Lecture – 9

Solution of Nonlinear Equations

In this chapter we will discuss the problem of identifying the roots of the equations and system of equations. The basic formulation of the problem in the simplest case is this:

Given a function ***f(x),*** we seek value ***x*** for which ***f(x) = 0***. Solution ***x*** is called a root of equation, or zero of function ***f***. Thus, the problem is known as root or zero finding.

We will be dealing with following two important cases:

1. Single nonlinear equation in one unknown, $\boldsymbol{f} : \Re \to \Re$, which has scalar x as solution such that ***f(x) = 0***.
2. System of n coupled nonlinear equations in n unknowns, $\boldsymbol{f} : \Re^{\boldsymbol{n}} \to \Re^{\boldsymbol{n}}$, which has vector x as solution for which all components of f are zero simultaneously.

Examples: (a) Nonlinear equation in one dimension:

$$\boldsymbol{x}^2 - 4\sin(\boldsymbol{x}) = 0,$$

for which x = 1.9 is one approximate solution.

(b) System of nonlinear equations in two dimensions:

$$\boldsymbol{x}_1^2 - \boldsymbol{x}_2 + 0.25 = 0$$
$$-\boldsymbol{x}_1^2 + \boldsymbol{x}_2^2 + 0.25 = 0$$

for which $\boldsymbol{x} = [0.5 \quad 0.5]^{\boldsymbol{T}}$ is solution vector.

Existence and Uniqueness of Solutions

It is often difficult to determine the existence or number solutions to nonlinear equations. Whereas for system of linear equations the number of solutions must be either zero, one or infinitely many, nonlinear equations can have any number of solutions. Thus, determining existence and uniqueness of solutions is more complicated for nonlinear equations than for linear equations.

Although it is difficult to make any assertion about solution of nonlinear equations, there are nevertheless some useful local criteria that guarantee existence of a solution. The simplest for these is for one-dimensional problems, for which the Intermediate Value theorem provides a sufficient condition for a solution, which says that if f is continuous on a closed interval [a,b], and c lies between f(a) and f(b), then there is a value $\boldsymbol{x}^* \in [\boldsymbol{a},\boldsymbol{b}]$ such that f(x*) = c. Thus, if f(a) and f(b) differ in sign, then by taking c = 0 in the theorem we cann conclude that there must be a root with in the interval [a,b]. Such an interval [a,b] for which the sign of f differs at its endpoints is called a bracket for a solution of the one-dimensional nonlinear equation f(x) = 0.

Note: There is no simple analog for n dimensions.

The nonlinear equations can have any number of solutions. It can have a simple as well as multiple roots.

Multiple Root

Nonlinear equation may have multiple root, where both function and derivatives are zero, i.e. f(x) = 0 and f"(x) = 0. Geometrically this property means that the curve defined by ***f*** has a horizontal tangent on the x-axis. More generally for a smooth function ***f*** if $\boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{f}'(\boldsymbol{x}^*) = \boldsymbol{f}''(\boldsymbol{x}^*) = \ldots = \boldsymbol{f}^{(\boldsymbol{m}-1)}(\boldsymbol{x}^*) = 0$, then ***x**** is a multiple root of multiplicity ***m.*** If *m =1* then ***x**** is called a simple root.

Sensitivity and Conditioning

The sensitivity of the root-finding problem for a given function is opposite to that for evaluating the function: if the function value is insensitive to the value of the argument then root will be sensitive. This property makes sense if ***f(x) = y***, finding x given y has opposite conditioning from finding y given x.

Absolute condition number of root finding problem for root x* of $\boldsymbol{f} : \Re \rightarrow \Re$ is $1/|\boldsymbol{f}'(\boldsymbol{x}^*)|$. This implies that the root is ill conditioned if tangent line is nearly horizontal. In particular condition number for multiple root is infinite. Absolute condition number of root finding problem for root x* of $\boldsymbol{f} : \Re^{\boldsymbol{n}} \rightarrow \Re^{\boldsymbol{n}}$ is $\|\boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x})\|$, where J_f is a Jacobian matrix of ***f***. If Jacobian matrix is nearly singular, root is ill conditioned.

What do we mean by approximate solution $\hat{\boldsymbol{x}}$ to nonlinear system? $\hat{\boldsymbol{x}}$ is an approximate solution if either $\|\boldsymbol{f}(\hat{\boldsymbol{x}})\| \approx 0$ or $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\| \approx 0$. $\|\boldsymbol{f}(\hat{\boldsymbol{x}})\| \approx 0$ corresponds to "small residual." $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\| \approx 0$ measures closeness to (usually unknown) true solution x*. A solution criterion is not necessarily "small" simultaneously. Small residual implies accurate solution only if problem is well-conditioned.

Convergence Rate and Stopping Criteria

Unlike linear equations, most nonlinear equations can not be solved in finite number of steps. Iterative methods are being used to solve nonlinear equations. The total cost of solving problem depends on both the cost per iterations and the number of iterations required. There is often a trade off between these two factors. To compare the effectiveness of iterative methods, we need to characterize their convergence rate.

