

Generation of Bessel Functions on High Speed Computers

The generation of values of sets of functions such as the Bessel functions $J_n(x)$ and $Y_n(x)$ is a problem which one encounters frequently in numerical calculations. The present paper is concerned with describing a method which has been found particularly useful on high speed computers. The basic idea is due to J. C. P. Miller and is described in detail [as applied to the Bessel functions $I_n(x)$] in the introduction to the second volume [1] of Bessel functions published by BAAS. The tables in the BA volume were computed by hand as were the tables by Fox [2]. However, the method is particularly suited to large scale computers. In preparing a general purpose routine to compute $J_n(x)$ and $Y_n(x)$ one is faced with various obstacles. The power series representation of $J_n(x)$ is useful for values of x which are not too large. In addition to the slowness of convergence the form of $Y_n(x)$ becomes complicated. The asymptotic expansions of $J_n(x)$ and $Y_n(x)$ are useful when x is large with respect to n . In the region where both n and x are large neither of these representations is useful and a third form must be used. It is clear that any program which employs these representations and which should be generally useful must automatically choose the proper form for computation. The method of Miller avoids these difficulties and provides a simple algorithm not only for $J_n(x)$ but for many other functions with a similar behavior.

Specifically, the method is as follows (the theory supporting the method is given in [1], p. xvii):

1. Choose k larger than the greater of n or x , assume $\bar{J}_{k+1} = 0, \bar{J}_k = \alpha \neq 0$, where α is some arbitrarily chosen constant.

2. If $z = 1/x$, generate the sequence $\bar{J}_{k-1}, \bar{J}_{k-2}, \dots, \bar{J}_1, \bar{J}_0$ from the recurrence relation

$$\bar{J}_{p-1} = 2pz\bar{J}_p - \bar{J}_{p+1}$$

starting with $p = k$. If k is chosen sufficiently large we will obtain $J_p = c\bar{J}_p$ for desired values of p ranging from $p = 0$ to $p = n$.

3. To determine the constant c use the functional relation

$$J_0 + 2\sum_{m=1}^{\infty} J_{2m} = 1.$$

The choice of k offers no difficulty since the calculation may be done iteratively. Thus, starting with a value of $k > \max(n, x)$ we perform the calculations in (2) and (3) once and then repeat with k increased by a fixed amount. The results obtained are compared in accordance with a preassigned tolerance and the process is repeated until the criteria for acceptance are satisfied. In practice, if the values of J_p are desired for $p = 0, 1, \dots, n$ the comparison need only be made at $p = n$. It is advisable to employ floating point arithmetic and choose α as small as possible.

4. To generate the functions $Y_n(x)$ we employ the following representation for $Y_0(x)$,

$$Y_0(x) = (2/\pi) \left[J_0(x) \left\{ \log \frac{x}{2} + \gamma \right\} + 2 \sum_{m=1}^{\infty} \frac{(-1)^{m-1} J_{2m}(x)}{m} \right]$$

where γ is Euler's constant. We note that the series part here employs the same values J_{2k} as occur in the calculation of the normalization constant c .

5. Compute $Y_1(x)$ from the relation

$$J_1(x)Y_0(x) - J_0(x)Y_1(x) = 2/\pi x.$$

6. Generate $Y_p(x)$ in a forward direction for $p = 1, 2, \dots, n - 1$ from the recurrence relation

$$Y_{p+1} = 2pzY_p - Y_{p-1}.$$

The method may be employed in a similar manner for other sets of functions. In the following we outline the procedure for some of the functions for which it may be employed.

Modified Bessel Functions $I_n(x)$ and $K_n(x)$. Here the recurrence relation (descending order) for $I_n(x)$ is

$$I_{p-1} = 2pzI_p + I_{p+1}$$

while the normalization can be obtained from the relation

$$I_0 + 2 \sum_{m=1}^{\infty} I_m = e^x.$$

While there is an analogous relation for $K_0(x)$ corresponding to that for Y_0 , the possible loss of significant digits makes it unfeasible to use. To compute $K_0(x)$ the integral representation

$$K_0(x) = \int_0^{\infty} \exp(-x \cosh u) du$$

will be convenient for all values of $x > .01$. (Alternatively one may use polynomial approximations given by Allen [3].) The relation

$$K_1I_0 + K_0I_1 = 1/x$$

will yield K_1 while the recurrence relation (ascending order)

$$K_{p+1} = 2pzK_p + K_{p-1}$$

may be used to generate as many values of K_{p+1} as may be desired.

Spherical Bessel Functions $j_n(x)$ and $n_n(x)$. The spherical Bessel Functions $j_n(x) = \sqrt{\pi/2x} J_{n+1/2}(x)$ may be generated with the aid of the recurrence relation (descending order)

$$j_{p-1} = (2p + 1)zj_p - j_{p+1}$$

with the normalization constant determined from the fact that $j_0 = \sin x/x$. To determine the function $n_n(x) = (-1)^{n+1} \sqrt{\pi/2x} J_{-(n+1/2)}(x)$ we have

$$n_0 = -\frac{\cos x}{x} \text{ and } n_1 = -\frac{\sin x}{x} - \frac{\cos x}{x^2}$$

and it is convenient to compute these functions directly and compute as many further values of n_p as desired with the previous recurrence relation (ascending order) which is also satisfied by n_p .

Modified Bessel Functions of Half-Integral Order. The functions

$$i_n = \sqrt{\pi/2x} I_{n+\frac{1}{2}}(x)$$

$$k_n = (-1)^{n+1} \sqrt{\pi/2x} K_{n+\frac{1}{2}}(x)$$

both satisfy the recurrence relation

$$(2p + 1)zi_p = i_{p-1} - i_{p+1}$$

and may be computed in a fashion entirely similar to the computation of the spherical Bessel functions. We have also

$$i_0 = \frac{\sinh x}{x}$$

$$k_0 = -\frac{\pi}{2x} e^{-x}$$

$$k_1 = \frac{\pi e^{-x}(x + 1)}{2x^2}.$$

IRENE A. STEGUN
MILTON ABRAMOWITZ

National Bureau of Standards
Washington, D. C.

1. British Association for the Advancement of Science, *Mathematical Tables, Volume X, Bessel Functions, Part II, Functions of Positive Integer Order*, Cambridge, University Press, 1952.
2. L. FOX, *A Short Table for Bessel Functions of Integer Order and Large Arguments*, Royal Society Shorter Mathematical Tables No. 3, Cambridge, 1954.
3. E. E. ALLEN, "Polynomial approximations to some modified Bessel functions," *MTAC*, v. 10, 1956, p. 162-164.

Remarks on the Disposition of Points in Numerical Integration Formulas

1. Introduction. Numerical integration formulas of degree 2 (i.e., exact for polynomials of at most degree 2) consisting of $n + 1$ equally weighted points have been developed for certain regions in n -dimensional Euclidean space. Thacher [1] discusses the equations which a formula of this type must satisfy for regions which are invariant under the group of linear transformations which leave the n -cube with vertices $(\pm a, \pm a, \dots, \pm a)$ invariant; we call these symmetric regions. Hammer and Stroud [2] give two such formulas for the n -simplex. Hammer [3] has shown that a set of $2n$ equally weighted points lying on the coordinate axes form a formula of degree 3 for any symmetric region.

In section 2 of this note we show that the formulas of degree 2 discussed by Thacher can be described geometrically. We also show that there is a similar class of formulas for the regular n -simplex. In section 3 we describe geometrically a wide class of formulas of degree 3 containing $2n$ points for symmetrical regions.