Some Concepts in Numerical Method for Solving Non-Linear Equation

[1]Shreedevi Kalyan and [2]Ramesh Kempepatil
Asst.Professsor, Sharnbasava University Kalaburagi, Karnataka.
kalyanshreedevi@gmail.com
rameshkempepatil@gmail.com

ABSTRACT

This paper will focus on the numerical methods involved in solving systems of nonlinear equations. First, we will study Powell's method for finding a local minimum of a function, where the function must be a real-valued function of a fixed number of Real-valued inputs. Second, we will examine Regular falsi method which is also called False-position method this method has been described as a generalization of the Bisection Method from which we can try for a better convergence, at the risk of a worse one, or none at all. And third, to calculate the higher order Lagrange's interpolation of the given function, we will study Neville's method we will also give working examples and application of Powell's method, Regular-Falsi method and Neville's method. Which are playing very important role in engineering and science and technology.

INTRODUCTION

Over the years, we have been taught on how to solve equations using various algebraic methods. These methods include the substitution method and the elimination method. Other algebraic methods that can be executed include the quadratic formula and factorization. In Linear Algebra, we learned that solving systems of linear and non-linear equations can be implemented by using row reduction as an algorithm. However, when these methods are not successful, we use the concept of numerical methods. Numerical Methods are used to approximate solutions of equations when exact solutions cannot be determined via algebraic methods. They construct successive approximations that converge to the exact solution of an equation or system of equations.

In this paper we focused on solving nonlinear equations involving with various variables. We used methods such as Powell's method, the Regular-Falsi method, and the Neville's method. Problems are only focused on solving nonlinear equations with different variable, rather than nonlinear equations with only one variables.

The goal of this paper is to examine three different numerical methods that are used to solve systems of nonlinear equations in several variables. These method we will look at is Powell's method. This will be followed by Regular-Falsi method, which is sometimes called a False-position method. Lastly, we will study the Neville's method that is used to calculate higher order Lagrange's interpolation polynomial of nonlinear equations. For each method, a breakdown of each numerical procedure will be provided. In addition, there will be some discussion of the convergence of the numerical methods, as well as the applications of each method.

After a discussion of each of the three methods, we will be given working examples for each method and its application.

BASIC DEFINITIONS:

1) LOCAL MINIMUM AND LOCAL MAXIMUM

A real valued function f defined on a domain x has an absolute maximum point at x^* if $f(x^*) \geq f(x)$ for all x in X. similarly the function has an absolute minimum point at x^* if $f(x^*) \leq f(x)$ for all x in X. the value of the function at maximum point is called the maximum value of the function and the value of the function at a minimum point is called the minimum value of the function.

Note: The maxima and minima of a function known collectively as extrema.

2) GOLDEN RATIO

The Golden ratio has unique mathematical properties. It is the only positive number whose square is one greater than itself. it is the only positive number whose reciprocal is one less than itself. It is also the found in limits and Fibonacci series

Its symbol is phi (ϕ) it is the special number approximately equals to 1.618

3) DIRECTION SET METHOD

Powell's method is also known as Direction set method in which by using linmin we move along the first direction to its minimum then from there along the second direction to its minimum and so on cycling through the whole set of directions as many times as necessary. This simple method is actually not too bad for many functions.

4) CONJUGATE DIRECTION

Having performed a line minimization along a direction we would like to choose a new direction V so that minimizing along V will not spoil the minimization along U. we can determine such a direction by using the Taylor approximation at a

$$f(a+x) \approx c - bx + \frac{1}{2} x^T Ax$$

The gradient of f near a given by $\nabla f(a+x) = Ax - b$

If we have just minimized along a direction u then the component of the gradient along u must be zero, thus the gradient itself is perpendicular to u

$$\nabla f . u = 0$$

As we now move along some direction V the gradient changes by

$$\delta(\nabla f) = Av$$

In order not to interfere with our u minimization we required that the gradient remain perpendicular to u, i.e. that the change in gradient itself be perpendicular to u. this is simply

$$u^T Av = 0$$

Two vectors u,v having this property are said to be conjugate. A set of vectors for which this holds for all pairs is a conjugate set.

5) BRACKETING MINIMUM

A minimum is bracketed by three points, if and is less than both and if this condition holds and the function continuous in the interval, than has a minimum for some, a very simple iterative minimum bracketing procedure follows start from any initial point.

6) BISECTION METHOD

The bisection method in mathematics is a root finding method that repeatedly bisects an interval and then select a subinterval in which a root must lie for further processing. It is very simple and robust method but it is also relatively slow.

7) SIMPLE BRACKETING METHOD

This is one of the simplest and reliable iterative method for the solution of non linear equation, this method is also known as binary chopping or half interval method.

8) UNDER INTERPOLATION

Interpolation is an estimation of a value within two known values in a sequence of values polynomial interpolation is a method of estimating value between known data points.

9) SOR METHOD:

In numerical linear algebra the method of successive over relaxation is a variant of the Gauss-seidel method for solving a linear system of equations. Resulting in faster convergence.

10) DIVIDED DIFFERENCE

Divided difference is an algorithm historically used for computing tables of logarithmic and trigonometric functions. Divided difference is a method can be used to calculate the coefficient in the interpolation polynomial in the Newton form.

11) LAGRANGE POLYNOMIAL

Lagrange polynomial are used for polynomial interpolation for a given set of points with no two value3s equal to the Lagrange polynomial is a polynomial of lowest degree that assumes each values and the corresponding values.

12) ZERO-ORDER APPROXIMATION

Zero-order approximation process a chemical reaction in which the rate of change of concentration is independent of the concentration. Zero-order approximation is an approximation of function by a constant.

13) LINEAR APPROXIMATION

Linear approximation is the process of finding the equation of a line that is the closest estimate of a function for a given value of x. linear approximation is also known as tangent linear approximation and it is used to simplify the formulas associated with trigonometric functions.

14) QUADRATIC APPROXIMATION

Quadratic approximation also uses the point to approximate nearby values but uses a parabola instead of just a tangent line to do so.

POWELL'S METHOD

INTRODUCTION

Powell's method strictly Powell's conjugate direction method, is an algorithm proposed by Michael J. D. Powell's for finding a local minimum of a function. The function need not be differentiable, and no derivatives are taken.

The function must be real valued function of a fixed number of real valued inputs. The caller passes in the initial point. The caller also passes in a set of initial search vectors. Typically N search vectors (say $\{s_1 \ldots s_n\}$) are passed in which are simply the normal aligned to each axis.

The method minimises the function by a bi-directional search along each search vector, in turn. The bi-directional line search along each search vector can be done by golden section search or Brent's method.

The method is useful for calculating the local minimum of a continuous but complex function, especially one without an underlying mathematical definition, because it is not necessary to take derivatives. The basic algorithm is simple; the complexity is in the linear searches along the search vectors, which can be achieved via Brent's method.

