A procedure is given for computation of e^x using tables of coefficients of the economized approximating polynomial over a range of positive and negative x. A related procedure that uses continued fractions is also discussed.

The exponential function was selected to test the effectiveness of table lookup methods in the computation of elementary functions. The number of multiplications or divisions required of standard methods is compared with the number required when table lookup is employed.

Computation of e^x with the use of large tables

by Kurt Spielberg

With the availability of continually increasing amounts of core storage, one might reasonably expect a trend toward programs which take advantage of large tables to reduce execution times. Examples of areas in which large tables have already been found necessary for the attainment of reasonable computation times are: large-scale weather forecasting, neutron transport and diffusion calculations, and bubble chamber analyses.

In this paper we investigate the effectiveness of large tables in the computation of the elementary function e^x. A number of suitably efficient approximations developed by means of Chebyshev techniques are given, so that the reader has a choice of various table sizes. The approximations are mainly economized polynomials (in the sense of References 1, 2, 3), but some continued-fraction approximations are provided for application in "short-arithmetic" routines.

The allocation of additional storage to tables is economically justified if it yields a sufficient reduction in computing time. Therefore, the table lookup procedure is compared with a standard procedure[4,5] that uses polynomials or continued fractions over a relatively large domain. Tables of modest size, say 256 entries, reduce the number of multiplications required in the central loop of the program from approximately 10 or 11 to 5 for high-precision and from 5 or 6 to 2 for low-precision subroutines.

Techniques similar to the one described in this paper have undoubtedly been used before, but only in conjunction with very small tables.

We may assume that the argument is given in normalized floating point form

$$x = b^s f \qquad \text{where} \qquad \frac{1}{b} \le f < 1$$

Here, b denotes the base of the number system, s the exponent (usually given in the characteristic of the floating-point number) and f the fraction. For the IBM 704, 709, 7090, 7094, we have $b = 2$; for SYSTEM/360, $b = 16$.

The reduction of the domain of x, which is fairly large, to a small domain in a derived variable is usually accomplished by multiplication of the argument with $\log_b e$. Thus,

$$e^x = b^{x \log_b e} = b^{M \pm F} = b^M b^{\pm F} \tag{1}$$

$$\text{where} \qquad -1 < F < 1 \qquad \text{or} \qquad 0 < F < 1$$

Here M stands for an integral part of $x\log_b e$ and F for the remaining fractional part. The domain of F, for which an approximation to b^F must be found, can easily be made to correspond to either $(-1, 1)$ or $(0, 1)$.

Even for the hexadecimal base of the SYSTEM/360, the choice $b = 2$ is of interest because one may wish to utilize approximations for 2^F or 2^{-F} only, rather than for 16^F or 16^{-F}. However, it then becomes necessary to examine the integer M modulo 4, e.g., by

$$2^M \cdot 2^F = 2^{4N+i} \cdot 2^F = 16^N \cdot 2^F \cdot 2^i \tag{2}$$

the multiplications with 2^i ($i = 0, 1, 2$, or 3) being executed explicitly. The multiplication with 16^N is, of course, accomplished by addition of N to the characteristic of 2^F, when the latter is available in floating-point hexadecimal format.

The choice $b = 16$ in Equation 1 has the merit of avoiding the clumsy procedure in Equation 2, but leads to the necessity of obtaining a good approximation to 16^F in, say, $(0, 1)$. In general, for a given degree of accuracy, approximations to b^F require an increasing number of coefficients as b increases.

We consider the functions $b^{\pm F} = b^{\pm(y+h)}$ in the interval $0 \le F < 1$, the argument F (for example, m binary digits or bits) being decomposed into the first n bits, denoted by y, and the remaining $m - n$ bits, denoted by h. The table consists of the value of $b^{\pm y}$, stored at the left endpoints

$$y_i = (i-1)\Delta \qquad \Delta = 2^{-n}; \qquad i = 1, 2, \cdots, 2^n$$

which describe a partitioning of $[0, 1)$ into 2^n subintervals.[6]

Expanding $b^{\pm(y+h)}$ about a typical point y_i, we have:

$$b^{\pm F} = b^{\pm y_i} \sum_{k=0}^{\infty} c_k h^k \tag{3}$$

$$c_k = (\pm 1)^k \frac{1}{k!} (\log_e b)^k$$

where h can be taken as a non-negative difference of F and y_i,

$$0 \leq h < 2^{-n} = \Delta$$

Let A_N stand for the absolute error (in magnitude) committed when the power series $\sum_{k=0}^{\infty} c_k h^k$ is approximated, say in accordance with a Chebyshev economization procedure,[1,2,3] by an N term polynomial $\sum_{k=0}^{N-1} a_k h^k$. The relative error in $b^{\pm F}$ is then given, after cancellation of $b^{\pm y_i}$, by $R_N = A_N / b^{\pm h}$. An upper bound on R_N for a given value of n is evidently given by:

$$R_N(n) \lesssim A_N b^{\Delta}$$

The factor b^{Δ} approaches 1 as n increases.

