

## Chebyshev Expansions for the Error and Related Functions

By J. L. Schonfelder

**Abstract.** New 30 decimal place Chebyshev expansions, which can be used to obtain approximations up to that accuracy, for the error function and its complement are presented. Bilinear and biquadratic mappings of the independent variable have been employed, in one case to improve convergence and in the other to increase the basic approximation range. The practical effects of these mappings are discussed, and the method used to generate the tabulated expansions is outlined. The expansions here presented form the basis of the routines in the S15 chapter of the NAG library.

**Introduction.** Clenshaw [1] presents Chebyshev expansions, accurate to twenty decimal places, 20D, for the error function,  $\text{erf}(x)$ . These expansions are of the form\*

$$\text{erf}(x) = \frac{x}{4} \sum_{r=0}^N a_r T_r(t), \quad |x| \leq 4, \quad t = 2\left(\frac{x}{4}\right)^2 - 1,$$

and

$$\text{erf}(x) = 1 - \frac{e^{-x^2}}{x} \sum_{r=0}^N a_r T_r(t), \quad x > 4, \quad t = 2\left(\frac{4}{x}\right)^2 - 1.$$

In the first expansion to obtain 20D accuracy 33 terms are required; in the second, 18 terms. Luke [2] presents expansions which differ from the above only in trivial detail but which use a break point of three rather than four. For these expansions, which are given up to 20D, 25 terms are required for the small  $x$  region and 22 for the large  $x$ . The S15 chapter of the NAG library had a requirement for routines to approximate the functions  $\text{erf}(x)$ ,  $\text{erfc}(x)$ ,  $P(x)$  and  $Q(x)$ , the error function, its complement and the cumulative normal distribution function and its complement.\*\* All of these functions can be obtained simply from the basic  $\text{erf}(x)$  approximations since

---

Received September 14, 1977.

AMS (MOS) subject classifications (1970). Primary 33A20, 41A10, 65D20, 60E05.

Key words and phrases. Special function approximation, error functions, Chebyshev expansions.

\*The symmetry of the function  $\text{erf}(-x) = -\text{erf}(x)$  means that the function need only be approximated for positive  $x$ .

Note. The notation  $\sum_{r=0}^N$  denotes the summation is performed with the term in  $r = 0$  divided by 2.

The Chebyshev polynomials  $T_r(t)$  are those defined on the range  $t \in (-1, +1)$  i. e.  $T_r(t) = \text{Cos}[r \arccos(t)]$ .

\*\*The notation for functions and their definitions is that used in M. ABRAMOWITZ & I. STEGUN, Editors, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Nat. Bur. Standards Appl. Math. Series, No. 55, U.S. Government Printing Office, Washington, D. C., 1965.

Copyright © 1978, American Mathematical Society

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x),$$

$$P(x) = \frac{1}{2} [1 + \operatorname{erf}(+x/\sqrt{2})] = \frac{1}{2} \operatorname{erfc}(-x/\sqrt{2}),$$

$$Q(x) = \frac{1}{2} [1 - \operatorname{erf}(x/\sqrt{2})] = \frac{1}{2} \operatorname{erfc}(x/\sqrt{2}).$$

However, serious loss of accuracy is involved in some regions of  $x$  in all these forms due to the subtraction. Therefore, it was decided to produce an approximation for  $\operatorname{erfc}$  directly. Also, the NAG library is implemented on machines with precisions ranging from 7D to 18D with a possible future requirement of up to 28D.\*\*\* Hence, it was decided to produce new expansions for up to 30D. Following a suggestion by Miller [3] it was decided to look at alternative mappings of the expansion variable  $t(x)$  to see if any significant improvement in convergence could be obtained.

In the next section the final expansion forms used are presented and tabulated. The effect of the bilinear and biquadratic mappings employed are discussed. In the following section a brief outline of the techniques employed to generate and check these expansions is provided.

**The Expansions.** For the error function itself, which has a zero at the origin, it is desirable to extract this zero explicitly. This allows relative accuracy to be maintained even for very small arguments. Hence, it was decided to use two approximation ranges on the pattern of Clenshaw [1]. However, the choice of subdivision at  $x = 4$  results in an excessively long small  $x$  expansion if 30D is required. The subdivision was, therefore, moved to a smaller value of  $x$ ;  $x = 2$  was in fact found to be adequate. It is to some extent questionable whether a break point as low as two is necessary. The expansions of Luke [2] which use three may well be superior for some purposes. However, the expansions presented here are fairly well balanced in the most important 8–18D accuracy range and the extra efficiency thus produced for arguments less than two, which are somewhat more common than larger arguments, could be useful in practice. The extra length thereby introduced into the large  $x$  expansion was compensated by use of a biquadratic mapping for  $x \rightarrow t$  to improve convergence.