Assume that we know how to minimize a function of one variable. If we start at a point p in N-dimensional space, and proceed from there in some Vector direction, then any function of N variables $f(p)$ can be minimized along the line n by our one-dimensional methods. Different methods will differ only by how they The next direction n to try The line minimization routine linmin is a black-box sub algorithm, whose definition is

Linmin: Given as input the vectors p and n and the function find the scalar

That minimizes

$f(p+\lambda n)$. Replace p by $p+\lambda n$ Replace n by λn. Done.

DIRECTION SET METHODS

- Obviously what we need is a better set of directions than the e_i's. All direction set methods consist of prescriptions for updating the set of directions as the method proceeds, attempting to come up with a set which either.
- Includes some very good directions that will take us far along narrow valleys, or else (more subtly).
- Includes some number of "non-interfering" directions with the special property that minimization along one is not "spoiled" by subsequent minimization along another, so that interminable cycling through the set of directions can be avoided.

CONJUGATE DIRECTION

- This concept of "non-interfering" directions, called conjugate directions, can be made mathematically explicit.
- Assume f is differentiable. If we minimize f along the direction u, then the gradient must b perpendicular to u at the line minimum
- Take some particular point p as the origin of coordinate system then any function f can be approximated by its Taylor series

$$f(x) = f(p) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \ldots$$

$$\approx c - b^t x + \frac{1}{2} x^t A x$$

Where $c = f(p), b = -\nabla f\big/_p$, A=Hessian matrix.

In the approximation, the gradient is easily calculated as $\nabla f = Ax - b$.

- This implies that the gradient will vanish - the function will be at an extremum at a value of x obtained by solving $A x - b$.
- How does the gradient f change as we move along some direction v ? $\delta(\nabla f) = A v$
- Suppose that we have moved along some direction u to a minimum and now propose to move along some new direction v . The condition that motion along v not spoil our minimization along u is just that the
- gradient stay perpendicular to u , i.e., that the change in the gradient be perpendicular to u : $0 = u^t \delta(\nabla f) = u^t A v$
- If this holds for two vectors u and v, they are said to be conjugate.

The procedure of searching for an optimum holding all Independent variables except one constant is not necessarily efficient (Left panel).

It would be much more e efficient to search along a set of directions that is conjugate to the objective function, as represented by z_1, z_2 In the figure on the right.

CONJUGATE SETS

- When the relation holds pair wise for all members of a set of vectors, they are said to be a conjugate set.
- If you do successive line minimization of a function along a conjugate set of directions, then you don't need to redo any of those directions.
- The idea behind a direction set method is to come up with a set of N linearly independent, mutually conjugate directions.
- Then, one pass of N line minimizations will put it exactly at the minimum of a quadratic form. For functions f that are not exactly quadratic forms, it won't be exactly at the minimum; but repeated cycles of N line minimizations will in due course converge quadratic ally to the minimum.

DESCRIPTION OF POWELL'S METHOD

Powell first discovered a direction set method that does produce N mutually conjugate directions (doesn't require us to Know any derivatives!).

Initialize the set of directions u_i to the basis vectors; $u_i = e_i, i = 1,....N$ now repeat the following sequence of steps until Stops decreasing:

- Save your starting position as p_0.
- For $i = 1,....N$, move p_{i-1} to the minimum along direction u_i and call this point p_i
- For $i = 1,....N-1$,set $u_i \leftarrow u_{i+1}$.
- Set $u_n \leftarrow p_N - p_0$.
- Move p_N to the minimum long direction u_n and call this point p_0.

POWELL'S METHOD

POWELL'S METHOD (HISTORY)

Powell (1964) showed that, for a quadratic form, k iterations of the above basic procedure produces a set of directions u_i whose last k members are mutually conjugate? Therefore, N iterations of the basic procedure, amounting to $N(N+1)$ line minimizations In all will exactly minimize a quadratic form [Brent, 1973] gives proofs of these statements in accessible form. Unfortunately, there is a problem with Powell's algorithm.

The procedure of throwing away, at each stage, u_n in favor of $p_N - p_0$ tends to produce sets of directions that “folds up on each other" and become linearly dependent. Once this happens, then the procedure finds the minimum of the function f only over a subspace of the full N -dimensional case; in other words, it gives the wrong answer.

POWELL'S METHOD IN PRACTICE

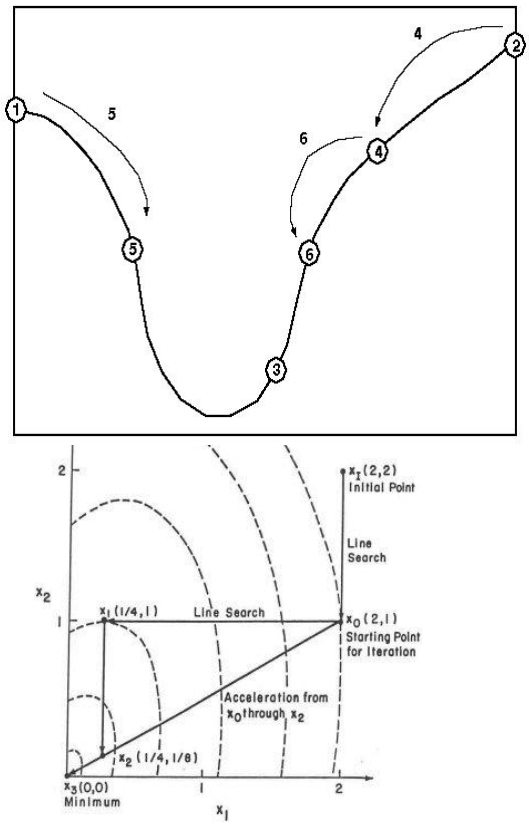
There are a number of ways to x up the problem of linear dependence in Powell's algorithm, among them:

We can reinitialize the set of directions u_i to the basis vector after every N or $N+1$ iterations of the basic procedure. This produces a serviceable method, which we commend if quadratic convergence is important for your application (i.e., if your functions are close to quadratic forms and if you Desire high accuracy).

We can give up the property of quadratic convergence in favor of a more heuristic scheme (due to Powell) which tries to find a few good directions along narrow valleys instead of N necessarily conjugate directions.

One-dimensional Minimization: Golden Section Search

Idea: successive bracketing of a minimum of a function f.



Minimum is originally bracketed by triplet {1, 3, 2}. f Is evaluated by 4,

which replaces 2; than at

5 $\rightarrow$ replaces 1; then at 6 $\rightarrow$ replaces 4

Rule: keep a centre point that is lower than the two outside points.

BRACKETING A MINIMUM

A minimum is bracketed when there is a triplet of points $a<b<c$ Such that $f(c)>f(b)>f(a)$. In the

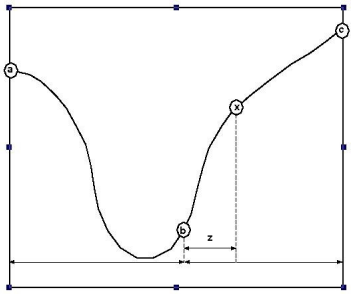
Figure above: {a, b, c} = {1, 3, 2}.

