Chapter 3

Evaluation of Functions

3.1 Power series and truncation error

In this section we deal with

(i) how the computer handles power series, and

(ii) the consequences when functions are evaluated using power series.

In general, many functions $f(x)$ can be represented as

$$f(x) = \sum_{j=0}^{\infty} c_j x^j \ , \tag{3.1}$$

at least on some interval enclosing $x = 0$. As one can imagine, special functions represented by power series will be *truncated*. The truncation carries an error which we discuss now.

Assume that we want to compute a quantity S, which can be expressed in a series expansion, $S = \sum_{j=0}^{n} a_j$ and set

$$S_n = \sum_{j=0}^{n} a_j, \quad R_n = S - S_n \ . \tag{3.2}$$

We call $\sum_{n+1}^{\infty} a_j$ the *tail* of the series; a_n is the *last included term* and a_{n+1} is the *first neglected term*. The *remainder* R_n with reversed sign is called the *truncation error* [the correction one has to make in order to eliminate the error].

In numerical computation, a series should be regarded as a finite expansion together with a remainder.

3.1.1 The exponential function

Let

$$e^x = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + \dots \ . \tag{3.3}$$

We know that if $x = 0$ we have $e^x = 1$, so $a_0 = 1$.

Now we differentiate the infinite series with respect to x, which of course leaves the left hand side of the equation unchanged:

$$e^x = a_1 + 2a_2 x + 3a_3 x^2 + 4a_4 x^3 + \dots \ . \tag{3.4}$$

Putting $x = 0$ this time reveals that $a_1 = 1$. Differentiating a second time gives:

$$e^x = 2a_2 + (3 \times 2)\, a_3 x + (4 \times 3)\, a_4 x^2 + \dots \ . \tag{3.5}$$

This time substituting $x = 0$ gives us $a_2 = \frac{1}{2}$, etc. This leads to:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots = \sum_{j=0}^{\infty} \frac{x^j}{j!} \tag{3.6}$$

By simply coding

$$\exp(-x) = \sum_{j=0}^{\infty} (-1)^j \frac{x^j}{j!} \ , \tag{3.7}$$

we obtain the results shown in Table 3.1.

Table 3.1: Numerical values for the negative exponential function using Eq. (3.7). The value 'NaN' indicates numerical overflow.

x	$\exp(-x)$	*Series*	*Number of terms in series*
0.0	1.00000E+00	1.00000E+00	1
10.0	4.53999E−05	4.53999E−05	44
20.0	2.06115E−09	4.87460E−09	72
30.0	9.35762E−14	−3.42134E−05	100
40.0	4.24835E−18	−2.21033E+00	127
50.0	1.92875E−22	−8.33851E+04	155
60.0	8.75651E−27	−8.50381E+08	171
70.0	3.97545E−31	NaN	171
80.0	1.80485E−35	NaN	171
90.0	8.19401E−40	NaN	171
100.0	3.72008E−44	NaN	171

For $x = 70$ we have an overflow problem represented by NaN. In other words, the calculation of a factorial of 171 is beyond the limit set by double precision.

One obvious way to reduce the argument is to reduce the size of the argument to write

$$\exp(-x) = [\exp(-x/2)]^2 \,, \tag{3.8}$$

as often as necessary. When applied k times, we get

$$\exp(-x) = [\exp(-x/2^k)]^{2^k} \,, \tag{3.9}$$

which allows for better handling of large $|x|$. However, this approach only delays the appearance of `NaN`to slightly larger values of x. As we will see later, recurrence relations are one of the best ways out of these limitations.

3.1.2 Horner's scheme

We frequently have to evaluate polynomials (truncated power series, etc.). Therefore it would be nice to find an efficient algorithm for their evaluation. You might think that this is a trivial matter, but we will show that there are several possibilities, one of which is more economical than the others. Consider the polynomial

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 \,. \tag{3.10}$$

- If we compute each power x^k by itself and multiply it by a_k, we have to perform

  $$n + (n{-}1) + \ldots 1 = \frac{(n{+}1)n}{2} \,, \tag{3.11}$$

  multiplications and n additions.

