Chapter 9
Applications in Numerical Analysis

Numerical methods are commonly used for solving mathematical problems that are formulated in science and engineering where it is difficult or impossible to obtain exact solutions. MATLAB has a large library of functions for numerically solving a wide variety of mathematical problems. This chapter explains a number of the most frequently used of these functions. It should be pointed out here that the purpose of this book is to show users how to use MATLAB. Some general information on the numerical methods is given, but the details, which can be found in books on numerical analysis, are not included.

The following topics are presented in this chapter: solving an equation with one unknown, finding a minimum or a maximum of a function, numerical integration, and solving a first-order ordinary differential equation.

9.1 SOLVING AN EQUATION WITH ONE VARIABLE

An equation with one variable can be written in the form $f(x) = 0$. A solution to the equation (also called a root) is a numerical value of x that satisfies the equation. Graphically, a solution is a point where the function $f(x)$ crosses or touches the x axis. An exact solution is a value of x for which the value of the function is exactly zero. If such a value does not exist or is difficult to determine, a numerical solution can be determined by finding an x that is very close to the solution. This is done by the iterative process, where in each iteration the computer determines a value of x that is closer to the solution. The iterations stop when the difference in x between two iterations is smaller than some measure. In general, a function can have zero, one, several, or an infinite number of solutions.

In MATLAB a zero of a function can be determined with the command (built-in function) `fzero` with the form:

```
x = fzero(function,x0)
```

Solution — The function to be solved. — A value of x close to where the function crosses the axis.

The built-in function `fzero` is a MATLAB function function (see Section 7.9), which means that it accepts another function (the function to be solved) as an input argument.

Additional details on the arguments of `fzero`:

- `x` is the solution, which is a scalar.
- `function` is the function to be solved. It can be entered in several different ways:
  1. The simplest way is to enter the mathematical expression as a string.
  2. The function is created as a user-defined function in a function file and then the function handle is entered (see Section 7.9.1).
  3. The function is created as an anonymous function (see Section 7.8.1) and then the name of the anonymous function (which is the name of the handle) is entered (see Section 7.9.1).

  (As explained in Section 7.9.2, it is also possible to pass a user-defined function and an inline function into a function function by using its name. However, function handles are more efficient and easier to use, and should be the preferred method.)

- The function has to be written in a standard form. For example, if the function to be solved is $xe^{-x} = 0.2$, it has to be written as $f(x) = xe^{-x} - 0.2 = 0$. If this function is entered into the `fzero` command as a string, it is typed as: `'x*exp(-x)-0.2'`.
- When a function is entered as an expression (string), it cannot include predefined variables. For example, if the function to be entered is $f(x) = xe^{-x} - 0.2$, it is not possible to define `b=0.2` and then enter `'x*exp(-x)-b'`.
- `x0` can be a scalar or a two-element vector. If it is entered as a scalar, it has to be a value of x near the point where the function crosses (or touches) the x axis. If `x0` is entered as a vector, the two elements have to be points on opposite sides of the solution. If $f(x)$ crosses the x axis, then $f(x0(1))$ has a different sign than $f(x0(2))$. When a function has more than one solution, each solution can be determined separately by using the `fzero` function and entering values for `x0` that are near each of the solutions.

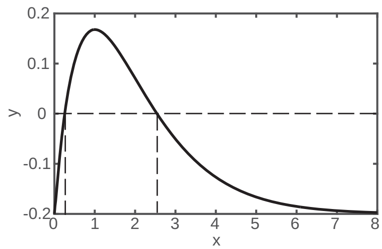
- A good way to find approximately where a function has a solution is to make a plot of the function. In many applications in science and engineering the domain of the solution can be estimated. Often when a function has more than one solution only one of the solutions will have a physical meaning.

Sample Problem 9-1: Solving a nonlinear equation

Determine the solution of the equation $xe^{-x} = 0.2$.

Solution

The equation is first written in the form of a function: $f(x) = xe^{-x} - 0.2$. A plot of the function, shown on the right, shows that the function has one solution between 0 and 1 and another solution between 2 and 3. The plot is obtained by typing




```
>> fplot('x*exp(-x)-0.2',[0 8])
```

in the Command Window. The solutions of the function are found by using the `fzero` command twice. First the equation is entered as a string expression, and a value of `x0` between 0 and 1 (`x0` = 0.7) is used. Second, the equation to be solved is written as an anonymous function, which is then used in `fzero` with `x0` between 2 and 3 (`x0` = 2.8). This is shown below:

```
>> x1=fzero('x*exp(-x)-0.2',0.7)
x1 =
    0.2592
>> F=@(x)x*exp(-x)-0.2
F =
    @(x)x*exp(-x)-0.2
>> fzero(F,2.8)
ans =
    2.5426
```

The function is entered as a string expression.

The first solution is 0.2592.

Creating an anonymous function.

Using the name of the anonymous function in `fzero`.

The second solution is 2.5426.

Additional comments:

- The `fzero` command finds zeros of a function only where the function crosses the x axis. The command does not find a zero at points where the function touches but does not cross the x axis.
- If a solution cannot be determined, `NaN` is assigned to `x`.
- The `fzero` command has additional options (see the Help Window). Two of the more important options are: