Subsections

Roots of Equations

Backeting Methods

Graphical Methods

A simple method for obtaining an estimate of the root of the equation $f(x)=0$ is to make a plot of the function and observe where it crosses the x axis.

The Bisection Method

In general, if $f(x)$ is real and continuous in the interval from x_l to x_u and $f(x_l)$ and $f(x_u)$ have opposite signs, that is

$$f(x_l)f(x_u) < 0 \tag{2.1}$$

then there is at least one real root between x_l and x_u.

The False-Position Method

A shortcoming of the bisection method

- equally dividing the interval
- no account for for the magnitudes of $f(x_l)$ and $f(x_u)$

An alternative method is to join $f(x_l)$ and $f(x_u)$ by a straight line and the intersection of this line with the x axis represents an improved estimate of the root. This mothod is called as method of false position, regula falsi, or linear interpolation method.

The false-position formula is

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)} \tag{2.2}$$

See Figure 5.14 in textbook

Open Methods

- bracketing method : the root is located within an interval prescribed by a lower and an upper bound.
- open method : require only a single starting value of x or two starting point that do not necessarily bracket the root.

Simple Fixed-point Iteration

Open methods employ a formula to predict the root. Such a formula can be develped for simple fixed-point iteration by rearranging the function $f(x) = 0$ so that s is on the left-hand side of the equation:

$$x = g(x) \tag{2.3}$$

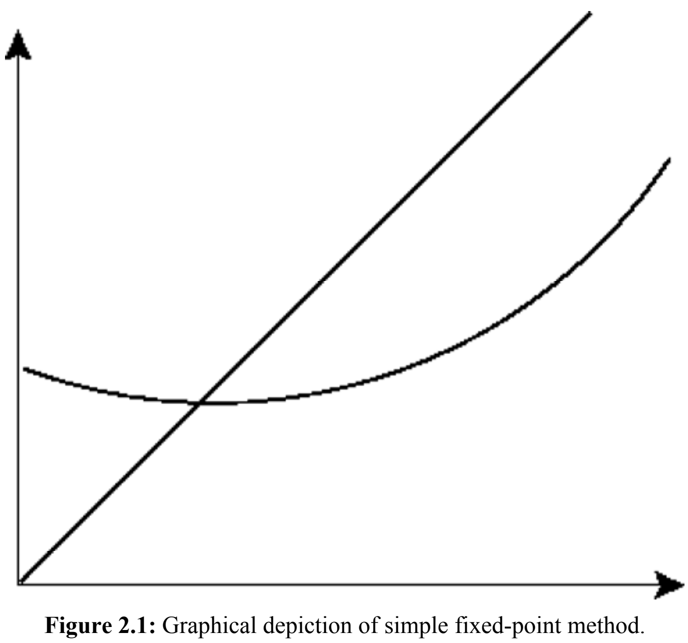

Figure 2.1: Graphical depiction of simple fixed-point method.

The Newton-Raphson Method

If the initial guess at the root is x_i, a tangent can be extended from the point $[x_i, x(x_i)]$. The point where this tangent crosses the x axis usaually represents an improved estimate of the root.

The Newton-Raphson formula is

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{2.4}$$

Pitfalls of the Newton-Raphson Method are shown in Figure 6.6

The Secant Method

A potential problem in implementing the Newton-Raphson method is the evaluation of the derivative. In Secant method the derivative is approximated by a backward finite divided difference

$$f'(x_i) \simeq \frac{f(x_{i-1}) - f(x_i)}{x_{i-1} - x_i} \tag{2.5}$$

The Secant formula is

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)} \tag{2.6}$$

The difference between the secant method and the false-position method is how one of the initial values is replaced by the new estimate.

Rather than using two arbitrary values to estimate the derivative, an alternative approach involves a fractional perturbation of the independent variable to estimate $f'(x)$,

$$f'(x_i) \simeq \frac{f(x_i + \delta x_i) - f(x_i)}{\delta x_i} \tag{2.7}$$

where δ is a small perturbation fraction. This approximation gives the following iterative equation:

$$x_{i+1} = x_i - \frac{\delta x_i f(x_i)}{f(x_i + \delta x_i) - f(x_i)} \tag{2.8}$$

Multiple Roots

Some difficulities in multiple roots problem

- no change in sign at even multiple roots
- $f(x)$ and $f'(x)$ go to zero at the root
- the Newton-Raphson method and secant method show linear, rather than quadratic, convergence for multiple roots.

