XC(J,I+1),FC(J,I+1),N2,XC(J,I+50),
     1  XC(J,I+51),FC(J,I+51),N3,XC(J,I+100),XC(J,I+101),FC(J,I+101)
  104 CONTINUE
   75 CONTINUE
C
C *** CALCULATE AND OUTPUT THE PROBABILITY THAT AN OUTLIER COMING FROM
```

```
   61 FORMAT('0',4X,'THE TRUE MINIMUM OUTLIER FOR THE INTEGRATION METHOD
     1=',D12.4)
      CALL ORDER(NSETS)
      WRITE(6,32)
      WRITE(6,33)
      WRITE(6,34)
      L1=0
      DO 137 L=1,NSETS
      L1=L1+1
      WRITE(6,31) (IND(J,L),XO(J,L),J=1,5)
      IF(L1.LT.50)GO TO 137
      L1=0
      WRITE(6,32)
      WRITE(6,33)
      WRITE(6,34)
  137 CONTINUE
   32 FORMAT('1',4X,'MINIMUM OUTLIER VALUES')
   33 FORMAT('-',3X,'DATA',1X,'*',2(4X,'ALPHA',4X,'*',1X,'DATA',1X,'*'),
     1  1X,'ALPHA CALC.',1X,'*',1X,'DATA',1X,'*',4X,'NORMAL',3X,
     2  '*',1X,'DATA',1X,'*',1X,'INTEGRATION',/,4X,'SET',2X,'*',4X,
     3  'KNOWN',4X,'*',1X,'SET',2X,'*',5X,'= 1',5X,'*',1X,'SET',
     4  2X,'*',1X,'W/ OUTLIER',2X,'*',1X,'SET',2X,'*',4X,'METHOD')
   34 FORMAT('+',85X,'*',1X,'SET',2X,'*',4X,'METHOD',/,3X,6('-'),
     1  4('*',13('-'),'*',6('-')),'*',13('-'))
      WRITE(6,61) X5
   31 FORMAT(' ',3X,I3,2X,4('*',2X,D9.3,2X,'*',1X,I3,2X),'*',2X,D9.3)
C
C *** SET INTERVALS FOR AND CALCULATE THE CUMULATIVE PROBABILITY THAT A
C *** GIVEN VALUE IS DETECTED AS AN OUTLIER IN THE GAMMA DISTRIBUTION
C *** WITH PARAMETERS ALPHA AND BETA (F-CUM)
C
      DO 71 J=1,5
      ICH(J)=0
      IF(XO(J,1).GT.1000.)GO TO 93
      DX=(XO(J,NSETS)-XO(J,1))/100.0D0
      IF(DX.GT.4999.)GO TO 93
      DO 90 ID=1,24
      P=5.0D0/10.0D0**(ID-4)
      Q=2.0D0/10.0D0**(ID-4)
      R=1.0D0/10.0D0**(ID-4)
      IF(DX.LT.P)DELX=Q
      IF(DX.LT.Q)DELX=R
      IF(DX.LT.R)DELX=P/10.0D0
      IF(DX.GT.R)GO TO 91
   90 CONTINUE
   91 NINT(J)=IFIX(SNGL((XO(J,NSETS)-XO(J,1))/DELX))+2
      NT=IFIX(SNGL(XO(J,1)/DELX))
      XC(J,1)=DFLOAT(NT)*DELX
      NFL=1
      FC(J,1)=0.0D0
      NIN=NINT(J)
      DO 94 I=1,NIN
      XC(J,I+1)=XC(J,I)+DELX
      NF=0
      XM=(XC(J,I+1)+XC(J,I))/2.0D0
      IF(NFL.GT.NSETS)GO TO 94
      DO 73 K=NFL,NSETS
      IF(XO(J,K).LE.XM)NF=NF+1
      IF(XO(J,K).GT.XM)GO TO 74
   73 CONTINUE
```

```
C *** THE GAMMA DISTRIBUTION WITH PARAMETERS ALPHAI AND BETAI IS
C *** DETECTED AS AN OUTLIER IN DATA COMING FROM THE GAMMA DISTRIBUTICN
C *** WITH PARAMETERS ALPHA ANC BETA
C
      VAR=ALPHA/BETA**2
      DO 72 L=1,11
      RLAM(L)=DFLOAT(L)*(1.+(.35*BETA*(4.-ALPHA))/(3.*ALPHA*1.07))
     1       /DSQRT(ALPHA)
      IF(IT.EQ.2)RLAM(L)=1.4*RLAM(L)
      AVG=RLAM(L)*ALPHA/BETA
      PARM(1)=AVG**2/VAR
      PARM(2)=AVG/VAR
      ALPH(L)=AVG**2/VAR
      BET(L)=AVG/VAR
      DO 72 J=1,5
      IF(L.EQ.1)GO TO 125
      IF(PROB(J,L-1).GE.0.995)GC TO 126
  125 PRCB(J,L)=0.0D0
      IF(ICH(J).EQ.1)GO TO 72
      NIN=NINT(J)
      DO 78 I=1,NIN
      GAM=PRCUM(XC(J,I+1))-PRCUM(XC(J,I))
   78 PROB(J,L)=PROB(J,L)+FC(J,I+1)*GAM
      GAM=PRCUM(XC(J,NINT(J)+1))
      PROB(J,L)=PROB(J,L)+(1.0D0-GAM)
      GO TO 72
  126 PROB(J,L)=1.0D0
   72 CCNTINUE
      DO 105 L=1,11
      PARM(1)=ALPH(L)
      PARM(2)=BET(L)
      DO 127 L1=1,NSETS
      CALL RANDU(IX,IY,YFL)
      IX=IY
  127 XSIM(L1)=PRCNT(YFL)
      DO 105 J=1,5
      PRSIM(J,L)=0.0D0
      DO 105 L1=1,NSETS
      IF(XSIM(L1).GT.X(J,L1))PRSIM(J,L)=PRSIM(J,L)+1.0D0/DFLOAT(NSETS)
  105 CONTINUE
      WRITE(6,79) ALPHA,BETA
      WRITE(6,80)
      WRITE(6,81) (RLAM(L),ALPH(L),BET(L),(PROB(J,L),J=1,5),L=1,11)
      WRITE(6,106) ALPHA,BETA
      WRITE(6,80)
      WRITE(6,81) (RLAM(L),ALPH(L),BET(L),(PRSIM(J,L),J=1,5),L=1,11)
   79 FORMAT('1',4X,'PROBABILITY THAT A DATA POINT GENERATED FROM THE GA
     1MMA DISTRIBUTION WITH PARAMETERS',/,5X,'ALPHA AND BETA WILL BE DET
     2ECTED AS AN OUTLIER IN THE GAMMA DISTRIBUTION WITH PARAMETERS',/,
     35X,'ALPHA =',F6.3,1X,'AND BETA =',D10.4)
   80 FORMAT('-',29X,'*',9X,'*',12X,'*',13X,'FISCHER''S METHOD',12X,'*',
     1  13X,'*',/,30X,'*',9X,'*',12X,2('*',4X,'ALPHA',4X),'*',1X,
     2  'ALPHA CALC.',1X,'*',4X,'NORMAL',3X,'*',1X,'INTEGRATION',/,23X,
     3  'LAMBDA',1X,'*',2X,'ALPHA',2X,'*',4X,'BETA',4X,'*',4X,'KNOWN',4X
     4  ,'*',5X,'= 1',5X,'*',1X,'W/ OUTLIER',2X,2('*',4X,'METHOD',3X),/,
     5  22X,8('-'),'*',9('-'),'*',12('-'),5('*',13('-')))
   81 FORMAT(' ',21X,21(F7.3,1X,'*',F8.3,1X,'*',D11.4,1X,5('*',F10.5,3X)
     1  /,22X))
  106 FORMAT('1',4X,'SIMULATED PROBABILITY THAT A DATA POINT GENERATED F
     1ROM THE GAMMA DISTRIBUTICN WITH PARAMETERS',/,5X,'ALPHA AND BETA W
```

```
     2ILL BE DETECTED AS AN OUTLIER IN THE GAMMA DISTRIBUTION WITH PARAM
     3ETERS',/,5X,'ALPHA =',F6.3,1X,'AND BETA =',D10.4)
 100 CONTINUE
     STOP
     END
C
C
C
C
C
      SUBROUTINE ORDER(NSETS)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ORD/ X(5,1000),IND(5,1000),XO(5,1000)
      DO 1 I=1,5
      IND(I,1)=1
      DO 1 L=2,NSETS
      L1=L-1
      DO 2 J=1,L1
      IF(X(I,L).LT.X(I,IND(I,J)))GO TO 3
    2 CONTINUE
    3 LJ=L-J
      IF(L.EQ.J)GO TO 1
      DO 5 K=1,LJ
    5 IND(I,L-K+1)=IND(I,L-K)
    1 IND(I,L-LJ)=L
      DO 4 J=1,5
      DO 4 L=1,NSETS
    4 XO(J,L)=X(J,IND(J,L))
      RETURN
      END
C
C
C
C
C
      SUBROUTINE RANDU(IX,IY,YFL)
      REAL*8 YFL
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
C
C
C
```

```
C
C
      REAL FUNCTION PRCNT*8 (X)
C***********************************************************************
C*                                                                     *
C*                              PRCNT                                  *
C*                                                                     *
C***********************************************************************
C*   THIS FUNCTION RETURNS THE CORRESPONDING LAMBDA VALUE AT THE       *
C*   X*(100) PERCENTILE FOR THE PRIOR DISTRIBUTION. THE FUNCTION USES  *
C*   THE ORTMI ROUTINE TO FIND THE ROOT OF F(LAMBDA)=CUM(LAMBDA)-X=0.  *
C*   /SUBPRM/ COMMON BLOCK IS USED TO PASS PARAMETERS TO THE FUNCTION  *
C*   PRFCT. PARAMETERS IN COMMON BLOCK /RTPASS/ ARE SET BY THE CALLING *
C*   ROUTINE AND ARE EXPLAINED IN THE ORTMI DOCUMENTATION.             *
C*                                                                     *
C*                                                                     *
C*   WRITTEN BY: S. D. HANSEN          5/80.                           *
C*                                                                     *
C***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/RTPASS/  XLI,XRI,EPS,IEND,IERR
      COMMON/SUBPRM/ ALPHA
      EXTERNAL PRFCT
      ALPHA=X
      CALL ORTMI (ANS,F,PRFCT,XLI,XRI,EPS,IEND,IERR)
      IF (IERR.NE.0) WRITE(6,100) IERR
 100  FORMAT(' ERROR IN PRCNT ROUTINE, IERR= ',I4)
      PRCNT=ANS
      RETURN
      END
C*
C*
C*
C*
      REAL FUNCTION PRFCT*8(P)
C***********************************************************************
C*                                                                     *
C*   THIS FUNCTION EVALUATES F(LAMBDA)=CUM(LAMBDA)-ALPHA FOR THE       *
C*   PRIOR DISTRIBUTION.                                               *
C*                                                                     *
C***********************************************************************
      REAL*8 PRCUM,ALPHA,P
      COMMON/SUBPRM/ ALPHA
      PRFCT=PRCUM(P)-ALPHA
C&&   WRITE (6,*) P,PRFCT
      RETURN
      END
```