The expansion for small  $x$  has the form

$$\operatorname{erf}(x) = xy(t),$$

where

$$y(t) = \sum_{r=0}^N a_r T_r(t), \quad t = 2\left(\frac{x}{2}\right)^2 - 1,$$

and

$$t \in (-1, +1) \quad \text{if } x \in (0, 2).$$

The coefficients  $a_r$  correct to 30D are tabulated below.†

\*\*\*8D is single precision on PDP10 machines and 18D is double precision on Univac. 28D is the possible double precision on CDC equipment. This work was done before the possibility of extended precisions on IBM 370 and ICL 2900 machines was considered and hence allowance has not been made for this 35D working. However, the expansions presented here could be extended to 35 or 40D if the need arises.

†The tables have been reproduced from card decks containing the coefficients produced directly from the generating programs. If any reader wishes to use these coefficients directly the author is willing to supply copies of these card decks.

TABLE 1

Y=ERF(X)/X X=(0,A) T=2(X/A)+2 -1 A=2

MAX. ORD.=	27	MAX. ACC.=	30
0		+1.483110564084803581889448079057R+0	
1		-3.01071073386594942470731046311R-1	
2		+6.8994830689831566246603180718R-2	
3		-1.3916271264722187682546525687R-3	
4		+2.420799522433463662891678239R-4	
5		-3.65863968584808644649382577R-5	
6		+4.886209844323190482828887568R-6	
7		-5.749256558035684835054215R-7	
8		+6.11324357843476469706758R-8	
9		-5.8991015312958434390846R-9	
10		+5.207009092068648240455R-10	
11		-4.23297587996554326810R-11	
12		+3.18811350666491749748R-12	
13		-2.23615501883226842738R-13	
14		+1.46732984799108492R-14	
15		-9.044001985381747R-15	
16		+5.25481371547092R-16	
17		-2.88742261222849R-17	
18		+1.504785187558R-18	
19		-7.4572892821R-20	
20		+3.522563810R-21	
21		-1.589446448R-22	
22		+6.8643658R-24	
23		-2.84257R-25	
24		+1.13068R-26	
25		-4.338R-28	
26		+1.68R-29	
27		-1.08R-30	

TABLE 2

Y=X\*EXP(X\*X)\*(1-ERF(X)) X=(2,INF) T=(K-X\*X)/(K-8+X\*X) K=10.5

MAX. ORD.=	43	MAX. ACC.=	30
0		+1.0779778520723831511688335910348R+0	
1		-2.6559890409148673372146500904R-2	
2		-1.487073146698099509605046333R-3	
3		-1.38040145414143859607708920R-4	
4		-1.1280303332287491498507366R-5	
5		-1.172869842743725224053739R-6	
6		-1.03476150393304615537382R-7	
7		-1.1899114085892438254447R-8	
8		-1.016222544989498640476R-9	
9		-1.37895716146965692169R-10	
10		-9.369613033737303335R-12	
11		-1.918809583959525349R-12	
12		-3.7573017201993707R-14	
13		-3.7053726026983357R-14	
14		+2.627565423490371R-15	
15		-1.121322876437933R-15	
16		+1.84136028922538R-16	
17		-4.9130256574886R-17	
18		+1.0704455167373R-17	
19		-2.6718936624058R-18	
20		+6.49326867976R-19	
21		-1.65399353183R-19	
22		+4.2605626604R-20	
23		-1.1255840765R-20	
24		+3.025617448R-21	
25		-8.29042146R-22	
26		+2.31049558R-22	
27		-6.5469511R-23	
28		+1.8842314R-23	
29		-5.504341R-24	
30		+1.630950R-24	
31		-4.89860R-25	
32		+1.49054R-25	
33		-4.5922R-26	
34		+1.4318R-26	
35		-4.516R-27	
36		+1.440R-27	
37		-4.64R-28	
38		+1.51R-28	
39		-5.0R-29	
40		+1.7R-29	
41		-6.0R-30	
42		+2.0R-30	
43		-1.0R-30	

The expansion for large  $x$  has the form

$$\operatorname{erf}(x) = 1 - \frac{e^{-x^2}}{x} y(t),$$

where

$$y(t) = \sum_{r=0}^{N'} a_r T_r(t), \quad t = (10.5 - x^2)/(2.5 + x^2),$$

and

$$t \in (-1, +1) \quad \text{if } x \in (2, \infty).$$

The coefficients  $a_r$  correct to 30D are tabulated above.

