Numerical Methods

King Saud University

Aims

In this lecture, we will . . .

- Introduce the topic of numerical methods
- Consider the Error analysis and sources of errors

Introduction

A numerical method which can be used to solve a problem will be called an algorithm. An algorithm is a complete and unambiguous set of procedures leading to the solution of a mathematical problem. The selection or construction of appropriate algorithms properly falls within the scope of numerical analysis. Having decided on a specific algorithm or set of algorithms for solving the problem, the numerical analyst should consider all the sources of error that may affect the results. Considering how much accuracy is required, estimate the magnitude of the round-off and discretization error, determine an appropriate step size or the number of iterations required, provide for adequate checks on the accuracy, and make allowance for corrective action in cases of nonconvergence.

Computer based solutions

The major steps involved to solve a given problem using a computer are:

- Modeling: Setting up a mathematical model, i.e., formulating the problem in mathematical terms, taking into account the type of computer one wants to use.
- Choosing an appropriate numerical method (algorithm) together with a preliminary error analysis (estimation of error, determination of steps, size etc.)
- Programming, usually starting with a flowchart showing a block diagram of the procedures to be performed by the computer and then writing, such as MATLAB and MAPLE programs.
- Operation or computer execution.
- Interpretation of results, which may include decisions to rerun if further data are needed.

In elementary calculus we learn how to differentiate and integrate to get exact answers to remarkably diverse range of realistic problems that could not be solved by purely algebraic methods. Unfortunately, from a practical point of view, the techniques of elementary (or even advanced) calculus alone are not adequate for solving calculus type problems such as solving polynomial equations of degree greater than four or even a simple equation such as

$$x = \cos x,$$

also, to evaluate integrals of type

$$\int_a^b e^{x^2} dx \quad \text{and} \quad \int_a^b \frac{\sin x}{x} dx; \qquad \text{etc.},$$

it is impossible to get the exact solutions of these problems. Even when an analytical solution can be found it may be of more theoretical than practical. Fortunately, one rarely needs exact answers.

Types of Numerical Methods

There are two basic types of numerical methods:

- Direct numerical
  which computes the solution to a problem in a finite number of steps. These methods would give the precise answer if they were performed in infinite precision arithmetic. Examples include Gaussian elimination, the LU factorization method for solving systems of linear equations. In practice, finite precision is used and the result is an approximation of the true solution (assuming stability). In the absence of rounding errors, direct methods would deliver an exact solution.
- An indirect (iterative) numerical methods
  which is a mathematical procedure that generates a sequence of improving approximate solutions for a class of problems. A specific implementation of an iterative method, including the termination criteria, is an algorithm of the iterative method.

An iterative method is called convergent if the corresponding sequence converges for given initial approximations. A mathematically rigorous convergence analysis of an iterative method is usually performed. A convergence test, often involving the residual, is specified in order to decide when a sufficiently accurate solution has (hopefully) been found. Even using infinite precision arithmetic these methods would not reach the solution within a finite number of steps (in general). Iterative methods are often the only choice for nonlinear equations. However, iterative methods are often useful even for linear problems involving a large number of variables, where direct methods would be prohibitively expensive (and in some cases impossible) even with the best available computing power. Examples include Newton's method, bisection method, and Jacobi iteration. In computational matrix algebra, iterative methods are generally needed for large problems.

An iterative method for the given problem converges means:- approximate values should come in side the given interval **I-** difference between two successive approximations should be small. Otherwise diverges. An iterative process may converge or diverge. If the divergence occurs, the procedure should be terminated because there may be no solution. We can restart the procedure by changing the initial approximation if necessary. But in the case of convergence we have to apply some stopping procedures to end the computations. In the following there are some more stopping criterion that can be used, each of them can be apply to any iterative technique considered in this chapter. By selecting a tolerance $\epsilon > 0$ and generate approximate solutions $x_1, x_2, \ldots, x_n$ until one of the following conditions is satisfied:

$$|x_n - x_{n-1}| < \epsilon \quad \text{or} \quad \frac{|x_n - x_{n-1}|}{|x_n|} < \epsilon, \qquad x_n \neq 0.$$

Sometimes difficulties can arise using any of these stopping criteria. For example, there exist sequence $\{x_n\}_0^\infty$ with the property that the differences $(x_n - x_{n-1})$ converge to zero while the sequence itself diverges. It is also possible for $f(x_n)$ to be close to zero while x_n differs significantly from α. Without additional knowledge about $f(x)$ or α, the above second inequality is the best stooping criterion to apply because it tests relative error. Also, one of the other stopping criteria is to use a fixed number of iterations, and then the final approximation x_n may be considered as the value of the required root. This type of stopping criteria is helpful when the convergence is very slow. It is important to note that in considering whether an iteration *converges* or not, it may be necessary to ignore the first few iterations since the procedure may appear diverge initially, even though it ultimately converges.

