Juergen Schwarz

# Estimation of Parameters and Fitness Tests in the Lognormal Distribution

**ABSTRACT** *This paper presents derivations of programs for the estimation of parameters and for performing fitness tests for the two- and three-parameter lognormal distribution. The utility of the procedures is demonstrated with the aid of an example. Since the lognormal distribution occurs frequently in practice, these procedures can be applied to many areas of science and engineering.*

**Keywords:** Statistics, logarithmic normal distribution, estimation of parameters, $\chi^2$ test, Kolmogorov-Smirnov test. **Hardware:** HP 9845B with structured programming ROM, possibly also with advanced programming ROM.

## 1 Introduction

When faced with the task of examining the statistical properties of random samples for which no probability model is known, one generally first tests for the presence of a normal distribution. This model is all the more likely to be valid when numerous quantities showing statistically independent variations additively constitute the series of measurements to be evaluated. Each quantity may exert only a slight influence on the overall result because the central limit theorem suggests a normal distribution. However, if the $\chi^2$ test and/or the Lilliefors modification of the Kolmogorov-Smirnov test for a normal distribution [ref.1] are rejected at a low significance level, it becomes necessary to search for another probability model. If the empirical density curve has negative skew, it would appear reasonable to test for the presence of a lognormal distribution. This distribution function is ascribable to the multiplicative interaction of many random effects. Practical applications include time studies, high voltage engineering, analytical chemistry, and, in the present case, semiconductor engineering.

## 2 The Two-Parameter Lognormal Distribution

A continuous random quantity $X$ which can assume all positive values is said to be lognormal distributed with the parameters $\mu$ and $\sigma$ if the random quantity $Y$ formed by the transformation $Y = \ln X$ has a normal distribution with the parameters $\mu$ and $\sigma$.

Hence $X$ has the density:

$$f(x) = \begin{cases} \dfrac{1}{x\,\sigma\sqrt{2\pi}} \exp\left\{ -\dfrac{1}{2} \left( \dfrac{\ln x - \mu}{\sigma} \right)^2 \right\} & \text{for } x > 0 \\ 0 & \text{for } x \leq 0. \end{cases} \tag{1}$$

*Juergen Schwarz, AEG Aktiengesellschaft, Geschaeftsbereich Komponenten, Fachbereich Leistungselektronik und Anlagenbau, Culemeyerstrasse 1, D-1000 Berlin 48, FRG.*

where $\mu$ can be regarded as the scale parameter and $\sigma$ as the shape parameter (see Figure 1). The distribution function:

$$F(x) = \int^{x} f(\zeta)\, d\zeta \qquad (2)$$

cannot be presented in closed form. However, on application of the distribution function $\Phi$ of the standardized normal distribution, it can be expressed as:

$$F(x) = \Phi \left( \frac{\ln x - \mu}{\sigma} \right) \qquad (3)$$

The likelihood method has proved its value as asymptotically expectation-true, efficient, consistent, and sufficient process for estimation of parameters [ref.2].

We start from a concrete random sample with the elements $x_i$ of size $n$ and a known distribution function of the population. In order to estimate the parameters of the distribution function the likelihood function of the concrete random sample is defined as:

$$L = f(x_1)\, f(x_2)\, f(x_3) \ldots f(x_n) = \prod_{i=1}^{n} f(x_i) \qquad (4)$$

This function, which is dependent upon both the random sample and the parameters of the distribution is a measure for the probability of occurrence of the random sample. This likelihood is regarded as a function of the unknown distribution parameter and the latter is determined such that L is maximized. To this end, equation (4) is partially differentiated with respect to these parameters and the resulting functions set equal to zero. It has proved convenient to simplify this calculation by basing further calculation on the logarithm of the likelihood function, since the product according to equation (4) then gives the sum:

$$\ln L = \sum_{i=1}^{n} \ln f(x_i) \qquad (5)$$

It can easily be proved that this log-likelihood function has its extreme values at the same places as equation (4). In the specific application, the log-likelihood function is found to be:

$$\ln L = \sum_{i=1}^{n} \ln \left[ \frac{1}{x_i \sigma \sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left( \frac{\ln x_i - \mu}{\sigma} \right)^2 \right\} \right] \qquad (6)$$

This affords:

$$\ln L = -\frac{n}{2} \ln(2\pi) - n \ln \sigma - \sum_{i=1}^{n} \left[ \ln x_i + \frac{(\ln x_i - \mu)^2}{2\sigma^2} \right] \qquad (7)$$

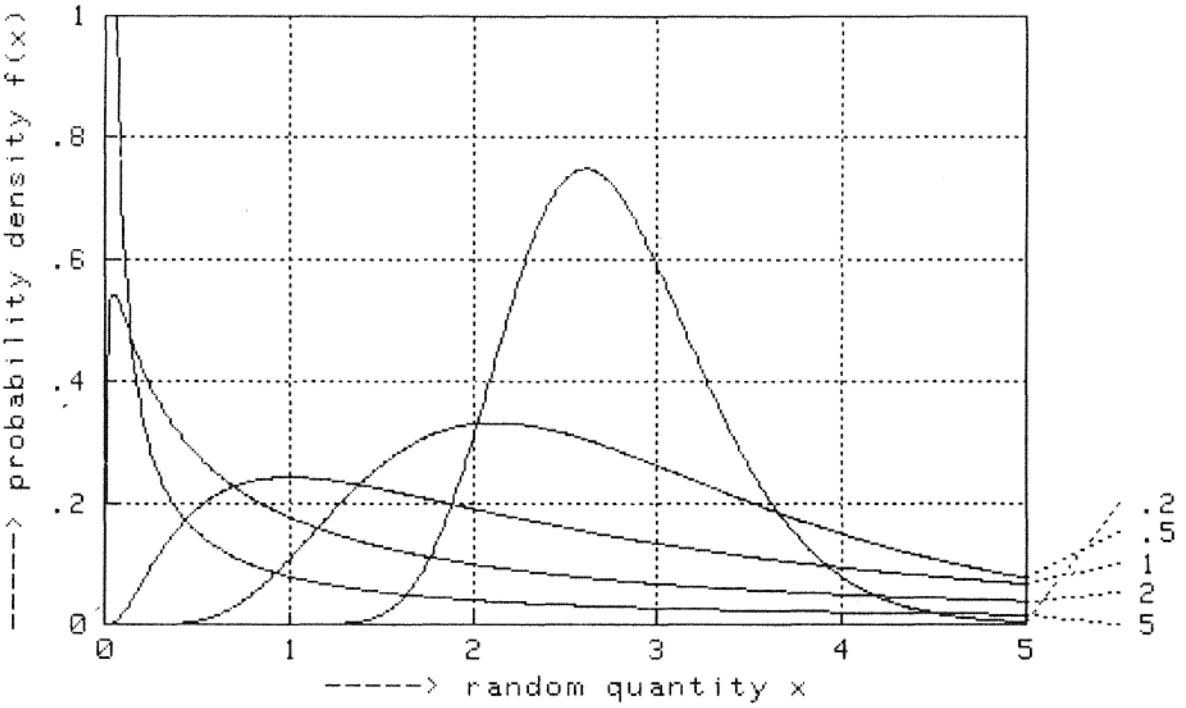
Differentiation with respect to $\mu$ yields:

