Reading Material

Handout on Runge-Kutta Method with Adaptive Step Size
(from *Numerical Recipes* by Press et al.)

From *Computational Physics*:

- Chapter 2: all sections
- Appendix A.1 and A.2
Sources of Errors

(Adapted from First Steps in Numerical Analysis by Hosking et al.)

The main sources of error in obtaining numerical solutions to mathematical problems are:

(a) the model – its construction usually involves simplifications and omissions;
(b) the data – there may be errors in measuring or estimating values;
(c) the numerical method – generally based on some approximation;
(d) the representation of numbers – for example, \( \pi \) cannot be represented exactly by a finite number of digits;
(e) the arithmetic – frequently errors are introduced in carrying out operations such as addition (\(+\)) and multiplication (\(\times\)).
Approximation to numbers

Although it may not seem so to the beginner, it is important to examine ways in which numbers are represented.

1. Number representation

We humans normally represent a number in decimal (base 10) form, although modern computers use binary (base 2) and also hexadecimal (base 16) forms. Numerical calculations usually involve numbers that cannot be represented exactly by a finite number of digits. For instance, the arithmetical operation of division often gives a number which does not terminate; the decimal (base 10) representation of $\frac{2}{3}$ is one example. Even a number such as 0.1 which terminates in decimal form would not terminate if expressed in binary form. There are also the irrational numbers such as the value of $\pi$, which do not terminate. In order to carry out a numerical calculation involving such numbers, we are forced to approximate them by a representation involving a finite number of significant digits ($S$). For practical reasons (for example, the size of the back of an envelope or the 'storage' available in a machine), this number is usually quite small. Typically, a 'single precision' number on a computer has an accuracy of only about 6 or 7 decimal digits (see below).

To five significant digits ($SS$), $\frac{2}{3}$ is represented by 0.66667, $\pi$ by 3.1416, and $\sqrt{2}$ by 1.4142. None of these is an exact representation, but all are correct to within half a unit of the fifth significant digit. Numbers should normally be presented in this sense, correct to the number of digits given.

If the numbers to be represented are very large or very small, it is convenient to write them in floating point notation (for example, the speed of light $2.99792 \times 10^8$ m/s, or the electronic charge $1.6022 \times 10^{-19}$ coulomb). As indicated, we separate the significant digits (the mantissa) from the power of ten (the exponent); the form in which the exponent is chosen so that the magnitude of the mantissa is less than 10 but not less than 1 is referred to as scientific notation.

In 1985 the Institute of Electrical and Electronics Engineers published a standard for binary floating point arithmetic. This standard, known as the IEEE Standard 754, had been widely adopted (it is very common on workstations used for scientific computation). The standard specifies a format for 'single precision' numbers and a format for 'double precision' numbers. The single precision format allows 32 binary digits (known as bits) for a floating point number with 23 of these bits allocated for the mantissa. In the double precision format the values are 64 and 52 bits, respectively. On conversion from binary to decimal, it turns out that
any IEEE Standard 754 single precision number has an accuracy of about six or seven decimal digits, and a double precision number an accuracy of about 15 or 16 decimal digits.

2 Round-off error

The simplest way of reducing the number of significant digits in the representation of a number is merely to ignore the unwanted digits. This procedure, known as chopping, was used by many early computers. A more common and better procedure is rounding, which involves adding 5 to the first unwanted digit, and then chopping. For example, \( \pi \) chopped to four decimal places (4D) is 3.1415, but it is 3.1416 when rounded; the representation 3.1416 is correct to five significant digits (5S). The error involved in the reduction of the number of digits is called round-off error. Since \( \pi \) is 3.14159..., we could remark that chopping has introduced much more round-off error than rounding.

3 Truncation error

Numerical results are often obtained by truncating an infinite series or iterative process (see Step 5). Whereas round-off error can be reduced by working to more significant digits, truncation error can be reduced by retaining more terms in the series or more steps in the iteration; this, of course, involves extra work (and perhaps expense!).
We have noted that a number is to be represented by a finite number of digits, and hence often by an approximation. It is to be expected that the result of any arithmetic procedure (any algorithm) involving a set of numbers will have an implicit error relating to the error of the original numbers. We say that the initial errors propagate through the computation. In addition, errors may be generated at each step in the algorithm, and we may speak of the total cumulative error at any step as the accumulated error.

1 Absolute error

The absolute error is the absolute difference between the exact number \( x \) and the approximate number \( x^* \); that is,

\[ e_{\text{abs}} = |x - x^*| \]

A number correct to \( n \) decimal places has

\[ e_{\text{abs}} \leq 0.5 \times 10^{-n} \]

we expect that the absolute error involved in any approximate number is no more than five units at the first neglected digit.