In conclusion **Iterative methods** are more common than **Direct methods** in numerical analysis. Some methods are direct in principle but are usually used as though they were not. For these methods the number of steps needed to obtain the exact solution is so large that an approximation is accepted in the same manner as for an iterative method.
The numerical methods deal with numbers. We exam the sources of various types of computational errors.

Error Analysis

An approximate number p is a number that differs but slightly from an exact number α. We write

$$p \approx \alpha.$$

By error E of an approximate number p, we mean the difference between the exact number α and its computed approximation p. Thus we define

$$E = \alpha - p. \tag{1}$$

If $\alpha > p$, the error E is positive, and if $\alpha < p$, the error E is negative. In many situations, the sign of the error may not be known and might even be irrelevant. Therefore, we define *absolute error* as

$$|E| = |\alpha - p|. \tag{2}$$

The *relative error* RE of an approximate number p is the ratio of the absolute error of the number to the absolute value of the corresponding exact number α.

Thus

$$RE = \frac{|\alpha - p|}{|\alpha|}, \qquad \alpha \neq 0. \tag{3}$$

If we approximate $\frac{1}{3}$ by 0.333, we have

$$E = \frac{1}{3} \times 10^{-3} \qquad \text{and} \qquad RE = 10^{-3}.$$

Note that relative error is generally a better measure of the extend of error than the actual error. But one should also note that relative error is undefined if the exact answer is equal to zero. Generally, we shall be interested in E (or sometimes $|E|$) rather than RE, but when the true value of a quantity is very small or very large, relative errors are more meaningful. For example, if the true value of a quantity is 10^{15}, and error of 10^{6} is probably not serious, but this is more meaningfully expressed by saying that $RE = 10^{-9}$. In actual computation of the relative error, we shall often replace the unknown true value by the computed approximate value.

Sometimes the quantity

$$\frac{|\alpha - p|}{|\alpha|} \times 100\%, \tag{4}$$

is defined as *percentage error*. From the above example, we have

$$PE = 0.001 \times 100 = 0.1\%.$$

In investigating the effect of the total error in various methods, we shall often mathematically derive an error, called, *error bound* and which is a limit on how large the error can be.

Sources of Errors

We will consider three types of errors which occur in a computation

- Human Error
  These types of errors arise when the equations of the mathematical model are formed, due to sources such as the idealistic assumptions made to simplify the model, inaccurate measurements of data, miscopying of figures, the inaccurate representation of mathematical constants (for example, if the constant π occurs in an equation, we must replace π by 3.1416 or 3.141593, etc.).

- Truncation Error
  This type of error is caused when we are forced to use mathematical techniques which give approximate, rather than exact, answer. For example, suppose that we use the Maclaurin's series expansion to represent $\sin x$, so that

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

  If we want a number that approximates $\sin(\frac{\pi}{2})$, we must terminate the expansion in order to obtain

$$\sin(\frac{\pi}{2}) = \frac{\pi}{2} - \frac{(\pi/2)^3}{3!} + \frac{(\pi/2)^5}{5!} - \frac{(\pi/2)^7}{7!} + E,$$

  where E is the truncation error introduced in the calculation. The Taylor series is the most important means used to derive numerical schemes and analysis truncation errors.

- Round-off Error
  This type of errors are associated with the limited number of digits numbers in the computer. For example, by rounding off 1.32463672 to six decimal places to give 1.324637. Any further calculation involving such a number will also contain an error. Round-off numbers according to following rules:
  1. If first discarded digit is less than 5, leave the remaining digits of number unchanged, that is, $48.47263 \approx 48.4726$.
  2. If the first discarded digit is exceeds 5, add 1 to the retained digit. For example, $48.4726 \approx 48.473$.
  3. If the first discarded digit is exactly 5 and there are nonzero among those discarded, add, 1 to the last retained digit. For example, $3.0554 \approx 3.06$.
  4. If the first discarded digit is exactly 5 and all other discarded digits are zero, the last retained digit is left unchanged if it is *even*, otherwise 1 is added to it. For example,

$$\begin{aligned} 3.05500 &\approx 3.06 \\ 3.04500 &\approx 3.04. \end{aligned}$$

To understand the nature of round-off errors, it is necessary to learn the ways numbers are stored and additions and subtractions are performed in a computer.

A solution is *correct within k decimal places* if the error is less than 0.5×10^{-k}.

If x^* is an approximation to x, then we say that x^* approximates x to k *significant digits* if k is the largest nonnegative integer for which $\left|\frac{x - x^*}{x}\right| < 5 \times 10^{-k}$.

Summary

In this lecture, we ...

- Introduced the topic of numerical methods
- Considered the Error analysis and sources of errors