Chapter 5. Evaluation of Functions

5.0 Introduction

The purpose of this chapter is to acquaint you with a selection of the techniques that are frequently used in evaluating functions. In Chapter 6, we will apply and illustrate these techniques by giving routines for a variety of specific functions. The purposes of this chapter and the next are thus mostly in harmony, but there is nevertheless some tension between them: Routines that are clearest and most illustrative of the general techniques of this chapter are not always the methods of choice for a particular special function. By comparing this chapter to the next one, you should get some idea of the balance between "general" and "special" methods that occurs in practice.

Insofar as that balance favors general methods, this chapter should give you ideas about how to write your own routine for the evaluation of a function which, while "special" to you, is not so special as to be included in Chapter 6 or the standard program libraries.

CITED REFERENCES AND FURTHER READING:

Fike, C.T. 1968, *Computer Evaluation of Mathematical Functions* (Englewood Cliffs, NJ: Prentice-Hall).

Lanczos, C. 1956, *Applied Analysis*; reprinted 1988 (New York: Dover), Chapter 7.

5.1 Series and Their Convergence

Everybody knows that an analytic function can be expanded in the neighborhood of a point x_0 in a power series,

$$f(x) = \sum_{k=0}^{\infty} a_k (x - x_0)^k \tag{5.1.1}$$

Such series are straightforward to evaluate. You don't, of course, evaluate the kth power of $x - x_0$ *ab initio* for each term; rather you keep the $k - 1$st power and update it with a multiply. Similarly, the form of the coefficients a is often such as to make use of previous work: Terms like $k!$ or $(2k)!$ can be updated in a multiply or two.

How do you know when you have summed enough terms? In practice, the terms had better be getting small fast, otherwise the series is not a good technique to use in the first place. While not mathematically rigorous in all cases, standard practice is to quit when the term you have just added is smaller in magnitude than some small ϵ times the magnitude of the sum thus far accumulated. (But watch out if isolated instances of $a_k = 0$ are possible!).

A weakness of a power series representation is that it is guaranteed *not* to converge farther than that distance from x_0 at which a singularity is encountered *in the complex plane*. This catastrophe is not usually unexpected: When you find a power series in a book (or when you work one out yourself), you will generally also know the radius of convergence. An insidious problem occurs with series that converge everywhere (in the mathematical sense), but almost nowhere fast enough to be useful in a numerical method. Two familiar examples are the sine function and the Bessel function of the first kind,

$$\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1} \tag{5.1.2}$$

$$J_n(x) = \left(\frac{x}{2}\right)^n \sum_{k=0}^{\infty} \frac{(-\frac{1}{4}x^2)^k}{k!(k+n)!} \tag{5.1.3}$$

Both of these series converge for all x. But both don't even start to converge until $k \gg |x|$; before this, their terms are increasing. This makes these series useless for large x.

Accelerating the Convergence of Series

There are several tricks for accelerating the rate of convergence of a series (or, equivalently, of a sequence of partial sums). These tricks will *not* generally help in cases like (5.1.2) or (5.1.3) while the size of the terms is still increasing. For series with terms of decreasing magnitude, however, some accelerating methods can be startlingly good. *Aitken's δ^2-process* is simply a formula for extrapolating the partial sums of a series whose convergence is approximately geometric. If S_{n-1}, S_n, S_{n+1} are three successive partial sums, then an improved estimate is

$$S'_n \equiv S_{n+1} - \frac{(S_{n+1} - S_n)^2}{S_{n+1} - 2S_n + S_{n-1}} \tag{5.1.4}$$

You can also use (5.1.4) with $n + 1$ and $n - 1$ replaced by $n + p$ and $n - p$ respectively, for any integer p. If you form the sequence of S'_i's, you can apply (5.1.4) a second time to *that* sequence, and so on. (In practice, this iteration will only rarely do much for you after the first stage.) Note that equation (5.1.4) should be computed as written; there exist algebraically equivalent forms that are much more susceptible to roundoff error.

