CHAPTER-I

MATHEMATICAL MODELLING

INTRODUCTION

GENERAL

Numerical methods are techniques by which mathematical problems are formulated so that they can be solved with arithmetic operations. Although there are many kinds of numerical methods, they have common characteristics: they invariably involve large numbers of tedious arithmetic calculations. It is little wonder that with the development of fast, efficient digital computers, the role of numerical methods in engineering problem solving has increased dramatically in recent years.

The reasons why we should study numerical methods are because:

1. Numerical methods are extremely powerful problem solving tools.
2. During your career, you may often have occasions to use commercially available prepackaged, or "canned", computer programs that involve numerical methods.
3. Many problems cannot be appreciated using canned programs.
4. They are efficient vehicles for learning to use computers.
5. They provide a vehicle for you to reinforce your understanding of mathematics.

Mathematical Modelling

A **mathematical model** is an abstract model that uses mathematical language to describe the behaviour of a system. Mathematical **models** are used particularly in the natural sciences and engineering disciplines (such as physics, biology, and electrical engineering) but also in the social sciences (such as economics, sociology and political science); physicists, engineers, computer scientists, and economists use mathematical models most extensively.

Often when engineers analyze a system to be controlled or optimized, they use a mathematical model. In analysis, engineers can build a descriptive model of the system as a hypothesis of how the system could work, or try to estimate how an unforeseeable event could affect the system. Similarly, in control of a system, engineers can try out different control approaches in simulations.

A mathematical model usually describes a system by a set of variables and a set of equations that establish relationships between the variables. The values of the variables can be practically anything; real or integer numbers, logical values or characters, for example. The variables represent some properties of the system, for example, measured system outputs often in the form of signals, timing data, counters, event occurrence (yes/no). The actual model is the set of functions that describe the relations between the different variables.

A mathematical model can be represented as a functional relationship of the form

Dependent variable = f(independent variables, parameters, forcing functions)

Where:

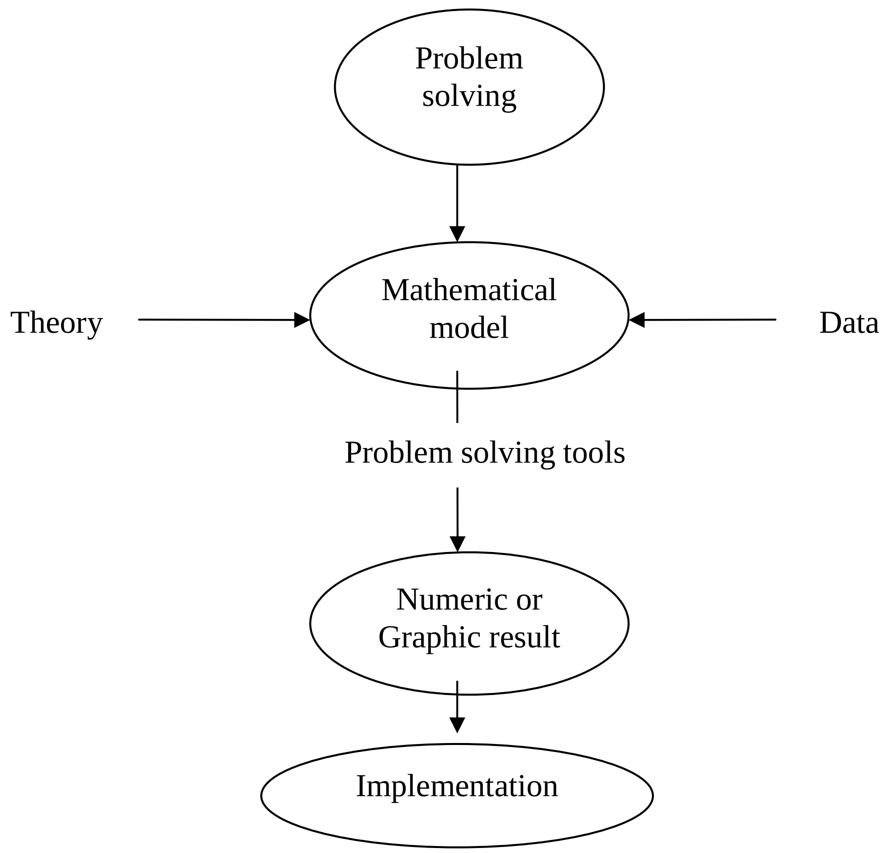
- The dependent (state) variable is a characteristic that usually reflects the behavior or state of the system.
- The independent (decision) variables are usually dimensions such as time and space, along which the system's behavior is determined.
- The parameters (constants) are reflective of the system's properties or composition.
- The forcing functions are external influences acting up on the system.