Another alternative is to define a new function $u(x)$,

$$u(x) = \frac{f(x)}{f'(x)} \tag{2.9}$$

An alternative form of the Newton-Raphson method:

$$x_{i+1} = x_i - \frac{u(x)}{u'(x)} \tag{2.10}$$

where $u'(x)$ is

$$u'(x) = \frac{f'(x)f'(x) - f(x)f''(x)}{\left[f'(x)\right]^2} \tag{2.11}$$

And finally

$$x_{i+1} = x_i - \frac{f(x_i)f'(x_i)}{\left[f'(x_i)\right]^2 - f(x_i)f''(x_i)} \tag{2.12}$$

Systems of Nonlinear Equations

The Newton-Raphson method can be used to solve a set of nonlinear equations. The Newton-Raphson method employ the derivative of a function to estimate its intercept with the axis of the independent variable. This estimate was based on a first-order Taylor series expansion. For example, we consider two variable case,

$$u(x, y) = 0 \tag{2.13}$$

$$v(x, y) = 0 \tag{2.14}$$

A first-order Taylor series expasion can be written as

$$u_{i+1} = u_i + (x_{i+1} - x_i)\frac{\partial u_i}{\partial x} + (y_{i+1} - y_i)\frac{\partial u_i}{\partial y} \tag{2.15}$$

$$v_{i+1} = v_i + (x_{i+1} - x_i)\frac{\partial v_i}{\partial x} + (y_{i+1} - y_i)\frac{\partial v_i}{\partial y} \tag{2.16}$$

The above equation can be rearranged to give

$$\frac{\partial u_i}{\partial x}x_{i+1} + \frac{\partial u_i}{\partial y}y_{i+1} = -u_i + x_i\frac{\partial u_i}{\partial x} + y_i\frac{\partial u_i}{\partial y} \tag{2.17}$$

(2.18)

$$\frac{\partial v_i}{\partial x} x_{i+1} + \frac{\partial v_i}{\partial y} y_{i+1} = -v_i + x_i \frac{\partial v_i}{\partial x} + y_i \frac{\partial v_i}{\partial y}$$

Finally

$$x_{i+1} = x_i - \frac{u_i \frac{\partial v_i}{\partial y} - v_i \frac{\partial u_i}{\partial y}}{\frac{\partial u_i}{\partial x} \frac{\partial v_i}{\partial y} - \frac{\partial u_i}{\partial y} \frac{\partial v_i}{\partial x}} \tag{2.19}$$

$$y_{i+1} = y_i - \frac{v_i \frac{\partial u_i}{\partial x} - u_i \frac{\partial v_i}{\partial x}}{\frac{\partial u_i}{\partial x} \frac{\partial v_i}{\partial y} - \frac{\partial u_i}{\partial y} \frac{\partial v_i}{\partial x}} \tag{2.20}$$

Roots of Polynomials

The roots of polynomials have the following properties

- For an nth-order equation, there are n real and complex roots.
- If n is odd, there is at least one real root.
- If complex roots exist, they exist in conjugate pairs.

Polynomials in Engineering and Science

Polynomial are used extensively in curve-fitting. However, another most important application is in characterizing dynamic system and, in particular, linear systems.

For example, we consider the following simple second-order ordinary differential equation:

$$a_2 \frac{d^2 y}{dt} + a_1 \frac{dy}{dt} + a_0 y = F(t) \tag{2.21}$$

where $F(t)$ is the forcing function. And the above ODE can be expressed as a system of 2 first-order ODEs by defining a new variable z,

$$z = \frac{dy}{dt} \tag{2.22}$$

This reduces the problem to solving

$$\frac{dz}{dt} = \frac{F(t) - a_1 z - a_0 y}{a_2} \tag{2.23}$$

$$\frac{dy}{dt} = z \tag{2.24}$$

In a simliar fashion, nth-order linear ODE can always be expressed as a system of n first-order ODEs.

The general solution of ODE equation deals with the case when the forcing function is set to zero.

$$a_2 \frac{d^2 y}{dt} + a_1 \frac{dy}{dt} + a_0 y = 0 \tag{2.25}$$

This equation gives something very fundamental about the system being simulated-that is, how the system reponds in the absence of external stimuli. The general solution to all unforced linear system is of the form $y = e^{rt}$.

$$a_2 r^2 e^{rt} + a_1 r e^{rt} + a_0 e^{rt} = 0 \tag{2.26}$$

or cancelling the exponential terms,

$$a_2 r^2 + a_1 r + a_0 = 0 \tag{2.27}$$

This polynomial is called as characteristic equation and these r's are referred to as eigenvalues.

$$\begin{matrix} r_1 \\ r_2 \end{matrix} = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2 a_0}}{a_0} \tag{2.28}$$

- overdamped case : all real roots
- critically damped case : only one root
- underdamped case : all complex roots

Computing with Polynomials

For nth-order polynomial calculation, general approach requires $n(n+1)/2$ multiplications and n additions. However, if we use a nested format, n multiplications and n additions are required.

```
DO j=n, 0
   p = p * x + a(j)
END DO
```

If you want to find all roots of a polynomial, you have to remove the found root before another processing. This removal process is referred to as polynomial deflation.

Conventional Methods

- Müller's method : projects a parabola through three points.
- Bairstow's method:
    1. guess a value for the root $x = t$
    2. divide the polynomial by the factor $x - t$
    3. determine whether there is a reminder. If not, the guess is was perfect and the root is equal to. If there is a reminder, the guess can be systematically adjusted and the procedure repeated until the reminder disappears.

Root Location with Libraries and Packages

- Matlab:
  - roots
  - poly
  - polyval
  - residue
  - conv
  - deconv
- IMSL:
  - ZREAL

Engineering Applications: Roots of Equations

See the textbook

Taechul Lee

2001-11-29