In the case of rapidly decreasing c_k we can obtain a good estimate for A_N, and with it an approximation for $R_N(n)$ as

$$R_N(n) \approx |2^{-2N+1} c_N 2^{-nN} b^{\Delta}| \tag{4}$$

This can be seen as follows. Introduce the variable x by means of $x = h/\Delta$, so that the domain of x is [0, 1). Under this transformation, the power series in Equation 3 goes into a power series $\chi(x)$,

$$\sum c_k h^k = \sum c_k \Delta^k x^k \equiv \chi(x)$$

To economize $\chi(x)$ we replace, in principle, the powers of x by the appropriate shifted Chebyshev polynomials $\tilde{T}_i(x)$ and discard all terms for which $i \geq N$. The error A_N is essentially given by the leading term discarded, which in view of

$$x^N = 2^{1-2N}[\tilde{T}_N(x) + \cdots]$$

becomes

$$|2^{1-2N} \tilde{T}_N(x) c_N \Delta^N| \lesssim |2^{1-2N} c_N \Delta^N|$$

From Equation 4 we can construct Table 1 which gives $R_N(n)$ as a function of n and N. This table allows a quick estimate of the number of coefficients required in the approximation polynomial for a given number of table entries and a specified relative error. For small values of N and n, the entries are not very reliable, since the c_k may not decrease sufficiently rapidly with increasing k. For the case $b = 16$, we set

$$|c_1| = \log_e 16 = 2.77258\ 87222\ 39781\ 23766\ 8928 \cdots$$

The approximations given in this paper were obtained from Chebyshev expansions in $T_n(x)$, valid in $-1 \leq x \leq +1$. This approach was chosen because a computer program, IB CTR, was available.[3] Basically, the program generates and tests polynomial-, rational-, and continued-fraction approximations $f^*(z)$ to given polynomials $f(z)$ in $-\alpha \leq z \leq \alpha$, with α arbitrary. The polynomial $f(z)$ consists of a sufficiently large number of terms of the power series of the function considered.

Table 1 Relative error as a function of n and N

n	N = 2	3	4	5	6	7	8	2^n number of table entries
4	(2) .224	(4) .322	(6) .350	(8) .307	(10) .219	(12) .136	(15) .731	16
8	(5) .741	(8) .669	(11) .453	(14) .246	(17) .110	(21) .428	(24) .145	256
9	(5) .184	(9) .831	(12) .282	(16) .762	(19) .172	(23) .333	(27) .563	512
10	(6) .459	(9) .103	(13) .175	(17) .238	(21) .268	(25) .259	(29) .219	1024
11	(6) .115	(10) .129	(14) .109	(19) .740	(23) .417	(27) .202	(32) .854	2048
12	(7) .286	(11) .162	(16) .684	(20) .231	(25) .652	(29) .158	(34) .334	4096
13	(8) .716	(12) .202	(17) .427	(22) .723	(26) .102	(31) .123	(36) .130	8192
14	(8) .179	(13) .252	(18) .267	(23) .226	(28) .159	(34) .963	(39) .509	16384

( ) Number of leading zeros

Given the approximation $f^*(z)$ in $-\alpha \leq z \leq \alpha$, we have to shift the interval to $[0, \Delta = 2\alpha]$ by means of the linear transformation

$$h = z + \alpha \rightarrow 0 \leq h \leq \Delta = 2\alpha \tag{5}$$

$$(16^h)^* = 16^\alpha \cdot (16^z)^* \tag{6}$$

If the absolute error in $(16^z)^*$ is denoted by A, the absolute error in $(16^h)^*$ becomes $16^\alpha \cdot A$. However, 16^α is quite close to 1 for the ranges considered.

In Tables 2, 3, 4, and 5 we give the coefficients a_i, $i = 0, 1, \cdots, N$, of a set of polynomial approximations to 16^h. They were selected to be useful in the two ranges of accuracy which are of most interest in the SYSTEM/360 context. The long range and short range arithmetic floating-point words of SYSTEM/360 have fractions of 56 and 24 bits respectively. Even for normalized arguments, the three leading bits may be zero, so that we will be interested in approximations of relative error between 2^{-57} and 2^{-54} ($0.7 \cdot 10^{-17}$ and $0.56 \cdot 10^{-16}$) and between 2^{-25} and 2^{-22} ($0.3 \cdot 10^{-7}$ and $0.24 \cdot 10^{-6}$), respectively. In the first range, the given coefficients were computed in floating-point double-precision mode on the 704. Since this arithmetic mode has no more than 54 significant bits, sufficient computational accuracy cannot be expected. For this contingency we are in a position to give "increments" Δa_i (the primary output of IB CTR) which

Table 2 Polynomial approximations to 16^h

First set of N numbers: approximation coefficients, $a_0, a_1, a_2, \cdots$
Second set of N numbers: increments, $\Delta a_0, \Delta a_1, \Delta a_2, \cdots$
Third set of N numbers: first coefficients of power series, $a_0^0, a_1^0, a_2^0, \cdots$

