PUBLIKACIJE ELEKTROTEHNIČKOG FAKULTETA UNIVERZITETA U BEOGRADU
PUBLICATIONS DE LA FACULTÉ D'ÉLECTROTECHNIQUE DE L'UNIVERSITÉ À BELGRADE

SERIJA: MATEMATIKA I FIZIKA — SÉRIE: MATHÉMATIQUES ET PHYSIQUE

№ 461 — № 497 (1974)

# 464. COMPUTER PROGRAMS FOR COMPUTATION OF THE HYPERBOLIC TANGENT FUNCTION*

*Dušan V. Slavić* and *Laslo L. Kraus*

In the present paper programs for computation of the hyperbolic tangent function of a real variable for the standard and extended precision of the computer IBM 1130 are suggested.

The hyperbolic tangent function is one of the few elementary mathematical functions which are infallibly present in a subroutine library of any computer. The large-scale application of computers requires a constant improvement of computer programs. The intention of this paper is to substitute the present programs for computation of the function $x \mapsto \operatorname{th} x$ by programs having incomparably higher accuracy and greater speed of computation.

Due to the specific manner of representation of real numbers in computers and to the properties of the hyperbolic tangent function, it is useful to adopt the next algorithm for computation of this function:

$$\operatorname{th} x \approx \begin{cases} x & (|x|<a), \\ R(x) & (a \leqq |x| < b), \\ T(x) & (b \leqq |x| < c), \\ \operatorname{sgn} x & (c \leqq |x|). \end{cases} \tag{1}$$

Here $R(x)$ is a polynomial or a rational function suitable for the desired accuracy of the given computer, and $T(x)$ is a transcendental expression based on the equation:

$$\operatorname{th} x = \frac{e^{2x}-1}{e^{2x}+1}. \tag{2}$$

Parameters $a$, $b$, $c$, depend on the computer characteristics, primarily on the desired accuracy. Having in view the requested accuracy, endeavours should be made to choose $a$ as large as possible and $b$ and $c$ as small as possible, in order to obtain the highest possible speed of computation of the function $x \mapsto \operatorname{th} x$.

The boundary between the rational and the transcendental approximation of the function $x \mapsto \operatorname{th} x$ is determined differently in various papers. Starting from the general mathematical considerations the boundary $b = 0.5 \log 3 =$

* Received December 17, 1973.

$=0.54930614433$ is obtained leading to numerous approximative expressions for $\operatorname{th} x$ in the segment $(-0.5 \log 3,\ 0.5 \log 3)$; see for example [1] (pp. 160, 183—185, 188—190, 196, 204) and [2] (pp. 31, 39). For this value of $b$ it is useful to adopt the transcedental approximation in the form (see for example [3], p. 99):

$$\operatorname{th} x=\left(1-\frac{2}{e^{2|x|}+1}\right)\operatorname{sgn} x .$$

Depending on the hardware and the software of the particular computer, the domain of application of the transcendental approximation can be extended. So, for example, the company CII uses the next formula [4]:

$$\operatorname{th} x=\begin{cases}2\left(0.5-1./(e^{2x}+1.)\right) & (x>b)\\ 2\left(1./(e^{-2x}+1.)-0.5\right) & (x<-b)\end{cases}$$

where $b=0.5 \log 2=0.34657359$.

Computer CDC 3600 uses formula (2) with $b=0.3$, see [4], p. 98.

When the characteristic of a real number is for the base 2, it is convenient to choose the boundaries of the form $2^{-m}$ ($m$ being an integer), because then the determination of the region wherein the particular argument lies is very simple and the selection of the approximation is easy. For this reason the value $b=0.5$ is adopted in [5]. In that paper for the extended precision of the computer IBM 1130 the formula

$$\operatorname{th} x=\left(0.25-\frac{2.}{e^{2|x|}+1.}+0.75\right)\operatorname{sgn} x \tag{3}$$

is used, ensuring the highest possible accuracy. However, formula (3) is unapplicable for $b<0.5$. Further, it is not necessary to make programs with possible maximum accuracy. Minor computational errors smaller than those occurring in computation of other elementary functions are to be accepted; especially if it contributes to greater speed of computation.