SUCCESSIVELY IMPROVE THIS BRACKETING

- Evaluate f at new x, between 1 and 3, or between 3 and 2.
- Choosing a point 4 between 3 and 2, we evaluate $f(4)$.
- $f(3)<f(4)$, so the new bracketing triplet of points is {1,2,3}
- Choose another point at 5, and evaluate $f(5)$.
- $f(3)<f(5)$, so the new bracketing triplet is {5, 3, 4}.

Middle point of new triplet is the best minimum achieved so far.

Important question: “How do we choose the new point x each time?"



Let b be some fraction w of the way between a and c

$$w = \frac{b-a}{c-a} \quad \text{And } 1-w = \frac{c-b}{c-a}.$$

Bracketing a Minimum

- If x is some additional fraction z beyond b, then $z = \frac{x-b}{c-a}$.
- The next bracketing segment will either be of length $w+z$ relative to the current one, or of length $1-w$.
- To minimize the worst case, choose z to make these equal $z = 1-2w$.
- This gives us the symmetric point to b in the interval (a, c) i.e. $|b-a| = |x-c|$ This implies that x lies in the larger of the two Segments ($z > 0$ only if w < 1/2).
- But where in the larger segment? Where did the value of w itself come from?
- ...presumably from the previous stage of applying the same strategy: if z is optimal, then so was w.
- Scale similarity: z should be the same fraction of the way from b to c (if that is the bigger segment) as was b from a to c

i.e. $w = \frac{z}{1-w}$.

- $w^2 - 3w + 1 = 0 \rightarrow w \approx 0.38197$, and $1-w \approx 0.61803$.
- Optimal interval has its middle point a fractional distance 0.38197 from one end, and 0.61803 from the other → golden section fractions.

POWELL'S METHOD:

Let X_0 be an initial guess at the location of the minimum of the function

$z = f(x_1, x_2 \ldots x_N)$. Assume that the partial derivatives of the function are not available. An intuitively appealing approach to approximating a minimum of the function *f* is to generate the next approximation X_1 by proceeding successively to a minimum of *f* along each of the *N*

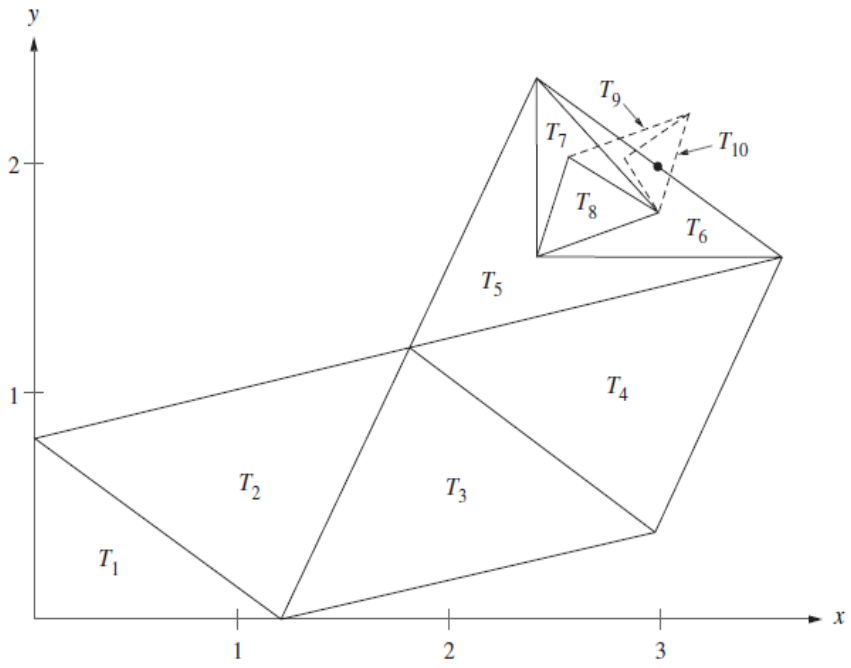


Fig.1 The sequence of triangles $\{T_k\}$ converging to the point (3, 2)

Points $X_0 = P_0$, P_1, $P_2 \ldots P_N = X_1$. Along each standard base vector the function f is a function of one variable. Thus the minimization of *f* requires the application of either the golden ratio or Fibonacci searches on an interval over which the function is unimodal. The iteration is then repeated to generate a sequence of points $\{X_k\}_{k=0}^{\infty}$. Unfortunately, the method is, in general, inefficient due to the geometry of multivariable functions. But the step from the point X_0 to the point X_1 is the first step of Powell's method.

The essence of Powell's method is to add two steps to the process described in the preceding paragraph. The vector $P_N - P_0$ represents, in some sense, the *average direction* moved during each iteration. Thus the point X_1 is determined to be the point at which the minimum of the function *f* occurs along the vector $P_N - P_0$. As before, *f* is a function of one variable along this vector and the minimization requires an application of the golden ratio or Fibonacci searches. Finally, since the vector $P_N - P_0$ was such a *good* direction, it replaces one of the direction vectors for the next iteration. The iteration is then repeated using the new set of direction vectors to generate a sequence of points $\{X_k\}_{k=0}^{\infty}$. The process is outlined below.

Let X_0 be an initial guess at the location of the minimum of the function

$z = f(x_1, x_2, \ldots, x_N)$, $\{E_k = [00\ldots01_k0\ldots0] : k = 1, 2, \ldots, N\}$ be the set of standard base vectors.

$$U = [U_1' U_2' \ldots U_N'] = [E_1' E_2' \ldots E_N']. \qquad (1)$$

and i=0.

(i) Set $P_0 = X_i$.

(ii) For k = 1, 2, . . . , N find the value of γ_k that minimizes f $(P_{k-1} + \gamma_k U_k)$ and set

$P_k = P_{k-1} + \gamma_k U_k$.

(iii) Set i = i + 1.

(iv) Set $U_j = U_{j+1}$ for j = 1, 2. . . N − 1. Set $U_N = P_N - P_0$.

(v) Find the value of γ that minimizes f ($P_0 + \gamma U_N$). Set $X_i = P_0 + \gamma U_N$

(vi) Repeat steps (i) through (v).

EXAMPLE

Use the process described in the preceding paragraph to find X_1 and X_2 for the function $f(x, y) = \cos(x) + \sin(y)$. Use the initial point $X_0 = (5.5, 2)$.