$$\frac{\partial \ln L}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (\ln x_i - \mu) = 0 \qquad (8)$$

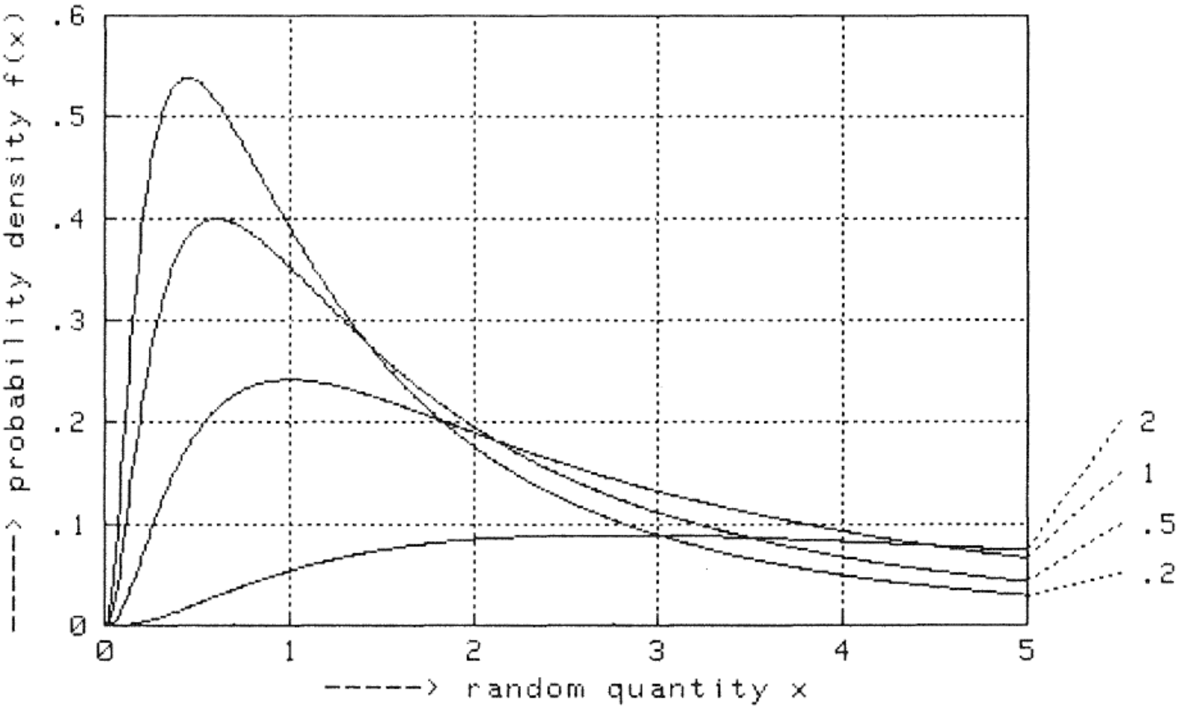
and differentiation with respect to $\sigma$ gives:

$$\frac{\partial \ln L}{\partial \sigma} = -\frac{n}{\sigma} + \sum_{i=1}^{n} \frac{(\ln x_i - \mu)^2}{\sigma^3} = 0 \qquad (9)$$

**Figure 1** *Graphical representation of the population density of the two-parameter lognormal distribution.*




a) constant scale parameter $\mu$ = 1 and variable shape parameter $\sigma$




b) constant shape parameter $\sigma$ = 1 and variable scale parameter $\mu$

The maximum likelihood estimator for the scale parameter follows immediately from equation (8):

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} ln\ x_i \tag{10}$$

and after multiplication of equation (9) by $\sigma^3$ the maximum likelihood estimator for the shape parameter is obtained as:

$$\hat{\sigma} = \sqrt{\frac{1}{n} \sum_{n=1}^{n} (ln\ x_i - \hat{\mu})^2} \tag{11}$$

These two equations correspond in principle to the two known estimators for the parameters of the normal distribution. However, the estimator for $\sigma$ is not unbiased, but only asymptotically unbiased. Replacement of $n$ in equation (11) by $n - 1$ affords an unbaised estimator [ref. 1]. Equations (10) and (11) are very easily calculated so that printing of a program listing is not necessary.

If fitness tests for lognormal distribution are also rejected and if the empirical distribution function plotted on a probability grid tends toward a straight line in the upper region but the straight line is bent up or down in the lower region, then we might be dealing with a three-parameter lognormal distribution.

