Numerical Methods

Introduction to Numerical Methods

Nonlinear Equations: Roots

- Objective is to find a solution of

  F(x) = 0

  Where *F* is a polynomial or a transcendental function, given explicitly.
- Exact solutions are not possible for most equations.
- A number $x \pm e$, ($e > 0$) is an approximate solution of the equation if there is a solution in the interval [x-e,x+e]. e is the maximum possible error in the approximate solution.
- With unlimited resources, it is possible to find an approximate solution with arbitrarily small e.

Bisection Method

Let *F(x)* be a continuous function and let *a* and *b* be real numbers such that *f(a)* and *f(b)* have opposite signs. Then there is a x* in interval [a,b] such that *F(x*) = 0*.

Then c = (a + b)/2 is an approximate solution with maximum possible error (b – a)/2.

If f(c) and f(a) have opposite signs then the solution x* is in the interval [a,c]. Then, again, d = (c + a)/2 is an approximate solution but with max possible error (b – a)/4.

Else the solution is in the interval [c,b]. The approximate solution now is (c+b)/2 with max possible error (b-a)/4.

Continuing this process n times we can reduce the max possible error to $(b-a)/2^n$.

Advantages

- Always convergent
- The root bracket gets halved with each iteration - guaranteed.

Drawbacks

- Slow convergence
- If one of the initial guesses is close to the root, the convergence is slower

Drawbacks (continued)

- If a function f(x) is such that it just touches the x-axis it will be unable to find the lower and upper guesses.
- Function changes sign but root does not exist

Regula Falsi Method

Algorithm

1. Let a and b be such that f(a)*f(b) < 0
2. Let c = (f(b)*a – f(a) * b) / (f(b) – f(a))
3. If f(a) * f(c) < 0 then b = c
   else a = c
4. If more accuracy is required go to step 2
5. Print the approximate solution (a + b)/2

Advantages

- It always converges.
- It does not require the derivative.
- It is a quick method.

Disadvantages

- One of the interval definitions can get stuck.
- It may slow down in unfavourable situations.
- Like Bisection, need an initial interval around the root.

x_i

Newton Raphson Method

- The Newton–Raphson formula consists geometrically of extending the tangent line at a current point until it crosses zero, then selecting the next guess to the abscissa of that zero crossing.
- This technique derives from the Taylor series expan sion of a function near a point.

Algorithm

1. Input xOld
2. xNew = xOld – f(xOld) / f’(xOld)
3. If not satisfied go to step 2
4. Print xNew

Advantages

- One of the fastest convergence method.
- Convergence on the root quadraticly.
- Near a root, the number of significant digits approximately doubles with each step.
- This leads to the ability of the Newton-Raphson Method to “polish” a root from another convergence technique.

Advantages (Cont..)

- Easy to convert to multiple dimensions.
- Can be used to improved a root found by other methods.

Disadvantages

- Must find the derivative.
- Poor global convergence properties.
- Dependent on initial approximation.
- May be too far from local root.
- Fails when there is zero derivative.
- May be there is indefinite loop.

Efficiency of N-R Method

- For efficiency the user provides the routine that evaluates both f(x) and its first derivative at the point x,
- The Newton-Raphson formula can also be applied using a numerical difference to approximate the true local derivative but this is not recommended,

Efficiency (Cont..)

- You are doing two function evaluation per step, so at best the super-linear order of convergence will be only square root of 2.
- If you take dx too small, you will be wiped out by round off, while if you take it too large, your order of convergence will be linear, no better than using the initial evaluation of derivative at indicated point for all subsequent steps.

Fixed point iteration method

- **<u>Fixed point</u> :** A point, say, **s** is called a fixed point if it satisfies the equation $\mathbf{x = g(x)}$.
- **<u>Fixed point Iteration</u> :** The transcendental equation $\mathbf{f(x) = 0}$ can be converted algebraically into the form $\mathbf{x = g(x)}$ and then using the iterative scheme with the recursive relation $\mathbf{x_{i+1} = g(x_i)}$, $\mathbf{i = 0, 1, 2, \ldots,}$
- with some initial guess $\mathbf{x_0}$ is called the fixed point iterative scheme.
-

Alogrithm

- **Algorithm - Fixed Point Iteration Scheme**Given an equation $f(x) = 0$
  Convert $f(x) = 0$ into the form $x = g(x)$
  Let the initial guess be x_0
  Do

  $$\mathbf{x_{i+1} = g(x_i)}$$

Limitation

Algorithm is valid while none of following is met:

- Fixing apriori the total number of iterations **N** .
- By testing the condition **| x_{i+1} - g(x_i) |** (where **i** is the iteration number) less than some tolerance limit, say epsilon, fixed apriori.

Algorithm (Example)

Convert the given equation in the form $x = g(x)$

Examples

$x^2 – 1 = 0$ can be written as

$x = 1 / x$ or

$x = x^2 + x –1$

$x^2 + x – 2 = 0$ can be written as

$x = 2 – x^2$

$x = sqrt(2 – x)$

Convergence

- **Condition for Convergence :** If **g(x)** and **g'(x)** are continuous on an interval **J** about their root **s** of the equation $\mathbf{x = g(x)}$, and if $\mathbf{|g'(x)|<1}$ for all **x** in the interval **J** then the fixed point iterative process $\mathbf{x_{i+1}=g(x_i), i = 0, 1, 2, \ldots}$, will converge to the root $\mathbf{x = s}$ for any initial approximation $\mathbf{x_0}$ belongs to the interval **J** .