The complementary error function,  $\operatorname{erfc}(x)$  has no zeros. It takes the value one at the origin. It asymptotes to zero for large positive  $x$  and to two for large negative  $x$ .

For  $x \geq 0$ ,  $\operatorname{erfc}(x) = e^{-x^2} y(x)$  and for  $x < 0$ ,  $\operatorname{erfc}(x) = 2 - e^{-x^2} y(-x)$ .

Hence, the whole region may be covered by approximating  $y(x)$  for positive  $x$ . That is by the expansion

$$y(x) = \sum' a_r T_r(t), \quad t = (x - 3.75)/(x + 3.75),$$

where  $t \in (-1, +1)$  if  $x \in (0, \infty)$ .

The coefficients  $a_r$  correct to 30D are tabulated below.

TABLE 3

Y=EXP(X\*X)\*ERFC(X) X=(0,INF) T=(X-K)/(X+K) K=3.75

MAX\_ORD.= 43 MAX\_ACC.= 30

0	+6.101430819232004179264658157568-1
1	+4.348412727125774718281828208888-1
2	+1.763511936436055011258402981238-1
3	-6.07107956092494148600512158258-2
4	+1.77120689956941144861471411918-2
5	-4.3211193855672938185998649688-3
6	+8.542166768870986788198320558-4
7	-1.271550906091627426288939408-4
8	+1.12481672436711894688470728-5
9	+3.130638854218209726301528-7
10	-2.7098806853776202200908868-7
11	+3.07376227014076884409598-8
12	+2.5156203848176229373148-9
13	-1.0289299213203191275908-9
14	+2.994405211994993936388-11
15	+2.605178968726693629088-11
16	-2.63483992417196938688-12
17	-6.4340450989063644388-13
18	+1.1245740180166344788-13
19	+1.72815333899860988-14
20	-4.26410169494237588-15
21	-5.4537197788019188-16
22	+1.5869760776167188-16
23	+2.089983784433488-17
24	-5.900526886940988-18
25	-9.4189338755488-19
26	+2.1497735647088-19
27	+4.666098500888-20
28	-7.243011862888-21
29	-2.387966824888-21
30	+1.91177535888-22
31	+1.20482568888-22
32	-6.72377888-25
33	-5.747997888-24
34	-4.28493888-25
35	+2.44856888-25
36	+4.3793888-26
37	-8.1518888-27
38	-3.0898888-27
39	+9.3888888-29
40	+1.7488888-28
41	+1.6888888-29
42	-8.0888888-30
43	-2.0888888-30

**Discussion.** Taking the bilinear transformation used for  $\operatorname{erfc}$  first, the entire semi-infinite range  $x \in (0, \infty)$  may be uniquely mapped onto the required finite range  $t \in (-1, +1)$  for the Chebyshev expansion by the bilinear transformation

$$t = (x - k)/(x + k), \quad k > 0.$$

The resulting Chebyshev expansions in terms of the variable  $t$  provide, by truncation, approximations which preserve most of the properties of the Chebyshev polynomial expansions [4]. In particular, they provide approximations which are near mini-max and which allow precision changes to be made simply by changes in truncation point. However, the resulting approximations are in fact rational approximations in terms of the variable  $x$ . A polynomial approximation in the variable  $x$  over the entire range is obviously impractical, if not impossible. However, the use of the rational variable  $t$  allows us to produce, in essence, a polynomial approximation with all its advantages but to gain at least some of the increased flexibility or rational approximations. In this case the flexibility has been used to extend the range of approximation. We have gained this flexibility to some extent by introducing a further parameter,  $k$ .

The value of  $k$  used in the above expansion was chosen essentially by trial and error. For any given precision there is a value of  $k$  which gives optimum convergence. This value is different for different precisions and so the value actually used is a compromise which gives near optimal convergence over the main precision range of interest, viz. 8D to 18D. Fortunately, the convergence is comparatively insensitive to  $k$  so the value that gives optimal convergence for about 12–14D is not far from optimal for 8D or 18D. Certainly, for 30D one could have obtained a shorter expansion but only at the expense of requiring more terms in the 8–10D truncations. As these expansions were intended for use in transportable software, a further constraint on the values of  $k$  was that it had to be a value that was exactly representable on any machine. This means that  $k$  must be representable by less than six decimal digits and it must be the sum of a limited number of adjacent powers of two. For instance, the value 3.75 actually used is equivalent to  $2^1 + 2^0 + 2^{-1} + 2^{-2}$  and, hence, should be represented exactly on any floating point system currently in use.