Mathematical modeling and Engineering problem solving

Knowing and understanding are prerequisites for the effective implementation of any tool. This is particularly true when using computers to solve engineering problems. Although they have great potential utility, computers are practically useless without a fundamental understanding of how engineering systems work.

Mathematical modelings have a rule in engineering problem solving. A mathematical model is defined as a formulation or equation that expresses the essential features of a physical system or process in mathematical terms.

The engineering problem solving process is described by the following diagram.



Classifying mathematical models

Mathematical models can be classified in several ways, some of which are described below.

1. **Linear vs. nonlinear**: Mathematical models are usually composed by variables, which are abstractions of quantities of interest in the described systems, and operators that act on these variables, which can be algebraic operators, functions, differential operators, etc. If all the operators in a mathematical model present linearity the resulting mathematical model is defined as linear. A model is considered to be nonlinear otherwise. In a mathematical programming model, if the objective functions and constraints are represented entirely by linear equations, then the model is regarded as a linear model. If one or more of the objective functions or constraints are represented with a nonlinear equation, then the model is known as a nonlinear model.

2. **Deterministic vs. probabilistic (stochastic)**: A deterministic model is one in which every set of variable states is uniquely determined by parameters in the model and by sets of previous states of these variables. Therefore, deterministic models perform the same way for a given set of initial conditions. Conversely, in a stochastic model, randomness is present, and variable states are not described by unique values, but rather by probability distributions.

3. **Static vs. dynamic**: A static model does not account for the element of time, while a dynamic model does. Dynamic models typically are represented with difference equations or differential equations.

4. **Lumped parameters vs. distributed parameters**: If the model is homogeneous (consistent state throughout the entire system) the parameters are lumped. If the model is heterogeneous (varying state within the system), then the parameters are distributed. Distributed parameters are typically represented with partial differential equations. It is usually appropriate to make some approximations to reduce the model to a sensible size. Engineers often can accept some approximations in order to get a more robust and simple model. For example Newton's second law of motion is an approximated model of the real world. Still, Newton's model is quite sufficient for most ordinary-life situations, that is, as long as particle speeds are well below the speed of light, and we study macro-particles only.

Newton formulated his second laws of motion, which states that the time rate of change of momentum of a body is equal to the resultant force acting on it.

The mathematical expression or model of the second law is the well-known equation

$$F = ma \qquad (1)$$

Where F = net force acting on the body (N)

m = mass of the object (kg)

a = its acceleration (m/s^2)

The second law can be rewritten in the format shown below by simply dividing both sides by m to give

$$a = F/m \qquad (2)$$

where: a = the dependent variable reflecting the system's behavior
F = the forcing function &
m = a parameter representing property of the system.

Eqn. (2) has several characteristic that are typical of mathematical models of the physical world:

1. It describes a natural process or system in mathematical terms.
2. It represents an idealization and simplification of reality
3. Finally, it yields reproductive results and consequently, can be used for predictive purposes.