- If we produce the powers successively and multiply at each level by the corresponding a_k, we have $2n$ multiplications and n additions.

- However, if we rewrite the polynomial in the form

  $$P(x) = (\ldots(((a_n x + a_{n-1})x + a_{n-2})x + a_{n-3})x + \ldots + a_1)x + a_0 \,, \tag{3.12}$$

  we have only n multiplications and additions!

This is *Horner's scheme*, the most economical method for the evaluation of polynomials. You should always apply it when you need to evaluate polynomials. Horner's scheme is readily used on a calculator, as you will easily convince yourself.

For example let us focus on the following polynomial

$$p(x) = 2x^3 + 3x^2 - 5x + 7 \,, \tag{3.13}$$

which is simply evaluated at $x = 2$ as $p(2) = 25$.

Factoring an x out of the first 3 terms (and reordering) provides the following result:

$$p(x) = 7 + x[-5 + x(3 + 2x)] \tag{3.14}$$

The nested structure above allows to write a 'do loop' to evaluate the polynomial. If the polynomial is stored in an array, such as $p = [7, -5, 3, 2]$, then computationally one is performing the following steps:

$$y = p(4) \quad \text{or} \quad y = 2 \tag{3.15}$$

$$y = p(3) + xy \quad \text{or} \quad y = 3 + 2x \tag{3.16}$$

$$y = p(2) + xy \quad \text{or} \quad y = -5 + x(3 + 2x) \tag{3.17}$$

$$y = p(1) + xy \quad \text{or} \quad y = -7 + x[-5 + x(3 + 2x)] \tag{3.18}$$

The above should remind you of the *synthetic division* or *synthetic substitution* you studied in college algebra.

The power of Horner's method – besides reducing the number of operations to $2n$ – lies in the fact that all of the steps above can be easily combined into:

Horner

```
n = 4
y = p(n)
do  i=n-1,1,-1
    y = p(i) + x*y
enddo
```

You feed the function `Horner`:

- the polynomial p as an array which holds the coefficients of p, in this case $[7, -5, 3, 2]$
- the value of x at which the polynomial is to be evaluated

. . . and the job is done.

We saw that

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots = \sum_{j=0}^{\infty} \frac{x^j}{j!} , \tag{3.19}$$

And thus,

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \ldots \tag{3.20}$$

$$= 1 + 1 + \frac{1}{2}\Big(1 + \frac{1}{3}\Big(1 + \frac{1}{4}\Big(1 + \frac{1}{5}\big(1 + \ldots\big)\Big)\Big)\Big) , \tag{3.21}$$

which can be calculated using Horner's scheme.

3.1.3 The power of series representations

Series can be very useful when dealing with integrals and differential equations. For example let us consider the following integral

$$I = \int_0^1 e^{-t^2} dt \, , \tag{3.22}$$

to be computed to 4 correct decimals.

The integral cannot be expressed in terms of elementary functions. However we can use the series expansion of the exponential function to obtain

$$I = \sum_{n=0}^{\infty} \int_0^1 \frac{(-1)^n t^{2n}}{n!} \, dt = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)\, n!} \simeq 0.7468 \, , \tag{3.23}$$

(with seven terms). The series converges quickly and the 8th term is already less than 2×10^{-5}.