For *alternating series* (where the terms in the sum alternate in sign), *Euler's transformation* can be a powerful tool. Generally it is advisable to do a small number $n-1$ of terms directly, then apply the transformation to the rest of the series beginning with the nth term. The formula (for n even) is

$$\sum_{s=0}^{\infty}(-1)^s u_s = u_0 - u_1 + u_2 \ldots - u_{n-1} + \sum_{s=0}^{\infty} \frac{(-1)^s}{2^{s+1}}[\Delta^s u_n] \tag{5.1.5}$$

Here Δ is the *forward difference operator*, i.e.,

$$\begin{aligned} \Delta u_n &\equiv u_{n+1} - u_n \\ \Delta^2 u_n &\equiv u_{n+2} - 2u_{n+1} + u_n \\ \Delta^3 u_n &\equiv u_{n+3} - 3u_{n+2} + 3u_{n+1} - u_n \qquad \text{etc.} \end{aligned} \tag{5.1.6}$$

Of course you don't actually do the infinite sum on the right-hand side of (5.1.5), but only the first, say, p terms, thus requiring the first p differences (5.1.6) obtained from the terms starting at u_n.

Euler's transformation can be applied not only to convergent series. In some cases it will produce accurate answers from the first terms of a series that is formally divergent. It is widely used in the summation of asymptotic series. In this case it is generally wise not to sum farther than where the terms start increasing in magnitude; and you should devise some independent numerical check that the results are meaningful.

There is an elegant and subtle implementation of Euler's transformation due to van Wijngaarden [1]: It incorporates the terms of the original alternating series one at a time, in order. For each incorporation it *either* increases p by 1, equivalent to computing one further difference (5.1.6), or else *retroactively* increases n by 1, without having to redo all the difference calculations based on the old n value! The decision as to which to increase, n or p, is taken in such a way as to make the convergence most rapid. Van Wijngaarden's technique requires only one vector of saved partial differences. Here is the algorithm:

```
#include <math.h>

void eulsum(float *sum, float term, int jterm, float wksp[])
```
Incorporates into `sum` the `jterm`'th term, with value `term`, of an alternating series. `sum` is input as the previous partial sum, and is output as the new partial sum. The first call to this routine, with the first `term` in the series, should be with `jterm=1`. On the second call, `term` should be set to the second term of the series, with sign opposite to that of the first call, and `jterm` should be 2. And so on. `wksp` is a workspace array provided by the calling program, dimensioned at least as large as the maximum number of terms to be incorporated.
```
{
    int j;
    static int nterm;
    float tmp,dum;

    if (jterm == 1) {                    Initialize:
        nterm=1;                         Number of saved differences in wksp.
        *sum=0.5*(wksp[1]=term);         Return first estimate.
    } else {
        tmp=wksp[1];
        wksp[1]=term;
        for (j=1;j<=nterm-1;j++) {       Update saved quantities by van Wijn-
            dum=wksp[j+1];                  gaarden's algorithm.
```

```
            wksp[j+1]=0.5*(wksp[j]+tmp);
            tmp=dum;
        }
        wksp[nterm+1]=0.5*(wksp[nterm]+tmp);
        if (fabs(wksp[nterm+1]) <= fabs(wksp[nterm]))      Favorable to increase p,
            *sum += (0.5*wksp[++nterm]);      and the table becomes longer.
        else                                   Favorable to increase n,
            *sum += wksp[nterm+1];             the table doesn't become longer.
    }
}
```

The powerful Euler technique is not directly applicable to a series of positive terms. Occasionally it is useful to convert a series of positive terms into an alternating series, just so that the Euler transformation can be used! Van Wijngaarden has given a transformation for accomplishing this [1]:

$$\sum_{r=1}^{\infty} v_r = \sum_{r=1}^{\infty} (-1)^{r-1} w_r \tag{5.1.7}$$

where

$$w_r \equiv v_r + 2v_{2r} + 4v_{4r} + 8v_{8r} + \cdots \tag{5.1.8}$$

Equations (5.1.7) and (5.1.8) replace a simple sum by a two-dimensional sum, each term in (5.1.7) being itself an infinite sum (5.1.8). This may seem a strange way to save on work! Since, however, the indices in (5.1.8) increase tremendously rapidly, as powers of 2, it often requires only a few terms to converge (5.1.8) to extraordinary accuracy. You do, however, need to be able to compute the v_r's efficiently for "random" values r. The standard "updating" tricks for sequential r's, mentioned above following equation (5.1.1), can't be used.