Figure 1 shows graphically the order of the polynomial approximation required in the expansions for  $e^{x^2} \operatorname{erfc}(x)$  to provide truncation errors less than  $0.5 \times 10^{-d}$  for  $d = 8, 10, 12, 14, 16, 18$  for various values of  $k$ . This graph shows that for each precision there is a fairly large region in which  $k$  may vary without the number of terms in the expansion changing from the optimum. This optimum valley moves slowly to the right as the precision increases. The value 3.75 was chosen as a reasonable compromise value, being clearly within the optimum valley for all precisions from 8–18D. In fact, 3.75 remains within the optimal valley for precisions up to 25D and is only just outside it at 30D.

For the large  $x$  range of the error function itself, the function  $y(x)$  we are actually approximating is dependent on  $x^2$  not  $x$ ; and hence, a biquadratic is more appropriate. In this case we have a mapping

$$t = (k - x^2)/(k - 8 + x^2), \quad k > 4,$$

which uniquely maps the region  $x \in (2, \infty)$  into the required region  $t \in (-1, +1)$ . It should be noted in this case that this mapping is, in fact, a simple generalization of the more usual mapping as employed by Clenshaw [1]. If  $k = 8$ , then the mapping collapses into the simple form

$$t = 2\left(\frac{2}{x}\right)^2 - 1,$$

in which case the expansion is equivalent to an expansion in terms of the modified range Chebyshev polynomials  $T_r^*(t)$  with the expansion variable  $4/x^2 \in (0, 1)$ .