## 3 Three-Parameter Lognormal Distribution

If, instead of $ln X$, the random quantity $Y = ln(X - x_0)$ resulting on transformation $ln(X - x_0)$ shows a normal distribution then we are indeed dealing with a three-parameter logarithmic distribution. Figure 2 shows the transformation of this distribution to a normal distribution. The probability density is given by:

$$f(x) = \begin{cases} \dfrac{1}{(x - x_0)\,\sigma\sqrt{2\pi}} \exp\left\{ -\dfrac{1}{2} \left( \dfrac{ln(x - x_0) - \mu}{\sigma} \right)^2 \right\} & \text{for } x > x_0 \\ 0 & \text{for } x \leq x_0 \end{cases} \tag{12}$$

and the distribution function is:

$$F(x) = \int_{x_0}^{x} f(\zeta)\ d\zeta = \Phi\left( \frac{ln(x - x_0) - \mu}{\sigma} \right) \tag{13}$$

The likelihood function according to equation (4) can again be used for estimation of parameters:

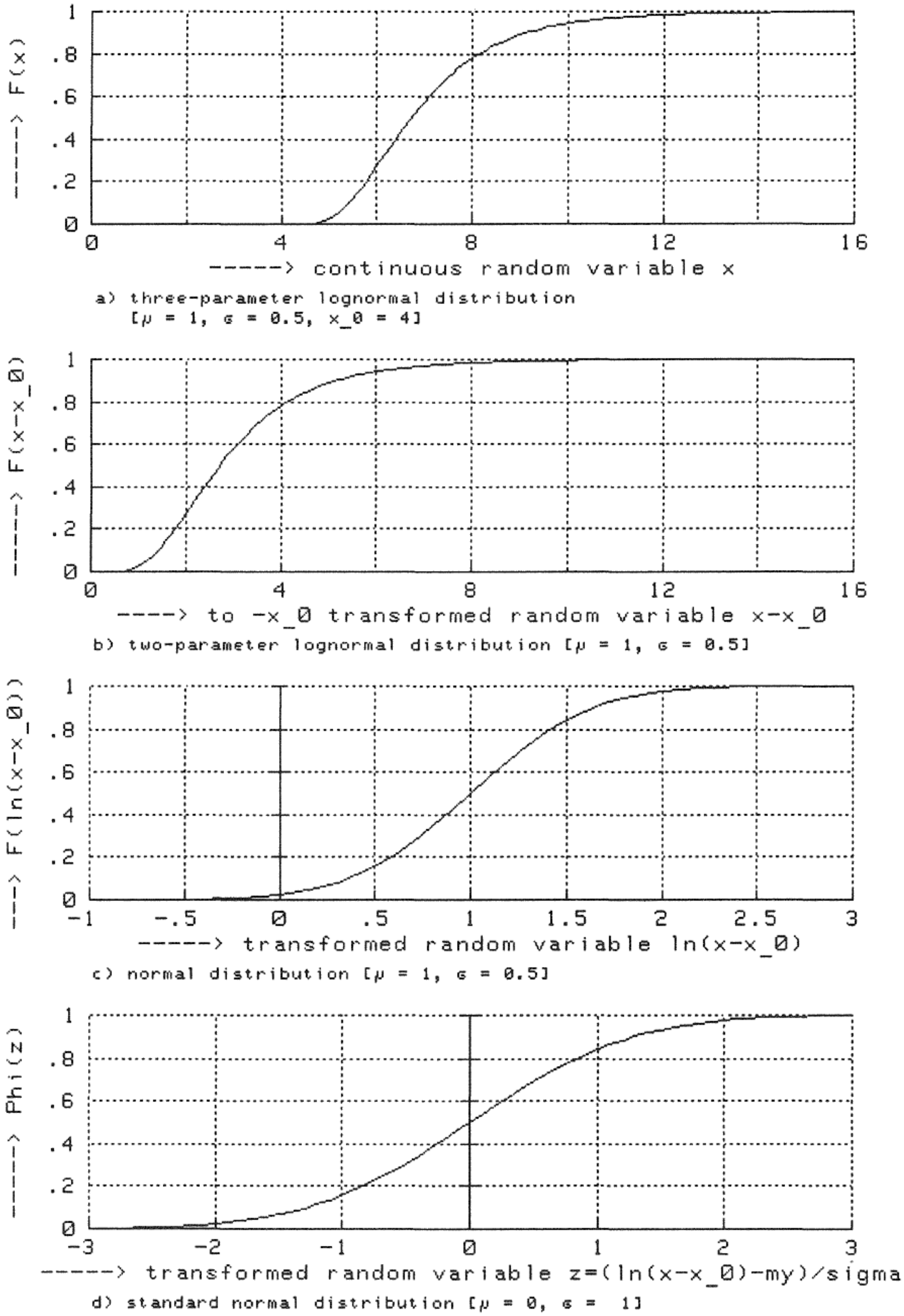
$$L = \prod_{i=1}^{n} \frac{1}{(x_i - x_0)\,\sigma\sqrt{2\pi}} \exp\left\{ -\frac{1}{2} \left( \frac{ln(x_i - x_0) - \mu}{\sigma} \right) \right\} \tag{14}$$

Taking logarithms and multiplying affords:

$$ln\ L = -\frac{n}{2} ln(2\pi) - n\ ln\ \sigma + \left(\frac{\mu}{\sigma^2} - 1\right) \sum_{i=1}^{n} ln(x_i - x_0) -$$

$$- \frac{1}{2\sigma^2} \sum_{i=1}^{n} ln^2 (x_i - x_0) - \frac{n}{2\sigma^2} \mu^2 \tag{15}$$

**Figure 2** *Transformation of the three-parameter lognormal distribution to the standard normal distribution. (Parameters of the initial distribution: $\mu = 1$, $\sigma = 0.5$, $x_o = 4$.)*



