Digital Computer Laboratory
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

To: S & EC Group and Charles Gaudette

From: Murray Watkins

Date: 16 December 1955

FUNCTION EVALUATION SUBROUTINES FOR REAL-TIME PARALLEL OPERATION

Consider a digital computer which operates on two accumulators in parallel - i.e., stores and manipulates a pair of words (a,b).

In such real-time applications as conversion from polar coordinates (r,θ) to cortesian coordinates (x,y), or vice versa, time for calculation can now be halved by evaluating a different fraction in each side of the accumulator.

For example:

(i) $(r,\theta) \rightarrow (x,y)$ via:

Calculate $\beta = 2/\pi\,\theta$, given θ in radians, or $\beta = \theta/90$, given θ in degrees, put α = fractional part of β and calculate $1-\alpha$.

Then via subroutine: $(\sin 1-\alpha, \sin\alpha)$

multiply by r (x,y)

(ii) $(x,y) \rightarrow (r,\theta)$ via:

if $x = y$, then $r = \sqrt{2}\cdot x$, $\alpha = 1/2$.

if $x \neq y$, calculate $u < 1$ where $u = x/y$ (or y/x)

Then via subroutine $(1/2\sqrt{1+u^2}, \{\tan^{-1}u\}\div u)$

Multiply by (y,u) : $(r/2, \alpha)$

This last subroutine could be table look-up with quartic interpolation (see Method #1c), or via summation of a truncated expansion for the function in a Fourier-Chebyshev series, using $T_n(x)$ over $(-1,1)$ or $T_n^*(x)$ over $(0,1)$ (see Method #2).

Method #1

The most straightforward method is simultaneous table look-up and interpolation (cf. Todd, Trans. Symp. App. Math., v2, pp. 102-4).

(a) Linear interpolation.

Store $f_i = f(x_i)$, $i = 0, 1, 2 \dots$ and δf_i. The calculation cycle is of minimum length, but to get accuracy to five significant figures, many tabular values must be stored. Nearly all the program is a lengthy table, so the program takes too long to read in and occupies too much space in core memory.

(b) Quadratic interpolation.

Store f_i and $\delta^2 f_i$ and use Everett's or Bessel's interpolation formula. Fewer table values need be stored (only a small percentage of those needed for (a) and this saving in length far outweighs the disadvantage of a longer calculation cycle.

(c)

Store f_i and $\delta_m^2 f_i = \delta^2 f_i - 0.18393\,\delta^4 f_i$. This yields nearly fourth-order accuracy, so still fewer tabular values need be stored (less than half the number for (b)), yet the calculation cycle is exactly the same as in (b). (The only penalty is more preliminary work in setting up the table, since $\delta_m^2 f_i$ is obtained only after calculating $\delta^2 f_i$ and $\delta^4 f_i$)

Everett's formula: $$f_p \doteq q f_0 + p f_1 + E_0^2 \delta^2 f_0 + E_1^2 \delta^2 f_1 + E_0^4 \delta^4 f_0 + E_1^4 \delta^4 f_1$$
$$= q f_0 + p f_1 + E_0^2 \delta_m^2 f_0 + E_1^2 \delta_m^2 f_1$$

Truncation error = $h^6 \left|\binom{p+3}{6}\right| \max |f^{VI}(x)|$.

where $p = 1/h(x - x_0)$, $q = \frac{1}{h}(x_1 - x) = 1 - p$, $E_0^2 = \frac{q(q^2-1)}{6}$, $E_1^2 = \frac{p(p^2-1)}{6}$

Here $\delta_m^2 = \delta^2 + k\delta^4$, $k = k(p) = \frac{E_0^4}{E_0^2} = \frac{(p+1)(p-3)}{20}$, approximately constant over $0 \le p \le 1$. (varies from -.15 to -.20).