An iterative method is said to be of order 'r' or has the rate of convergence 'r', if 'r' is the largest positive real number for which there exists a finite constant $\boldsymbol{C} \neq 0$ such that

$|\varepsilon_{\boldsymbol{k}+1}| \leq \boldsymbol{C}|\varepsilon_{\boldsymbol{k}}|^{\boldsymbol{r}}$, where $\varepsilon_{\boldsymbol{k}} = \boldsymbol{x}_{\boldsymbol{k}} - \boldsymbol{x}^*$ is the error in the kth iteration. C is the asymptotic error constant usually depends on the derivatives of ***f(x)*** at ***x = x*. x**** is the true solution.

For methods that maintain interval known to contain solution (e.g. Bisection method), rather than specific approximate value for solution, take error to be length on interval containing solution.

Some particular cases of interest:

	Convergence Rate	**Digits gained per iteration**
R = 1	Linear (C<1)	Constant
R > 1	Superlinear	Increasing
R = 2	Quadratic	Double

One way to interpret the distinction between linear and superlinear is that, asymptotically a linear convergent sequence gains a constant number of additional correct digits per iteration, whereas a super linearly convergent sequence gains as increasing number of additional correct digits with each iterations. For example, a quadratically convergent method doubles the number of correct digits with each iteration.

A convergence theorem may tell us that an iterative scheme will converge for a given problem, and how rapidly it will do so, but that does not specifically address the issue of when to stop iterating and declare the resulting approximate solution to be good enough. Devising a suitable stopping criterion is a complex and subtle issue for number of reasons. We may know in principle that the error $\|\varepsilon_k\|$ is becoming small, but since we do not know the true solution, we have no way of knowing $\|\varepsilon_k\|$ directly. The reasonable surrogate is the relative change in successive iterates $\|x_{k+1} - x_k\|/\|x_k\|$. If this is small enough, we stop. To ensure that the problem has actually been solved, one may also want to verify that the residual $\|f(x_k)\|$ is suitably small. $\|x_{k+1} - x_k\|/\|x_k\|$ and $\|f(x_k)\|$ are not necessarily small simultaneously. They depend on the condition of the problem. In addition, all of these criteria are affected by the relative scaling of the components of both the argument x and the function f, and possibly other problem dependent features as well. For all these reasons, a full proof-stopping criterion can be difficult to achieve and complicated to state.

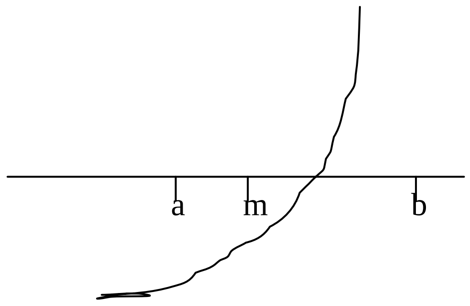
Bisection algorithm

In finite precision arithmetic, there may be no machine number ***x**** such that ***f(x*)*** is exactly zero. An alternative is to seek a very short interval [a, b] in which ***f*** has change in sign. The intermediate value theorem of calculus makes sure that bracket has a value for function goes to zero.

The bisection method makes use of Intermediate Value Theorem. It begins with an initial bracket and successively reduces its length until the solution has been isolated as accurately as desired (or the arithmetic precision will permit). At each iteration, the function is evaluated at the midpoint of the current interval, and half of the interval can be discarded, depending on the sign of the function at the mid point.

Algorithm:

```
while ((b-a) > tol) do
        m = a + (b –a) / 2
        if sign(f(a)) = sign(f(m)) then
              a = m
        else
              b = m
        end
end
```



Implementation issues:

Note a couple of points in the above algorithm that are designed to avoid the pitfalls when it is implemented in finite-precision, floating-point arithmetic. First, perhaps the most obvious formula for computing the midpoint m of the interval [a, b] is ***m = (a+b)/2***. However, with this formula the result in finite-precision arithmetic is not even guaranteed o fall within the interval [a, b] (in two-digit, decimal arithmetic, for example, this formula gives a "midpoint" 0.7 for the interval [0.67, 0.69]. Moreover, the intermediate quantity ***a+b*** could overflow in extreme case, even though the mid point is well defined and should be computable. A better alternative is the formula ***m = a + (b –a)/2***, which cannot overflow and is guaranteed to fall within the interval [a, b], provided and b have the same sign (as will normally the case unless the root happens to be near zero). Second, testing whether two function values $f(x_1)$ and $f(x_2)$ agree in sign mathematically equivalent to testing whether the product $f(x_1)\ .\ f(x_2)$ is positive or negative. In floating point arithmetic, however such implementation is risky because the product could easily underflow when function values are small, which they will be as we approach the root. A safer alternative is to use the sign function explicitly, where ***sign(x)*** = 1 if $x \geq 0$ and ***sign(x)*** = -1 if $x < 0$.

Convergence Analysis

The bisection method makes no use of the magnitude of the function values, only their signs. As a result, bisection is certain to converge but does so rather slowly. Specifically, at each successive iteration the length of the interval containing the solution, and hence a bound on the possible error, is reduced by half. This means that the bisection method is linearly convergent, with r = 1 and C = 0.5.

If the permissible error is ε, the approximate number of iterations required may be determined from the relation

$$\frac{b-a}{2^n} \leq \varepsilon \quad \text{or} \quad n \geq \frac{\log(b-a) - \log\varepsilon}{\log 2}$$

Cost of the method: It requires one function value evaluation per iteration.