Differentiation with respect to $\mu$ and $\sigma$ and setting equal to zero affords relations analogous to equations (10) and (11) for the two parameters:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} ln\,(x_i - \hat{x}_0) \tag{16}$$

$$\hat{\sigma} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} [ln\,(x_i - \hat{x}_0) - \hat{\mu}]^2} \tag{17}$$

Differentiation with respect to $x_0$ furnishes:

$$\frac{\partial\, ln\, L}{\partial\, x_0} = (\frac{\mu}{\sigma^2} - 1) \sum_{i=1}^{n} \frac{1}{x_i - x_0} - \frac{1}{\sigma^2} \sum_{i=1}^{n} \frac{ln(x_i - x_0)}{x_i - x_0} = 0 \tag{18}$$

After multiplication by $-\sigma^2$ and introduction of equation (16) and of:

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} ln^2(x_i - x_0) - \frac{1}{n^2} \left[ \sum_{i=1}^{n} ln(x_i - x_0) \right]^2 \tag{19}$$

into equation (18) the result is:

$$g(\hat{x}_0) = \left\{ n \sum_{i=1}^{n} \left[ ln^2 (x_i - \hat{x}_0) - ln(x_i - \hat{x}_0) \right] - \left[ \sum_{i=1}^{n} ln(x_i - \hat{x}_0) \right]^2 \right\} \cdot$$

$$\cdot \sum_{i=1}^{n} \frac{1}{x_i - \hat{x}_0} + n^2 \sum_{i=1}^{n} \frac{ln(x_i - \hat{x}_0)}{x_i - \hat{x}_0} = 0 \tag{20}$$

a non-linear estimator for $x_0$ which depends only upon $\hat{x}_0$. If this function is designated as $g(\hat{x}_0)$, then the estimated value $\hat{x}_0$ is obtained by determining the zero of $g(\hat{x}_0)$. It is worth knowing that $g(\hat{x}_0)$ assumes the following limited values:

$$\lim_{\hat{x}_0 \to -\infty} g(\hat{x}_0) = 0$$

$$\lim_{\substack{\hat{x}_0 - \min x_i \to -0 \\ i = 1(1)n}} g(\hat{x}_0) = -\infty \tag{21}$$

In order to use the Newton procedure, the derivative of $g(\hat{x}_0)$ is given as:

$$\frac{d\, g(\hat{x}_0)}{d\, \hat{x}_0} = n^2 \sum_{i=1}^{n} \frac{ln(x_i - \hat{x}_0) - 1}{x_i - \hat{x}_0} + \left\{ n \sum_{i=1}^{n} [\, ln^2(x_i - \hat{x}_0) - ln(x_i - \hat{x}_0)] \right.$$

$$- \left[ \sum_{i=1}^{n} ln(x_i - \hat{x}_0) \right]^2 \Bigg\} \sum_{i=1}^{n} \frac{1}{(x_i - \hat{x}_0)^2} + \sum_{i=1}^{n} \frac{1}{(x_i - \hat{x}_0)} \cdot$$

$$\cdot \Bigg\{ n \sum_{i=1}^{n} \left[ \frac{1 - 2ln(x_i - \hat{x}_0)}{x_i - \hat{x}_0} \right] + 2 \left[ \sum_{i=1}^{n} ln(x_i - \hat{x}_0) \right] \left[ \sum_{i=1}^{n} \frac{1}{x_i - \hat{x}_0} \right] \Bigg\} \quad (22)$$

Listing 1 shows a SUB-program in which the above algorithm is implemented. Starting from the extreme values of the random sample the REPEAT-UNTIL loop of lines 360 to 410 approach the smallest value of the random sample until the derivative according to equation (22) becomes negative. The value of Delta = 0.5 selected here will, of course, depend upon the distribution and has so far proven to be good. Subsequently, the zero of equation (20) is determined by the Newton method. The shape and scale parameters are calculated in lines 490 and 500 and the following lines serve for calculation of the function and its derivative according to (20) and (22). If MAT commands are implemented in the computer, then computing time can be saved. Listing 2 shows details of a program using MAT commands which is functionally equivalent to Listing 1. By way of explanation, it is mentioned that the scalar product of the two vectors *X*(*) and *Y*(*) are calculated with the function DOT (*X*, *Y*).

## 4 Fitness Tests

In a fitness test the null hypothesis:

$H_0$: the parent population shows a lognormal distribution with the parameters $\hat{\mu}$, $\hat{\sigma}$, $\hat{x}_0$

is tested against the alternative hypothesis:

$H_1$: the parent population does not show a lognormal distribution with the parameters $\hat{\mu}$, $\hat{\sigma}$, $\hat{x}_0$

at the significance level $\alpha$. The known tests are $\chi^2$ and Kolmogorov-Smirnov. However, neither test proves that the population has (in the present case) a lognormal distribution; passing the test means only that the test does not rule out a lognormal distribution.

### 4.1 The $\chi^2$ Test

The test is performed with:

$$t = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{n\,p_i} \quad (23)$$

where $n_i$ is the absolute class frequency of the $i$-th class in an appropriately selected classification $I_1, I_2, \ldots, I_k$ with $I_i = [a_i, a_{i+1}]$, $i = 1(1)k$ ($k$ is the number of classes), and:

$$p_i = \int_{a_i}^{a_{i+1}} F(\zeta\,; \hat{\mu}, \hat{\sigma}, \hat{x}_0)\, d\zeta \quad (24)$$

is the hypothetical probability of the elements in the respective classes. A useful orientation value for the number of classes is:

$$k \approx 5\, l_g n \quad (25)$$

It should be observed that $n\,p_i$, in all classes is $\geq 5$. The hypothesis $H_0$ is rejected [ref.1] if:

$$t > \chi^2_{m\,;\,1-\alpha} \quad (26)$$

$m$ is the number of degree of freedom. In the present case it holds that:

$$m = \begin{cases} k - 2 \text{ for the two-parameter lognormal distribution} \\ k - 3 \text{ for the three-parameter lognormal distribution} \end{cases} \quad (27)$$