```
      REAL FUNCTION PRIOR*8(X)
C*************************************************************************
C*                                                                       *
C* THIS SUBROUTINE EVALUATES THE PRIOR DENSITY FUNCTION FOR THE FAIL-    *
C* URE RATE CASES. DISTRIBUTION PARAMETER VALUES AND FUNCTION DEFINING   *
C* INDEX ARE PASSED TO SUBROUTINE THROUGH THE COMMON BLOCK /PRIR/        *
C*                                                                       *
C* SUBROUTINES NEEDED: DXP                                               *
C*                                                                       *
C* SUBROUTINE ARGUMENTS:                                                 *
C*    X       = FAILURE RATE VALUE AT WHICH DENSITY FUNCTION IS DESIRED  *
C*                                                                       *
C* COMMON BLOCK /PRIR/ VARIABLES:                                        *
C*    PARM  = DISTRIBUTION PARAMETER VECTORS: PARM(1)=ALPHA, PARM(2)=    *
C*            BETA (=TAU FOR GAMMA), PARM(3)=MINIMUM LAMBDA OR SHIFT     *
C*            PARAMETER, PARM(4)=MAXIMUM LAMBDA PARAMETER (USED ONLY     *
C*            FOR THE BETA DISTRIBUTION)                                 *
C*    IPRIR = DISTRIBUTION INDEX; =1 GAMMA, =2 LOGNORMAL, =3 WEIBULL,    *
C*            =4 LOGBETA                                                 *
C*                                                                       *
C* WRITTEN BY J.K. SHULTIS 3/79.  MODIFIED 6/79, 7/79.                   *
C*                                                                       *
C*************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/PRIR/PARM(4),RNG(2),IPRIR
      REAL*8 X,DLGAMA,DEXP,DLOG,DABS,DXP
      PRIOR=0.0D0
      IF(X.LE.PARM(3)) RETURN
C*****
C***** GAMMA DENSITY FUNCTION:  (LAMBDA-THETA)=GAMMA(ALPHA,TAU)
C***** PARAMETERS: ALPHA>0,  TAU>0,  THETA>0;   THETA<LAMBDA<INFINITY
      GO TO (10,20,30,40),IPRIR
   10 ARG=X-PARM(3)
      AA=PARM(1)
      BB=(AA-1.0D0)*DLOG(ARG)+AA*DLOG(PARM(2))-PARM(2)*ARG
     1-DLGAMA(AA)
      PRIOR=DXP(BB)
      RETURN
C*****
C***** LOGNORMAL DENSITY FUNCTION: LOG(LAMBDA-THETA)=NORMAL(ALPHA,BETA)
C***** PARAMETERS: -INF<ALPHA<INF, BETA>0, THETA>0;  LAMBDA>0
   20 ARG=DLOG(X-PARM(3))
      B2=PARM(2)**2
      BB=-ARG-0.5D0*DLOG(6.28318530717959*B2)-0.5D0*((ARG-PARM(1))**2)/B
     12
      PRIOR=DXP(BB)
      RETURN
C*****
C***** WEIBULL DENSITY FUNCTION: (LAMBDA-THETA)=WEIBULL(ALPHA,BETA)
C***** PARAMETERS: ALPHA,BETA,THETA>0;  LAMBDA>THETA>0
   30 BB=PARM(2)
      ARG=(X-PARM(3))/PARM(1)
      AA=DLOG(BB)-DLOG(PARM(1))+(BB-1.0D0)*DLOG(ARG)-ARG**BB
      PRIOR=DXP(AA)
      RETURN
C*****
C***** FOUR PARAMETER LOGBETA:  LOG(LAMBDA)=BETA(ALPHA,BETA,A,B)
C***** PARAMETERS: ALPHA,BETA>0;  -INF<A<LOG(LAMBDA)<B<INF
   40 A=PARM(3)
      B=PARM(4)
      DL=DLOG(X)
      IF((DL.GE.B).OR.(DL.LE.A)) RETURN
```

```
      AA=PARM(1)
      BB=PARM(2)
      ARG= (DLGAMA(AA+BB)-DLGAMA(BB)-DLGAMA(AA)) - DL +
     1(-(AA+BB-1.0D0)*DLOG(B-A) + (AA-1.0D0)*DLOG(DL-A)+(BB-1.D0)*DLOG(B
     2-DL))
      PRIOR=DXP(ARG)
      RETURN
      END
C        SUBROUTINE DRTMI
C
C        PURPOSE
C           TO SOLVE GENERAL NONLINEAR EQUATIONS OF THE FORM FCT(X)=0
C           BY MEANS OF MUELLER-S ITERATION METHOD.
C
C        USAGE
C           CALL DRTMI (X,F,FCT,XLI,XRI,EPS,IEND,IER)
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT.
C
C        DESCRIPTION OF PARAMETERS
C           X      - RESULTANT ROOT OF EQUATION FCT(X)=0.
C           F      - RESULTANT FUNCTION VALUE AT ROOT X.
C           FCT    - NAME OF THE EXTERNAL FUNCTION SUBPROGRAM USED.
C           XLI    - INPUT VALUE WHICH SPECIFIES THE INITIAL LEFT BOUND
C                    OF THE ROOT X.
C           XRI    - INPUT VALUE WHICH SPECIFIES THE INITIAL RIGHT BOUND
C                    OF THE ROOT X.
C           EPS    - INPUT VALUE WHICH SPECIFIES THE UPPER BOUND OF THE
C                    ERROR OF RESULT X.
C           IEND   - MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED.
C           IER    - RESULTANT ERROR PARAMETER CODED AS FOLLOWS
C                     IER=0 - NO ERROR,
C                     IER=1 - NO CONVERGENCE AFTER IEND ITERATION STEPS
C                             FOLLOWED BY IEND SUCCESSIVE STEPS OF
C                             BISECTION,
C                     IER=2 - BASIC ASSUMPTION FCT(XLI)*FCT(XRI) LESS
C                             THAN OR EQUAL TO ZERO IS NOT SATISFIED.
C
C        REMARKS
C           THE PROCEDURE ASSUMES THAT FUNCTION VALUES AT INITIAL
C           BOUNDS XLI AND XRI HAVE NOT THE SAME SIGN. IF THIS BASIC
C           ASSUMPTION IS NOT SATISFIED BY INPUT VALUES XLI AND XRI, THE
C           PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER=2.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           THE EXTERNAL FUNCTION SUBPROGRAM FCT(X) MUST BE FURNISHED
C           BY THE USER.
C
C        METHOD
C           SOLUTION OF EQUATION FCT(X)=0 IS DONE BY MEANS OF MUELLER-S
C           ITERATION METHOD OF SUCCESSIVE BISECTIONS AND INVERSE
C           PARABOLIC INTERPOLATION, WHICH STARTS AT THE INITIAL BOUNDS
C           XLI AND XRI. CONVERGENCE IS QUADRATIC IF THE DERIVATIVE OF
C           FCT(X) AT ROOT X IS NOT EQUAL TO ZERO. ONE ITERATION STEP
C           REQUIRES TWO EVALUATIONS OF FCT(X). FOR TEST ON SATISFACTORY
C           ACCURACY SEE FORMULAE (3,4) OF MATHEMATICAL DESCRIPTION.
C           FOR REFERENCE, SEE G. K. KRISTIANSEN, ZERO OF ARBITRARY
C           FUNCTION, BIT, VOL. 3 (1963), PP.205-206.
C
C     ..................................................................
```

```
      SUBROUTINE DRTMI(X,F,FCT,XLI,XRI,EPS,IEND,IER)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C
C     PREPARE ITERATION
      IER=0
      XL=XLI
      XR=XRI
      X=XL
      TOL=X
      F=FCT(TOL)
      IF(F)1,16,1
    1 FL=F
      X=XR
      TOL=X
      F=FCT(TOL)
      IF(F)2,16,2
    2 FR=F
      IF(DSIGN(1.0D0,FL)+DSIGN(1.0D0,FR))25,3,25
C
C     BASIC ASSUMPTION FL*FR LESS THAN 0 IS SATISFIED.
C     GENERATE TOLERANCE FOR FUNCTION VALUES.
    3 I=0
      TOLF=100.*EPS
C
C
C     START ITERATION LOOP
    4 I=I+1
C
C     START BISECTION LOOP
      DO 13 K=1,IEND
      X=.5*(XL+XR)
      TOL=X
      F=FCT(TOL)
      IF(F)5,16,5
    5 IF(DSIGN(1.0D0,F)+DSIGN(1.0D0,FR))7,6,7
C
C     INTERCHANGE XL AND XR IN ORDER TO GET THE SAME SIGN IN F AND FR
    6 TOL=XL
      XL=XR
      XR=TOL
      TOL=FL
      FL=FR
      FR=TOL
    7 TOL=F-FL
      A=F*TOL
      A=A+A
      IF(A-FR*(FR-FL))8,9,9
    8 IF(I-IEND)17,17,9
    9 XR=X
      FR=F
C
C     TEST ON SATISFACTORY ACCURACY IN BISECTION LOOP
      TOL=EPS
      A=DABS(XR)
      IF(A-1.)11,11,10
   10 TOL=TOL*A
   11 IF(DABS(XR-XL)-TOL)12,12,13
   12 IF(DABS(FR-FL)-TOLF)14,14,13
   13 CONTINUE
```

```
C       END OF BISECTION LOOP
C
C       NO CONVERGENCE AFTER IEND ITERATION STEPS FOLLOWED BY IEND
C       SUCCESSIVE STEPS OF BISECTION OR STEADILY INCREASING FUNCTION
C       VALUES AT RIGHT BOUNDS. ERROR RETURN.
        IER=1
     14 IF(DABS(FR)-DABS(FL))16,16,15
     15 X=XL
        F=FL
     16 RETURN
C
C       COMPUTATION OF ITERATED X-VALUE BY INVERSE PARABOLIC INTERPOLATION
     17 A=FR-F
        DX=(X-XL)*FL*(1.+F*(A-TOL)/(A*(FR-FL)))/TOL
        XM=X
        FM=F
        X=XL-DX
        TOL=X
        F=FCT(TOL)
        IF(F)18,16,18
C
C       TEST ON SATISFACTORY ACCURACY IN ITERATION LOOP
     18 TOL=EPS
        A=DABS(X)
        IF(A-1.)20,20,19
     19 TOL=TOL*A
     20 IF(DABS(DX)-TOL)21,21,22
     21 IF(DABS(F)-TOLF)16,16,22
C
C       PREPARATION OF NEXT BISECTION LOOP
     22 IF(DSIGN(1.0D0,F)+DSIGN(1.0D0,FL))24,23,24
     23 XR=X
        FR=F
        GO TO 4
     24 XL=X
        FL=F
        XR=XM
        FR=FM
        GO TO 4
C       END OF ITERATION LOOP
C
C
C       ERROR RETURN IN CASE OF WRONG INPUT DATA
     25 IER=2
        RETURN
        END
        REAL FUNCTION PRCUM*8(X)
C***********************************************************************
C*                                                                     *
C* THIS SUBROUTINE EVALUATES THE PRIOR CUMULATIVE FUNCTION FOR THE     *
C* FAILURE RATE CASES. DISTRIBUTION PARAMETER VALUES AND FUNCTION DE-  *
C* FINITION INDEX ARE PASSED TO SUBROUTINE THROUGH THE COMMON BLOCK    *
C* /PRIR/.                                                             *
C*                                                                     *
C*   SUBROUTINES NEEDED:  GAMIC, MDBETA                                *
C*                                                                     *
C* SUBROUTINE ARGUMENTS:                                               *
C*     X       = FAILURE RATE VALUE AT WHICH DENSITY FUNCTION IS DESIRED *
C*                                                                     *
C* COMMON BLOCK /PRIR/ VARIABLES:                                      *
```