SOLUTION

Let U=$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $P_0 = X_0 = (5.5, 2)$. when $i = 1$

The function,

$$f(P_0 + \gamma_1 U_1) = f\left((5.5, 2) + \gamma_1 (1, 0)\right)$$

$$= f(5.5 + \gamma_1, 2)$$

$$= \cos(5.5 + \gamma_1) + \sin(2)$$

Has a minimum at $\gamma_1 = -2.3584042$. Thus $P_1 = (3.1415958, 2)$. When $i = 2$

The function

f ($P_1 + \gamma_2 U_2$) = f ((3.1415958, 2) + γ_2(0, 1))

= f (3.1415982, 2 + γ_2)

= cos (3.1415982) + sin (2 + γ_2)

Has a minimum at $\gamma_2 = 2.7123803$. Thus $P_2 = (3.1415958, 4.7123803)$.

Set $U_2' = (P_2 - P_0)'$

U = $\begin{bmatrix} 0 & -2.3584042 \\ 1 & 2.7123803 \end{bmatrix}$

The function

f ($P_0 + \gamma U_2$) = f ((5.5, 2) + γ (−2.3584042, 2.7123803))

= f (5.5 − 2.3584042γ , 2 + 2.7123903γ)
= cos (5.5 − 2.3584042γ) + sin (2 + 2.7123803γ)

Has a minimum at γ = 0.9816697. Thus X_1 = (3.1848261, 4.6626615).

Set $P_0 = X_1$. When i = 1 the function

f ($P_0 + \gamma_1 U_1$) = f ((3.1848261, 4.6626615) + γ_1(0, 1))

= f (3.1848261, 4.6626615 + γ_1)
= cos(3.1848261) + sin(4.6626615 + γ_1)

Has a minimum at $\gamma_1 = 0.0497117$. Thus $P_1 = (3.1848261, 4.7123732)$.

When i = 2 the function

$f(P_1 + \gamma_2 U_2) = f((3.1848261, 4.7123732) + \gamma_2(-2.3584042, 2.7123809))$

$= f(3.1848261 - 2.3584042\gamma_2, 4.7123732 + 2.7123809\gamma_2)$
$= \cos(3.1848261 - 2.3584042\gamma_2) + \sin(4.7123732 + 2.7123809\gamma_2)$

Has a minimum at $\gamma_2 = 0.0078820$. Thus $P_2 = (3.1662373, 4.7337521)$.

Set $U_2' = (P_2 - P_0)'$ and

$$U = \begin{bmatrix} -2.3584042 & -0.0185889 \\ 2.7123803 & 0.0710906 \end{bmatrix}$$

The function

$f(P_0 + \gamma U_2) = f((3.1848261, 4.6626615) + \gamma(-0.0185889, 0.0710906))$

$= f(3.1848261 - 0.0185889\gamma, 4.6626615 + 0.0710906\gamma)$
$= \cos(3.1848261 - 0.0185889\gamma) + \sin(4.6626615 + 0.0710906\gamma)$

Has a minimum at $\gamma = 0.8035684$. Thus X2 = (3.1698887, 4.7197876).

The function f (x, y) = cos(x) + sin(y) has a relative minimum at the point

P = (π, 3π/2). The graph of f is shown in Figure2. Figure3. Shows a contour plot of the function f and the relative positions of the points X0, X1, and X2.

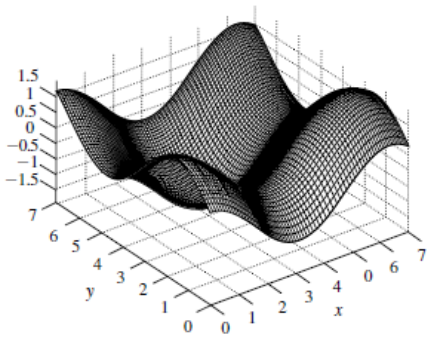


Figure 2. The graph of f(x,y)=cos(x)+sin(y).

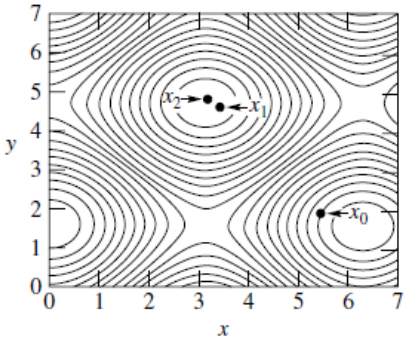


Figure 3. The contour graph of f(x,y)=cos(x)+sin(y).

In step (IV) of the previous process the first vector U_1 was discarded and the average direction vector $P_N - P_0$ was added to the list of direction vectors. In fact, it would be better to discard the vector U_r along which the greatest decrease in *f* occurred. It seems reasonable that the vector U_r is a large component of the average direction vector

$U_N = P_N - P_0$. Thus, as the number of iterations increase, the set of direction vectors will tend to become linearly dependent. When the set becomes linearly dependent one or more of the directions will be lost and it is likely that the set of points $\{X\}_{k=0}^{\infty}$ will not converge to the point at which the local minimum occurs. Furthermore, in step (iv) it was assumed that the average direction vector represented a good direction in which to continue the search. But that may not be the case.

RESULT OF POWELL'S METHOD

(i) Set $P_0 = X_i$.
(ii) For $k = 1, 2, \ldots, N$ find the value of γ_k that minimizes $f(P_{k-1} + \gamma_k U_k)$ and set $P_k = P_{k-1} + \gamma_k U_k$.

(iii) Set r and U_r equal to the maximum decrease in f and the direction of the maxi-mum decrease, respectively, over all the direction vectors in step (ii).
(iv) Set $i = i + 1$.
(v) If $f(2P_N - P_0) \geq f(P_0)$ or $2(f(P_0) - 2f(P_N) + f(2P_N - P_0))(f(P_0) - f(P_N) - r)^2 \geq r(f(P_0) - f(2P_N - P_0))^2$, then set $X_i = P_N$ and return to step (i). Otherwise go to step (Vi).

(Vi) Set $U_r = P_N - P_0$.

(Vii) Find the value of γ that minimizes $f(P_0 + \gamma U_r)$. Set $X_i = P_0 + \gamma U_r$.

(Viii) Repeat steps (i) through (vii).

If the conditions in step (v) are satisfied, then the set of direction vectors is left unchanged. The first inequality in step (v) indicates that there is no further decrease in the value of f in the average direction $P_N - P_0$. The second inequality indicates that the decrease in the function f in the direction of greatest decrease U_r was not a major part of the total decrease in f in step (ii). If the conditions in step (v) are not satisfied, then the direction of greatest decrease U_r is replaced with the average direction from step (ii); $P_N - P_0$. In step (vii) the function is minimized in this direction. Stopping criteria based on the magnitudes $X_i - X_{i-1}$ or $f(X_i)$ are typically found in steps (v) and (vii).

APPLICATIONS

1) Powell's method can be applied to the optimization of a flow injection system configuration.
2) The performance of this method has been compared with the modified simplex method the system studied is the determination of ammonia, based on indophenols blue reaction.
3) Powell's method has been used for the evaluation of the surge arrester models parameters. the proper modelling of metal oxides surge arresters and the right selection of equivalent circuit parameters are very significant issues, since quality and reliability of lightning performance studies can be improved with the more efficient representation of the arresters dynamic behaviour.