Method # 2 -

$$\sin\frac{\pi x}{2} = x\Big[1.27628 - .28526\,T_1^*(x^2) + .00912\,T_2^*(x^2) - .00014\,T_3^*(x^2)\Big]$$

$$= 1.57080\,x - .64596\,x^3 + .07969\,x^5 - .00468\,x^7 + .00016\,x^9$$

$$= 1.21615\,P_1(x) - .25489\,P_3(x) + .00920\,P_5(x) - .00016\,P_7(x)$$

$$\cos\frac{\pi x}{2} = .47200 - .49940\,T_1^*(x^2) + .01114\,T_2^*(x^2) - .00138\,T_3^*(x^2) + .00019\,T_4^*(x^2) - .00003\,T_5^*(x^2)$$

$$= 1 - 1.23370\,x^2 + .25367\,x^4 - .02086\,x^6 + .00092\,x^8 - .00002\,x^{10}$$

$$= .63662 - .68758\,P_2(x) + .051778\,P_4(x) - .00133\,P_6(x) + .00002\,P_8(x)$$

where $T_n^*(x) = T_{2n}(2x^2-1) = \cos(n\cos^{-1}(2x^2-1))$, Chebyshev polynomial;

$P_n(x) = \dfrac{1}{2^n n!}\dfrac{d^n}{dx^n}(x^2-1)^n$, Legendre polynomial.

DGL · 115

$$\frac{\tan^{-1} u}{u} = .88137 - .10589\,T_1^*(x^2) + .01114\,T_2^*(x^2)$$

$$- .00138\,T_3^*(x^2) + .00019\,T_4^*(x^2) - .00003\,T_5^*(x^2)$$

$$= 1 - .33334x^2 + .20018x^4 - .13793x^6 + .10831x^8$$

$$\sqrt{1-w} = 1 - 2\left[\tfrac{1}{3}T_2(w) + \tfrac{1}{15}T_4(w) + \tfrac{1}{35}T_6(w) + \tfrac{1}{63}T_8(w) + \dots + \tfrac{1}{4n^2-1}T_{2n}(w)\right]$$

whence $\sqrt{1+u^2} = \dfrac{\sqrt{1-u^4}}{\sqrt{1-u^2}}$, or, if $w = 2(1-u^2)$, $\frac{1}{\sqrt{2}}\sqrt{1+u^2} = \sqrt{1-w}$.

Also, $\sqrt{1-x^2} = \dfrac{\pi}{2}\Big[\tfrac{1}{2} - \Big(\tfrac{5}{16}P_2(x) + \tfrac{9}{128}P_4(x) + \tfrac{65}{2048}P_6(x) + \tfrac{595}{32{,}768}P_8(x)$

$$+ \dots + \frac{4m+1}{(2m-1)(2m+2)}\left(\frac{1}{2^m}\cdot\frac{1.3.5\ (2m-1)}{m!}\right)^2 P_{2m}(x) + \dots\Big)\Big].$$

$$(1-x)^\rho = 2^\rho \sum_{n=0}^{\infty} \frac{2n+1}{n+1+\rho}\,\frac{(-\rho)_n}{(1+\rho)_n}\,P_n(x), \text{ where } (\alpha)_n = \frac{\Gamma(n+\alpha)}{\Gamma(\alpha)} = \alpha(\alpha+1)\dots(\alpha+n-1),$$

$(\alpha)_0 = 1$.

$$= 2^\rho\,\Gamma(\alpha+\rho+1)\sum_{n=0}^{\infty}\frac{\Gamma(2n+\alpha+\beta+1)\,\Gamma(n+\alpha+\beta+1)}{\Gamma(n+\alpha+1)\,\Gamma(n+\alpha+\beta+2+\rho)}(-\rho)_n\,P_n^{\alpha,\beta}(x)$$

where $P_n^{\alpha,\beta}(x) = 2^{-n}\sum_{m=0}^{\infty}\binom{n+\alpha}{m}\binom{n+\beta}{n-m}(x-1)^{n-m}(x+1)^m$,

Jacobi polynomial. $P_n^{-\frac{1}{2},-\frac{1}{2}}(x) = T_n(x)$.

$$= 2^\rho\,\Gamma(\rho+\tfrac{1}{2})\sum_{n=0}^{\infty}\frac{(2n)!}{\Gamma(n+\frac{1}{2})}\cdot\frac{n!\,(-\rho)_n}{\Gamma(n+1+\rho)}\,T_n(x)$$

Note $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,dt$, the Gamma-function;

$\Gamma(z+1) = z\cdot\Gamma(z)$; $\Gamma(\tfrac{1}{2}) = \sqrt{\pi}$.