2 Relative error

The relative error is the ratio of the absolute error to the absolute exact number; that is,

\[ e_{\text{rel}} = \frac{e_{\text{abs}}}{|x|} \leq \frac{e_{\text{abs}}}{|x^*| - e_{\text{abs}}} \]

(Note that the upper bound follows from the triangle inequality; thus

\[ |x^*| = |x + x^* - x| \leq |x| + |x^* - x| \]

so that \(|x| \geq |x^*| - e_{\text{abs}}\).) If \( e_{\text{abs}} \ll |x^*| \), then

\[ e_{\text{rel}} \approx \frac{e_{\text{abs}}}{|x^*|} \]

A decimal number correct to \( n \) significant digits has

\[ e_{\text{rel}} \leq 5 \times 10^{-n} \]
Approximation to functions

An important procedure in Analysis is to represent a given function as an infinite series of terms involving simpler or otherwise more appropriate functions. Thus, if \( f \) is the given function, it may be represented as the series expansion

\[
f(x) = a_0 \phi_0(x) + a_1 \phi_1(x) + \cdots + a_n \phi_n(x) + \cdots
\]

involving the set of functions \( \{ \phi_j \} \). Mathematicians have spent a lot of effort in discussing the convergence of series; that is, in defining conditions for which the partial sum

\[
s_n(x) = a_0 \phi_0(x) + a_1 \phi_1(x) + \cdots + a_n \phi_n(x)
\]

approximates the function value \( f(x) \) ever more closely as \( n \) increases. In Numerical Analysis, we are primarily concerned with such convergent series; computing the sequence of partial sums is an approximation process in which the truncation error may be made as small as we please by taking sufficient terms into account.

1 The Taylor series

The most important expansion to represent a function is the Taylor series. If \( f \) is suitably smooth in the neighbourhood of some chosen point \( x_0 \) we have

\[
f(x) = f(x_0) + hf'(x_0) + \frac{h^2}{2!} f''(x_0) + \cdots + \frac{h^n}{n!} f^{(n)}(x_0) + R_n
\]

where

\[
f^{(k)}(x_0) = \frac{d^k f}{dx^k}
\]

\( h = x - x_0 \) denotes the displacement from \( x_0 \) to point \( x \) in the neighbourhood, and the remainder term is

\[
R_n = \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(\xi)
\]

for some point \( \xi \) between \( x_0 \) and \( x \). (This is known as the Lagrange form of the remainder; its derivation may be found in Section 8.7 of Thomas and Finney (1992) cited in the Bibliography.) Note that the \( \xi \) in this expression for \( R_n \) may be written as \( \xi = x_0 + \theta h \), where \( 0 < \theta < 1 \).

The Taylor expansion converges for \( x \) within some range including the point \( x_0 \), a range which lies within the neighbourhood of \( x_0 \) mentioned above. Within this range of convergence, the truncation error due to discarding terms after the \( x^n \) term (equal to the value of \( R_n \) at point \( x \)) can be made smaller in magnitude than
any positive constant by choosing \( n \) sufficiently large. In other words, by using \( R_n \) to decide how many terms are needed, one may evaluate the function at any point in the range of convergence as accurately as the accumulation of round-off error permits.

From the viewpoint of the numerical analyst, it is most important that the convergence be fast enough. For example, if we consider \( f(x) = \sin x \) we have

\[
\begin{align*}
  f'(x) &= \cos x \\
  f''(x) &= -\sin x \\
  \vdots \\
  \text{etc.}
\end{align*}
\]

and the expansion (about \( x_0 = 0 \)) for \( n = 2k - 1 \) is given by

\[
\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots + \frac{(-1)^{k-1}x^{2k-1}}{(2k-1)!} + R_{2k-1}
\]

with

\[
R_{2k-1} = \frac{(-1)^k x^{2k+1}}{(2k+1)!} \cos \xi
\]

Note that this expansion has only odd-powered terms so, although the polynomial approximation is of degree \( (2k - 1) \), it has only \( k \) terms. Moreover, the absence of even-powered terms means that the same polynomial approximation is obtained with \( n = 2k \), and hence \( R_{2k-1} = R_{2k} \); the remainder term \( R_{2k-1} \) given above is actually the expression for \( R_{2k} \). Since \( |\cos \xi| \leq 1 \), then

\[
|R_{2k-1}| \leq \frac{|x|^{2k+1}}{(2k+1)!}
\]

if \( 5D \) accuracy is required, it follows that we need only take \( k = 2 \) at \( x = 0.1 \), and \( k = 4 \) at \( x = 1 \) (since \( 9! = 362,880 \)). On the other hand, the expansion for the natural (base \( e \)) logarithm,

\[
\ln(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots + \frac{(-1)^{n-1}x^n}{n} + R_n
\]

where

\[
R_n = \frac{(-1)^n x^{n+1}}{(n + 1)(1 + \xi)^{n+1}}
\]
is less suitable. Although only $n = 4$ terms are needed to give $5D$ accuracy at $x = 0.1$, $n = 13$ is required for $5D$ accuracy at $x = 0.5$, and $n = 19$ gives just $1D$ accuracy at $x = 1$!

Further, we remark that the Taylor series is not only used extensively to represent functions numerically, but also to analyse the errors involved in various algorithms.