As another example, consider the differential equation $y'' = -xy$, with initial conditions $y(0) = 1, y'(0) = 0$. Here again the solution cannot be simply expressed in terms of elementary functions. One can find a power series solution by writing

$$y(x) = \sum_{n=0}^{\infty} c_n x^n \, . \tag{3.24}$$

Differentiating twice will lead to ($y'' = -xy$)

$$\begin{aligned} & 2\,c_2 + 6\,c_3 x + 12\,c_4 x^2 + \ldots + (m+2)(m+1)c_{m+2} x^m + \ldots \\ & \quad = -c_0 x - c_1 x^2 - c_2 x^3 - \ldots - c_{m-1} x^m - \ldots \end{aligned} \tag{3.25}$$

which implies

$$c_2 = 0, \qquad (m+2)(m+1)c_{m+2} = -c_{m-1}, \qquad m \geq 1 \, . \tag{3.26}$$

The initial conditions allow us to conclude that $c_0 = 1$, $c_1 = 0$. Thus $c_n = 0$, if n is not a multiple of 3, and using the recursion one arrives at

$$y(x) = 1 - \frac{x^3}{6} + \frac{x^6}{180} - \frac{x^9}{12960} + \ldots \, , \tag{3.27}$$

which can be evaluated using Horner's method.

Conclusion: Series expansions are a very important tool in numerical calculations. Solutions to differential equations can often be expressed in terms of series expansions.

3.2 Recurrence relations

Recurrence formulae are used, whenever possible, as ways of representing series and reducing errors and calculation time. Many useful functions satisfy recurrence relation, e.g.,

$$\begin{aligned}
s_n &= -s_{n-1}\frac{x}{n} \ , && (3.28)\\
J_{n+1}(x) &= \frac{2n}{x}J_n(x) - J_{n-1}(x) \ , && (3.29)\\
nE_{n+1} &= e^{-x} - xE_n(x) \ , && (3.30)
\end{aligned}$$

where the functions are $(-1)^n x^n/n!$ (the terms in the power series expansion (3.7) for the exponential function), the Bessel functions of the first kind, and exponential integrals, respectively.

Note: In [NR90], there are two routines '`recur1`' and '`recur2`' that handle first-and second-order linear recurrences. The recurrences are implemented as trivial do-loops.

Let us go back to the exp($-x$) function where direct methods lead to `NaN`s. This problem can be immediately dealt with by using the recurrence formula (3.28) above. The results are shown in Table 3.2. We did get rid of the `NaN`s but the results still are way off! This illustrates the challenges one faces when dealing with series!

Table 3.2: Like in Table 3.1, but using the recurrence formula (3.28) instead. *[I am not so sure this is what you get using 3.28. (Rachid to check)]*

x	exp($-x$)	*Series*	*Number of terms in series*
0.0	1.00000E+00	1.00000E+00	1
10.0	4.53999E−05	4.53999E−05	44
20.0	2.06115E−09	4.87460E−09	72
30.0	9.35762E−14	−3.42134E−05	100
40.0	4.24835E−18	−2.21033E+00	127
50.0	1.92875E−22	−8.33851E+04	155
60.0	8.75651E−27	−8.50381E+08	182
70.0	3.97545E−31	−3.2979605E+13	209
80.0	1.80485E−35	9.1805682E+16	237
90.0	8.19401E−40	−5.0516254E+21	264
100.0	3.72008E−44	−2.9137556E+25	291

3.2.1 Numerical calculation of erf x

In many physics applications involving the normal probability distribution function, integrals of the form $\int_a^x \exp(-t^2/2)\,dt$ appear. This integral can not be solved in terms of standard transcendental and algebraic functions, so a new special function called the error function is introduced:

$$\operatorname{erf} x = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt \; . \tag{3.31}$$

One means of evaluating the error function is the truncated power series. For small $|x|$, a good and efficient way to evaluate the $\operatorname{erf} x$ is to use the truncated power series:

$$\operatorname{erf} x = \frac{2x}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)\, k!} \, x^{2k} \simeq \frac{2x}{\sqrt{\pi}} \sum_{k=0}^{N-1} \frac{(-1)^k}{(2k+1)\, k!} \, x^{2k} \; . \tag{3.32}$$