Tables of $\chi^2$ distribution are to be found, *e.g.*, in Sachs [ref.1]; equations for a simple approximation in Bruhn [ref.3] and for accurate calculation in Mardia and Zemroch [ref.4]. Listing 3 shows a SUB-program for calculating test quantities for given parameters of the lognormal distribution. Class limits are calculated in lines 350 to 390 on the basis of the extreme values of the empirical distribution, while in the lines up to 500 the elements of the random sample are assigned to the individual classes. This is followed by calculation of the hypothetical probability in the individual classes. Those classes within the lines 590 to 740 for which $n\,p_i < 5$ are assigned to the next class. The residue provides a test quantity according to equation (23). Lines 520, 540, 550, and 570 call the functional subprogram FNPhi which supplies the distribution function of the normal distribution. It is reproduced in Listing 4, being based on an approximation equation for the Gaussian error integral FNErf.

## 4.2 The Kolmogorov-Smirnov Test

The maximum deviation between the empirical and the hypothetical theoretical distribution function serves as test quantity:

$$t = \max_{x_0 < x < \infty} | F_n(x) - F_0(x) | \tag{28}$$

The empirical distribution function is given by:

$$F_n(x) = \begin{cases} 0 & \text{for } x < x_i \\ \frac{i}{n} & \text{for } x_i < x \leq x_{i+1}, i = 1(1)n - 1 \\ 1 & \text{for } x > x_n \end{cases} \tag{29}$$

where $x_i$ are the elements of random sample ordered by size. Since equation (29) gives a step curve, only the conditions at the corner points have to be examined, so that:

$$t = \max_{i = 1(1)n} \left[ \frac{i}{n} - F_0(x_i), F_0(x_i) - \frac{i-1}{n} \right] \tag{30}$$

affords the desired quantity. Critical values $K_{n;\alpha}$ for the Kolmogorov-Smirnov test can be taken from corresponding tabulations [ref.1]. The hypothesis $H_0$ is rejected if:

$$t > K_{n;\alpha} \tag{31}$$

Listing 5 shows part of a main program in which the test quantity is determined according to equation (30) (line 2120). Moreover, the procedure for drawing the empirical distribution function as a polygon of cumulative frequency on a probability grid of the three-parameter lognormal distribution is also shown (line 2100). The function subprogram for calculating the inverse function of the normal distribution function shown in Listing 6 is used for this purpose. Lines 2160 and 2170 plot the line of the theoretical distribution function.

## 5 Example

Figures 3 and 4 show an example taken from semiconductor engineering. Conclusions regarding the distribution function of the population are to be drawn from empirically determined data on the forward voltage drop of thyristors at a given on-state current. This distribution function is an important basis for thermal dimensioning of rectifiers using thyristors. Figure 3 shows the numerical results after tests for normal distribution and two-parameter lognormal distribution were rejected.

The Kolmogorov-Smirnov test is accepted at a high level of significance; however, it affords very conservative results in this case because it was conceived for an independent distribution with the critical values used here and not for empirical parameters estimated from the random sample [ref.1]. Nevertheless, the author does not know any limited critical values for the three-parameter lognormal distribution analogous to the Lilliefors modification for the normal distribution. Moreover, the $\chi^2$ test is also not rejected at the 2.5% level. It should be borne in mind that relatively large random samples such as in the present case can only rarely be fitted satisfactorily to the theoretical distributions.

**Figure 3** *Numerical data of evaluation of a random sample for determination of the empirical parameters of the distribution and performing two fitness tests of the distribution with these parameters.*

Device: Thyristor T 2200 N 3600 ... 4200 (AEG AG)

Statistical Evaluation:

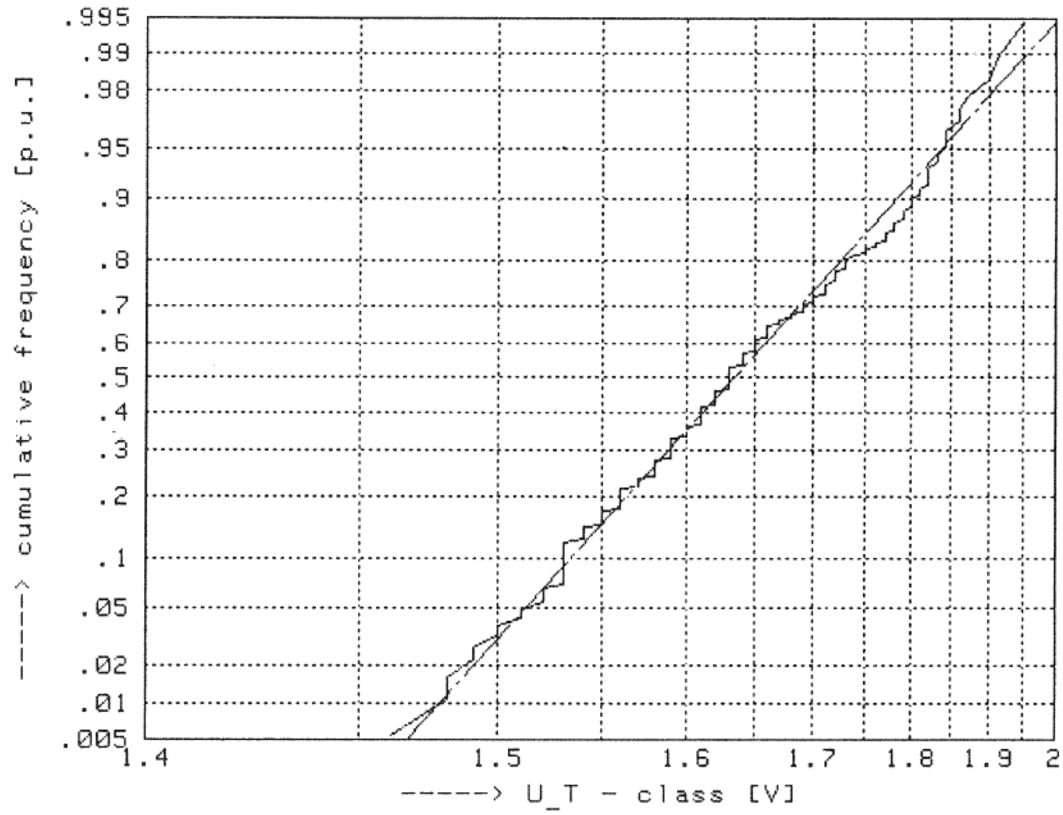
| | | | |
|---|---|---|---|
| empirical position parameter of the distribution | U_0 | = | 1.327 V |
| empirical scale parameter of the distribution | μ_UT | = | -1.181 |
| empirical shape parameter of the distribution | σ_UT | = | 0.309 |
| sample mean of U_T | x_quer | = | 1.649 V |
| sample standard derivation of U_T | s | = | 0.100 V |
| sample minimum of U_T | x_min | = | 1.460 V |
| sample maximum of U_T | x_max | = | 1.950 V |
| sample range of variation of U_T | R | = | 0.490 V |