Company Hewlett-Packard has adopted for their computer 2100 A the value $b=1/8$ and formula (2) (see for example [6]). More than hundred thousand of tests have shown that formula (2) and value $b=1/8$ can be used also in the IBM 1130 for computations in extended precision. Moreover, with some modifications, in standard precision of the IBM 1130 this formula can be used even for $b=1/32$.

Such small values of the parameter $b$ enable relatively simple realisation of the rational approximation of the function $x \mapsto \operatorname{th} x$. Instead of the complicated approximations by rational functions based on the simplifications of the continued fraction developments

$$\operatorname{th} x=\frac{x}{1+}\,\frac{x^2}{3+}\,\frac{x^2}{5+}\,\frac{x^2}{7+}\,\frac{x^2}{9+}\cdots$$

or

$$\operatorname{th} x=x-\frac{x^3}{3+x^2+}\,\frac{x^2}{5+}\,\frac{x^2}{7+}\,\frac{x^2}{9+}\cdots,$$

see [7], p. 85, or on high order polynomial (as for example by CDC 3600), the error of the polynomial approximation can be reduced to the least possible value even by 7th and 3rd order polynomials for the extended and standard precision of the IBM 1130, respectively.

By careful writing of the program, the 7th order polynomial for the extended precision can be realized by four multiplications and three additions only (i.e., with 4 calls to the program EMPY and 3 calls to the program EADD). The 3rd order polynomial for the standard precision is realized by integer arithmetic operations only (i.e., without any call to the programs FMPY and FADD).

The boundary $a$ between the monomial and polynomial approximations is determined from the request that $(a-\operatorname{th} a)/a$ is less than the allowed error of computation. That means that $a=2^{-14}$ for the extended precision and $a=2^{-12}$ for the standard precision of the IBM 1130.

The boundary $c$ between the transcendental approximation and the asymptotic region is determined from the request that $1-\operatorname{th} c$ is less than the allowed error of the computation. That means that $c=16$ for the extended precision and $c=8$ for the standard precision of the IBM 1130.

In [8], Chapter 2, p. 88, the next values for these parameters: $a=0$, $b=0$, $c=32$, are adopted. See [9], pp. 96—100.

By [10] the hyperbolic tangent function in the IBM 1130 (Disk Monitor System, Version 2, Modification 11) is computed by the next algorithm:

$$»\operatorname{Tanh}(x)=\frac{e^{2x}-1}{e^{2x}+1}$$

$$\begin{array}{lll} \text{for} & x \geqq 32 & \operatorname{Tanh}(x)=1 \\ & x \leqq -32 & \operatorname{Tanh}(x)=-1« , \end{array}$$

compare with [10], p. 105.

However, investigations of the programs ETANH/ETNH and FTANH/FTNH have shown other algorithms.

The algorithm for the extended precision is

$$\operatorname{th} x=\begin{cases} \dfrac{(e^{2x}+1.)+(-2.)}{e^{2x}+1.} & (|x|<16), \\ 1. & (x \geqq 16), \\ -1. & (x \leqq -16). \end{cases}$$

The algorithm for the standard precision is

$$\operatorname{th} x=\begin{cases} \dfrac{(e^{2x}+1.)+(-2.)-\delta\dfrac{(e^{2x}+1.)+(-2.)-\delta_1}{(e^{2x}+1.)-\delta_2}}{(e^{2x}+1.)-\delta} & (|x|<16), \\ 1. & (x \geqq 16), \\ -1. & (x \leqq -16), \end{cases}$$

where:

$\delta$ — is the value of the binary digits in the second half of the second word of the mantissa of the expression $(e^{2x}+1.)$,

$\delta_1$ — is the value of the binary digits in the second word of the mantissa of the expression $(e^{2x}+1.)+(-2.)$,

$\delta_2$ — is the value of the binary digits in the second word of the mantissa of the expression $(e^{2x}+1.)$.