The equation above is a so-called alternating power series and is theoretically convergent for all x. If one truncates the series at some high order term $N-1$, the error ϵ that is made is smaller than the first neglected term:

$$|\epsilon| \leq \frac{(-1)^N}{(2N+1)\, N!} \, x^{2N} \; . \tag{3.33}$$

The main question is: What should the value of N be for a given x when evaluating the error function? This question is complicated by the presence of the factorial, which is not a built-in function in Fortran. However, even if there were a factorial function in Fortran, the best way to evaluate $E(k)$ is not to directly compute $E(k) = \frac{(-1)^k}{(2k+1)k!} x^{2k}$. Rather, you can use the following recursive relation (which you should derive)

$$\frac{E(k)}{E(k-1)} = R(k) = -\frac{(2k-1)\, x^2}{(2k+1)\, k}, \quad \text{with} \quad E(0) = 1 \; . \tag{3.34}$$

Therefore we compute $E(k)$ as

```
expression = E(0)
do k=1,N
    expression = expression * R(k)
enddo
```

The zeroth-order term $E(0) = 1$ is used to initialize the sum. Look through the formulae given above to see what equations you need to put in the loop for calculating $E(0)$ and $R(k)$.

Run the program for different values of N and compare to what *Maple or Mathematica* (which has a built in function) gives.

Note: If x is significantly larger than 1 (5 or larger), the series will converge very slowly. Other series expansions (asymptotic series) can be used in that case. If $|x| > 5$, it is generally safe to assume that $\operatorname{erf} x \simeq 1$ and return this value without calculating the sum.

3.3 Continued fractions

3.3.1 Euclid's algorithm

Some functions cannot be well approximated by a power series, but can well by a quotient of power series. In general, a *continued fraction* is an expression in the form

$$r = a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3 + \cfrac{b_4}{a_4 + \ldots}}}} \, , \tag{3.35}$$

where a_i and b_i may be any kind of numbers, variables, or functions. With all $b_i = 1$ we have *simple continued fractions*

$$r = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \ldots}}}} \, . \tag{3.36}$$

A more convenient notation is $r = [a_0, a_1, a_2, a_3, \ldots]$. Often, a_0 (the integer part of r) is separated from the rest of the coefficients with a semicolon: $r = [a_0; a_1, a_2, a_3, \ldots]$.

If the number of terms is infinite, r is called an *infinite continued fraction*. A *terminating fraction* can be evaluated backwards by the recurrence relations (*Euclid's algorithm*),

$$y_1 = a_{n-1} + b_n/a_n, \quad y_2 = a_{m-2} + b_{n-1}/y_1, \ldots \quad r_n = y_n = a_0 + b_1/y_{n-1} \, . \tag{3.37}$$

For example, let us apply Euclid's algorithm to finding the greatest common divisor of, say, 1387 and 3796. Begin by dividing the smaller number 1387 into the larger one (3796) and keeping track of the remainder.

$$3796 = 1387 \times 2 + 1022 \tag{3.38}$$

which also can be written as

$$\frac{3796}{1387} = 2 + \frac{1022}{1387} \tag{3.39}$$

$$= 2 + \frac{1}{\frac{1387}{1022}} . \tag{3.40}$$

The key step of the algorithm is exactly this: *Keep dividing the smaller number 1387 into the larger one.* Thus, let us continue

$$\frac{1387}{1022} = 1 + \frac{365}{1022} \tag{3.41}$$

$$= 1 + \frac{1}{\frac{1022}{365}} . \tag{3.42}$$

In other words,

$$\begin{aligned} \frac{3796}{1387} &= 2+\frac{1022}{1387} && (3.43)\\ &= 2+\frac{1}{1+\frac{1}{\frac{1022}{365}}}\,. && (3.44) \end{aligned}$$

Further

$$\begin{aligned} 1022 &= 365\times 2+292 && (3.45)\\ 365 &= 292\times 1+73 && (3.46)\\ 292 &= 73\times 4 && (3.47) \end{aligned}$$