```
C*     PARM  = DISTRIBUTION PARAMETER VECTORS: PARM(1)=ALPHA, PARM(2)=   *
C*             BETA (=TAU FOR GAMMA), PARM(3)=MINIMUM LAMBDA PARAMETER   *
C*             OR SHIFT PARAMETER, PARM(4)=MAXIMUM LAMBDA PARAMETER      *
C*             (USED ONLY FOR THE BETA DISTRIBUTICN)                     *
C*     IPRIR = DISTRIBUTION INDEX; =1 GAMMA, =2 LOGNORMAL, =3 WEIBULL,   *
C*             =4 LOGBETA                                                *
C*                                                                       *
C* WRITTEN BY J.K. SHULTIS, 3/79.  MODIFIED 6/79, 7/79.                  *
C*                                                                       *
C*************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 XXX,AAA,BBB,PROB
      COMMON/PRIR/PARM(4),RNG(2),IPRIR
      REAL*8 X, YY(1), DLGAMA, DEXP, DLOG, DABS
      PRCUM=0.0D0
      GO TO (10,20,30,40),IPRIR
C*****
C***** GAMMA CUMULATIVE DISTRIBUTION
   10 IF(X.LE.PARM(3)) RETURN
      ARG=X-PARM(3)
      ALPHA=PARM(1)
      TAU=PARM(2)*ARG
      CALL GAMIC(TAU,ALPHA,1.D-7,1,YY,NZ)
      IF (NZ.NE.0) WRITE(6,11)
   11 FORMAT(' PROBABLE INACCURATE RESULT FOR THE GAMMA CUMULATIVE DISTR
     1IBUTION')
      PRCUM=YY(1)
      RETURN
C*****
C***** CUMULATIVE OF THE LOGNORMAL DISTRIBUTION
   20 IF(X.LE.PARM(3)) RETURN
      XL=(-DLOG(X-PARM(3))+PARM(1))/PARM(2)
      XLA=DABS(XL)/1.414213562373095
      A=DERF(XLA)
      PRCUM=0.5D0*(1.0D0-DSIGN(A,XL))
      RETURN
C*****
C***** WEIBULL CUMULATIVE DISTRIBUTION
   30 IF(X.LE.PARM(3)) RETURN
      PRCUM=1.0D0
      ARG=((X-PARM(3))/PARM(1))**PARM(2)
      IF(ARG.GT.36.0D0) RETURN
      PRCUM=1.0D0-DEXP(-ARG)
      RETURN
C*****
C***** CUMULATIVE OF THE LOGBETA DISTRIBUTION
   40 B=PARM(4)
      A=PARM(3)
      DL=DLOG(X)
      IF((B-DL)*(DL-A)) 41,41,42
   42 XXX=(DL-A)/(B-A)
      AAA=PARM(1)
      BBB=PARM(2)
      CALL MDBETA(XXX,AAA,BBB,PROB,IER)
      PRCUM=PROB
      RETURN
   41 IF(DL.LE.A) RETURN
      IF(DL.GE.B) PRCUM=1.D0
      RETURN
      END
```

```
      SUBROUTINE MDBETA(X, P, Q, PROB, IER)
C
C***********************************************************************
C*
C*   FUNCTION:       EVALUATE THE INCOMPLETE BETA DISTRIBUTION FUNCTION
C*
C*   PARAMETERS:
C*   X     - VALUE TO WHICH FUNCTION IS TO BE INTERGRATED. X MUST BE IN TH
C*           RANGE (0,1) INCLUSIVE.
C*   P     - INPUT (1ST) PARAMETER (MUST BE GREATER THAN 0)
C*   Q     - INPUT (2ND) PARAMETER (MUST BE GREATER THAN 0)
C*   PROB  - OUTPUT PROBABILITY THAT A RANDOM VARIABLE FROM A BETA DISTRIB
C*              HAVING PARAMETERS P AND Q WILL BE LESS THAN OR EQUAL TO Q.
C*   IER   - ERROR PARAMETER.
C*              IER = 0 INDICATES A NORMAL EXIT
C*              IER = 1 INDICATES THAT X IS NOT IN THE RANGE (0,1) INCLUSIV
C*              IER = 2 INDICATES THAT P AND/OR Q IS LESS THAN OR EQUAL TO
C*
C* CODE BASED ON SIMILAR CODE BY N. BOSTEN AND E.BATTISTE AS MODIFIED BY
C* M. PIKE AND J. HOO.
C*
C***********************************************************************
      DOUBLE PRECISION PS, PX, Y, P1, DP, INFSUM, CNT, WH, XB,
     * DQ, C, EPS, EPS1, ALEPS, FINSUM, PQ, DA, DLGAMA
C DOUBLE PRECISION FUNCTION DLGAMA
C MACHINE PRECISION
      DATA EPS/1.D-6/
C SMALLEST POSITIVE NUMBER REPRESENTABLE
      DATA EPS1/1.D-78/
C NATURAL LOG OF EPS1
      DATA ALEPS/-179.6016D0/
C CHECK RANGES OF THE ARGUMENTS
      Y = X
      IF ((X.LE.1.0) .AND. (X.GE.0.0)) GO TO 10
      IER = 1
      GO TO 140
   10 IF ((P.GT.0.0) .AND. (Q.GT.0.0)) GO TO 20
      IER = 2
      GO TO 140
   20 IER =0
      IF (X.GT.0.5) GO TO 30
      INT = 0
      GO TO 40
C SWITCH ARGUMENTS FOR MORE EFFICIENT USE OF THE POWER
C SERIES
   30 INT = 1
      TEMP = P
      P = Q
      Q = TEMP
      Y = 1.D0 - Y
   40 IF (X.NE.0. .AND. X.NE.1.) GO TO 60
C SPECIAL CASE - X IS 0. OR 1.
   50 PROB = 0.
      GO TO 130
   60 IB = Q
      TEMP = IB
      PS = Q - FLOAT(IB)
      IF (Q.EQ.TEMP) PS = 1.D0
      DP = P
      PQ=Q
```

```
      PX = DP*DLOG(Y)
      PQ = DLGAMA(DP+DQ)
      P1 = DLGAMA(DP)
      C = DLGAMA(DQ)
      D4 = DLOG(DP)
C DLGAMA IS A FUNCTION WHICH CALCULATES THE DOUBLE
C PRECISION LOG GAMMA FUNCTION
      XB = PX + DLGAMA(PS+DP) - DLGAMA(PS) - D4 - P1
C SCALING
      IB = XB/ALEPS
      INFSUM = 0.D0
C FIRST TERM OF A DECREASING SERIES WILL UNDERFLOW
      IF (IB.NE.0) GO TO 90
      INFSUM = DEXP(XB)
      CNT = INFSUM*DP
C CNT WILL EQUAL DEXP(TEMP)*(1.D0-PS)I*P*Y**I/FACTORIAL(I)
      WH = 0.0D0
   80 WH = WH + 1.D0
      CNT = CNT*(WH-PS)*Y/WH
      IF (CNT.LT.EPS1/EPS) GOTO 90
      XB = CNT/(DP+WH)
      INFSUM = INFSUM + XB
      IF (XB/EPS.GT.INFSUM) GO TO 80
C DLGAMA IS A FUNCTION WHICH CALCULATES THE DOUBLE
C PRECISION LOG GAMMA FUNCTION
   90 FINSUM = 0.D0
      IF (DQ.LE.1.D0) GO TO 120
      XB = PX+ DQ*DLOG(1.D0-Y) + PQ - P1 - DLOG(DQ) - C
C SCALING
      IB = XB/ALEPS
      IF (IB.LT.0) IB = 0
      C = 1.D0/(1.D0-Y)
      CNT = DEXP(XB-FLOAT(IB)*ALEPS)
      PS = DQ
      WH = DQ
  100 WH =WH -1.D0
      IF (WH.LE.0.0D0) GO TO 120
      PX = (PS*C)/ (DP+WH)
      IF (PX.GT.1.0D0) GO TO 105
      IF (CNT/EPS.LE.FINSUM.OR.CNT.LE.EPS1/PX) GO TO 120
  105 CNT =CNT*PX
      IF (CNT.LE.1.D0) GO TO 110
C RESCALE
      IB = IB - 1
      CNT = CNT*EPS1
  110 PS =WH
      IF (IB.EQ.0) FINSUM = FINSUM + CNT
      GO TO 100
  120 PROB =FINSUM + INFSUM
  130 IF (INT.EQ.0) GO TO 140
      PROB = 1.0 - PROB
      TEMP = P
      P = Q
      Q = TEMP
  140 RETURN
      END
      SUBROUTINE GAMIC(X,ALPHA,REL,N,Y,NZ)
C***********************************************************************
C
C     WRITTEN BY D.E. AMOS AND S.L. DANIEL, NOVEMBER, 1974.
```

```
C      EXTRACTED FROM THE SANDIA LABORATORY'S MATHEMATICAL
C      LIBRARY AND ADAPTED FOR THE IBM 370 SYSTEM BY J. K. SHULTIS
C      JULY, 1979.
C
C      REFERENCE SC-DR-72 0303
C
C      ABSTRACT
C          GAMIC COMPUTES AN N MEMBER SEQUENCE OF INCOMPLETE GAMMA
C          FUNCTIONS NORMALIZED SO THAT AT X=INFINITY, THE INCOMPLETE
C          GAMMA FUNCTION HAS THE VALUE 1. THE SEQUENCE IS DENOTED BY
C
C          Y(K)=INCGAMMA(ALPHA+K-1,X)/GAMMA(ALPHA+K-1),  K=1,2,...,N
C
C          AND IS COMPUTED TO A RELATIVE ERROR REL OR BETTER WHERE ALPHA
C          .GT.0. IF ALPHA+N-1.GE.X, THE LAST MEMBER IS COMPUTED BY THE
C          CONFLUENT HYPERGEOMETRIC SERIES WITH THE OTHER MEMBERS
C          COMPUTED BY BACKWARD RECURSION ON A TWO-TERM FORMULA,
C
C            Y(K-1)=Y(K)+DEXP((ALPHA+K-1)*DLOG(X)-X-DLGAMA(ALPHA+K)).
C
C          IF ALPHA+N-1.LT.X, AN INTEGER M IS ADDED SO THAT
C          ALPHA+N-1+M.GE.X AND THE FIRST PROCEDURE IS APPLIED. SPECIAL
C          PROCEDURES APPLY FOR ALPHA.EQ.1 OR AN UNDERFLOW OCCURS OR
C          X EXCEEDS A CRITICAL VALUE, APTEST, WHERE ALL MEMBERS ARE 1.
C          TO THE WORD LENGTH OF THE CDC 6600. GAMIC USES DLGAMA.
C
C      DESCRIPTION OF ARGUMENTS
C
C          INPUT
C            X        - ARGUMENT, X.GE.0.0
C            ALPHA    - PARAMETER, ALPHA.GT.0.0
C            REL      - RELATIVE ERROR TOLERANCE, REL=1.E-S FOR S
C                         SIGNIFICANT DIGITS
C            N        - NUMBER OF GAMMA FUNCTIONS IN THE SEQUENCE
C                         BEGINNING AT PARAMETER ALPHA, N.GE.1
C
C          OUTPUT
C            Y        - A VECTOR CONTAINING AN N MEMBER SEQUENCE
C                         Y(K)= INCGAMMA(ALPHA+K-1,X)/GAMMA(ALPHA+K-1),
C                         K=1,...,N TO A RELATIVE ERROR REL.
C            NZ       - UNDERFLOW FLAG
C                         NZ.EQ.0, A NORMAL RETURN
C                         NZ.NE.0, UNDERFLOW, Y(K)=0.0, K=N-NZ+1,N RETURNED
C
C      ERROR CONDITIONS
C          IMPROPER INPUT PARAMETERS - A FATAL ERROR
C          UNDERFLOW - A NON-FATAL ERROR.
C
C**************************************************************************
       IMPLICIT REAL*8(A-H,O-Z)
       REAL*8 X ALPHA,REL,Y,DLOG,DEXP,DFLOAT
       DIMENSION Y(1)
       DIMENSION AA(6),BB(6),CC(5)
C
       DATA   AA             / 1.18399941922176D+00, 3.30888136276361D+02,
      1 1.04930832947926D+04, 3.78420325596908D+04, 1.57586618187374D+02,
      2 1.30569632410551D+03/
       DATA   BB             / 1.02652821626751D+00, 9.29753107520368D+03,
      1 6.53848923630220D+06, 2.89543295992889D+08, 8.16836456953161D+03,
      2 4.12237656364399D+06/
```