Here we assume $\delta$, $\delta_1$, $\delta_2$ to be in non-packed form, i.e. that they are in three-word form in FAC.

```
       EPR
       ENT        ETH
       ENT        ETHN
A      BSS        3
B      BSS        2
ETH    DC
       LIBF       EGETP
ETHN   DC
       LD       3 125
       S          F
       BSC    L   D,Z+
       LDD        H+1
       AND      3 126
       OR         G+1
       STD      3 126
       LD         G
       STO      3 125
C      BSC    I   ETHN
D      A          I
       BSC    I   ETHN,Z+
       S          J
       BSC    L   E,-
       LIBF       ESTO
       DC         B
       LIBF       EMPY
       DC         B
       LIBF       ESTO
       DC         A
       S          K
       STO      3 125
       LIBF       SNR
       LIBF       EADD
       DC         L
       LIBF       EMPY
       DC         A
       LIBF       EADD
       DC         M
       LIBF       EMPY
       DC         A
       LIBF       EMPY
       DC         B
       LIBF       EADD
       DC         B
       MDX        C
E      A          M
       STO      3 125
       CALL       EXPN
       LIBF       ESTO
       DC         B
       LIBF       EADD
       DC         G
       LIBF       ESTO
       DC         A
       LIBF       ELD
       DC         B
       LIBF       EADD
       DC         H
       LIBF       EDIV
       DC         A
       MDX        C
F      DC         133
G      XFLC       1.
H      XFLC       -1.
I      DC         19
J      DC         12
K      DC         4
L      DC         126
       DC         17502
M      XFLC       -0.3333343
       END
```

```
       SPR
       ENT        FTH
       ENT        FTHN
Z      BSS        1
FTH    BSS        1
       LIBF       FGETP
FTHN   DC
       LD       3 125
       S          P
       BSC    L   C,Z+
       LD         Q+1
       STO      3 125
       LDD        O
       OR         R
       AND      3 126
       OR         Q
A      STD      3 126
B      BSC    I   FTHN
C      A          S
       BSC    I   FTHN,Z+
       S          T
       BSC    L   F,-
       LD         U
       S        3 125
       STO        D
       LD       3 126
       STO        Z
D      DC
       STO        D
       M          Z
       M          D
       M          V
       SRT        4
       AD       3 126
       MDX        A
F      A          W
       STO      3 125
       CALL       FXPN
       LIBF       FADD
       DC         Q
       LIBF       FSTO
       DC         Z
       LD       3 127
       AND        X
       STO        D
       LIBF       FADD
       DC         R
       LD         O
       S          D
       M        3 126
       D          Z
       SRT        16
       AD       3 126
       STD      3 126
       LIBF       FDIV
       DC         Z
       MDX        B
O      DC         0
Q      DEC        1.
R      DEC        -2.
S      DC         15
T      DC         7
U      DC         /18FB
V      DC         /FAAC
W      DC         125
X      DC         255
P      DC         132
       END
```

Figure 1

This algorithm enables reduction of the constant $b$ even to $2^{-5}$ with still satisfying accuracy.

We note, that the author of the IBM's programs did not introduce regions of the monomial and polynomial approximation (i.e. $a=0$, $b=0$, in his programs). A large relative error and low speed of computation are stemming therefrom. These disadvantages are removed by programs ETH/ETHN and FTH/FTHN, suggested on Figure 1.

Both programs follow the general algorithm given in (1). Parameters $a$, $b$, $c$, are those already chosen in the preceding discussion.

In the program ETH/ETHN for $2^{-3} \leqq |x| < 2^{4}$ the formula (2) is used, while for $2^{-14} \leqq |x| < 2^{-3}$ the formula

$$x \mapsto R(x) = x - 0.3333343\,x^3 + 0.13353\,x^5 - 0.0625\,x^7$$

is adopted.

In the program FTH/FTHN for $2^{-5} \leqq |x| < 2^{3}$ the formula

$$\operatorname{th} x \approx \frac{(e^{2x}+1.)+(-2.)-\delta\,\dfrac{(e^{2x}+1.)+(-2.)-\delta_1}{(e^{2x}+1.)-\delta_2}}{(e^{2x}+1.)-\delta}$$

is used, where $\delta$, $\delta_1$, $\delta_2$ are variables as previously defined. For $2^{-14} \leqq |x| < 2^{-5}$ the formula

$$\operatorname{th} x \approx x - 0.3333\,x^3$$

is adopted.