Model evaluation

An important part of the modeling process is the evaluation of an acquired model. *How do we know if a mathematical model describes the system well?* This is not an easy question to answer. Usually the engineer has a set of measurements from the system which are used in creating the model. Then, if the model was built well, the model will adequately show the relations between system variables for the measurements at hand. The question then becomes: *How do we know that the measurement data are a representative set of possible values?* Does the model describe well the properties of the system between the measurement data (interpolation)? Does the model describe well events outside the measurement data (extrapolation)? A common approach is to split the measured data into two parts; training data and verification data. The training data are used to *train* the model, that is, to estimate the model parameters. The verification data are used to evaluate model performance. Assuming that the training data and verification data are not the same, we can assume that if the model describes the verification data well, then the model describes the real system well. However, this still leaves the *extrapolation question* open. How well does this model describe events outside the measured data? Consider again the above model. Newton made his measurements without advanced equipment, so he could not measure properties of particles travelling at speeds close to the speed of light. Likewise, he did not measure the movements of molecules and other small particles, but macro particles only. It is then not surprising that his model does not extrapolate well into these domains, even though his model is quite sufficient for ordinary life physics.

Approximations and Round –off errors

Numerical technique yield usually estimates that are close to the exact analytical solution. There is a discrepancy, or error, because the numerical method involved an approximation. For many applied engineering problems we cannot compute exactly the error associated with our numerical methods. In these cases we must settle for approximations or estimates of the errors.

Numerical errors arise from the use of approximations to represent exact mathematical operations and quantities. These include truncation errors, which is the discrepancy introduced by the fact that numerical methods may employ approximations to represent exact mathematical operations and quantities, and round-off errors, which result when numbers having limited significant figures are used to represent exact numbers.

Absolute and Relative Errors

The relationship between the exact, or true, result and the approximation can be formulated as:-

True value = approximation value + error;

Absolute error = true value – approximation value

A shortcoming of this definition is that it takes no account of the order of the magnitude of the value under examination. One way to account for the magnitudes of the quantities being evaluated is to normalize the error to the true value, as in

$$E_t = \frac{\text{true error}}{\text{true value}}$$ where E_t - true fractional relative error

$$\varepsilon_t = \frac{\text{true error}}{\text{true value}} * 100\%$$ where ε_t - is the true percent relative error.

The relative error is sometimes is important to get a feeling on how significant an error is. However, in real-world applications, we will obviously not know the true answer a prior. For these situations, an alternative is to normalize the error using the best available estimate of the true value.

Still in some applications the solution of a given problem is obtained through successive approximation. In such cases the value at the end of any iteration apart from the last approximation of the final approximate solution.

Hence in such cases an estimate of the error, at the end of iteration, is made based on approximate values of the present and previous iteration. Therefore, Absolute error E_a and relative error E_r can be calculated as:-

E_a = present approximation – previous approximation

$$E_r = \frac{(\text{present approximation} - \text{previous approximation})}{\text{present approximation}}$$

An approximate percentage relative error can be calculated as; $\left| \frac{(X^{new} - X^{old})}{X^{new}} \right| * 100\%$

If the approximation is greater than the true value, the error is negative. Often, when performing computations, we may not be concerned with the sign of the error, but we are interested in whether the percent absolute value is lower than the given error tolerance value or not.

Truncation errors and the Taylor series

Truncation errors are those that result from using an approximation in place of an exact mathematical procedure. Mathematical formulation that is used widely in numerical methods to express functions in an approximate fashion is the Taylor series.