```
      DATA   CC               / 4.30952856710482D+05, 8.27988256743362D+09,
     1 2.41944468684445D+12, 4.21722873236008D+05, 7.56593802747116D+09/
      DATA SLOG/-160.0D0/
      DATA  CON14             / 3.32361913C191650+1/
      DATA  SCALE             /1.D-60/
      DATA  ASCL              /1.D-16/
C
      IF(REL.LE.0.) GO TO 91
      IF(N.LT.1) GO TO 92
      IF(ALPHA.LE.0.0) GO TO 93
      NZ=0
      IF(X) 94,10,20
   10 DO 11 I=1,N
   11 Y(I)=0.
      RETURN
C
C     IF X.GE.XLIM(ALPHA+N-1), THEN Y(K)=1., K=1,2,...,N
C
   20 NN=N
      RX=1./X
      NBAR=0
      APN=ALPHA+DFLOAT(N)-1.
      APTEST=APN
      AM1=ALPHA-1.
      IF(APN.LE.1.) GO TO 22
   21 IF(X.LE.APTEST) GO TO 40
      IF(APTEST.GT.200.) GO TO 25
      S1=((AA(1)*APTEST+AA(2))*APTEST+AA(3))*APTEST+AA(4)
      S2=(APTEST+AA(5))*APTEST+AA(6)
      GO TO 226
   25 IF(APTEST.GT.10000.) GO TO 36
      S1=((BB(1)*APTEST+BB(2))*APTEST+BB(3))*APTEST+BB(4)
      S2=(APTEST+BB(5))*APTEST+BB(6)
  226 XLIM=S1/S2
C
      IF(X.GE.XLIM) GO TO 26
      GO TO 35
   36 S1=((APTEST+CC(1))*APTEST+CC(2))*APTEST+CC(3)
      S2=(APTEST+CC(4))*APTEST+CC(5)
      GO TO 226
   26 DO 27 I=1,N
   27 Y(I)=1.
      RETURN
C
   22 IF(ALPHA.NE.1.) GO TO 32
      IF(X.GT.CON14) GO TO 26
      IF(X.LT.0.1) GO TO 40
      Y(NN)=1.-DEXP(-X)
      RETURN
   32 NBAR=1
      APTEST=APN+1.
      GO TO 21
   35 NBAR=X-APTEST+5.+DFLOAT(NBAR)
   40 ABAR=APN+DFLOAT(NBAR)
      XLOG=DLOG(X)
      A1=1.
      SUM=1.
      ABK=ABAR+1.
   80   A1 = A1*X/ABK
      SUM = SUM + A1
```

```
      IF(A1.LT.REL) GO TO 100
      ABK=ABK+1.
      GO TC 80
  100 YY=SUM*SCALE
      D=ABAR
      IF(NBAR.EQ.0) GO TO 110
  105 CONTINUE
      DO 106 K=1,NBAR
      XOD=X/D
      IF(XOD.LT.ASCL) GO TO 106
      YY=XOD*YY+SCALE
  106 D=D-1.
C
      IF(NZ.NE.0) GO TO 114
 110  E= -X+D*XLOG-DLGAMA(D+1.)
  114 IF(E.GE.SLOG) GO TO 120
      Y(NN)=0.
      NZ=NZ+1
      NN=NN-1
      IF(NN.EQ.0) RETURN
      NBAR=1
      APN=APN-1.
      E=E+DLOG(D*RX)
      GO TC 105
 120  EXE = DEXP(E)
      Y(NN)=(EXE/SCALE)*YY
      NM1=NN-1
      IF(NM1.EQ.0) RETURN
      F=EXE*APN*RX
      KK = NN
      AK=DFLOAT(NN)+AM1
      DO 125 K=1,NM1
      Y(KK-1)=Y(KK)+F
      KK=KK-1
      AK=AK-1.
      F=F*AK*RX
  125 CONTINUE
      RETURN
C
   91 WRITE(6,191)
  191 FORMAT('IN GAMIC, IMPROPER INPUT FOR REL.')
      RETURN
   92 WRITE(6,192)
  192 FORMAT('IN GAMIC, IMPROPER INPUT FOR N.')
      RETURN
   93 WRITE(6,193)
  193 FORMAT('IN GAMIC, IMPROPER INPUT FOR ALPHA.')
      RETURN
   94 WRITE(6,194)
  194 FORMAT('IN GAMIC, IMPROPER INPUT FOR X.')
      RETURN
      END
C***************************************************************************
C*     DXP FUNCTION
C***************************************************************************
C*
C*  THIS FUNCTION ERROR CHECKS THEN EVALATES DEXP(X).
C*
C*
C*
```