$n = 8$ $N = 6$ $R = (17).110$*			
.99999 99999 99999 9	2.7725 88722 23980 1	3.8436 24111 28507 4	3.5522 63020 64986 0
2.4622 08432 35539 5	1.3727 71781 50757 0		
.34086 50235 53789 (16)	−.87806 55976 11774 (13)	.77780 95084 19347 (10)	−.28297 06962 41580 (07)
.36134 25338 63430 (05)	.16773 26288 78453 (05)		
.99999 99999 99999 7	2.7725 88722 23988 9	3.8436 24111 20729 3	3.5522 63048 94693 0
2.4622 04818 93005 7	1.3727 70104 18128 2		
$n = 9$ $N = 5$ $R = (15).223$*			
.99999 99999 99999 9	2.7725 88722 23783 5	3.8436 24119 31080 5	3.5522 51538 47731 5
2.4689 17769 28121 9			
−.11394 38381 43381 (14)	.42774 67967 23492 (11)	−.47765 25780 51647 (08)	.16285 09419 48967 (05)
.90499 76719 34894 (06)			
1.0000 00000 00000 1	2.7725 88722 23355 8	3.8436 24124 08733 1	3.5522 49909 96789 5
2.4689 16864 28354 6			
$n = 10$ $N = 5$ $R = (17).237$*			
.99999 99999 99999 8	2.7725 88722 23965 9	3.8436 24112 34014 7	3.5522 60103 36372 8
2.4655 76924 42791 9			
−.35567 57125 06744 (16)	.26706 55503 59738 (12)	−.59656 09736 60950 (09)	.40701 78530 84898 (06)
.22594 33280 62708 (06)			
.99999 99999 99999 9	2.7725 88722 23939 2	3.8436 24112 93670 8	3.5522 59696 34587 5
2.4655 76698 48459 1			
$n = 12$ $N = 4$ $R = (16).684$			
.99999 99999 99999 8	2.7725 88722 24874 0	3.8436 23927 84778 0	3.5534 65449 60428 9
.47851 36546 89676 (15)	−.89595 70477 02938 (11)	.36693 35421 72204 (07)	.25440 35275 12205 (07)
.99999 99999 99999 1	2.7725 88722 25770 0	3.8436 23891 15442 6	3.5534 65424 16393 7
$n = 13$ $N = 4$ $R = (17).427$			
.99999 99999 99999 9	2.7725 88722 24090 0	3.8436 24065 47697 4	3.5528 64144 84317 3
.29903 12720 91388 (16)	−.11198 21947 52305 (11)	.91729 50693 76185 (08)	.63590 11990 48984 (08)
.99999 99999 99999 8	2.7725 88722 24202 0	3.8436 24056 30402 4	3.5528 64138 48416 1
$n = 18$ $N = 3$ $R = (17).616$			
.99999 99999 99999 8	2.7725 88722 21070 3	3.8436 44437 61998 1	
−.18486 65106 88162 (16)	.96923 10094 58279 (11)	.89576 28133 94029 (11)	
.99999 99999 99999 8	2.7725 88722 20101 1	3.8436 44437 61102 4	

* Used in testing exponential subroutine
( ) Number of leading zeros

have to be added to the corresponding terms in the power series, say a_i^0, to give rise to the coefficients of the approximation polynomials:

$$a_i = a_i^0 + \Delta a_i \tag{7}$$

The a_i^0 are the coefficients of the power series of $16^{\pm z} = 16^{\pm \alpha x}$, after the transformation given in Equations 5 and 6. The a_i can then be computed easily on a desk calculator. The addition

Table 3 Lower accuracy polynomial approximations to 16^h

N approximation coefficients, a_0, a_1, a_2, $\cdots$

$n = 2 \quad N = 6$	$R = (06).220$*				
.99999 98958	2.7726 18480	3.8422 52315	3.5752 50124	2.2902 30896	1.9405 89840
$n = 3 \quad N = 6$	$R = (08).201$				
.99999 99986	2.7725 89515	3.8435 50606	3.5547 48806	2.4244 75201	1.6257 23021
$n = 3 \quad N = 5$	$R = (06).139$				
1.0000 00095	2.7725 50756	3.8460 31204	3.4991 82789	2.9325 10894	
$n = 4 \quad N = 5$	$R = (08).333$				
1.0000 00003	2.7725 86526	3.8439 03933	3.5398 25509	2.6861 00767	
$n = 5 \quad N = 4$	$R = (07).210$				
.99999 99809	2.7726 08210	3.8405 18784	3.7099 70193		
$n = 6 \quad N = 4$	$R = (08).123$				
.99999 99988	2.7725 91114	3.8428 60363	3.6301 53834		
$n = 7 \quad N = 3$	$R = (07).548$				
1.0000 00053	2.7724 65732	3.8855 16414			
$n = 8 \quad N = 3$	$R = (08).673$				
1.0000 00007	2.7725 58104	3.8645 04054			

* Used in testing exponential subroutine
( ) Number of leading zeros

of the increments Δa_i from Equation 7 produces coefficients of sufficient accuracy. It should be noted that this will be necessary for the leading coefficients only, since the terms of higher order $a_k h^k$ will be small because of the smallness of h.

For the convenience of the reader, we give the a_i^0 to 16-digit accuracy whenever the Δa_i are given. In Tables 6 and 7, we also supply the leading coefficients of the power series expansion of $16^{\pm x}$ and a set of powers $16^{\pm\Delta}$. These tables are intended to assist the reader in deciding whether or not his desk-calculator computations are on the right track.

For any combination of n and N, the constants in the tables are given in the order $a_0, a_1, \cdots, a_{N-1}$; $\Delta a_0, \Delta a_1, \cdots, \Delta a_{N-1}$; $a_0^0, a_1^0, \cdots, a_{N-1}^0$. The fact that the first coefficients a_0, Δa_0, a_0^0 do not, in general, satisfy Equation 7 reflects, in the author's estimation, the inaccuracy of the double-precision arithmetic utilized in arriving at the table entries. Digits in parentheses denote the number of leading zeros after the decimal point.