FIGURE 1

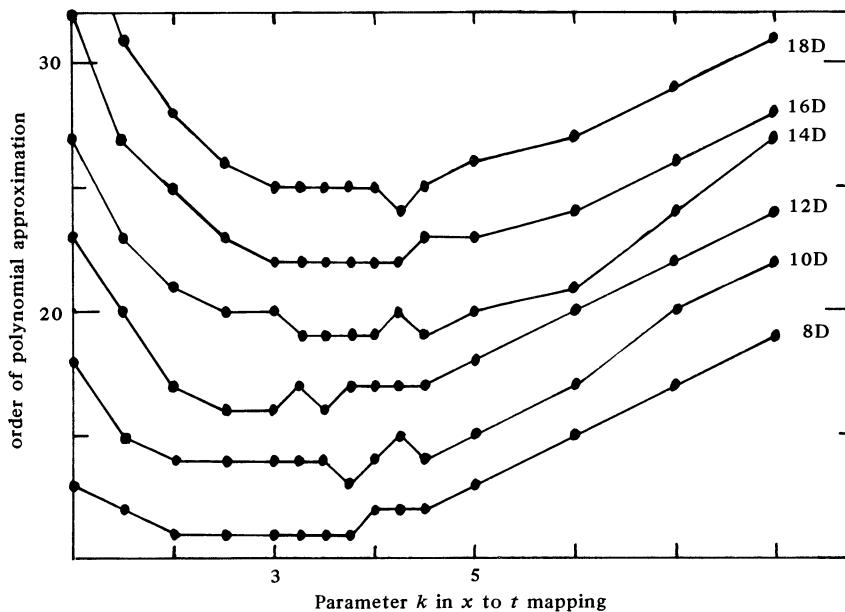


FIGURE 2

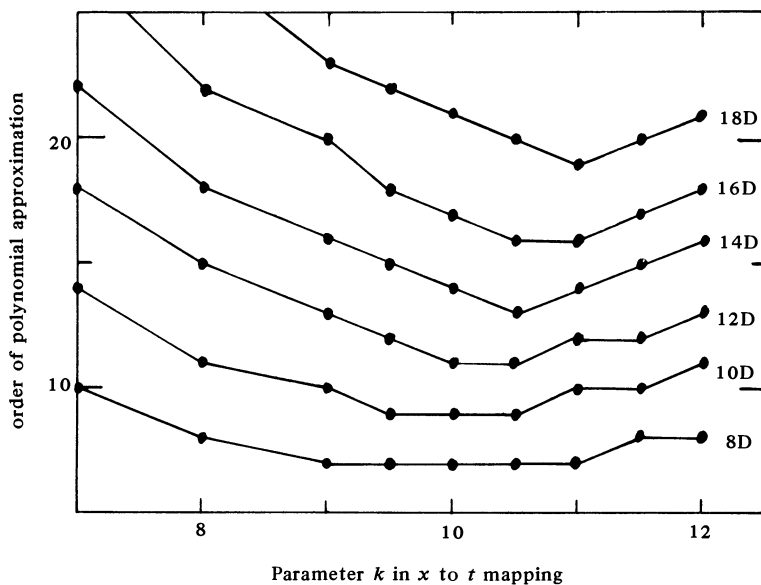


Figure 1 shows the order of polynomial required, to provide each of the precisions indicated, plotted against the mapping parameter  $k$  for the expansion of  $y(x) = e^{x^2} \operatorname{erfc}(x)$ ,  $t = (x - k)/(x + k)$  and  $x \in (0, \infty)$ .

N.B. In both Figures 1 and 2 the lines joining the measured points have no real significance. They are included as a guide to the eye.

Figure 2 shows the order of polynomial required, to provide each of the precisions indicated, plotted against the mapping parameter  $k$ , for the expansion of  $y(x) = [1 - \operatorname{erf}(x)] x e^{x^2}$ ,  $t = (k - x^2)/(k - 8 + x^2)$  and  $x \in (2, \infty)$ .

Figure 2 shows a plot of the order of the polynomial approximation required to give a truncation error of less than  $0.5 \times 10^{-d}$  for  $d = 8, 10, 12, 14, 16, 18$  for various values of  $k$ . Note the behavior is similar to the previous case except that the width of the optimal valley is now precision dependent; by and large it is smaller for higher precision. Also the degree of improvement possible over the normal expansions,  $k = 8$  is greater for greater precisions, a desirable feature. The compromise value of  $k$  chosen in this case was 10.5. Unlike the situation with  $\operatorname{erfc}$ , this value of  $k$  is not optimal for precisions like 25D or 30D. A value of  $k = 12$  or more would be needed to give optimal convergence at these precisions, and this would mean the lower precisions would be well away from optimal. However,  $k = 10.5$  still provides a considerable improvement, eight terms, over the normal  $k = 8$  expansions at 30D.

The convergence of the Chebyshev expansion, as with all such mathematical expansions is governed by the closeness of the singularities of the function being expanded to the expansion region. The major effect of mappings such as these is to allow us to move the singularities further away from the expansion region. The mapping may also weaken the effect of the singularities by modifying the strength of the singularity as well as moving it. This fact explains the reason why a similar mapping is not of any significant use for the small  $x$  range in approximating  $\operatorname{erf}(x)$ . The generalization of the mapping actually used is

$$t = \frac{2(x^2 - k)}{(2 - k)x^2 + 2k}, \quad 0 < k < 4.$$

If  $k = 2$ , this collapses to the normal mapping as used. In the case  $k = 2$  the expansion region is symmetrically located with respect to the singularities. Therefore, any different value of  $k$  destroys this symmetry and, hence, brings the singularities closer to the expansion region, hence making the convergence worse rather than better.

**Generation of Expansions.** The three expansions tabulated in the previous section were generated making use of the fact that the three functions involved satisfy linear differential equations with polynomial coefficients. For small  $x$ ,  $\operatorname{erf}(x)$  we have

$$(4 + 4t)\ddot{y} + (14 + 8t)\dot{y} + 4y = 0, \quad y(-1) = 2/\sqrt{\pi} \quad \text{and} \quad y'(-1) = -4/3\sqrt{\pi}.$$

For large  $x$ ,  $\operatorname{erf}(x)$  we have

$$\left[ \frac{k}{k-4} + \frac{8t}{k-4} + \frac{16-k}{k-4} t^2 + \frac{8-k}{k-4} t^3 \right] \dot{y} + [(1-2k) + (17-2k)t] y \\ = 2k/\sqrt{\pi} + (16-2k)t/\sqrt{\pi} \quad \text{and} \quad y'(-1) = 1/\sqrt{\pi}.$$

For  $\operatorname{erfc}(x)$  we have

$$[1 - 3t + 3t^2 - t^3] \dot{y} + [-4k^2 - 4k^2t]y = -4k/\sqrt{\pi} + 4kt/\sqrt{\pi} \quad \text{and} \quad y(+1) = 0.$$