CONCLUSION

Hence the Powell's method an algorithm proposed by Michael J. D. Powell for finding a local minimum of a function. The function need not be Differentiable and no derivatives are taken by using this method we can calculate the local minimum of a continuous but complex function, especially one without an underlying mathematical definition, because it is not necessary to take derivative.

REGULAR-FALSI METHOD

(False-Position Method of Solving a Nonlinear Equation)

INTRODUCTION

The bisection method was described as one of the simple bracketing methods of solving a nonlinear equation of the general form

$$f(x) = 0 \qquad (1)$$

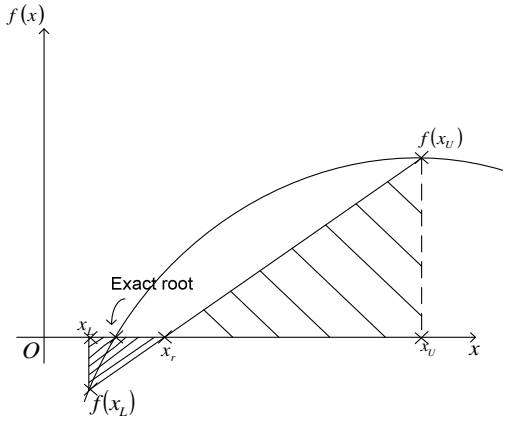


Figure 1: False-Position Method the above nonlinear equation can be stated as finding the value of *x* such that Equation (1) is satisfied.

In the bisection method, we identify proper values of x_L (lower bound value) and x_U (upper bound value) for the current bracket, such that

$$f(x_L)\,f(x_U) < 0. \tag{2}$$

The next predicted/improved root x_r can be computed as the midpoint between x_L and x_U as

$$x_r = \frac{x_L + x_U}{2} \tag{3}$$

The new upper and lower bounds are then established, and the procedure is repeated until the convergence is achieved (such that the new lower and upper bounds are sufficiently close to each other).

However, in the example shown in Figure 1, the bisection method may not be efficient because it does not take into consideration that $f(x_L)$ is much closer to the zero of the function $f(x)$ as compared to $f(x_U)$. In other words, the next predicted root x_r would be closer to x_L (in the example as shown in Figure 1), than the mid-point between x_L and x_U. The false-position method takes advantage of this observation mathematically by drawing a secant from the function value at x_L to the function value at x_U, and estimates the root as where it crosses the *x*-axis.

FALSE-POSITION METHOD

The Regular-Falsi method is a numerical method for estimating the roots of a polynomial f(x). A value x replaces the midpoint in the bisection method and serves as the new approximation of a root of f(x). The objective is to make convergence faster.

Based on two similar triangles, shown in Figure 1, one gets

$$\frac{0 - f(x_L)}{x_r - x_L} = \frac{0 - f(x_U)}{x_r - x_U} \tag{4}$$

From Equation (4), one obtains

$$(x_r - x_L) f(x_U) = (x_r - x_U) f(x_L)$$

$$x_U f(x_L) - x_L f(x_U) = x_r \{ f(x_L) - f(x_U) \}$$

The above equation can be solved to obtain the next predicted root x_m as

$$x_r = \frac{x_U f(x_L) - x_L f(x_U)}{f(x_L) - f(x_U)} \quad (5)$$

The above equation, through simple algebraic manipulations, can also be expressed as

$$x_r = x_U - \frac{f(x_U)}{\left\{\frac{f(x_L) - f(x_U)}{x_L - x_U}\right\}} \quad (6)$$

Or,

$$x_r = x_L - \frac{f(x_L)}{\left\{\frac{f(x_U) - f(x_L)}{x_U - x_L}\right\}} \quad (7)$$

Observe the resemblance of Equations (6) and (7) to the secant method.

FALSE-POSITION ALGORITHM

The steps to apply the false-position method to find the root of the equation $f(x) = 0$ are as follows.

1. Choose x_L and x_U as two guesses for the root such that $f(x_L) f(x_U) < 0$, or in other words, $f(x)$ changes sign between x_L and x_U.

2. Estimate the root, x_r of the equation $f(x) = 0$ as

$$x_r = \frac{x_U f(x_L) - x_L f(x_U)}{f(x_L) - f(x_U)}$$

3. Now check the following

If $f(x_L) f(x_r) < 0$, then the root lies between x_L and x_r; then $x_L = x_L$ and $x_U = x_r$.

If $f(x_L) f(x_r) > 0$, then the root lies between x_r and x_U; then $x_L = x_r$ and $x_U = x_U$.

If $f(x_L) f(x_r) = 0$, then the root is x_r. Stop the algorithm.

4. Find the new estimate of the root

$$x_r = \frac{x_U f(x_L) - x_L f(x_U)}{f(x_L) - f(x_U)}$$

Find the absolute relative approximate error as

$$|\in_a| = \left| \frac{x_r^{new} - x_r^{old}}{x_r^{new}} \right| \times 100$$

Where,

x_r^{new} = estimated root from present iteration

x_r^{old} = estimated root from previous iteration

5. Compare the absolute relative approximate error $|\in_a|$ with the pre-specified relative error tolerance $\in_s$. If $|\in_a| > \in_s$, then go to step 3, else stop the algorithm. Note one should also check whether the number of iterations is more than the maximum number of iterations allowed. If so, one needs to terminate the algorithm and notify the user about it.

Note that the false-position and bisection algorithms are quite similar. The only difference is the formula used to calculate the new estimate of the root x_r.

EXAMPLE-1

You are working for "DOWN THE TOILET COMPANY" that makes floats for ABC commodes. The floating ball has a specific gravity of 0.6 and has a radius of 5.5cm. You are asked to find the depth to which the ball is submerged when floating in water. The equation that gives the depth x to which the ball is submerged under water is given by

$$x^3 - 0.165x^2 + 3.993\times10^{-4} = 0$$

Use the false-position method of finding roots of equations to find the depth x to which the ball is submerged under water. Conduct three iterations to estimate the root of the above equation. Find the absolute relative approximate error at the end of each iteration, and the number of significant digits at least correct at the end of third iteration.

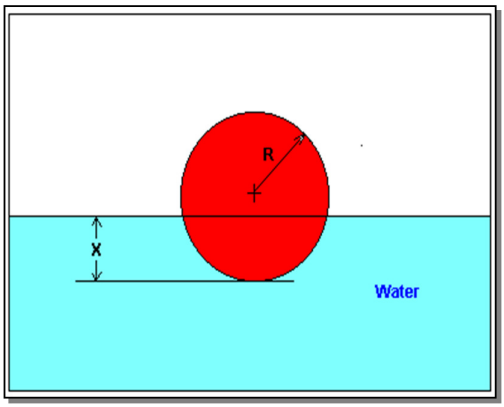


Fig.2 Floating ball problem.