Taylor series provide a means to predict a function value at one point in terms of the function value and its derivatives at another point. In particular, the theorem states that any smooth function can be approximated as a polynomial.
The complete Taylor series expansion is expressed as

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 + \frac{f'''(x_i)}{3!}(x_{i+1} - x_i)^3 + ... + R_n \qquad \textbf{(1-1)}$$

Note that Eq.1-1 is an infinite series; the reminder term is included to account all the terms from n+1 to infinity

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x_{i+1} - x)^{n+1} \qquad \textbf{(1-2)}$$

In general the n^{th}-order Taylor series expansion will be exact for an n^{th} order polynomial. For other differentiable and continuous functions, such as exponentials and sinusoids, a finite number of terms will not yield an exact estimate. The inclusion of few more terms will result in an approximation that is close enough to the true value for practical purposes. The assessment of how many terms are required to get “close enough” is based on the remainder term of the expansion. Equation 1-2 is useful to get insight to truncation errors. We can choose how far away from x we want to evaluate f(x), and we can control the number of terms we include in the expansion. Accordingly Eqn.1-2 is usually expressed as

$$R_n = O(h^{n+1}) \qquad \textbf{(1-3)}$$

where $h = x_{i+1} - x_i$,the nomenclature $O(h^{n+1})$ means that the truncation error is of the order of h^{n+1}.

That is, the error is proportional to the step size h raised to the $(n+1)^{th}$ power.

Example 1

Use zero – through 3rd order Taylor series expansion, Find f (6), use a base point x = 4.

$f(x) = x^3+3x^2+2x+5$

Solution

$f(4)=125$, $f'(4)=74$, $f''(4)=30$, $f'''(4)=6$ and all other higher derivatives of $f(x)$ at $x=4$ are zero.

$$f(x+h)=f(x)+f'(x)h+f''(x)\frac{h^2}{2!}+f'''(x)\frac{h^3}{3!}+\cdots$$

$$x=4\ ;\ h=6-4\ =2$$

Since fourth and higher derivatives of $f(x)$ are zero at $x=4$. The Taylor series for f(x) at point x=4 is given by

$$f(4+2)=f(4)+f'(4)2+f''(4)\frac{2^2}{2!}+f'''(4)\frac{2^3}{3!}$$

$$f(6)=125+74(2)+30\left(\frac{2^2}{2!}\right)+6\left(\frac{2^3}{3!}\right)$$

$$=125+148+60+8$$

$$=341$$

Example 2

The Taylor series for e^x at point $x=0$ is given by

$$e^x=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\frac{x^5}{5!}+\cdots$$

What is the truncation (true) error in the representation of e^1 if only four terms of the series are used?

Solution

a) If only four terms of the series are used, then

$$e^x\approx 1+x+\frac{x^2}{2!}+\frac{x^3}{3!}$$

$$e^1\approx 1+1+\frac{1^2}{2!}+\frac{1^3}{3!}$$

$$=2.66667$$

The truncation (true) error would be the unused terms of the Taylor series, which then are

$$E_t=\frac{x^4}{4!}+\frac{x^5}{5!}+\cdots$$

$$=\frac{1^4}{4!}+\frac{1^5}{5!}+\cdots$$

$$\cong 0.0516152$$

b) As you can see in the previous example that by taking more terms, the error bounds decrease and hence you have a better estimate of e^1.

How many terms it would require to get an approximation of e^1 within a magnitude of true error of less than 10^{-6}?

Solution

Using $(n+1)$ terms of the Taylor series gives an error bound of

$$R_n(x) = \frac{(x-h)^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

$$x = 0, h = 1, f(x) = e^x$$

$$R_n(0) = \frac{(0-1)^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

$$= \frac{(-1)^{n+1}}{(n+1)!} e^c$$

Since

$$x < c < x + h$$

$$0 < c < 0 + 1$$

$$0 < c < 1$$

$$\frac{1}{(n+1)!} < |R_n(0)| < \frac{e}{(n+1)!}$$

So if we want to find out how many terms it would require to get an approximation of e^1 within a magnitude of true error of less than 10^{-6},

$$\frac{e}{(n+1)!} < 10^{-6}$$

$$(n+1)! > 10^6 e$$

$$(n+1)! > 10^6 \times 3$$ (as we do not know the value of e but it is less than 3).

$$n \geq 9$$

So 9 terms or more will get e^1 within an error of 10^{-6} in its value.