The Kolmogorov-Smirnov test of the lognormal distribution with the empirical parameters described above is not rejected with the level of significance of $\alpha \geq 0.20$. The size of test is D_n = 0.050 and the critical value K_n_α is K = 0.080. The size of the random sample is n = 180.

The $\chi^2$-test of goodness of fit of the lognormal distribution with the empirical parameters shown above ist not rejected with a level of significance of $\alpha \geq 0.025$.
The test quantity is t = 12.72 and the critical value $\chi^2$_m_α is $\chi^2$ = 14.45. The number of classes ist n_classe = 9. Therefore the number of degree of freedom is m = 6.

**Figure 4** *Empirical distribution function and distribution function of the population estimated by the maximum likelihood method.*

U_T - distribution of 180 thyristors of typ T 2200 N 3600 ... 4200 (AEG AG)




Remark: probability grid of the three-parameter lognormal distribution with position parameter U_0 = 1.327 V.

Figure 4 shows the graphical representation for the present example, as obtained with the program given in Listing 5. The fit is seen to be good.

## References


[1] Lothar Sachs: “Angewandte Statistik.” Springer-Verlag, Berlin, Heidelberg, New York, Tokyo (1984); see also Lothar Sachs: “Applied Statistic.” Springer-Verlag, New York, Heidelberg, Berlin, Tokyo (1984).

[2] Gisela Härtler: “Statistische Methoden für die Zuverlässigkeitsanalyse.” Verlag Technik, Berlin (1983).

[3] Jörn Bruhn: “Statistik für programmierbare Taschenrechner (UPN).” Anwendung programmierbarer Taschenrechner Vol. 21. Friedrich Vieweg & Sohn, Braunschweig, Wiesbaden (1984).

[4] K.V. Mardia and P.J. Zemroch: “Tables of the F- and Related Distributions with Algorithmus.” Academic Press, London, New York, San Francisco (1978).


**Listing 1** *SUB program for estimation of parameters for the three-parameter lognormal distribution.*

```
10    SUB Lognormal(INTEGER N,REAL X(*),My,Sigma,X_0,Eps)
20    !
30    ! SUB program for estimation of empirical parameters for the
40    !            three-parameter lognormal distribution
50    !            with the maximum-likelihood-method
60    !
70    ! input parameters:
80    !         n          ...  size of sample
90    !         x(1:n)     ...  one-dimensional array of the sample
100   !         eps        ...  relative accuracy for the solution for x_0
110   !
120   ! output parameters:
130   !         my         ...  estimate for the scale parameter
140   !         sigma      ...  estimate for the shape parameter
150   !         x_0        ...  estimate for the position parameter
160   !
170   ! Referenc: Gisela Härtler: "Statistische Methoden für die
180   !           Zuverlässigkeitsanalyse". Verlag Technik Berlin (1983).
190   !
200   ! (c)      1985 by Jürgen Schwarz          version/date:     2.0 / 12.08.85
210   ! filename:                 englLN          memory:        cartridges 77/78
220   ! computer: HP 9845 B with SP ROM           language:              HP-BASIC
230   !
240     INTEGER I
250     REAL Derivation,Delta,Function,X_min,X_max,X_0_alt,Sum_ln_kr
260     REAL Sum_1,Sum_2,Sum_cross,Sum_cross_2,Sum_ln,Sum_ln_2
270     REAL X_cross(1:N),X_cross_2(1:N),X_ln(1:N)
280     !
290     IF (N<3) OR (Eps<=0) THEN SUBEXIT                        ! bad data check
300     REDIM X(1:N)
310     X_min=X_max=X(1)
320     FOR I=2 TO N
330       X_max=MAX(X_max,X(I))
340       X_min=MIN(X_min,X(I))
350     NEXT I
360     Delta=.5
370     REPEAT
380       X_0=X_min-Delta*(X_max-X_min)          ! initial value for the iteration
390       GOSUB Function
400       Delta=.5*Delta
```

```
410       UNTIL Derivation<0
420       !
430       REPEAT
440         X_0_alt=X_0
450         X_0=X_0_alt-Function/Derivation
460         GOSUB Function
470       UNTIL ABS(X_0_alt-X_0)<Eps*(X_max-X_min)
480       !
490       My=Sum_ln/N                                   ! empirical scale parameter
500       Sigma=SQR((Sum_ln_2-Sum_ln*Sum_ln/N)/N)       ! empirical shape parameter
510       SUBEXIT
520       !
530       ! !
540       !
550   Function:   ! SUB program for the conditional equation for x_0 [f(x_0)]
560               ! and there first derivation [d f(x_0)/d x_0]
570       PRINT USING "#,5A,XMZ.DDDE,10X";"x_0 =",X_0
580       !
590       Sum_cross=Sum_cross_2=Sum_ln=Sum_ln_2=Sum_ln_kr=0
600       FOR I=1 TO N
610         X_cross(I)=1/(X(I)-X_0)                    ! vector: 1/(x-x_0)
620         Sum_cross=Sum_cross+X_cross(I)             ! sum 1/(x-x_0)
630         X_cross_2(I)=X_cross(I)*X_cross(I)         ! vector: 1/(x-x_0)^2
640         Sum_cross_2=Sum_cross_2+X_cross_2(I)       ! sum 1/(x-x_0)^2
650         X_ln(I)=LOG(X(I)-X_0)                      ! vector: ln(x-x_0)
660         Sum_ln=Sum_ln+X_ln(I)                      ! sum ln(x-x_0)
670         Sum_ln_2=Sum_ln_2+X_ln(I)*X_ln(I)          ! sum (ln(x-x_0))^2
680         Sum_ln_kr=Sum_ln_kr+X_ln(I)*X_cross(I)     ! sum ln(x-x_0)/(x-x_0)
690       NEXT I
700       !
710       Sum_1=N*(Sum_ln_2-Sum_ln)-Sum_ln*Sum_ln
720       Function=Sum_1*Sum_cross+N*N*Sum_ln_kr
730       !
740       Derivation=Sum_1*Sum_cross_2
750       Sum_1=Sum_2=0
760       FOR I=1 TO N
770         Sum_1=Sum_1+(1-2*X_ln(I))*X_cross(I)
780         Sum_2=Sum_2+(X_ln(I)-1)*X_cross_2(I)
790       NEXT I
800       Derivation=Derivation+N*N*Sum_2
810       Derivation=Derivation+(N*Sum_1+2*Sum_ln*Sum_cross)*Sum_cross
820       !
830       PRINT USING "#,8A,XMZ.DDDE,5X";"f(x_0) =",Function
840       PRINT USING "11A,XMZ.DDDE";"d f/d x_0 =",Derivation
850       RETURN
860     SUBEND
```

**Listing 2** *Details from a SUB program serving the same purpose as Listing 1 but using MAT commands.*

```
10      SUB Lognormal_mat(INTEGER N,REAL X(*),My,Sigma,X_0,Eps)


310       MAT SEARCH X(*),MAX;X_max
320       MAT SEARCH X(*),MIN;X_min


460       My=Sum_ln/N                                   ! empirical scale parameter
470       Sigma=SQR(Sum_ln_2/N-My*My)                   ! empirical shape parameter
```

```
560      MAT X_asterisk=(-X_0)+X                                  ! vector: (x-x_0)
570      MAT X_cross=(1)/X_asterisk                               ! vector: 1/(x-x_0)
580      MAT X_cross_2=X_cross.X_cross                            ! vector: 1/(x-x_0)^2
590      MAT X_ln=LOG(X_asterisk)                                 ! vector: ln(x-x_0)
600      !
610      Sum_cross=SUM(X_cross)
620      Sum_ln=SUM(X_ln)
630      Sum_ln_2=DOT(X_ln,X_ln)                                  ! sum (ln(x-x_0))^2
640      !
650      Sum=N*(Sum_ln_2-Sum_ln)-Sum_ln*Sum_ln
660      Function=Sum*Sum_cross+N*N*DOT(X_ln,X_cross)
670      !
680      MAT Help=(-1)+X_ln
690      Derivation=Sum*SUM(X_cross_2)+N*N*DOT(Help,X_cross_2)
700      MAT Help=(-2)*X_ln
710      MAT Help=(1)+Help
720      Sum=N*DOT(Help,X_cross)+2*Sum_ln*Sum_cross
730      Derivation=Derivation+Sum*Sum_cross
```

**Listing 3** *SUB program for performing the $\chi^2$ test.*

```
10     SUB Chi2_lognormal(INTEGER N,N_class,REAL X(*),My,Sigma,X_0,Test)
20     !
30     ! SUB program for the chi-squared test of goodness of fit with
40     !             the previously calculated empirical parameter of the
50     !             three-parameter logarithmic normal distribution
60     !
70     ! input parameters:
80     !          n          ...  size of sample
90     !          x(1:n)     ...  one-dimensional array of the sample
100    !          my         ...  scale parameter of the distribution
110    !          sigma      ...  shape parameter of the distribution
120    !          x_0        ...  position parameter of the distribution
130    ! output parameters:
140    !          n_class   ... number of class of the chi-squared test of fit
150    !          test      ... test value of the chi-squared test of fit
160    !
170    ! (c)      1985 by Jürgen Schwarz           version/date:    2.0 / 26.06.84
180    ! filename:                  englLN         memory:        cartridges 77/78
190    ! computer: HP 9845 B with SP ROM           language:               HP-BASIC
200    !
210      INTEGER I,J,N_class(1:20)
220      REAL X_max,X_min,Difference,X_stroke(1:N)
230      REAL Limits(1:19),P_class(1:20)
240      !
250      IF N<10 THEN SUBEXIT                                     ! bad data check
260      REDIM X(1:N)
270      !
280      X_max=X_min=X(1)
290      FOR I=1 TO N
300        X_max=MAX(X_max,X(I))
310        X_min=MIN(X_min,X(I))
320        X_stroke(I)=LOG(X(I)-X_0)
330      NEXT I
340      !
350      N_class=MIN(20,INT(5*LGT(N))-1)  ! orientation for the number of classes
360      REDIM Limits(1:N_class-1),P_class(1:N_class),N_class(1:N_class)
370      FOR I=1 TO N_class-1
380        Limits(I)=I*LOG((X_max-X_0)/(X_min-X_0))/N_class+LOG(X_min-X_0)
390      NEXT I
400      FOR I=1 TO N
```

```
410        IF X_stroke(I)>=Limits(N_class-1) THEN
420          N_class(N_class)=N_class(N_class)+1
430        ELSE
440          J=0
450          REPEAT
460            J=J+1
470          UNTIL X_stroke(I)<Limits(J)
480          N_class(J)=N_class(J)+1
490        END IF
500      NEXT I
510      !
520      P_class(1)=N*FNPhi(Limits(1),My,Sigma)
530      FOR I=2 TO N_class-1
540        P_class(I)=FNPhi(Limits(I),My,Sigma)
550        P_class(I)=N*(P_class(I)-FNPhi(Limits(I-1),My,Sigma))
560      NEXT I
570      P_class(N_class)=N*(1-FNPhi(Limits(N_class-1),My,Sigma))
580      !
590      WHILE P_class(1)<5
600        N_class=N_class-1
610        P_class(1)=P_class(1)+P_class(2)
620        N_class(1)=N_class(1)+N_class(2)
630        FOR I=2 TO N_class
640          P_class(I)=P_class(I+1)
650          N_class(I)=N_class(I+1)
660        NEXT I
670        REDIM N_class(1:N_class),P_class(1:N_class)
680      END WHILE
690      WHILE P_class(N_class)<5
700        N_class=N_class-1
710        P_class(N_class)=P_class(N_class)+P_class(N_class+1)
720        N_class(N_class)=N_class(N_class)+N_class(N_class+1)
730        REDIM N_class(1:N_class),P_class(1:N_class)
740      END WHILE
750      !
760      Test=0
770      FOR I=1 TO N_class
780        Difference=N_class(I)-P_class(I)
790        Test=Test+Difference*Difference/P_class(I)
800      NEXT I
810    SUBEND
```

**Listing 4** *Auxiliary program for calculating the distribution function of the normal distribution.*

```
10     DEF FNErf(REAL X)
20     !
30     ! Functional subprogram for calculation of the Gaussian error
40     ! integral with the accuracy of ≤ 1.5E-7
50     !
60     ! (c)      1984 by Jürgen Schwarz          version/date:     2.0 / 27.03.84
70     ! filename:                 englLN          memory:         cartridges 77/78
80     ! computer: HP 9845 B with SP ROM          language:                HP-BASIC
90     !
100    ! Referenc:  Milton Abramowitz and Irene A. Stegun: "Handbook of Mathe-
110    !            matical Functions". National Bureau of Standards. Applied
120    !            Mathematics Series No. 55. United States Department of
130    !            Commerce June 1964.
140    !
150      REAL T,P
160      !
170      T=1/(1+.3275911*ABS(X))
```

```
180      P=(((1.061405429*T-1.453152027)*T+1.421413741)*T-.284496736)*T
190      RETURN SGN(X)*(1-(P+.254829592)*T*EXP(-X*X))
200    FNEND
210    DEF FNPhi(REAL X,My,Sigma)
220    !
230    ! Functional subprogram for calculation of the value of the distribution
240    ! function Phi(X;my,sigma) of the normal distribution with known X,
250    ! known expected value my and known standard derivation sigma of the
260    ! distribution parameters
270    !
280    ! (c)      1984 by Jürgen Schwarz          version/date:    2.0 / 26.05.84
290    ! filename:                 englLN         memory:       cartridges 77/78
300    ! computer: HP 9845 B with SP ROM          language:              HP-BASIC
310    !
320      RETURN .5*(1+FNErf((X-My)/(SQR(2)*Sigma)))
330    FNEND
```

**Listing 5** *Calculation of the test quantity for the Kolmogorov-Smirnov test and plotting of the empirical and the theoretical distribution.*

```
2030   MAT X=(-U_0)+U_t    ! subtraction of the vektor with the position parameter
2040   MAT X=LOG(X)        ! logarithmic of the vektor [ln(x-x_0)]
2050   MAT SORT X          ! sorting the vector
2060   !
2070   D=0                 ! zero the test value for the Kolmogorov-Smirnov test
2080   MOVE X(1),FNInv_phi(1/(N+1))        ! set the pen into the basic position
2090   FOR I=1 TO N
2100     DRAW X(I),FNInv_phi(I/(N+1))      ! move to the next plot position
2110     Phi=FNPhi(X(I),My_ut,Sigma_ut)    ! compute the test value for the
2120     D=MAX(D,I/N-Phi,Phi-(I-1)/N)      ! Kolmogorov-Smirnov test
2130   NEXT I
2140   !
2150   LINE TYPE 6
2160   MOVE My_ut+Sigma_ut*Vtl_extr,Vtl_extr  ! plot the distribution function
2170   DRAW My_ut-Sigma_ut*Vtl_extr,-Vtl_extr ! with the empirical parameters
```

**Listing 6** *Functional subprogram for calculating the inverse function of the distribution function of the normal distribution.*

```
10     DEF FNInv_phi(REAL Phi)
20     !
30     ! Functional subprogram for calculation of the inverse function of the
40     ! distribution function of the standard normal distribution [x=x(phi)]
50     !
60     ! (c)      1984 by Jürgen Schwarz          version/date:    2.0 / 27.03.84
70     ! filename:                 englLN         memory:       cartridges 77/78
80     ! computer: HP 9845 B with SP ROM          language:              HP-BASIC
90     !
100    ! Referenc:  Jörn Bruhn: "Statistik für programmierbare Taschenrechner".
110    !            Friedr. Vieweg & Sohn Braunschweig und Wiesbaden (1983).
120    !
130      REAL Phi_asterisk,S,S_asterisk,Numerator,Denominator
140      !
150      IF (Phi>=1) OR (Phi<=0) THEN PAUSE                      ! bad data check
160      Phi_asterisk=ABS(2*Phi-1)
170      S_asterisk=2/(1-Phi_asterisk)
180      S=SQR(LOG(S_asterisk*S_asterisk))
190      Numerator=(.010328*S+.802853)*S+2.515517
200      Denominator=((.001308*S+.189269)*S+1.432788)*S+1
210      !
220      RETURN SGN(Phi-.5)*(S-Numerator/Denominator)
230    FNEND
```