Solution of Nonlinear Equations using Numerical Methods

Soupfoo

Introduction

Numerical Methods are a set of techniques used to approximate solutions to mathematical problems that are too complex to solve using analytical methods. The solutions obtained using numerical methods are often close to the true value with tolerable error, and in some cases, they may even be exactly equal to the true solution. In engineering, mathematics and economics, we often encounter nonlinear relationships that are difficult to solve (or even unsolvable) using algebraic or symbolic manipulations. These types of problems typically require iterative techniques to find approximate solutions. Some of the widely used numerical methods for solving nonlinear equations include:

1. Bisection method
2. Secant method
3. Newton-Raphson method

Bisection method

Bisection method, also known as binary chopping or half-interval method, is an iterative method used for solving nonlinear equations. It is a really simple method but it is relatively slow compared to other methods. It is often used to obtain a rough approximation which is then used as an initial guess for faster methods.

Iteration steps

Let **x1** and **x2** be initial guesses and **f(x)** be a continuous function in the interval [**x1, x2**]. The functional values **f(x1)** and **f(x2)** should have opposite signs, i.e **f(x1).f(x2)** $<$ **0**.

1. Assume a tolerable error, **E**.
2. If **f(x1)** $<$ **0**, set **a** $=$ **x1** and **b** $=$ **x2**. Otherwise set **a** $=$ **x2** and **b** $=$ **x1**.

3. Calculate the midpoint of the interval, `c = (a + b)/2`.
4. If **|b - a| < E** or **|f(c)| < E**, stop iterating.
5. If **f(c) < 0**, set **a = c** for next iteration. Otherwise, set **b = c**.
6. Go to step 3.

The final value of **c** is the required solution of the equation.

Example

`f(x) = x^3 + x - 3`

Take the interval **[1, 2]**, i.e. **a = 1** and **b = 2**. Let the tolerable error be **E = 0.0001**.

f(1) = 1^3 + 1 - 3 = -1 and **f(2) = 2^3 + 2 - 3 = 7**

The initial condition **f(a).f(b) < 0** is satisfied.

As the function is continuos in the interval **[1, 2]**, the root must be between 1 and 2.

Iteration	a	b	c	f(c)
1	1.000000	2.000000	1.500000	1.875000
2	1.000000	1.500000	1.250000	0.203125
3	1.000000	1.250000	1.125000	-0.451172
4	1.125000	1.250000	1.187500	-0.137939
5	1.187500	1.250000	1.218750	0.029022
6	1.187500	1.218750	1.203125	-0.055340
7	1.203125	1.218750	1.210938	-0.013381
8	1.210938	1.218750	1.214844	0.007765
9	1.210938	1.214844	1.212891	-0.002822
10	1.212891	1.214844	1.213867	0.002468
11	1.212891	1.213867	1.213379	-0.000177
12	1.213379	1.213867	1.213623	0.001145
13	1.213379	1.213623	1.213501	0.000484
14	1.213379	1.213501	1.213440	0.000153
15	1.213379	1.213440	1.213409	-0.000012
16	1.213409	1.213440	1.213425	0.000071
17	1.213409	1.213425	1.213417	0.000029

After 17 iterations, it is clear that the interval converges to **1.2134**. Therefore, the root of **f(x)** is **1.2134**.

Secant method

Secant method is similar to bisection method; it uses two initial guesses but does not require that they bracket the root. At each iteration, the root is estimated using the recurrence relation,

```
X_n+1  =  X_n - f(X_n) * (X_n - X_n-1) / (f(X_n) - f(X_n-1))
```

Iteration steps

Let **x1** and **x2** be initial guessses and **f(x)** be a continuous function in the interval [**x1, x2**]. The functional values **f(x1)** and **f(x2)** need not have opposite signs. However, having opposite signs ensures convergence.

1. Assume a tolerable error, **E**.
2. Let **a = x1** and **b = x2**.
3. Calculate new approximation, `c = b - f(b) * (b - a) / (f(b) - f(a))`
4. If |**c - b**| < **E** or |**f(c)**| < **E**, stop iterating.
5. Set **a = b, b = c** and go to step 3.

The final value of **c** is the required solution of the equation.

Example

```
f(x) = x*ln(x) - 2.4
```

Let the initial guesses be **a = 3, b = 4** and tolerable error be **E = 0.0001**.

Iteration	a	b	c	f(c)
1	3.000000	4.000000	2.601734	0.087721
2	4.000000	2.601734	2.561616	0.009555
3	2.601734	2.561616	2.556712	0.000043
4	2.561616	2.556712	2.556690	0.000000
5	2.556712	2.556690	2.556690	0.000000

After 5 iterations, the value converges to **2.55669**, which is the root of given function.

Newton-Raphson method

Newton-Raphson method is also an iterative numerical method used to approximate the roots of a real-valued function **f(x) = 0**. It is widely used in numerical analysis due to it's fast convergence and efficiency. The simplest version of this method requires a function **f(x)**, it's derivative **f'(x)** (where **f'(x) =/= 0**) and an initial guess **x_0**.

Iteration steps

1. Assume a tolerable error, **E**.
2. Take an initial guess, **x__0**.
3. Refine the estimate using the formula, `x_n+1 = x_n - f(x_n) / f'(x_n)`
4. If |**x__n+1 - x__n**| < **E** or |**f(x__n+1)**| < **E**, stop iterating.
5. Set **x__n = x__n+1** and go to step 3.

The final value of **x__n** is the best estimate.

Example

`f(x) = 3x^2 - e^(-x)`

The derivative of this function is `f'(x) = 6x + e^(-x)`

Take an initial guess, **x__0 = 1** and tolerable error be **E = 0.0001**

Iteration	x__n	x__n+1	f(x__n+1)
1	1.000000	0.586657	0.476315
2	0.586657	0.469802	0.037015
3	0.469802	0.459054	0.000310
4	0.459054	0.458962	-0.000000

After 4 iterations, the value of **f(x__n+1)** is less than **E**. Hence, the best estimated root is **0.458962**.

Conclusion

Numerical methods like Bisection, Secant and Newton-Raphson methods are essential for solving nonlinear equations that are often difficult to solve using traditional analytical techniques. Each method has its advantages and limitations. Bisection method is simple and guarantees convergence, though it is slower. Secant and Newton-Raphson methods are faster but they may not always converge and depend on good initial guesses. While the solutions may not be exact, they are sufficiently accurate for practical use. Numerical methods provide effective tools for solving complex problems that would otherwise be difficult or impossible to solve.