```
      REAL FUNCTION DXP*8(X)
      REAL*8 DEXP,X,DABS
      IF (DABS(X).GT.170.0D0) GOTO 10
      DXP=DEXP(X)
      RETURN
   10 IF (X) 20,20,30
   20 DXP=0.0D0
      RETURN
   30 DXP=1.0D70
      RETURN
      END
C
C
C***********************************************************************
C*
C*                              SGN
C*
C***********************************************************************
C
C
C    THIS FUNCTION RETURNS A NEGATIVE 1.0D0 FOR NEGATIVE X, OR A POSITIVE
C    1.0D0 FOR VALUES OF X GREATER THAN OR EQUAL TO 0.0D0.
C
C
      REAL FUNCTION SGN*8 (X)
      REAL*8 X
      IF (X) 100,200,200
 100  SGN=-1.0D0
      RETURN
 200  SGN=1.0D0
      RETURN
      END
```

The following is a program for generating power curves for several methods of detecting a single outlier in (F_i,T) data by simulation with the parameters alpha and beta calculated by the prior matching moments method (PMMM).

In the driving program:

NDATA = number of data points per data set less one

NSETS = number of data sets analyzed

PRSIM(J,L) = points in the power curve

The subroutine 'RANDU' is for generating a random number between 0 and 1, and comes from IBM Manual C20-8011.

The subroutine 'RF' is for generating a random F_i from a random number between 0 and 1.

The subroutine 'HCUM' is for calculating the cumulative distribution of the compound model (See Eq. 3.19).

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SNGL
      COMMON /CUMH/ A,B,T,NDATA
      COMMON /FGEN/ ALPH(25),BET(25),T1,IX,IY
      DIMENSION F(1000,50),PRSIM(5,25),A1(1000),B1(1000),A2(1000),
     1     B2(1000),RLAM(25),NFAIL(25)
C
C *** INITIALIZE MAIN PROGRAM
C
      NDATA=19
      NDATA1=NDATA+1
      NSETS=263
      T=10000.0D0
      T1=T
C
C *** INITIALIZE SUBPROGRAM 'RANDU'
C
      IX=384766237
      DO 1 I=1,43
      CALL RANDU(IX,IY,YFL)
    1 IX=IY
C
C *** SET VALUES OF THE GAMMA PARAMETERS ALPHA AND BETA
C
      DO 100 IA=1,5
      IF(IA.EQ.1)ALPHA=1.0D0
      IF(IA.EQ.2)ALPHA=1.25D00
      IF(IA.EQ.3)ALPHA=1.5D00
      IF(IA.EQ.4)ALPHA=2.0D0
      IF(IA.EQ.5)ALPHA=4.0D0
      DO 100 IB=1,5
      BETA=ALPHA*10.0D0**(3.5D00+DFLOAT(IB)*0.5D00)
C
C *** READ AND OUTPUT DATA
C
      READ(9,300) ((F(J,I),I=1,NDATA),J=1,NSETS)
      READ(9,300) DUMMY
      WRITE(6,301)
      WRITE(6,302) (J,(F(J,I),I=1,NDATA),J=10,NSETS,10)
C
C *** SET VALUE OF LAMBDA AND CORRESPONDING VALUES OF GAMMA PARAMETERS
C *** ALPHAI AND BETAI
C
      DO 18 L=1,15
   18 NFAIL(L)=0
      VAR=ALPHA/BETA**2
      DO 2 L=1,15
      RLAM(L)=DEXP((DLOG(BETA)+DLOG(5.0D0-ALPHA)-6.8D00)/1.4D00)*(1.0D0
     1 +(ALPHA-4.0D0)*(BETA-ALPHA*1.0D4)/(BETA*(-6.0D0)))*DFLOAT(L)
     2 /15.0D0
      IF(IB.EQ.2)RLAM(L)=RLAM(L)/(1.0D0+(4.0D0-ALPHA)*0.1D00)
      IF(IB.EQ.3)RLAM(L)=RLAM(L)/(1.0D0+(4.0D0-ALPHA)*0.15D00)
      IF(IB.EQ.4)RLAM(L)=RLAM(L)/(1.0D0+(4.0D0-ALPHA)*0.5D00)
      AVG=RLAM(L)*ALPHA/BETA
      ALPH(L)=AVG**2/VAR
```

```
      BETI(L)=AVG/VAR
C
C *** INITIALIZE PROBABILITIES, GENERATE OUTLIER, AND FIND MAXIMUM
C *** NUMBER CF FAILURES
C
      DO 3 I=1,5
    3 PRSIM(I,L)=0.0D0
      DO 4 J=1,NSETS
      FS=RF(L)
      FMD=0.0D0
      DO 5 I=1,NDATA
      IF(F(J,I).GT.FMD)FMD=F(J,I)
    5 CONTINUE
      FM=FMD
      IF(FS.GT.FM)FM=FS
      IF(FM.EQ.0.0D0)GO TO 4
C
C *** TRY TO DETECT OUTLIER WITH GAMMA PARAMETERS ALPHA AND BETA KNOWN
C
      A=ALPHA
      B=BETA
      PHCUM=HCUM(FM)**NDATA1
      IF(PHCUM.GE.0.95000)PRSIM(1,L)=PRSIM(1,L)+1.0DC/DFLOAT(NSETS)
C
C *** TRY TO DETECT OUTLIER WITH GAMMA PARAMETERS ALPHA AND BETA
C *** CALCULATED USING POTENTIAL OUTLIER
C
      SUM=FS/T
      DO 6 I=1,NDATA
    6 SUM=SUM+F(J,I)/T
      AVG1=SUM/DFLOAT(NDATA1)
      VAR1=(FS/T-AVG1)**2
      DO 7 I=1,NDATA
    7 VAR1=VAR1+(F(J,I)/T-AVG1)**2
      VAR1=VAR1/DFLOAT(NDATA)
      A1(J)=AVG1**2/VAR1
      B1(J)=AVG1/VAR1
      A=A1(J)
      B=B1(J)
      PHCUM=HCUM(FM)**NDATA1
      IF(PHCUM.GE.0.95000)PRSIM(2,L)=PRSIM(2,L)+1.0D0/DFLOAT(NSETS)
C
C *** TRY TO DETECT OUTLIER WITH GAMMA PARAMETERS ALPHA AND BETA
C *** CALCULATED WITHOUT USING POTENTIAL OUTLIER
C
      AVG2=(DFLOAT(NDATA1)*AVG1-FM/T)/DFLOAT(NDATA)
      VAR2=0.0D0
      DO 8 I=1,NDATA
    8 VAR2=VAR2+(F(J,I)/T-AVG2)**2
      IF(FM.GT.FS)VAR2=VAR2-(FM/T-AVG2)**2+(FS/T-AVG2)**2
      VAR2=VAR2/DFLOAT(NDATA-1)
      IF(VAR2.EQ.0.0D0)GO TO 15
      A2(J)=AVG2**2/VAR2
      B2(J)=AVG2/VAR2
      A=A2(J)
      B=B2(J)
      IF(B.LE.0.0D0)GO TO 15
      PHCUM=HCUM(FM)**NDATA1
      IF(PHCUM.GE.0.95000)PRSIM(3,L)=PRSIM(3,L)+1.0DC/DFLOAT(NSETS)
```

```
      GO TO 16
   15 A2(J)=0.0D0
      B2(J)=0.0D0
      NFAIL(L)=NFAIL(L)+1
C
C *** TRY TO DETECT OUTLIER USING BINOMIAL METHOD AND COMPOUND MODEL
C
   16 NDP=NDATA
      NF1=IFIX(SNGL(FMD+1.0D0))
      IF(FS.GT.FMD)GO TO 9
      FMD1=0.0D0
      NDP=0
      F1=0.0D0
      DO 10 I=1,NDATA
      IF(F(J,I).LT.FM)NDP=NDP+1
      IF(F(J,I).GT.F1.AND.F(J,I).LT.FM)F1=F(J,I)
   10 CONTINUE
      IF(FS.LT.FM)NDP=NDP+1
      IF(FS.GT.F1.AND.FS.LT.FM)F1=FS
      NF1=IFIX(SNGL(F1+1.0D0))
    9 P=0.0D0
      DO 11 I=1,NF1
      FP=DFLOAT(I-1)
      HM=DLGAMA(FP+A1(J))+FP*DLOG(T)+A1(J)*DLOG(B1(J))-DLGAMA(A1(J))
     1     -DLGAMA(FP+1.0D0)-(FP+A1(J))*DLOG(T+B1(J))
   11 P=P+DEXP(HM)
      CD2=DFLOAT(NDATA+2)
      PXF=0.0D0
      DO 12 I=NDP,NDATA1
      X=DFLOAT(I)
      R=DLGAMA(CD2)-DLGAMA(X+1.0D0)-DLGAMA(CD2-X)+X*DLOG(P)
     1     +(CD2-X-1.0D0)*DLOG(1.0D0-P)
   12 PXF=PXF+DEXP(R)
      IF(PXF.LE.0.05D00)PRSIM(4,L)=PRSIM(4,L)+1.0D0/DFLOAT(NSETS)
C
C *** TRY TO DETECT OUTLIER USING BINOMIAL MEHTOD AND HOMOGENEOUS MODEL
C
      P=0.0D0
      DO 13 I=1,NF1
      FP=DFLOAT(I-1)
      POI=FP*DLOG(AVG1*T)-AVG1*T-DLGAMA(FP+1.0D0)
   13 P=P+DEXP(POI)
      PXF=0.0D0
      DO 14 I=NDP,NDATA1
      X=DFLOAT(I)
      R=DLGAMA(CD2)-DLGAMA(X+1.0D0)-DLGAMA(CD2-X)+X*DLOG(P)
     1     +(CD2-X-1.0D0)*DLOG(1.0D0-P)
   14 PXF=PXF+DEXP(R)
      IF(PXF.LT.0.05D00)PRSIM(5,L)=PRSIM(5,L)+1.0D0/DFLOAT(NSETS)
    4 CONTINUE


    2 CONTINUE
C
C *** OUTPUT RESULTS
C
      WRITE(6,303) ALPHA,BETA
      WRITE(6,304)
      WRITE(6,305) (RLAM(L),ALPH(L),BET(L),(PRSIM(J,L),J=1,5),L=1,15)
```

```
      WRITE(6,306)
      DO 17 L=1,15
      IF(NFAIL(L).EQ.0)GO TO 17
      WRITE(6,307) RLAM(L),NFAIL(L)
   17 CONTINUE
  100 CONTINUE
C
C *** FORMATS
C
  300 FORMAT(100(100F3.0/))
  301 FORMAT('1',4X,'DATA *',/,5X,'SET   *',51X,'DATA',/,4X,6('-'),
     1       '*',100('-'))
  302 FORMAT(' ',4X,250(I3,2X,'*',19F5.0,/,10X,'*',/,5X))
  303 FORMAT('1',4X,'SIMULATED PROBABILITY THAT A DATA POINT GENERATED F
     1ROM THE GAMMA DISTRIBUTION WITH PARAMETERS',/,5X,'ALPHA AND BETA W
     2ILL BE DETECTED AS AN OUTLIER IN THE GAMMA DISTRIBUTION WITH PARAM
     3ETERS',/,5X,'ALPHA =',F6.3,1X,'AND BETA =',D11.4)
  304 FORMAT('-',8X,'*',2(12X,'*'),11X,'INTEGRATION METHOD',12X,'*',
     1  6X,'BINOMIAL METHOD',/,9X,'*',12X,'*',12X,'*  PARAMETERS ',
     2  2('*  PARMS CALC '),'*   COMPOUND   * HOMOGENEOUS',/,'  LAMBDA *
     3   ALPHA    *     BETA     *     KNOWN     *  W/ OUTLIER * W/O OUTLIER
     4',2('*     MODEL     '),/,1X,8('-'),'*',12('-'),'*',12('-'),5('*',
     5  13('-')))
  305 FORMAT(' ',25(F7.3,' *',D11.4,' *',D11.4,1X,5('*',F10.5,3X),/,1X))
  306 FORMAT('-')
  307 FORMAT(' ','FOR LAMBDA =',F8.3,'THE INTEGRATION METHOD (PARMS CALC
     1 W/O OUTLIER) FAILED',I5,' TIMES')
      STOP
      END
```

```
      REAL FUNCTION HCUM*8(FM)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SNGL
      COMMON /CUMH/ A,B,T,NDATA
      NFM1=IFIX(SNGL(FM))
      HCUM=0.0D0
      DO 1 I=1,NFM1
      F=DFLOAT(I-1)
      H=DLGAMA(F+A)+F*DLOG(T)+A*DLOG(B)-DLGAMA(A)-DLGAMA(F+1.0D0)
     1      -(F+A)*DLOG(T+B)
    1 HCUM=HCUM+DEXP(H)
      RETURN
      END
```

```
      REAL FUNCTION RF*8(L)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /FGEN/ ALPH(25),BET(25),T,IX,IY
      DIMENSION XC(1001)
      XC(1)=0.0D0
      A=ALPH(L)
      B=BET(L)
      J=2
      DO 1 I=1,1000
      F=DFLOAT(I-1)
      P=DLGAMA(F+A)+F*DLOG(T)+A*DLOG(B)-DLGAMA(A)-DLGAMA(F+1.0D0)
     1      -(F+A)*DLOG(T+B)
      XC(I+1)=XC(I)+DEXP(P)
      IF(XC(I+1).GE.0.9999D00)GO TO 2
    1 J=J+1
    2 CALL RANDU(IX,IY,YFL)
      IX=IY
      IF(YFL.GT.0.9999D00)GO TO 3
      IF(YFL.GT.XC(J))GO TO 3
      DO 4 I=2,J
      IF(YFL.LE.XC(I))RF=DFLOAT(I-2)
      IF(YFL.LE.XC(I))GO TO 5
    4 CONTINUE
      GO TO 5
    3 RF=DFLOAT(J-1)
    5 RETURN
      END
```

```
      SUBROUTINE RANDU(IX,IY,YFL)
      REAL*8 YFL
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
```

The following is a program for generating power curves for several methods of detecting a single outlier in (F_i,T) data by simulation with the parameters alpha and beta calculated by the marginal matching moments method (MMMM).

In the driving program:

NDATA = number of data points per data set less one

NSETS = number of data sets analyzed

PRSIM(J,L) = points in the power curve

The subroutine 'RANDU' is for generating a random number between 0 and 1, and comes from IBM Manual C20-8011.

The subroutine 'RF' is for generating a random F_i from a random number between 0 and 1.

The subroutine 'HCUM' is for calculating the cumulative distribution of the compound model (See Ep. 3.19).