Finally, Table 8 gives continued fraction approximations to $16^{\pm z}$, for low-precision subroutines. Here z lies in the interval $-\Delta/2 \leq z \leq \Delta/2$. The transformation to $0 \leq h \leq \Delta$ is trivial and can be left to the reader. It would not be meaningful to give continued fraction approximations for the larger precision range of interest, since again the double-precision arithmetic used in arriving at these approximations was not accurate enough. In

Table 4 Polynomial approximations to 16^{-h}

First set of N numbers: approximation coefficients, $a_0, a_1, a_2, \cdots$
Second set of N numbers: increments, $\Delta a_0, \Delta a_1, \Delta a_2, \cdots$
Third set of N numbers: first coefficients of power series, $a_0^0, a_1^0, a_2^0, \cdots$

$n = 8 \quad N = 6 \quad R = (17).109$*			
.99999 99999 99999 8	−2.7725 88722 23976 0	3.8436 24111 28563 9	−3.5522 62889 02641 5
2.4622 08686 50661 3	−1.3579 84301 98948 8		
.33772 37682 39469 (16)	−.87032 74451 87467 (13)	.77153 25688 77960 (10)	−.28112 51574 66996 (07)
.36069 08954 34581 (05)	−.16592 58152 13235 (05)		
.99999 99999 99999 8	−2.7725 88722 23967 3	3.8436 24111 20848 7	−3.5522 62860 91390 0
2.4622 05079 59766 0	−1.3579 82642 73133 6		
$n = 9 \quad N = 5 \quad R = (15).221$*			
.99999 99999 99999 7	−2.7725 88722 23784 5	3.8436 24103 41591 9	−3.5522 51583 72241 5
2.4555 84189 95559 8			
.11343 42174 27525 (14)	−.42598 16078 12399 (11)	.47603 86921 92820 (08)	−.16267 46640 40865 (05)
.90011 01627 60475 (06)			
.99999 99999 99998 6	−2.7725 88722 23358 5	3.8436 24098 65553 2	−3.5522 49956 97577 5
2.4555 83289 84543 5			
$n = 10 \quad N = 5 \quad R = (17).237$*			
.99999 99999 99999 7	−2.7725 88722 23965 9	3.8436 24110 35329 0	−3.5522 60109 01936 1
2.4589 10142 70730 7			
.35487 94305 64288 (16)	−.26651 39290 17594 (12)	.59555 22948 14314 (09)	−.40679 75042 86287 (06)
.22533 23899 75764 (06)			
.99999 99999 99999 7	−2.7725 88722 23939 3	3.8436 24109 75773 9	−3.5522 59702 22185 7
2.4589 09917 37491 7			
$n = 12 \quad N = 4 \quad R = (16).683$			
.99999 99999 99999 6	−2.7725 88722 23082 6	3.8436 23927 94091 4	−3.5510 60917 26426 2
.47825 92242 87784 (15)	−.89555 91370 62224 (11)	.36687 14529 59622 (07)	−.25423 13796 41909 (07)
.99999 99999 99999 1	−2.7725 88722 22187 0	3.8436 23891 25377 0	−3.5510 60891 84112 3
$n = 13 \quad N = 4 \quad R = (17).427$			
.99999 99999 99999 8	−2.7725 88722 23866 1	3.8436 24065 48861 6	−3.5516 61878 69683 0
.29895 17625 91314 (16)	−.11195 73253 37243 (11)	.91721 74578 54901 (08)	−.63568 60142 14195 (08)
.99999 99999 99999 8	−2.7725 88722 23754 1	3.8436 24056 31644 1	−3.5516 61872 33997 0
$n = 18 \quad N = 3 \quad R = (17).616$			
.99999 99999 99999 8	−2.7725 88722 20170 4	3.8436 03785 19664 4	
.18486 51257 19881 (16)	−.96922 75924 12394 (11)	.89575 33393 31146 (11)	
.99999 99999 99999 8	−2.7725 88722 20101 1	3.8436 03785 18768 6	

* Used in testing exponential subroutine
( ) Number of leading zeros

the case of continued fractions, the calculations are too complex to allow a modification of the coefficients on a desk calculator.

The given approximations have the form

$$16^{\pm z} = H_0 + \cfrac{G_1}{z + H_1 + \cfrac{G_2}{z + H_2 + \cfrac{G_3}{z + H_3}}}$$

for $N = 7$. The coefficients are given in the order $H_0, G_1, H_1, G_2 \cdots$,

Table 5 Lower accuracy polynomial approximations to 16^{-h}

N approximation coefficients, a_0, a_1, a_2, $\cdots$

$n = 2$ $N = 6$	$R = (06)\ 110$*				
.99999 99452	−2.7725 72806	3.8428 71828	−3.5391 74835	2.3579 84098	−.97029 49200
$n = 3$ $N = 6$	$R = (08).156$				
.99999 99990	−2.7725 88143	3.8435 69674	−3.5503 87128	2.4328 37713	−1.1495 59772
$n = 3$ $N = 5$	$R = (07).932$				
.99999 99305	−2.7725 60734	3.8418 15544	−3.5110 95049	2.0735 98339	
$n = 4$ $N = 5$	$R = (08).279$				
.99999 99977	−2.7725 86836	3.8433 81565	−3.5413 09708	2.2587 32506	
$n = 5$ $N = 4$	$R = (07).193$				
.99999 99823	−2.7725 70608	3.8407 14159	−3.4020 57667		
$n = 6$ $N = 4$	$R = (08).118$				
.99999 99989	−2.7725 86416	3.8428 84780	−3.4762 47221		
$n = 7$ $N = 3$	$R = (07).535$				
.99999 99476	−2.7724 67787	3.8022 57809			
$n = 8$ $N = 3$	$R = (08).666$				
.99999 99934	−2.7725 58361	3.8828 75667			

* Used in testing exponential subroutine
( ) Number of leading zeros

the upper and lower signs correspond to 16^{+z} and 16^{-z}, respectively. The approximation to $16^{\pm z}$ for $n = 5$, $N = 4$, however, has the form

$$16^{+z} = H_0 + Az + \frac{G_1}{z + H_1}$$

where A is listed in last place.

All approximations were generated by the IB CTR program from functions of the form $f(\alpha x)$, where $-1 \leq x \leq 1$ and

$$\alpha = \frac{\Delta}{2} = 2^{-n-1}$$

They were compared with the power series at 100 equispaced points in $-1 \leq x \leq 1$. In all cases the errors did not exceed the a priori estimates computed by IB CTR. The errors (the R's) quoted in the tables, take the transformations in Equations 5 and 6 into account. In the case of the continued fractions, the error estimates are somewhat too pessimistic. The user of the approximations will find them to be as much as five to ten times more accurate.

Table 6 Power series coefficients of $16^{\pm x}$

i	a_i^0			
0	1.00000	00000	00000	
1	$\pm$2.77258	87222	39781	
2	3.84362	41113	45610	
3	$\pm$3.55226	29545	48579	
4	2.46224	10515	52888	
5	$\pm$1.36535	63541	94271	
6	.63092	86049	12909	6

The approximations indicated in the tables by an asterisk were used for testing purposes in exponential subroutines for SYSTEM/360. These subroutines were tested in various ranges of the argument at 100 points for each range. The absolute error of $(1 - e^x e^{-x})^*$, which can easily be shown to be as much as $R^+ + R^-$, was taken as a measure of the error (the terms $R^\pm$ represent the relative error for e^x when the approximation of $16^{\pm h}$ is used). For the other approximations, several spot checks were made using a desk calculator. In all cases, the results were found to be as accurate as could be expected. That is, the errors were either less than $R^+ + R^-$, or could be explained by inaccuracies in the table entries.

The appendices reproduce assembly listings of two SYSTEM/360 codes for e^x which illustrate some of the logical procedures necessary for the table lookup. In programming the tests described above, the inversions were suppressed.

A word of caution about the construction of the table is in order. It is clear that the results of the lookup subroutines will be no more accurate than the table entries. In general, computing these entries on the SYSTEM/360 for use in the lookup subroutines is not quite adequate due to accumulation of round-off error. For convenience, nonetheless, the table entries were calculated on SYSTEM/360 with the aid of the 7090 SUPPAK simulator. Therefore, for the more precise approximations, the tests cannot be considered conclusive as far as the last 2 digits are concerned. This difficulty did not arise for the less precise approximations of Tables 3, 5, and 8.

In order to assess the efficiency of the table lookup approximations, we compare them with standard approximations for 2^x in the interval $-1/2 \leq x \leq 1/2$. The choice of 2^x (rather than 16^x) for the standard approximations is indicated because of faster convergence of the power series. This choice leads, as already discussed, to some coding difficulties which should be at least as great as those encountered in the table-lookup procedure advocated in this paper.

As a measure of efficiency, we may safely take P_M (or P_D), the number of multiplications (or divisions) required in coding

Table 7 Powers of 16: $16^{\pm\Delta}$

n	$\Delta = 2^{-n}$			$16^{+\Delta}$			$16^{-\Delta}$			
5	(1)	.3125		1.09050	77326	65258	.91700	40432	04671	2
6	(1)	.15625		1.04427	37824	27414	.95760	32806	98573	6
7	(2)	.78125		1.02189	71486	54117	.97857	20620	87700	1
8	(2)	.39062	5	1.01088	92860	51700	.98922	80131	93975	5
9	(2)	.19531	25	1.00542	99011	12803	.99459	94234	83633	2
10	(3)	.97656	25	1.00271	12750	50202	.99729	60560	85470	1
11	(3)	.48828	125	1.00135	47198	92108	.99864	71128	90970	1
12	(3)	.24414	0625	1.00067	71306	93066	.99932	33275	02650	7
13	(3)	.12207	03125	1.00033	85080	52682	.99966	16064	96243	6
14	(4)	.61035	15625	1.00016	92397	05302	.99983	07889	31929	1

( ) Number of leading zeros

Table 8 Continued fraction approximations to $16^{\pm x}$

Order of coefficients for $n = 2, 3, 4, 7, 8$: $H_0, G_1, H_1, G_2, \cdots$
For $n = 5$: H_0, G_1, H_1, A