In Tables 1. through 4. a comparative survey of characteristics of the IBM's programs (ETNH and FTNH) the programs suggested in the present paper (ETHN and FTHN) for computation of the function $x \mapsto \operatorname{th} x$ in the IBM 1130 (in extended and standard precision, respectively) is given.

Table 1

| K | 0—93 | 94—97 | 98—113 | 114—125 | 126—131 | 132 | 133—255 |
|---|---|---|---|---|---|---|---|
| ETNH | 5.6—5.8 | 5.6—8.2 | 7.6—8.2 | 7.4—8.0 | 7.3—8.0 | 5.6—8.0 | 0.156 |
| ETHN | 0.06 | 0.06 | 0.06 | 4.5—5.5 | 7.4—8.1 | 5.7—8.1 | 0.112 |

Table 2

| K | 0—93 | 94—102 | 103—116 | 117—123 | 124—131 | 132 | 133—255 |
|---|---|---|---|---|---|---|---|
| FTNH | 3.9—4.0 | 3.9—4.6 | 4.3—4.6 | 4.3—4.6 | 4.1—4.3 | 3.9—4.2 | 0.15 |
| FTHN | 0.06 | 0.06 | 0.06 | 0.24—0.26 | 4.1—4.3 | 0.11 | 0.11 |

Table 3

| K | 0—119 | 120 | 121 | 122 | 123 | 124 | 125 | 126—128 | 129—132 | 133—255 |
|---|---|---|---|---|---|---|---|---|---|---|
| ETANH | >9 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 3 | 0 |
| ETH | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 3 | 2 | 0 |

Table 4

| K | 0—113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121—122 | 123 | 124—132 | 133—255 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FTANH | >9 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 2 | 1 | 0 |
| FTH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 |

In Tables, K is the characteristic of the real argument $X$ represented in form $X = M \cdot 2^{K-128}$ where $M$ is a normalized mantissa, $0.5 \leq |M| < 1$ (except for $X = 0$, when $M = 0$).

Tables 1 and 2 show the time of computation in the above programs in milliseconds for the various values of the characteristic K of the argument $X$. The storage cycle of the machine is 3.6 $\mu$sec.

Tables 3 and 4 show the number of inaccurate binary digits of the mantissa of the result of computation for the same programs.

From Tables 1 through 4 it is evident that programs suggested in the present paper (ETH/ETHN and FTH/FTHN) do not have greater error of computation than IBM's programs (ETANH/ETNH and FTANH/FTNH), and that they have, in the wide region of the argument, considerably smaller error and greater speed of computation.

* * *

D. S. Mitrinović, J. Lyness, W. J. Cody, H. Unger and E. Peschl have read this article in manuscript and have made some valuable remarks and suggestions.

## REFERENCES

1. C. T. Fike: *Computer Evaluation of Mathematical Functions.* New Jersey 1968.
2. IBM System/360, FORTRAN IV Library Subprograms, File Number 5360—25, Form c28—6596—4.
3. Б. П. Демидович и И. А. Марон: *Осново вычислительной математики.* Москва 1970.
4. CDC 3600, *Computer Systems Library Functions*, Palo Alto, California, 1966.
5. D. V. Slavić: *The general computer algorithm for the computation of the hyperbolic tangents function.* FCIP'72, e 1, 1—5. Bled 1972.
6. Hewlett-Packard (2100 A) Relocatable Subroutines 02116—91780. California 1970.
7. M. Abramowitz, I. A. Stegun (ed.): *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* New York 1968.
8. ICL FORTRAN, Compiler Libraries, 1900 Series. First ed. Sep. 1969.
9. J. F. Hart, ... : *Computer Approximations.* New York — London — Sydney 1968.
10. IBM 1130 Subroutine Library. Order № GC 26—5929—7.