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SNGL
      COMMON /CUMH/ A,B,T,NDATA
      COMMON /FGEN/ ALPH(25),BET(25),T1,IX,IY
      DIMENSION F(1000,50),PRSIM(5,25),A1(1000),B1(1000),A2(1000),
     1     B2(1000),RLAM(25),NFAIL(25),NFLR(5,25)
C
C *** INITIALIZE MAIN PROGRAM
C
      NDATA=19
      NDATA1=NDATA+1
      NSETS=263
      T=10000.0D0
      T1=T
C
C *** INITIALIZE SUBPROGRAM 'RANDU'
C
      IX=384766237
      DO 1 I=1,43
      CALL RANDU(IX,IY,YFL)
    1 IX=IY
C
C *** SET VALUES OF THE GAMMA PARAMETERS ALPHA AND BETA
C
      DO 100 IA=1,5
      IF(IA.EQ.1)ALPHA=1.0D0
      IF(IA.EQ.2)ALPHA=1.25D0
      IF(IA.EQ.3)ALPHA=1.50D0
      IF(IA.EQ.4)ALPHA=2.0D0
      IF(IA.EQ.5)ALPHA=4.0D0
      DO 100 IB=1,5
      BETA=ALPHA*10.0D0**(3.50D0+DFLOAT(IB)*0.50D0)
C
C *** READ AND OUTPUT DATA
C
      READ(5,300) ((F(J,I),I=1,NDATA),J=1,NSETS)
      READ(5,300) DUMMY
      WRITE(6,301)
      WRITE(6,302) (J,(F(J,I),I=1,NDATA),J=10,NSETS,10)
C
C *** SET VALUE OF LAMBDA AND CORRESPONDING VALUES OF GAMMA PARAMETERS
C *** ALPHAI AND BETAI
C
      DO 22 J=1,5
      DO 22 L=1,15
   22 NFLR(J,L)=0
      DO 18 L=1,15
   18 NFAIL(L)=0
      VAR=ALPHA/BETA**2
      DO 2 L=1,15
      RLAM(L)=DEXP((DLOG(BETA)+DLOG(5.0D0-ALPHA)-6.80D0)/1.40D0)*(1.0D0
     1  +(ALPHA-4.0D0)*(BETA-ALPHA*1.0D4)/(BETA*(-6.0D0)))*DFLOAT(L)
     2  /15.0D0
      IF(IB.EQ.2)RLAM(L)=RLAM(L)/(1.0D0+(4.0D0-ALPHA)*0.10D0)
      IF(IB.EQ.3)RLAM(L)=RLAM(L)/(1.0D0+(4.0D0-ALPHA)*0.150D0)
```

```
      IF(IB.EQ.4)RLAM(L)=RLAM(L)/(1.0D0+(4.0D0-ALPHA)*0.5D00)
      AVG=RLAM(L)*ALPHA/BETA
      ALPH(L)=AVG**2/VAR
      BET(L)=AVG/VAR
C
C *** INITIALIZE PROBABILITIES, GENERATE OUTLIER, AND FIND MAXIMUM
C *** NUMBER OF FAILURES
C
      DO 3 I=1,5
    3 PRSIM(I,L)=0.0D0
      DO 4 J=1,NSETS
      FS=RF(L)
      FMD=0.0D0
      DO 5 I=1,NDATA
      IF(F(J,I).GT.FMD)FMD=F(J,I)
    5 CONTINUE
      FM=FMD
      IF(FS.GT.FM)FM=FS
      IF(FM.EQ.0.0D0)GO TO 4
C
C *** TRY TO DETECT OUTLIER WITH GAMMA PARAMETERS ALPHA AND BETA KNOWN
C
      A=ALPHA
      B=BETA
      PHCUM=HCUM(FM)**NDATA1
      IF(PHCUM.GE.0.95D00)PRSIM(1,L)=PRSIM(1,L)+1.0D0/DFLOAT(NSETS)
C
C *** TRY TO DETECT OUTLIER WITH GAMMA PARAMETERS ALPHA AND BETA
C *** CALCULATED USING POTENTIAL OUTLIER
C
      SUM=FS/T
      DO 6 I=1,NDATA
    6 SUM=SUM+F(J,I)/T
      AVG1=SUM/DFLOAT(NDATA1)
      VAR1=(FS/T-AVG1)**2
      DO 7 I=1,NDATA
    7 VAR1=VAR1+(F(J,I)/T-AVG1)**2
      VAR1=VAR1/DFLOAT(NDATA)
      C=VAR1-AVG1/T
      IF(C.LE.1.0D-50)GO TO 21
      A1(J)=AVG1**2/(VAR1-AVG1/T)
      B1(J)=AVG1/(VAR1-AVG1/T)
      A=A1(J)
      B=B1(J)
      PHCUM=HCUM(FM)**NDATA1
      IF(PHCUM.GE.0.95D00)PRSIM(2,L)=PRSIM(2,L)+1.0D0/DFLOAT(NSETS)
      GO TO 19
   21 A1(J)=0.0D0
      B1(J)=0.0D0
      NFLR(2,L)=NFLR(2,L)+1
      NFLR(4,L)=NFLR(4,L)+1

C
C *** TRY TO DETECT OUTLIER WITH GAMMA PARAMETERS ALPHA AND BETA
C *** CALCULATED WITHOUT USING POTENTIAL OUTLIER
C
   19 AVG2=(DFLOAT(NDATA1)*AVG1-FM/T)/DFLOAT(NDATA)
      VAR2=0.0D0
      DO 8 I=1,NDATA
```

```
    8 VAR2=VAR2+(F(J,I)/T-AVG2)**2
      IF(FM.GT.FS)VAR2=VAR2-(FM/T-AVG2)**2+(FS/T-AVG2)**2
      VAR2=VAR2/DFLOAT(NDATA-1)
      IF(VAR2.EQ.0.0D0)GO TO 15
      C=VAR2-AVG2/T
      IF(C.LE.1.0D-50)GO TO 20
      A2(J)=AVG2**2/(VAR2-AVG2/T)
      B2(J)=AVG2/(VAR2-AVG2/T)
      A=A2(J)
      B=B2(J)
      IF(B.LE.0.0D0)GO TO 15
      PHCUM=HCUM(FM)**NDATA1
      IF(PHCUM.GE.0.95D00)PRSIM(3,L)=PRSIM(3,L)+1.0D0/DFLOAT(NSETS)
      GO TO 16
   15 A2(J)=0.0D0
      B2(J)=0.0D0
      NFAIL(L)=NFAIL(L)+1
      GO TO 16
   20 A2(J)=0.0D0
      B2(J)=0.0D0
      NFLR(3,L)=NFLR(3,L)+1
C
C *** TRY TO DETECT OUTLIER USING BINOMIAL METHOD AND COMPOUND MODEL
C
   16 NOP=NDATA
      IF(A1(J).EQ.0.0D0)GO TO 4
      NF1=IFIX(SNGL(FM0+1.0D0))
      IF(FS.GT.FM0)GO TO 9
      FM01=0.0D0
      NOP=0
      F1=0.0D0
      DO 10 I=1,NDATA
      IF(F(J,I).LT.FM)NOP=NOP+1
      IF(F(J,I).GT.F1.AND.F(J,I).LT.FM)F1=F(J,I)
   10 CONTINUE
      IF(FS.LT.FM)NOP=NOP+1
      IF(FS.GT.F1.AND.FS.LT.FM)F1=FS
      NF1=IFIX(SNGL(F1+1.0D0))
    9 P=0.0D0
      DO 11 I=1,NF1
      FP=DFLOAT(I-1)
      HM=DLGAMA(FP+A1(J))+FP*DLOG(T)+A1(J)*DLOG(B1(J))-DLGAMA(A1(J))
     1     -DLGAMA(FP+1.0D0)-(FP+A1(J))*DLOG(T+B1(J))
   11 P=P+DEXP(HM)
      CD2=DFLOAT(NDATA+2)
      PXF=0.0D0
      DO 12 I=NOP,NDATA1
      X=DFLOAT(I)
      R=DLGAMA(CD2)-DLGAMA(X+1.0D0)-DLGAMA(CD2-X)+X*DLOG(P)
     1     +(CD2-X-1.0D0)*DLOG(1.0D0-P)
   12 PXF=PXF+DEXP(R)
      IF(PXF.LE.0.05D00)PRSIM(4,L)=PRSIM(4,L)+1.0D0/DFLOAT(NSETS)
C
C *** TRY TO DETECT OUTLIER USING BINOMIAL MEHTOD AND HOMOGENEOUS MODEL
C
      P=0.0D0
      DO 13 I=1,NF1
      FP=DFLOAT(I-1)
      PO1=FP*DLOG(AVG1*T)-AVG1*T-DLGAMA(FP+1.0D0)
```

```
   13 P=P+DEXP(POI)
      PXF=0.000
      DO 14 I=NOP,NDATA1
      X=DFLOAT(I)
      R=DLGAMA(CD2)-DLGAMA(X+1.0D0)-DLGAMA(CD2-X)+X*DLOG(P)
     1       +(CD2-X-1.0D0)*DLOG(1.0D0-P)
   14 PXF=PXF+DEXP(R)
      IF(PXF.LT.0.05D00)PRSIM(5,L)=PRSIM(5,L)+1.0D0/DFLOAT(NSETS)
    4 CONTINUE
    2 CONTINUE
C
C *** OUTPUT RESULTS
C
      N=0
   24 WRITE(6,303) ALPHA,BETA
      IF(N.EQ.1)WRITE(6,308)
      WRITE(6,304)
      WRITE(6,305) (RLAM(L),ALPH(L),BET(L),(PRSIM(J,L),J=1,5),L=1,15)
      WRITE(6,306)
      DO 17 L=1,15
      IF(NFAIL(L).EQ.0)GO TO 17
      WRITE(6,307) RLAM(L),NFAIL(L)
   17 CONTINUE
      IF(N.GT.0)GO TO 100
      N=1
      DO 23 J=1,5
      DO 23 L=1,15
      IF(J.EQ.3)GO TO 25
      IF(NFLR(J,L).EQ.NSETS)GO TO 23
      PRSIM(J,L)=PRSIM(J,L)*DFLOAT(NSETS)/DFLOAT(NSETS-NFLR(J,L))
      GO TO 23
   25 N1=NFLR(J,L)+NFAIL(L)
      IF(N1.EQ.NSETS)GO TO 23
      PRSIM(J,L)=PRSIM(J,L)*DFLOAT(NSETS)/DFLOAT(NSETS-N1)
   23 CONTINUE
      GO TO 24
  100 CONTINUE
C
C *** FORMATS
C
  300 FORMAT(100(100F3.0/))
  301 FORMAT('1',4X,'DATA *',/,5X,'SET   *',51X,'DATA',/,4X,6('-'),
     1       '*',100('-'))
  302 FORMAT(' ',4X,250(I3,2X,'*',19F5.0,/,10X,'*',/,5X))
  303 FORMAT('1',4X,'SIMULATED PROBABILITY THAT A DATA POINT GENERATED F
     1ROM THE GAMMA DISTRIBUTION WITH PARAMETERS',/,5X,'ALPHA AND BETA W
     2ILL BE DETECTED AS AN OUTLIER IN THE GAMMA DISTRIBUTION WITH PARAM
     3ETERS',/,5X,'ALPHA =',F6.3,1X,'AND BETA =',D11.4)
  304 FORMAT('-',8X,'*',2(12X,'*'),11X,'INTEGRATION METHOD',12X,'*',
     1  6X,'BINOMIAL METHOD',/,9X,'*',12X,'*',12X,'*  PARAMETERS ',
     2  2('*  PARMS CALC '),'*   COMPOUND   * HOMOGENEOUS',/,'  LAMBDA *
     3    ALPHA    *     BETA     *     KNOWN     *  W/ OUTLIER * W/O OUTLIER
     4',2('*     MODEL     '),/,1X,8('-'),'*',12('-'),'*',12('-'),5('*',
     5  13('-')))
  305 FORMAT(' ',25(F7.3,' *',D11.4,' *',D11.4,1X,5('*',F10.5,3X),/,1X))
  306 FORMAT('-')
  307 FORMAT(' ','FOR LAMBDA =',F8.3,'THE INTEGRATION METHOD (PARMS CALC
     1 W/O OUTLIER) FAILED',I5,' TIMES')
  308 FORMAT(' ','EXCLUDING DATA SETS WHICH GAVE NEGATIVE PARAMETER VALU
```