SOLUTION

From the physics of the problem, the ball would be submerged between $x = 0$ and $x = 2R$,

Where,

R = radius of the ball,

That is

$$0 \le x \le 2R$$

$$0 \le x \le 2(0.055)$$

$$0 \le x \le 0.11$$

Let us assume

$$x_L = 0,\ x_U = 0.11$$

Check if the function changes sign between x_L and x_U

$$f(x_L) = f(0) = (0)^3 - 0.165(0)^2 + 3.993\times10^{-4} = 3.993\times10^{-4}$$
$$f(x_U) = f(0.11) = (0.11)^3 - 0.165(0.11)^2 + 3.993\times10^{-4} = -2.662\times10^{-4}$$

Hence

$$f(x_L)f(x_U) = f(0)f(0.11) = \left(3.993\times10^{-4}\right)\left(-2.662\times10^{-4}\right) < 0$$

Therefore, there is at least one root between x_L and x_U, that is between 0 and 0.11.

Iteration 1

The estimate of the root is

$$x_r = \frac{x_U f(x_L) - x_L f(x_U)}{f(x_L) - f(x_U)}$$
$$= \frac{0.11\times3.993\times10^{-4} - 0\times\left(-2.662\times10^{-4}\right)}{3.993\times10^{-4} - \left(-2.662\times10^{-4}\right)}$$
$$= 0.0660$$

$$f(x_r) = f(0.0660)$$
$$= (0.0660)^3 - 0.165(0.0660)^2 + \left(3.993\times10^{-4}\right)$$
$$= -3.1944\times10^{-5}$$

$$f(x_L)f(x_r) = f(0)f(0.0660) = (+)(-) < 0$$

Hence, the root is bracketed between x_L and x_r, that is, between 0 and 0.0660. So, the lower and upper limits of the new bracket are $x_L = 0, x_U = 0.0660$, respectively.

Iteration 2

The estimate of the root is

$$x_r = \frac{x_U f(x_L) - x_L f(x_U)}{f(x_L) - f(x_U)}$$
$$= \frac{0.0660\times3.993\times10^{-4} - 0\times\left(-3.1944\times10^{-5}\right)}{3.993\times10^{-4} - \left(-3.1944\times10^{-5}\right)}$$
$$= 0.0611$$

The absolute relative approximate error for this iteration is

$$\in_a = \left|\frac{0.0611 - 0.0660}{0.0611}\right|\times100 \cong 8\%$$

$$f(x_r) = f(0.0611)$$
$$= (0.0611)^3 - 0.165(0.0611)^2 + \left(3.993\times10^{-4}\right)$$
$$= 1.1320\times10^{-5}$$

$$f(x_L)f(x_r) = f(0)f(0.0611) = (+)(+) > 0$$

Hence, the lower and upper limits of the new bracket are $x_L = 0.0611,\ x_U = 0.0660$, respectively.

Iteration 3

The estimate of the root is

$$x_r = \frac{x_U f(x_L) - x_L f(x_U)}{f(x_L) - f(x_U)}$$
$$= \frac{0.0660 \times 1.132 \times 10^{-5} - 0.0611 \times \left(-3.1944 \times 10^{-5}\right)}{1.132 \times 10^{-5} - \left(-3.1944 \times 10^{-5}\right)}$$
$$= 0.0624$$

The absolute relative approximate error for this iteration is

$$\in_a = \left|\frac{0.0624 - 0.0611}{0.0624}\right| \times 100 \cong 2.05\%$$

$$f(x_r) = -1.1313 \times 10^{-7}$$

$$f(x_L)f(x_r) = f(0.0611)f(0.0624) = (+)(-) < 0$$

Hence the lower and upper limits of the new brackets are

$$x_L = 0.0611,\ x_U = 0.0624$$

All iterations results are summarized in Table 1. To find how many significant digits are at least correct in the last iterative value

$$|\in_a| \le 0.5 \times 10^{2-m}$$
$$2.05 \le 0.5 \times 10^{2-m}$$
$$m \le 1.387$$

The number of significant digits at least correct in the estimated root of 0.0624 at the end of 3rd iteration is 1.

Table1**:** Root of $f(x) = x^3 - 0.165x^2 + 3.993 \times 10^{-4} = 0$ for false-position method.

EXAMPLE-2

Find the root of

Iteration	x_L	x_U	x_r	$\|\in_a\|\%$	$f(x_m)$
1	0.0000	0.1100	0.0660	----	-3.1944×10^{-5}
2	0.0000	0.0660	0.0611	8.00	-1.1320×10^{-5}
3	0.0611	0.0660	0.0624	2.05	-1.1313×10^{-7}

$f(x) = (x-4)^2(x+2) = 0$, using the initial guesses of $x_L = -2.5$ and $x_U = -1.0$, and a pre-specified tolerance of $\in_s = 0.1\%$.

SOLUTION

The individual iterations are not shown for this example, but the results are summarized in Table 2. It takes five iterations to meet the pre-specified tolerance.

Table 2: Root of $f(x)=(x-4)^2(x+2)=0$ for false-position method.

Iteration	x_L	x_U	$f(x_L)$	$f(x_U)$	x_r	$\|\in_a\|\%$	$f(x_m)$
1	-2.5	-1	-21.13	25.00	-1.813	N/A	6.319
2	-2.5	-1.813	-21.13	6.319	-1.971	8.024	1.028
3	-2.5	-1.971	-21.13	1.028	-1.996	1.229	0.1542
4	-2.5	-1.996	-21.13	0.1542	-1.999	0.1828	0.02286
5	-2.5	-1.999	-21.13	0.02286	-2.000	0.02706	0.003383

To find how many significant digits are at least correct in the last iterative answer,

$$|\in_a| \le 0.5\times 10^{2-m}$$

$$0.02706 \le 0.5\times 10^{2-m}$$

$$m \le 3.2666$$

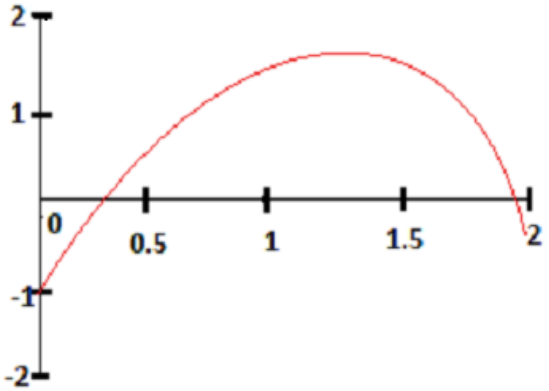
Hence, at least 3 significant digits can be trusted to be accurate at the end of the fifth iteration.

EXAMPLE-3

Find the root of $3x+\sin(x)-\exp(x)=0$.

SOLUTION

The graph of this equation is given in the figure.



Iteration No	A	B	c	F(a)*f(c)
1	0	0.5	0.376	1.38(+ve)
2	0.376	0.5	0.36	-0.102(-ve)
3	0.376	0.36	0.36	-0.085(-ve)

From this it's clear that there is a root between 0 and 0.5 and also another root between 1.5 and 2.0. now let us consider the root between f(x) in the interval [0, 0.5] where f(0)*f(0.5) is less than zero and use the regular-falsi scheme to obtain the zero of f(x)=0.

So one of the roots of $3x+\sin(x)-\exp(x)=0$ is approximately 0.36.

Note: although the length of the interval is getting smaller in each iteration, it is possible that it may not go to zero. If the graph $y=f(x)$ is concave near the root 's', one of the endpoints becomes fixed and the other end marches towards the root.

APPLICATION

1 .Regular-falsi method prediction of trace quantities of air pollutants produced by combustion reaction such as those found in industrial point sources.
2. The regular falsi method may be applied in its modified form. The regular falsi method is modified through the introduction of and under-interpolant factor which accelerates convergence of the numerical procedure in a manner similar to the maqualirian method of successive over-relaxation (SOR).

CONCLUSION

The regular falsi method is a method of solving non-linear equations by using this method one can try for a better convergence-rate, at the risk of a worse one at all. This is the most numerical equations solving method and usually it converges faster than Bisection method.

NEVILLE'S METHOD

INTRODUCTION

In mathematics **Neville's algorithm** is an algorithm used for polynomial interpolation that was derived by the mathematician **Eric Harold Neville.**

Neville's algorithm is based on the **Newton form** of the interpolating polynomial and the recursion relation for the **divided differences.**

DEFINITION:

Let f be a function whose values at the n points $x_0, x_1, ...x_n$ are known. Let $\{m_1, m_2, ...m_k\}$ be a set of k distinct integers from the set {0, 1, 2 …n}. Let $P_{m_1,m_2,...m_k,}(x)$ stand for the Lagrange polynomial that agrees with the function f at the k points $x_{m_1}, x_{m_2}, ...x_{m_k}$, i.e.,

$$P_{m_1,m_2,.....m_k}(x_{m_1})=f(x_{m_1}),\quad P_{m_1,m_2,.....m_k}(x_{m_2})=f(x_{m_2}),....,\ P_{m_1,m_2,.....m_k}(x_{m_k})=f(x_{m_k})\ .$$

Naturally, $P_{m_1,m_2,....m_k}(x)$ is the only polynomial of degree (k – 1) that passes through the k points $(x_{m_1}, f(x_{m_1})),....,(x_{m_k}, f(x_{m_k}))$.

The idea of the Neville's method is to use Lagrange polynomials of lower powers recursively in order to compute Lagrange polynomials of higher powers. This is useful, for example, if you have the Lagrange polynomial based on some set of data points $(x_i, f(x_k)),\ k=0,1,...n,$ and you get a new data point, $(x_{n+1}, f(x_{n+1}))$.

NEVILLE'S METHOD

Neville's method can be applied in the situation that we want to interpolate f(x) at a given point x = p with increasingly higher order Lagrange interpolation polynomials.

For concreteness, consider three distinct points $x_0, x_1,$ and x_2 at which we can evaluate f(x) exactly $f(x_0), f(x_1), f(x_2)$.From each of these three points we can construct an order zero (constant) "polynomial" to approximate $f(p)$ $f(p)\approx P_0(p)=f(x_0)$ (1)

$$f(p)\approx P_1(p)=f(x_1) \quad (2)$$

$$f(p)\approx P_2(p)=f(x_2) \quad (3)$$

$$f(p) \approx P_3(p) = f(x_3) \tag{4}$$

Of course this isn't a very good approximation so we turn to first order Lagrange polynomial

$$f(p) \approx P_{0,1}(p) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \tag{5}$$

$$f(p) \approx P_{1,2}(p) = \frac{x - x_2}{x_1 - x_2} f(x_1) + \frac{x - x_1}{x_2 - x_1} f(x_2) \tag{6}$$

There is also $P_{0,2,}$ but we won't concern ourselves with that one.

If we note that $f(x_i) = P_i(x),$ we find

$$P_{0,1}(p) = \frac{x - x_1}{x_0 - x_1} P_0(p) + \frac{x - x_0}{x_1 - x_0} P_1(p) \tag{7}$$

$$= \frac{(x - x_1)P_0(p) - (x - x_0)P_1(p)}{x_0 - x_1} \tag{8}$$

And similarly

$$P_{1,2}(p) = \frac{(x - x_2)P_1(p) - (x - x_1)P_2(p)}{x_1 - x_2} \tag{9}$$

In general we want to multiply $P_i(x)$ by $(x - x_j)$ where $j \neq i$ (i.e., x_j is a point that is NOT interpolated by $P_i(x)$). We take the difference of two such products and divide by the difference between the added points. The result is a polynomial $P_{i,i-1}$ of one degree higher than either of the two used to construct it and that interpolates all the points of the two constructing polynomials combined. This idea can be extended to construct the third order polynomial $P_{0,1,2}$.

$$P_{0,1,2}(p) = \frac{(p - x_2)P_{0,1}(p) - (p - x_0)P_{1,2}(p)}{x_0 - x_2} \tag{10}$$

A little algebra will convince you that

$$P_{0,1,2}(p) = \frac{(p - x_1)(p - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(p - x_0)(p - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(p - x_0)(p - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \tag{11}$$

Which is just the 3rd order Lagrange polynomial interpolating the points $x_0, x_1, x_2.$ this shouldn't surprise you since this is the unique third order polynomial interpolating these three points.

EXAMPLE-1

We are given the function

$$f(x) = \frac{1}{x} \tag{1}$$

We want to approximate the value $f(3)$.

First we evaluate the function at the three points

i	(x_i)	$f(x_i)$
0	2	0.5
1	2.5	0.4
2	4	0.25

(2)

SOLUTION

We can first make three separate zero-order approximations

$$f(3) \approx P_0(3) = f(x_0) = 0.5 \quad (3)$$

$$f(3) \approx P_1(3) = f(x_1) = 0.4 \quad (4)$$

$$f(3) \approx P_2(3) = f(x_2) = 0.25 \quad (5)$$

From these we proceed to construct $P_{0,1}$ and $P_{1,2}$ by using the Neville formula

$$f(3) \approx P_{0,1}(3) = \frac{(3-x_1)P_0(3)-(3-x_0)P_1(3)}{x_0-x_1} = \frac{(3-2.5)0.5-(3-2)0.4}{2-2.5} = 0.3 \quad (6)$$

$$f(3) \approx P_{1,2}(3) = \frac{(3-x_2)P_1(3)-(3-x_1)P_2(3)}{x_1-x_2} = \frac{(3-4)0.4-(3-2.5)0.25}{2.5-4} = 0.35 \quad (7)$$

So we can add these numbers to our table

i	x_i	P_i	$P_{i,i-1}$
0	2	0.5	
1	2.5	0.2	0.3
2	4	0.25	0.35

(8)

Finally we can compute $P_{0,1,2}$ using $P_{0,1}$ and $P_{1,2}$

i	x_i	P_i	$P_{i,i-1}$	$P_{i,i-1,i-2}$
0	2	0.5		
1	2.5	0.2	0.3	
2	4	0.25	0.35	0.325

$$f(3) \approx P_{0,1,2}(3) = \frac{(3-x_2)P_{0,1}(3)-(3-x_0)P_{1,2}(3)}{x_0-x_2}$$

$$= \frac{(3-4)(0.3)-(3-2)(0.35)}{(2-4)}$$

$$= 0.325 \tag{9}$$

If we find ourselves in the unusual situation that you know $P_{0,1,2}$ and one of $P_{0,1,}$ or $P_{1,2}$ but not the other, you can always rearrange Eq. 9 to suit your purposes.

Example-2

Suppose we have the following table of data for f(x):

X	$f(x)$
10.1	0.17537
22.2	0.37784
32.0	0.52992
41.6	0.66393
50.5	0.63608

Approximate f (27.5) using polynomial interpolation.

SOLUTION

Linear approximation:

$$P_1(27.5) = \frac{(27.5-32.0)}{(22.2-32.0)} f(22.2) + \frac{(27.5-22.2)}{(32.0-22.2)} f(32.0) \approx 0.46009$$

Quadratic approximation:

$$P_2(27.5) = \frac{(27.5-22.2)}{(10.1-22.2)} f(22.2) + \frac{(27.5-32.0)}{(10.1-32.0)} f(0.1)$$

$$\frac{(27.5-10.1)(27.5-32.0)}{(22.2-10.1)(22.2-32.0)} f(22.2) + \frac{(27.5-10.1)(27.5-22.2)}{(32.0-10.1)(32.0-22.2)} \approx 0.46141$$

There are also two potential cubic interpolating polynomials and a single quadratic polynomial.

EXAMPLE -3

Find the approximate the function f(x) at 3.The given points are

$$f(x_0) = f(1) = -6$$
$$f(x_1) = f(2) = 2$$
$$f(x_2) = f(4) = 12.$$

Using $p_{i,i}(x)$, the approximations for $P_{0,0}(3)$, $P_{1,1}(3)$ and $P_{2,2}(3)$.

SOLUTION

$$P_{0,1}(3) = \frac{(x_1 - x)P_{0,0}(x) + (x - x_0)P_{1,1}(x)}{x_1 - x_0}$$

$$= \frac{(2-3)(-6)+(3-1)(2)}{2-1}$$

$$= \frac{6+4}{1}$$

$$= 10$$

$$P_{1,2}(3) = \frac{(x_2 - x)P_{1,1}(x) + (x - x_1)P_{2,2}(x)}{x_2 - x_1}$$

$$= \frac{(4-3)(2)+(3-2)(12)}{4-2}$$

$$= \frac{2+12}{2}$$

$$= 7$$

From these two values we find $P_{0,2}(3)$ to be

$$P_{0,2}(3) = \frac{(x_2 - x)P_{0,1}(x) + (x - x_0)P_{1,2}(x)}{x_2 - x_0}$$

$$= \frac{(4-3)(10)+(3-1)(7)}{4-1}$$

$$= \frac{10+14}{3}$$

$$= \frac{24}{3}.$$

APPLICATION

- The highest degree polynomial does not necessarily deliver the smallest error.
- The Lagrange interpolating Polynomial of degree k does not help us find the one of degree k+1.
- The highest degree polynomial does not necessary deliver the smallest error.
- There is a method for using to derive higher order interpolations.

CONCLUSION

Neville's method can be applied increasingly higher order Lagrange interpolation polynomial.

REFERENCES

1. Powell M.J.D. (1964) "An efficient method for finding the minimum of a function of several variables without calculating derivatives". Computer Journal 7(2): 155-162.
2. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., (2007). "Section10.7 Direction set (Powell's) Methods in Multidimensions". pp.286-289. New York: Cambridge Universitry Press.
3. Mathews, John H. "Module for Powell search method for a minimum". Fullerton Retrieved 16 June 2017. California State University.
4. Brent, Richard P. (1973). "Section 7.3., Powell's algorithm". N.J. Prentice-Hall.
5. Jean-Luc Chabert. (Berlin Springer, 1999), "A History of Algorithms, from the Pebble to the Microchip". pp. 86-91.
6. Joseph Needham (1 January 1959) "Science and civilisation in China and the Sciences of the Heavens and the Earth". pp.147. Cambridge University Press. Volume 3.

7. Schwartz, R.K. (2004). “Calculation by Double False Position”.
8. Ford. J.A. (1995). “Improved algorithms of Illinois-type for the Numerical Solution of Nonlinear Equation”. University of Essex Press. CSM 257.
9. Dahlguist Germund, Bjorck, Ake, (2003) (1974). “Numerical Methods”. (pp.231-232)
10. Dowell, M, Jarratt, P, (1971). “A Modified Regular Falsi Method for completing the root of an equation”. BIT. 11(2), 168-174.
11. Galdino, Sergio (2011). “A family of Regular Falsi root-finding Methods”. Producing of 2011 world congress Engineering and Technology. Retrieved 9 September 2016.
12. Galdino Sergio (2011). “A family of Regular Falsi root-finding Methods”. Retrieved 11 July 2017.
13. Press. William. Saul Teukolsky. William Vetterling. Brain flannery (1992). “Polynomial Interpolation and Extrapolation”. The Art of Scientific Computing Cambridge University press.
14. J.N. Lyness and C.B. Molar. Van Der Monde (1966). “Numerical Differentiation, Numerische Mathematic”. pp. 458-464.
15. Atkinson, K.E. (1978). An Introduction to Numerical Analysis. Non-Linear System of Equations pp.88 - 95 . Canada: John Wiley and Sons.
16. Press, W.H., Flannery, B.P., Teukolsky, S.A., vetterling, W.T. (1988). Numerical Recipes in C. Powell’s method for system of non-linear equations. Pp286-289. New York: Cambridge University Press.
17. Burden, R.L., Faires, J.D (2005). Numerical Analysis. Numerical solutions of non-linear system of equations, pp.597-640. Belmount: Thomson Brooks/cole.
18. Burden, R.L., Faires, J.D (2005). Numerical Analysis. Boundary-value Problems for Ordinary differential Equations, pp.641-685. Belmount: Thomson Brook /Cole
19. Galdino, Sergio (2011). “A family of Regular Falsi root-finding Methods”. Producing of 2011 world congress Engineering and Technology. Retrieved 9 September 2015.
20. Burden, R.L., Faires, J.D (2005). Numerical Analysis. Special Type of Matrices, pp.380-413. Belmount.