$n = 2$ $N = 7$ $R^1 = (08).523$ $R^2 = (08).192$			
$-.98915\ 47947\ 11220$	$\mp 8.5732\ 34616\ 69475$	$\mp 4.3296\ 87747\ 97325$	$6.5159\ 25907\ 06117$
$\pm.39311\ 70161\ 43720$ (02)	$1.3011\ 89733\ 82176$	$\pm.39338\ 30795\ 90640$ (02)	
$n = 3,$ $N = 5$ $R^1 = (06).141$ $R^2 = (07).827$			
$1.0015\ 03444\ 34067$	$\pm 4.3373\ 01763\ 73720$	$\mp 2.1659\ 36834\ 86781$	$1.5620\ 03161\ 78210$
$\mp.54184\ 54955\ 71694$ (03)			
$n = 4$ $N = 5$ $R^1 = (08).409$ $R^2 = (08).256$			
$1.0003\ 75482\ 03752$	$\pm 4.3303\ 87266\ 57632$	$\mp 2.1645\ 16326\ 84734$	$1.5612\ 70867\ 60306$
$\mp.13540\ 10962\ 99561$ (03)			
$n = 5$ $N = 4$ $R^1 = (07).560$ $R^2 = (08).639$*			
$-3.5001\ 17190\ 31934$	$\mp 4.8694\ 12933\ 22660$	$\mp 1.0820\ 63582\ 05132$	$\mp 1.3862\ 40087\ 95073$
$n = 7$ $N = 3$ $R^1 = (06).186$ $R^2 = (07).268$			
$-.99999\ 51122\ 68675$	$\mp 1.4426\ 98566\ 05599$	$\mp.72135\ 10459\ 38893$	
$n = 8$ $N = 3$ $R^1 = (07).232$ $R^2 = (08).333$*			
$-.99999\ 87781\ 25393$	$\mp 1.4426\ 95922\ 24857$	$\mp.72134\ 84018\ 24546$	

* Used in testing exponential subroutine
( ) Number of leading zeros
R^1, theoretical relative error; R^2, numerical relative error

the subroutine. The multiplication for the table lookup will be traded off against the multiplication by 2^i shown in Equation 2. The time of the lookup itself, which requires one access to a stored table, depends upon the operational context, but usually will be small.

The following numbers are taken from standard approximations also produced by IB CTR. (N' identifies continued fraction approximations.)

$R = (17)\ .31 \quad N = 12 \quad P_M = 11$

$R = (15)\ .22 \quad N = 11 \quad P_M = 10$

$R = (08)\ .19 \quad N = 7 \quad P_M = 6$

$R = (09)\ .47 \quad N' = 7 \quad P_D = 3$

$R = (07)\ .78 \quad N = 6 \quad P_M = 5$

$R = (08)\ .79 \quad N' = 6 \quad P_M = 1 \quad P_D = 2$

For the approximations given in this paper, the corresponding numbers, of course, depend upon the size of the table. We summarize the results as follows:

High precision

When the number of table entries (2^n) increases from 256 to 8192, P_M is found to decrease from 5 to 3.

Low precision

When the number of table entries (2^n) increases from 4 to 256, P_M is found to decrease from 5 to 2 and P_D from 3 to 1.

For small values of n one would, of course, keep the tables in high speed memory and could expect closer agreement in efficiency between the two approaches compared.

CITED REFERENCES AND FOOTNOTE

1. C. Lanczos, *Applied Analysis*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 438–468 (1956).
2. "Tables of Chebyshev polynomials $S_n(x)$ and $C_n(x)$," *Applied Mathematics Series*, No. 9, National Bureau of Standards (December 1952).
3. K. Spielberg, "The representation of power-series in terms of polynomials, rational approximations, and continued fractions," *Journal of the Association for Computing Machinery* **8**, 613–627 (1961).
4. E. G. Kogbetliantz, "Computation of e^N for $-\infty < N < +\infty$ using an electronic computer," *IBM Journal of Research* and *Development* **1**, 2, 110–115 (April 1957).
5. D. Cantor, G. Estrin, and R. Turn, "Logarithmic and exponential function evaluation in a variable structure computer," *IRE Transactions on Electronic Computers EC* **11**, 155–164 (April 1962).

6. The convention used to represent the domain of an argument, say F, is that (0, 1) indicates $0 < F < 1$ and [0, 1) indicates $0 \leq F < 1$.

```
                                  *
001F40  79 00 C 0C8               EXP    CE    E1,MAX
001F44  47 2F 0 000                      BC    2,0(R15)        ERROR RETURN, BRANCH ON FIRST OPERAND HIGH
001F48  79 00 C 0D0                      CE    E1,MIN
001F4C  47 20 C 016                      BC    2,CONT          BRANCH ON FIRST OPERAND HIGH
001F50  1B 00                            SR    E1,E1           RETURN WITH NORMAL ZERO
001F52  47 FF 0 004                      BC    15,4(R15)
001F56  6C 00 C 0D8               CONT   MD    E1,LOGE
001F5A  68 20 C 0E0                      LD    E2,ZERO1        ZERO WITH CHARACTERISTIC 64 + 14
001F5E  2E 20                            AWR   E2,E1           GENERATES UNNORMALIZED M IN E2
001F60  47 80 C 0A0                      BC    8,P1            BRANCH ON M ZERO
001F64  60 20 C 0B8                      STD   E2,M            M + 4 CONTAINS FIXED POINT M, M CONTAINS
                                  *                            SIGN OF ARGUMENT
001F68  6A 20 C 0E8                      AD    E2,ZERO2        NORMALIZE M
001F6C  2B 02                            SDR   E1,E2           M + F - F = M (MAY BE NEGATIVE)
001F6E  6E 00 C 0E8               P3     AW    E1,ZERO2        FIXES POINT OF F AFTER BIT 7
001F72  60 00 C 0C0                      STD   E1,C1           F WITH POINT FIXED
001F76  94 00 C 0C0                      NI    C1,00           SET CHARACTERISTIC TO ZERO
001F7A  58 10 C 0C0                      L     R1,C1           ABSOLUTE VALUE OF F, POINT AFTER BIT 7
001F7E  88 10 0 010                      SRL   R1,16           FIRST 8 BITS OF F(=Y+H) FOR TABLE LOOK
001F82  89 10 0 003                      SLL   R1,3            ADDRESS FULL WORDS (8 BYTES LONG EACH)
001F86  94 00 C 0C1                      NI    C1+1,CO         SET Y ZERO
001F8A  96 40 C 0C0                      OI    C1,64           16,0 IN CHAR.,, UNNORMALIZED H FOR POLYN
001F8E  68 20 C 0C0                      LD    E2,C1           H
001F92  6A 20 C 0E8                      AD    E2,ZERO2        NORMALIZE H
001F96  98 57 C 0A8               P2     LM    R5,R7,XW        SET 3 REGISTERS TO 0,8,46
001F9A  68 00 C 0F0                      LD    E1,C5
001F9E  2C 02                     P4     MDR   E1,E2           EVALUATE POLYNOMIAL
001FA0  6A 05 C 0F8                      AD    E1,C4(R5)
001FA4  87 56 C 05E                      BXLE  R5,R6,P4
001FA8  6C 01 C 240                      MD    E1,TAB(R1)      MULTIPLICATION WITH TABLE VALUES GIVES 16,F
001FAC  91 FF C 0B9                      TM    M+1,X'FF'       TEST WHETHER M ZERO
001FB0  47 50 C 08C                      BC    5,P6            YES
001FB4  58 20 C 0BC                      L     R2,M+4          ABSOLUTE VALUE OF M (MAY BE ZERO)
001FB8  89 20 0 018                      SLL   R2,24           SHIFT ABS(H) INTO CHARACTERISTIC
001FBC  70 00 C 0C0                      STE   E1,C1
001FC0  5A 20 C 0C0                      A     R2,C1           ADD M TO CHAR. OF 16,F
001FC4  50 20 C 0C0                      ST    R2,C1
001FC8  78 00 C 0C0                      LE    E1,C1           FINAL RESULT IF X WAS POSITIVE
001FCC  91 80 C 0B8               P6     TM    M,X'80'         TEST SIGN OF ARGUMENT
001FD0  47 8F 0 004                      BC    8,4(R15)        IF POSITIVE , RETURN
001FD4  68 20 C 118                      LD    E2,FONE         IF NEGATIVE, INVERT
001FD8  2D 20                            DDR   E2,E1
001FDA  28 C2                            LDR   E1,E2
001FDC  47 FF 0 004                      BC    15,4(R15)       RETURN
001FE0  60 00 C 0B8               P1     STD   E1,M            NORMALIZED NON-ZERO NUMBER, SIGN OF ARG.
001FE4  47 F0 C 02E                      BC    15,P3

                                  *            CONSTANTS AND STORAGE
                         000000   E1     EQU   0
                         000002   E2     EQU   2
                         000004   E3     EQU   4
                         000006   E4     EQU   6
                         000001   R1     EQU   1
                         000002   R2     EQU   2
                         000003   R3     EQU   3
                         000004   R4     EQU   4
                         000005   R5     EQU   5
                         000006   R6     EQU   6
                         000007   R7     EQU   7
                         00000C   R12    EQU   12
                         00000F   R15    EQU   15
001FE8  00000000                  XW     DC    A(0)
001FEC  00000008                         DC    A(8)
001FF0  00000020                         DC    A(32)
001FF8                            M      DS    D
002000                            C1     DS    D
002008  +42.AD000000000000        MAX    DC    D'173'              173
002010  -42.AD000000000000        MIN    DC    D'-173'             -173
002018  +40.50551D94AE0BF8        LOGE   DC    D'.36067376022224085'
002020  4E00000000000000          ZERO1  DC    X'4E00000000000000'   ZERO WITH CHAR. 64+14
002028  4000000000000000          ZERO2  DC    X'4000000000000000'
002030  +41.15F6DF8B272590        C5     DC    D'1.372771781507570'
002038  +41.276534AB4E6DFA        C4     DC    D'2.462208432355395'
002040  +41.38D611BFC7B37E               DC    D'3.552263020649860'
002048  +41.3D7F7BFF016221               DC    D'3.843624111285074'
002050  +41.2C5C85FDF47437               DC    D'2.772588722239801'