```
  1ES')
   WRITE(6,309)
309 FORMAT('1')
   STCP
   END
```

```
      REAL FUNCTION RF*8(L)
      IMPLICIT REAL*8 (A-H,C-Z)
      COMMCN /FGEN/ ALPH(25),BET(25),T,IX,IY
      DIMENSICN XC(1001)
      XC(1)=0.0D0
      A=ALPH(L)
      B=BET(L)
      J=2
      DO 1 I=1,1000
      F=DFLCAT(I-1)
      P=DLGAMA(F+A)+F*DLOG(T)+A*DLOG(B)-DLGAMA(A)-DLGAMA(F+1.0D0)
     1       -(F+A)*DLCG(T+B)
      XC(I+1)=XC(I)+DEXP(P)
      IF(XC(I+1).GE.0.9999000)GC TC 2
    1 J=J+1
    2 CALL RANCU(IX,IY,YFL)
      IX=IY
      IF(YFL.GT.0.9999000)GC TO 3
      IF(YFL.GT.XC(J))GC TO 3
      DO 4 I=2,J
      IF(YFL.LE.XC(I))RF=DFLCAT(I-2)
      IF(YFL.LE.XC(I))GO TO 5
    4 CONTINUE
      GO TC 5
    3 RF=DFLCAT(J-1)
    5 RETURN
      END
```

```
      REAL FUNCTION HCUM*8(FM)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SNGL
      COMMON /CUMH/ A,B,T,NDATA
      NFM1=IFIX(SNGL(FM))
      HCUM=0.0D0
      DO 1 I=1,NFM1
      F=DFLOAT(I-1)
      H=DLGAMA(F+A)+F*DLOG(T)+A*DLOG(B)-DLGAMA(A)-DLGAMA(F+1.D00)
     1      -(F+A)*DLOG(T+B)
    1 HCUM=HCUM+DEXP(H)
      RETURN
      END
```

```
      SUBROUTINE RANDU(IX,IY,YFL)
      REAL*8 YFL
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
```

The following is a program for generating power curves for several methods of detecting a single outlier in an exponential distribution by the method described in App. A and by simulation.

In the driving program:

NDATA = number of data points per data set less one

NSETS = number of data sets analyzed

X(J,I) = x_{ci} of App. A for the various methods

FCUM(J,K) = $F(x_b)$ of App. A for the various methods

PRCUM(J,L) = points in the power curve for the method of App. A

PRSIM(J,L) = points in the power curve for the simulation method

The subroutine 'ORDER1' is for ordering the data of a data set.

The subroutine 'ORDER2' is for ordering the critical points at which an outlier is detected (the values of x_{ci} in App. A).

The subroutine 'RANDU' is for generating a random number between 0 and 1, and comes from IBM Manual C20-8011.

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SNGL
      COMMON /ORD1/ DATA(1250,30),D(30),NDATA
      COMMON /ORD2/ X(4,1250),XC(4,1250),INC(4,1250)
      DIMENSION XC(4,302),FC(4,302),PROB(4,25),PRSIM(4,25),XSIM1(1250)
     1   ,XSIM2(1250),RLAM1(25),RLAM2(25),BET1(25),BET2(25),NINT(4)
C
C *** INITIALIZE MAIN PROGRAM
C
      NDATA=19
      NSETS=263
C
C *** INITIALIZE SUBPROGRAM 'RANDU'
C
      IX=157366247
      DO 1 I=1,43
      CALL RANDU(IX,IY,YFL)
    1 IX=IY
C
C *** SET VALUES OF THE EXPONENTIAL PARAMETER BETA
C
      DO 100 IB=1,5
      BETA=0.1000**(IB+2)
C
C *** INPUT AND OUTPUT DATA
C
      READ(9,300) ((DATA(L,I),I=1,NDATA),L=1,NSETS)
      READ(9,300) DUMMY
      WRITE(6,301)
      WRITE(6,302) (L,(DATA(L,I),I=1,NDATA),L=15,NSETS,15)
C
C *** SET CRITICAL VALUES OF THE TEST STASTICS
C
C *** USE 'DO 100 IT=1,2' TO USE BOTH 95% AND 99% CONFIDENCE LEVELS
      IT=1
      IF(IT.EQ.2)GO TO 3
      TFU=0.2705
      TFL=0.0001317
      TDU=0.573
      TDL=0.051
      CUM=0.95000**(1.000/DFLOAT(NDATA+1))
      GO TO 4
    3 TFU=0.3297
      TFL=0.00002632
      TDU=0.687
      TDL=0.082
      CUM=0.99000**(1.000/DFLOAT(NDATA+1))
C
C *** PRELIMINARY CALCULATIONS
C
    4 DO 2 L=1,NSETS
      SUM1=0.0D0
      DO 5 I=1,NDATA
    5 SUM1=SUM1+DATA(L,I)
      CALL ORDER1(L)
```

```
C
C *** CALCULATE MINIMUM UPPER OUTLIER VIA FISCHER'S METHOD
C
      X(1,L)=TFU*SUM1/(1.0D0-TFU)
C
C *** CALCULATE MINIMUM UPPER OUTLIER VIA DIXON'S METHOD
C
      X(2,L)=(D(NDATA)-TDU*D(1))/(1.0D0-TDU)
C
C *** CALCULATE MAXIMUM OUTLIER VIA FISCHER'S METHOD
C
      X(3,L)=TFL*SUM1/(1.0D0-TFL)
C
C *** CALCULATE MAXIMUM LOWER OUTLIER VIA DIXON'S METHOD
C
      X(4,L)=(D(1)-TDL*D(NDATA))/(1.0D0-TDL)
      IF(X(4,L).LT.0.0D0)X(4,L)=0.0D0
    2 CONTINUE
C
C *** CALCULATE THE TRUE CRITICAL OUTLIER VALUES
C
      XL=-DLOG(CUM)/BETA
      XU=-DLOG(1.0D0-CUM)/BETA
C
C *** ORDER AND OUTPUT CRITICAL OUTLIER VALUES
C
      WRITE(6,303)
      WRITE(6,304)
      WRITE(6,305) (L,(X(J,L),J=1,4),L=15,NSETS,15)
      WRITE(6,306) XL,XU
      CALL ORDER2(NSETS)
      WRITE(6,303)
      WRITE(6,307)
      L1=0
      DO 6 L=1,NSETS
      L1=L1+1
      WRITE(6,308) (IND(J,L),XO(J,L),J=1,4)
      IF(L1.LT.50)GO TO 6
      L1=0
      WRITE(6,306) XL,XU
      WRITE(6,303)
      WRITE(6,307)
    6 CONTINUE
      WRITE(6,306) XL,XU
C
C *** SET INTERVALS FOR AND CALCULATE THE CUMULATIVE PROBABILITY THAT A
C *** GIVEN VALUE IS DETECTED AS AN OUTLIER IN THE EXPONENTIAL
C *** DISTRIBUTION WITH PARAMETER BETA (F-CUM)
C
      DO 7 J=1,4
      DX=(XO(J,NSETS)-XO(J,1))/100.0D0
      DO 8 I=1,29
      P=5.0D0/10.0D0**(I-15)
      Q=0.40D0*P
      R=0.20D0*P
      IF(DX.LT.P)DELX=Q
      IF(DX.LT.Q)DELX=R
      IF(DX.LT.R)DELX=P/10.0D0
      IF(DX.GT.R)GO TO 9
```

```
    8 CONTINUE
    9 NINT(J)=IFIX(SNGL((XO(J,NSETS)-XO(J,1))/DELX))+2
      IF(NINT(J).EQ.2)GO TO 7
      NT=IFIX(SNGL(XO(J,1)/DELX))
      XC(J,1)=DFLOAT(NT)*DELX
      NIN=NINT(J)
      DO 10 I=1,NIN
   10 XC(J,I+1)=XC(J,I)+DELX
      IF(J.GT.2)GO TO 11
      NFL=1
      FC(J,1)=0.0D0
      DO 12 I=1,NIN
      NF=0
      XM=(XC(J,I+1)+XC(J,I))/2.0D0
      IF(NFL.GT.NSETS)GO TO 12
      DO 13 K=NFL,NSETS
      IF(XO(J,K).LE.XM)NF=NF+1
      IF(XO(J,K).GT.XM)GO TO 14
   13 CONTINUE
   14 NFL=NFL+NF
   12 FC(J,I+1)=FC(J,I)+DFLOAT(NF)/DFLOAT(NSETS)
      GO TO 7
   11 NFL=NSETS
      FC(J,NIN+1)=0.0D0
      DO 15 I=1,NIN
      NF=0
      XM=(XC(J,NIN+2-I)+XC(J,NIN+1-I))/2.0D0
      IF(NFL.LT.1)GO TO 15
      DO 16 K=1,NSETS
      IF(XO(J,NFL+1-K).GE.XM)NF=NF+1
      IF(XO(J,NFL+1-K).LT.XM)GO TO 17
   16 CONTINUE
   17 NFL=NFL-NF
   15 FC(J,NIN+1-I)=FC(J,NIN+2-I)+DFLOAT(NF)/DFLOAT(NSETS)
    7 CONTINUE
      DO 18 J=1,4
      IF(NINT(J).EQ.2)GO TO 18
      NIN=NINT(J)+2
      DO 18 I=NIN,300
      FC(J,I)=0.0D0
   18 XC(J,I)=0.0D0
C *** OUTPUT F-CUM
      DO 19 J=1,4
      IF(NINT(J).EQ.2)GO TO 19
      WRITE(6,309)
      IF(J.EQ.1)WRITE(6,310)
      IF(J.EQ.2)WRITE(6,311)
      IF(J.EQ.3)WRITE(6,312)
      IF(J.EQ.4)WRITE(6,313)
      WRITE(6,314)
      DO 20 I=1,50
      K=I
      IF(J.LT.3)K=I+1
      N1=I
      N2=I+50
      N3=I+100
      WRITE(6,315) N1,XC(J,I),XC(J,I+1),FC(J,K),N2,XC(J,I+50),
     1  XC(J,I+51),FC(J,K+50),N3,XC(J,I+100),XC(J,I+101),FC(J,K+100)
   20 CONTINUE
```

```
      IF(NINT(J).LE.150)GO TO 19
      WRITE(6,309)
      IF(J.EQ.1)WRITE(6,310)
      IF(J.EQ.2)WRITE(6,311)
      IF(J.EQ.3)WRITE(6,312)
      IF(J.EQ.4)WRITE(6,313)
      WRITE(6,314)
      DO 21 I=151,200
      K=I
      IF(J.LT.3)K=I+1
      N1=I
      N2=I+50
      N3=I+100
      WRITE(6,315) N1,XC(J,I),XC(J,I+1),FC(J,K),N2,XC(J,I+50),
     1  XC(J,I+51),FC(J,K+50),N3,XC(J,I+100),XC(J,I+101),FC(J,K+100)
   21 CONTINUE
   19 CONTINUE
C
C *** CALCULATE THE PROBABILITY THAT AN OUTLIER COMING FROM THE
C *** EXPONENTIAL DISTRIBUTION WITH PARAMETER BETAI IS DETECTED
C *** AS AN OUTLIER IN DATA COMING FROM THE EXPONENTIAL DISTRIBUTION
C *** WITH PARAMETER BETA
C
      DO 22 L=1,15
      RLAM1(L)=1.0D0*DFLOAT(L)
      RLAM2(L)=0.01D00-DFLOAT(L-1)*0.000678D00
      BET1(L)=BETA/RLAM1(L)
      BET2(L)=BETA/RLAM2(L)
      DO 23 J=1,2
      IF(L.EQ.1)GO TO 24
      IF(PROB(J,L-1).GT.0.999)GO TO 25
   24 PROB(J,L)=0.0D0
      NIN=NINT(J)
      DO 26 I=1,NIN
      GAM=DEXP(-XC(J,I)*BET1(L))-DEXP(-XC(J,I+1)*BET1(L))
   26 PROB(J,L)=PROB(J,L)+FC(J,I+1)*GAM
      GAM=DEXP(-XC(J,NIN+1)*BET1(L))
      PROB(J,L)=PROB(J,L)+GAM
      GO TO 23
   25 PROB(J,L)=1.0D0
   23 CONTINUE
      DO 27 J=3,4
      IF(NINT(J).EQ.2)GO TO 33
      IF(L.EQ.1)GO TO 28
      IF(PROB(J,L-1).GT.0.999)GO TO 29
   28 PROB(J,L)=0.0D0
      NIN=NINT(J)
      DO 30 I=1,NIN
      GAM=DEXP(-XC(J,I)*BET2(L))-DEXP(-XC(J,I+1)*BET2(L))
   30 PROB(J,L)=PROB(J,L)+FC(J,I)*GAM
      GAM=1.0D0-DEXP(-XC(J,1)*BET2(L))
      PROB(J,L)=PROB(J,L)+GAM
      GO TO 27
   29 PROB(J,L)=1.0D0
      GO TO 27
   33 PROB(J,L)=0.0D0
   27 CONTINUE
   22 CONTINUE
C
```

```
C *** CALCULATE THE SIMULATED PROBABILITY THAT AN OUTLIER COMING FROM
C *** THE EXPONENTIAL DISTRIBUTION WITH PARAMETER BETAI IS DETECTED
C *** AS AN OUTLIER IN DATA COMING FROM THE EXPONENTIAL DISTRIBUTION
C *** WITH PARAMETER BETA
C
      DO 31 L=1,15
      DO 32 I=1,NSETS
      CALL RANDU(IX,IY,YFL)
      IX=IY
      XSIM1(I)=-DLOG(1.0D0-YFL)/BET1(L)
   32 XSIM2(I)=-DLOG(1.0D0-YFL)/BET2(L)
      DO 31 J=1,2
      PRSIM(J,L)=0.0D0
      PRSIM(J+2,L)=0.0D0
      DO 31 I=1,NSETS
      IF(XSIM1(I).GT.X(J,I))PRSIM(J,L)=PRSIM(J,L)+1.0D0/DFLOAT(NSETS)
      IF(XSIM2(I).LT.X(J+2,I))PRSIM(J+2,L)=PRSIM(J+2,L)+1.0D0
     1      /DFLOAT(NSETS)
   31 CONTINUE
C
C *** OUTPUT THE ABOVE CALCULATED PROBABILITIES
C
      WRITE(6,316) BETA
      WRITE(6,317)
      WRITE(6,318) (RLAM1(L),BET1(L),PROB(1,L),PROB(2,L),RLAM2(L),
     1      BET2(L),PROB(3,L),PROB(4,L),L=1,15)
      WRITE(6,319) BETA
      WRITE(6,317)
      WRITE(6,318) (RLAM1(L),BET1(L),PRSIM(1,L),PRSIM(2,L),RLAM2(L),
     1      BET2(L),PRSIM(3,L),PRSIM(4,L),L=1,15)
  100 CONTINUE
C
C *** FORMAT STATEMENTS
C
  300 FORMAT(100(100D12.4/))
  301 FORMAT('1',4X,'DATA',1X,'*',/,5X,'SET  *',51X,'DATA',/,4X,
     1  6('-'),'*',100('-'))
C *** CHANGE FORMAT 302 TO ACCOMODATE 'NDATA'
  302 FORMAT(' ',4X,100(I3,2X,'*',10D10.3,/,10X,'*',9D10.3,/,10X,
     1      '*',/,5X))
  303 FORMAT('1',4X,'CRITICAL OUTLIER VALUES')
  304 FORMAT('-',5X,'DATA',1X,2('*  FISCHER''S  *   DIXON''S   '),/,
     1  6X,'SET',2X,2('*     UPPER     '),2('*     LOWER     '),/,5X,6('-'),
     2  4('*',13('-')))
  305 FORMAT(' ',5X,100(I3,2X,4('*',2X,D9.3,2X),/,6X))
  306 FORMAT('0',4X,'THE TRUE CRITICAL UPPER AND LOWER OUTLIER VALUES AR
     1E',D12.4,2X,'AND',D12.4)
  307 FORMAT('-',5X,'DATA *  FISCHER''S  * DATA *   DIXON''S   * DATA',
     1  ' *  FISCHER''S  * DATA *   DIXON''S',/,6X,'SET  ',
     2  2('*    UPPER    * SET  '),'*     LOWER    * SET  *     LOWER',/,
     3  5X,6('-'),3('*',13('-'),'*',6('-')),'*',13('-'))
  308 FORMAT(' ',5X,I3,2X,3('*',D11.3,'  *',I4,2X),'*',D11.3)
  309 FORMAT('1',4X,'CUMULATIVE PROBABILITY THAT A GIVEN VALUE IS DETECT
     1ED AS AN OUTLIER')
  310 FORMAT(' ',4X,'UPPER OUTLIER USING FISCHER''S METHOD')
  311 FORMAT(' ',4X,'UPPER OUTLIER USING DIXON''S METHOD')
  312 FORMAT(' ',4X,'LOWER OUTLIER USING FISCHER''S METHOD')
  313 FORMAT(' ',4X,'LOWER OUTLIER USING DIXON''S METHOD')
  314 FORMAT('-',2(5X,'*    LOWER    *    UPPER    *',11X,'*'),5X,
```

```
     1  '*   LOWER   *   UPPER   *',/,1X,2(' INT *',2('   LIMIT   *'),
     2  '   F-CUM   *'),' INT ',2('*   LIMIT   '),'*   F-CUM',/,1X,
     3  2(5('-'),'*',11('-'),'*',11('-'),'*',11('-'),'*'),5('-'),'*',
     4  11('-'),'*',11('-'),'*',11('-'))
315 FORMAT(' ',2(I4,' *',D10.3,' *',D10.3,' *',F10.3,' *'),I4,' *',
     1  D10.3,' *',D10.3,' *',D10.3)
316 FORMAT('1',4X,'PROBABILITY THAT A DATA POINT COMING FROM THE EXPON
     1ENTIAL DISTRIBUTION WITH PARAMETER ',/,5X,'BETA WILL BE DETECTED A
     2S AN OUTLIER IN THE EXPONENTIAL DISTRIBUTION WITH PARAMETER BETA =
     3',D10.4)
317 FORMAT('-',4X,2(8X,'*',12X,'*  FISCHER''S  *   DIXON''S',13X),/,
     1  6X,'LAMBDA *    BETA',4X,2('*    UPPER    '),11X,'LAMBDA *',4X,
     2  'BETA',4X,2('*    LOWER    '),/,5X,2(8('-'),'*',12('-'),'*',
     3  13('-'),'*',13('-'),10X))
318  FORMAT(' ',4X,25(F7.3,' *',D11.4,' *',F10.5,'   *',F10.5,13X,
     1  F7.5,' *',D11.4,' *',F10.5,'   *',F10.5,/,5X))
319 FORMAT('1',4X,'SIMULATED PROBABILITY THAT A DATA POINT COMING FROM
     1 THE EXPONENTIAL DISTRIBUTION WITH PARAMETER',/,5X,'BETA WILL BE D
     2ETECTED AS AN OUTLIER IN THE EXPONENTIAL DISTRIBUTION WITH PARAMET
     3ER BETA =',D10.4)
      STOP
      END


      SUBROUTINE ORDER1(L)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CRD1/ DATA(1250,30),D(30),NDATA
      DIMENSION I(50)
      I(1)=1
      DO 1 J=2,NDATA
      J1=J-1
      DO 2 K=1,J1
      IF(DATA(L,J).LT.DATA(L,I(K)))GO TO 3
    2 CONTINUE
    3 JK=J-K
      IF(J.EQ.K)GO TO 1
      DO 4 K=1,JK
    4 I(J-K+1)=I(J-K)
    1 I(J-JK)=J
      DO 5 J=1,NDATA
    5 D(J)=DATA(L,I(J))
      RETURN
      END
```

```
      SUBROUTINE ORDER2(NSETS)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CRD2/ X(4,1250),XC(4,1250),IND(4,1250)
      DO 1 I=1,4
      IND(I,1)=1
      DO 1 L=2,NSETS
      L1=L-1
      DO 2 J=1,L1
      IF(X(I,L).LT.X(I,IND(I,J)))GO TO 3
    2 CONTINUE
    3 LJ=L-J
      IF(L.EQ.J)GO TO 1
      DO 5 K=1,LJ
    5 IND(I,L-K+1)=IND(I,L-K)
    1 IND(I,L-LJ)=L
      DO 4 J=1,4
      DO 4 L=1,NSETS
    4 XC(J,L)=X(J,IND(J,L))
      RETURN
      END
```

```
      SUBROUTINE RANDU(IX,IY,YFL)
      REAL*8 YFL
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
```