These equations can be used to find the set of coefficients  $a_r$  such that  $y = \sum_{r=0}^N a_r T_r(t)$  satisfies the equations plus boundary conditions to within a required tolerance using methods described in Fox and Parker [5]. The precise details of the programs involved will be published elsewhere [6]. The general principle is as follows. The coefficients  $a_r$  can be shown to satisfy an infinite set of linear equations

$$A\underline{a} = \underline{b},$$

where the first few rows of the matrix  $A$  and the right-hand side  $\underline{b}$  are determined by the boundary conditions. The subsequent rows are then determined by the differential equations, given that the coefficient  $a_r$  tends to zero rapidly for large  $r$ . The infinite set may be handled approximately by assuming  $a_r = 0$  for all  $r > N$ . The finite set can be conveniently solved by using the Crout LU decomposition and forward and backward substitution [7]. In fact, as the decomposition is independent of order, the process can be done iteratively. At any order  $N$  the next row and column of  $A$  can be generated and the decomposition extended from order  $N - 1$  to  $N$ , making use of the  $N - 1$  decomposition already done. The forward substitution can also be similarly extended. The first step of the backward substitution, which is trivial, gives a reasonable estimate of the value of the coefficient  $a_N$  added to the set. The process can then be terminated if two successive coefficients are less than the required tolerance. Apart from the first few rows, the matrix  $A$ , is mainly diagonally dominant and, hence, the Crout reduction technique is highly stable. The programs performing this calculation have all been written in Algol 68 using the multiple length arithmetic facility, mlaritha [8]. This allows arithmetic to be done to any required precision. The above expansions were generated with a basic working precision of at least 36D. The Crout reduction process was terminated when two successive coefficients were less than  $10^{-33}$ . The resulting coefficients were then rounded to 30D and stored in a card image file and punched as an actual card deck,<sup>††</sup> the correctness of the coefficients as verified by summing the series at selected points, points for which the value of the series was known exactly or could be calculated by some other means. The adequacy of the working tolerances was also checked by running the same programs with the tolerances increased to a working precision of 48D and stopping criteria at  $10^{-42}$ , and no effect was found on the 30D coefficients.

**Conclusion.** The main motivation for the production of these expansions was their use as the basis for transportable function approximation routines in the NAG library. However, the expansions themselves may well find application directly in other aspects of numerical computing, and it is for that reason that they are presented here.

The Computer Centre  
The University of Birmingham  
Birmingham B15 2TT, England

<sup>††</sup>See footnote<sup>†</sup>.



1. C. W. CLENSHAW, "Chebyshev series for mathematical functions" in *National Physical Laboratory Mathematical Tables*, Vol. 5, Her Majesty's Stationery Office, London, 1962.
2. Y. L. LUKE, *The Special Functions and their Approximations*, Vols. 1, 2 (especially Vol. 2, pp. 323–324), Academic Press, New York, 1969.  
Y. L. LUKE, *Mathematical Functions and their Approximations*, Academic Press, New York, 1976, pp. 123–124.  
Y. L. LUKE, *Algorithms for the Computation of Mathematical Functions*, Academic Press, New York, 1978.
3. G. F. MILLER, Private communication.
4. J. L. SCHONFELDER, "The NAG library and its special functions chapter," *International Computing Symposium 1974* (A. Günther et al., Editors), North-Holland, Amsterdam; American Elsevier, New York, 1974.  
J. L. SCHONFELDER, "Special functions in the NAG library," *Software for Numerical Mathematics* (D. J. Evans, Editor), (Proc. Conf., 1973), Academic Press, London and New York, 1974, pp. 285–300.  
J. L. SCHONFELDER, "The production of special function routines for a multi-machine library," *Software—Practice and Experience*, vol. 6, 1976, pp. 71–82.
5. L. FOX & I. B. PARKER, *Chebyshev Polynomials in Numerical Analysis*, Oxford Univ. Press, London and New York, 1968.
6. J. L. SCHONFELDER, "Generation of high precision Chebyshev expansions." (Manuscript in preparation.)
7. J. H. WILKINSON & C. REINSCH, *Linear Algebra*, Handbook for Automatic Computation, Vol. 2 (Grundlehren der math. Wissenschaften, Band 186, vol. 2), Springer-Verlag, Berlin and New York, 1971.
8. J. L. SCHONFELDER & J. T. THOMASON, "Applications support by direct language extension—An arbitrary precision arithmetic facility in Algol 68," *International Computing Symposium 1975* (Proc. Sympos., Antibes, 1975), E. Gelenbe and D. Potier (Editors), North-Holland, Amsterdam; American Elsevier, New York, 1975.