002058  +41.10000000000000        FONE   DC    D'1.C'          ALSO FIRST COEFFICIENT
                                         EJECT
```

```
                                 *
                                 *                  COMPUTE E,X AS E,X IF X NEGATIVE, AS E,-X IF X POS.
002060   79 00 C 1E8             EXPM    CE         E1,MAXM
002064   47 2F 0 000                     BC         2,0(R15)        ERROR RETURN, BRANCH ON FIRST OPERAND HIGH
002068   79 00 C 1F0                     CE         E1,MINM
00206C   47 20 C 136                     BC         2,CONTM         BRANCH ON FIRST OPERAND HIGH
002070   1B 00                           SR         E1,E1           RETURN WITH NORMAL ZERO
002072   47 FF 0 004                     BC         15,4(R15)
002076   6C 00 C 1F8             CONTM   MD         E1,LOGEM
00207A   68 20 C 200                     LD         E2,ZERM1        ZERO WITH CHARACTERISTIC 64+14
00207E   2E 20                           AWR        E2,E1           GENERATES UNNORMALIZED M IN E2
002080   47 80 C 1C2                     BC         8,P1M           BRANCHES ON M ZERO
002084   60 20 C 1D8                     STD        E2,MM           M+4 CONTAINS FIXED POINT M, M CONTAINS
                                 *                                  SIGN OF ARGUMENT
002088   6A 20 C 208                     AD         E2,ZERM2        NORMALIZE M
00208C   2B 02                           SDR        E1,E2           M+F - M = F (MAY BE NEGATIVE)
00208E   6E 00 C 208             P3M     AW         E1,ZERM2        FIXES POINT OF F AFTER BIT 7
002092   60 00 C 1E0                     STD        E1,C1M          F WITH POINT FIXED
002096   94 00 C 1E0                     NI         CIM,00          CHARACTERISTIC AND SIGN SET ZERO
00209A   58 10 C 1E0                     L          R1,C1M          ABSOLUTE VALUE OF F, POINT AFTER BIT 7
00209E   88 10 0 010                     SRL        R1,16           SHIFT OUT H
0020A2   89 10 0 003                     SLL        R1,3            ADDRESS FULL WORDS (8 BYTES LONG EACH)
0020A6   94 00 C 1E1                     NI         C1M+1,00        SET Y ZERO
0020AA   96 40 C 1E0                     OI         C1M,64          16,0 IN CHAR.,, UNNORMALIZED H FOR POLYN
0020AE   68 20 C 1E0                     LD         E2,C1M          UNNORMALIZED H
0020B2   6A 20 C 208                     AD         E2,ZERM2        NORMALIZE H
0020B6   98 57 C 1CC                     LM         R5,R7,XWM
0020BA   68 00 C 210                     LD         E1,C5M          COEFFICIENTS ARE FOR FUNCTION E,-X
0020BE   2C 02                   P4M     MDR        E1,E2
0020C0   6A 05 C 218                     AD         E1,C4M(R5)
0020C4   87 56 C 17E                     BXLE       R5,R6,P4M
0020C8   6C 01 C A40                     MD         E1,TABM(R1)     TABLE CONTAINS VALUES OF 16,-I/256
0020CC   91 FF C 1D9                     TM         MM+1,X'FF'      TEST WHETHER M ZERO
0020D0   47 50 C 1AE                     BC         5,P6M           YES
0020D4   58 20 C 1DC                     L          R2,MM+4         ABSOLUTE VALUE OF M (MAY BE ZERO)
0020D8   89 20 0 018                     SLL        R2,24
0020DC   13 22                           LCR        R2,R2           SUBTRACT ABSOLUTE VALUE OF M
0020DE   70 00 C 1E0                     STE        E1,C1M
0020E2   5A 20 C 1E0                     A          R2,C1M          ADD -M TO CHAR. TO GENERATE 16,-F-M
0020E6   50 20 C 1E0                     ST         R2,C1M
0020EA   78 00 C 1E0                     LE         E1,C1M          FINAL RESULT IF X WAS NEGATIVE
0020EE   91 80 C 1D8             P6M     TM         MM,X'80'        TEST SIGN OF ARGUMENT
0020F2   47 1F 0 004                     BC         1,4(R15)        IF NEGATIVE, BRANCH
0020F6   68 20 C 238                     LD         E2,FONEM        IF POSITIVE, INVERT
0020FA   2D 20                           DDR        E2,E1
0020FC   28 02                           LDR        E1,E2
0020FE   47 FF 0 004                     BC         15,4(R15)
002102   60 00 C 1D8             P1M     STD        E1,MM           NORMALIZED NON-ZERO NUMBER, SIGN OR ARG.
002106   47 F0 C 14E                     BC         15,P3M

                                 *                  CONSTANTS AND STORAGE

00210C   00000000                XWM     DC         A(C)
002110   00000008                        DC         A(8)
002114   00000020                        DC         A(32)
002118                           MM      DS         D
002120                           C1M     DS         D
002128   +42.AD000000000000      MAXM    DC         D'173'
002130   -42.AD000000000000      MINM    DC         D'-173'
002138   +40.50551D94AE0BF8      LOGEM   DC         D'.3606737602222405'
002140   4E00000000000000        ZERM1   DC         X'4E00000000000000'
002148   4000000000000000        ZERM2   DC         X'4000000000000000'
002150   -41.15BA4DBF586B79      C5M     DC         D'-1.357984301989488'
002158   +41.276534EF87917D      C4M     DC         D'2.462208686506613'
002160   -41.38D6119C729B5C              DC         D'-3.552262889026415'
002168   +41.3D7F7DFF016C12              DC         D'3.843624111285639'
002170   -41.2C5C85FDF4737F              DC         D'-2.772588722239760'
002178   +41.10000000000000      FONEH   DC         D'1.'
002180                           TAB     DS         256D
002980                           TABM    DS         256D
                                         END        TPO0            WILL REQUIRE BASE REGISTER
```