DETECTION OF OUTLIERS

in FAILURE DATA

by

DONALD ROBERT GALLUP

B.S., KANSAS STATE UNIVERSITY, 1979

AN ABSTRACT OF

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Nuclear Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1981

ABSTRACT

The objective of this work was to investigate and determine the best ways to detect outliers in failure rate data for continuously operating nuclear power plant components. Three specific types of data were investigated: 1) data which were distributed according to a gamma distribution; 2) data which were of the form (F_i, T_i), where F_i and T_i were the number of failures and test time, respectively, of component i; and 3) time-to-failure data, i.e., data which were distributed according to an exponential distribution.

Three methods of outlier detection in gamma samples were investigated; 1) Fisher's method; 2) a normal conversion method; and 3) an integration method. The normal conversion method was found to be the most powerful method of single outlier detection for all cases investigated. Two methods of outlier detection in (F_i, T_i) data were studied: 1) the cumulative marginal method and 2) the binomial method (which could only be used when all T_i were the same). For larger failure rates, $\bar{\lambda} > 5 \times 10$ hr^{-1}, the cumulative marginal mehtod using the prior matching moment method (PMMM) of parameter estimation without including the potential outlier was found to be the most powerful method of single outlier detection For smaller failure rates, $\hat{\bar{\lambda}} > 5 \times 10^{-5}$ hr^{-1}, and all T_i the same, the most powerful method of single outlier detection was found to be the binomial method used in conjunction with the homogeneous failure model. When all T_i are not the same, the most powerful method for detecting an outlier is the same method used for $\bar{\lambda} > 5 \times 10^{-5}$ hr^{-1}. Two methods of outlier detection in time-to-failure data were investigated: 1) Fisher's method and 2) Dixon's method. Fisher's method was found to be superior to Dixon's method of single outlier detection for every case investigated

All of the above results were obtained at a theoretical confidence level of 95%. However, the actual confidence levels varied, and this fact could have compromized the "fairness" of the comparisons made between the various tests.