Actually, Euler's transformation is a special case of a more general transformation of power series. Suppose that some known function $g(z)$ has the series

$$g(z) = \sum_{n=0}^{\infty} b_n z^n \tag{5.1.9}$$

and that you want to sum the new, unknown, series

$$f(z) = \sum_{n=0}^{\infty} c_n b_n z^n \tag{5.1.10}$$

Then it is not hard to show (see [2]) that equation (5.1.10) can be written as

$$f(z) = \sum_{n=0}^{\infty} [\Delta^{(n)} c_0] \frac{g^{(n)}}{n!} z^n \tag{5.1.11}$$

which often converges much more rapidly. Here $\Delta^{(n)} c_0$ is the nth finite-difference operator (equation 5.1.6), with $\Delta^{(0)} c_0 \equiv c_0$, and $g^{(n)}$ is the nth derivative of $g(z)$. The usual Euler transformation (equation 5.1.5 with $n = 0$) can be obtained, for example, by substituting

$$g(z) = \frac{1}{1+z} = 1 - z + z^2 - z^3 + \cdots \tag{5.1.12}$$

into equation (5.1.11), and then setting $z = 1$.

Sometimes you will want to compute a function from a series representation even when the computation is *not* efficient. For example, you may be using the values obtained to fit the function to an approximating form that you will use subsequently (cf. §5.8). If you are summing very large numbers of slowly convergent terms, pay attention to roundoff errors! In floating-point representation it is more accurate to sum a list of numbers in the order starting with the smallest one, rather than starting with the largest one. It is even better to group terms pairwise, then in pairs of pairs, etc., so that all additions involve operands of comparable magnitude.

CITED REFERENCES AND FURTHER READING:

Goodwin, E.T. (ed.) 1961, *Modern Computing Methods*, 2nd ed. (New York: Philosophical Library), Chapter 13 [van Wijngaarden's transformations]. [1]

Dahlquist, G., and Bjorck, A. 1974, *Numerical Methods* (Englewood Cliffs, NJ: Prentice-Hall), Chapter 3.

Abramowitz, M., and Stegun, I.A. 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, Volume 55 (Washington: National Bureau of Standards; reprinted 1968 by Dover Publications, New York), §3.6.

Mathews, J., and Walker, R.L. 1970, *Mathematical Methods of Physics*, 2nd ed. (Reading, MA: W.A. Benjamin/Addison-Wesley), §2.3. [2]

5.2 Evaluation of Continued Fractions

Continued fractions are often powerful ways of evaluating functions that occur in scientific applications. A continued fraction looks like this:

$$f(x) = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cfrac{a_4}{b_4 + \cfrac{a_5}{b_5 + \cdots}}}}} \tag{5.2.1}$$

Printers prefer to write this as

$$f(x) = b_0 + \frac{a_1}{b_1 +}\;\frac{a_2}{b_2 +}\;\frac{a_3}{b_3 +}\;\frac{a_4}{b_4 +}\;\frac{a_5}{b_5 +}\cdots \tag{5.2.2}$$

In either (5.2.1) or (5.2.2), the a's and b's can themselves be functions of x, usually linear or quadratic monomials at worst (i.e., constants times x or times x^2). For example, the continued fraction representation of the tangent function is

$$\tan x = \frac{x}{1 -}\;\frac{x^2}{3 -}\;\frac{x^2}{5 -}\;\frac{x^2}{7 -}\cdots \tag{5.2.3}$$

Continued fractions frequently converge much more rapidly than power series expansions, and in a much larger domain in the complex plane (not necessarily including the domain of convergence of the series, however). Sometimes the continued fraction converges best where the series does worst, although this is not