Omitting parentheses (as is customary), we may write

$$\begin{aligned} \frac{3796}{1387} &= 2+\frac{1022}{1387} && (3.48)\\ &= 2+\frac{1}{1+\frac{1}{\frac{1022}{365}}} && (3.49)\\ &= 2+\frac{1}{1}+\frac{1}{2}+\frac{1}{\frac{365}{292}} && (3.50)\\ &= 2+\frac{1}{1}+\frac{1}{2}+\frac{1}{1+\frac{1}{4}}. && (3.51) \end{aligned}$$

Finally we get, $3796/1387 = [2;1,2,1,4]$. The above explains why the terms a_i are usually called *quotients*.

3.3.2 Examples

For example, the computation of the continued fraction of

$$\pi = 3.14159\,26535\,89793\,23846\,26433\,83279\,50288\,4197$$

leads to $\pi = [3;7,15,1,292,1,1\ldots]$.

The exponential function A beautiful non-simple continued fraction for e is given by

$$e = 2+\cfrac{1}{1+\cfrac{1}{2+\cfrac{2}{3+\cfrac{3}{3+\ldots}}}}\,. \qquad (3.52)$$

Show that rational powers of e can be represented as

$$\begin{aligned} e^{1/2} &= [1;1,1,1,5,1,1,9,1,1,13,\ldots]\,, && (3.53)\\ e^{1/3} &= [1;2,1,1,8,1,1,14,1,1,20,\ldots]\,, && (3.54)\\ e^{1/4} &= [1;3,1,1,11,1,1,19,1,1,27,\ldots]\,, && (3.55)\\ e^{1/5} &= [1;4,1,1,14,1,1,24,1,1,34,\ldots]\,. && (3.56)\end{aligned}$$

3.3.3 Solutions of ordinary differential equations

The method of continued fractions is applicable to linear second-order ordinary differential equations. It yields a solution in terms of a continued fraction. The idea (which was proposed by Leonhard Euler) is to find a simple recurrence pattern that allows one to express the logarithmic derivative of the solution to an ordinary differential equation in terms of a continued fraction.

(a) Consider the linear second-order ordinary differential equation of the form

$$y = Q_0(x)y' + P_1(x)y''\,. \tag{3.57}$$

By differentiating n times with respect to x, show that

$$y^{(n)} = Q_n(x)y^{(n+1)} + P_{n+1}(x)y^{(n+2)}\,, \tag{3.58}$$

where

$$Q_n = \frac{Q_{n-1} + P'_n}{1 - Q'_{n-1}} \quad \text{and} \quad P_{n+1} = \frac{P_n}{1 - Q'_{n-1}}\,. \tag{3.59}$$

(b) Next, divide the original equation by y' and, hence, develop the continued fraction representation of y/y':

$$\frac{y}{y'} = Q_0 + \frac{P_1}{\frac{y'}{y''}} = Q_0 + \frac{P_1}{Q_1 + \frac{P_2}{\frac{y''}{y^{(3)}}}} = Q_0 + \frac{P_1}{Q_1 + \frac{P_2}{Q_2 + \frac{P_3}{\frac{y^{(3)}}{y^{(4)}}}}} = \ldots\,. \tag{3.60}$$

This process can be continued indefinitely. Either it terminates, in which case it represents the reciprocal of the logarithmic derivative, or it does not terminate, in which case it converges under certain reasonable conditions.

Example: Apply this method to the equation $xy'' - xy' - y = 0$, by evaluating the sequence of approximate solutions:

$$\text{(i)} \quad -\frac{x^2+2}{x}\,, \qquad \text{(ii)} \quad -\frac{x^3+5x}{x^2+3}\,, \qquad \text{(iii)} \quad -\frac{x^4+9x^2+8}{x^3+7x}\,. \tag{3.61}$$

3.3.4 Summary

- Continued fractions provide, in some sense, a series of best estimates for an irrational number.
- Functions can also be written as continued fractions, providing a series of better and better rational approximations.
- Continued fractions have also proved useful in the proof of certain properties of numbers such as e (the exponential fraction) and π.

3.4 Appendix

3.4.1 Factorial function

The factorial function is defined as (see Appendix A for how to make a call to a function from you main program):

factorial

```
recursive function factorial(n) result(nfact)
    implicit none
    integer, intent(in) :: n
    double precision :: nfact   ! integer would quickly overflow
    if (n > 0) then
        nfact = n * factorial(n-1)
    else
        nfact = 1
    endif
endfunction factorial
```

Question 12 *Truncating The Exponential Power Series*

For any power series expansion, the accuracy of a polynomial truncation depends upon the number of terms included in the expansion. Since it is impractical to include an infinite number of terms (at which point the precision is perfect), a compromise has to be made in choosing a sufficient number of terms to achieve the desired accuracy. However, in truncating a **Maclaurin series**, the chosen degree of polynomial is always going to best represent the function close to $x = 0$. The further away from $x = 0$, the worse the approximation becomes, and more terms are needed to compensate.

(i) In a Table, compare the accuracy of first, second and third degree polynomial approximations to the function $f(x) = e^x$ for $0 \leq x \leq 1$. Then compare the accuracy of polynomial approximations by potting $f(x) = e^x$ and the polynomials of degrees $n = 2, 3, 4, 5, 6, 7, 8, 9, 10$.

N.B.: A **Maclaurin series** is a Taylor series expansion of a function about 0.

Question 13 *The Geometric Progression*

The numbers:

$$j = 1, 2, 4, 8, ..., 256 \tag{3.62}$$

form a finite sequence generated by the

$$j = 2^i \ , \ \text{where } i = 0, 1, 2, 3, ..., 8 \ . \tag{3.63}$$

Here, the formula is just 2 raised to a power, the value of which is defined by each element of the domain. In general a Geometric Progression is defined as

$$S_{\mathrm{n}} = a, a\, x^1, a\, x^2, a\, x^3, ..., a\, x^{n-1} \ . \tag{3.64}$$

(i) Show that the compact form is $S_{\mathrm{n,c}} = a\left(\frac{1-x^n}{1-x}\right)$. *Hint: make use of successive factorization of x.*

(ii) For the Geometric Progression given by $j = 1, 2, 4, 8, ..., 256$, give the corresponding a, x and n. What is the corresponding compact form, $S_{\mathrm{n,c}}$?

Question 14 *Hydrogen orbital function*

The radial part of the 3*s* atomic orbital for hydrogen has the same form as the expression:

$$R(r) = \left(2\, r^2 - 18\, r + 27\right) \times \exp^{-\frac{x}{3}} \ . \tag{3.65}$$

(i) Rewrite the expression above by replacing the exponential part of the function, $\exp^{-\frac{x}{3}}$, with the first two terms of its power series expansion.

(ii) Compare the 2 expressions by plotting the two functions in the range $0 \leq r \leq 20$. Use different line-styles and color for each function (include the gnuplot script you wrote).

(iii) In this example, you will find that the polynomial approximation to the form of the radial wave function gives an excellent fit for small values of r (i.e. close to the nucleus). How many terms in the exponential power series expansion are needed to get a good fit up to $r = 5$?

Question 15 *Lab exercise*

In the lab session corresponding to this section, you will be reproducing the results shown in Tables 3.1 and 3.2 for the exponential function $\exp(-x)$:

(a) Reproduce the results shown in Table 3.1.

(b) Reduce the size of the argument by writing

$$\exp(-x) = [\exp(-x/2^k)]^{2^k}, \tag{3.66}$$

and show the results for $k = 1$ and $k = 2$. The results should appear in Table 3.1 by adding two extra columns (one for $k = 1$ and one for $k = 2$).

(c) Using the recurrence relation for the power series for $\exp(-x)$, reproduce the results shown in Table 3.2.

Note: in your code, make use of the Horner's scheme when evaluating the exponential function.