NUMERICAL SOLUTION OF NONLINEAR EQUATIONS

By S. S. Krishna Murthy

(Central Instruments and Services Laboratory, Indian Institute of Science, Bangalore-12 India)

(Received : March 17, 1967)

1. Abstract

There are many methods available to obtain approximate roots of a given system of nonlinear equation. Newton's method, method of steepest descent and iterative method are some of important methods with the help of which one can improve the accuracy of approximation. A method is presented here which involves less arithmetic than Newton's and steepest descent methods on the whole. Moreover, the present method does not demand the evaluation of partial derivatives.

2. Introduction

Let $(X_1, X_2, \cdots X_n)$ be the exact root of the system of equations

$$f_i \equiv f_i(x_1, x_2, \cdots x_n) = 0 \qquad [2.1]$$

for $\qquad i = 1, 2, 3, \cdots n$

Let $(x^*_1, x^*_2, \cdots x_n^*)$ be an approximation to the actual solution $(X_1, X_2, \cdots X_n)$.

$x_1^*, x_2^*, x_3^*, \cdots x_n^*$ are now each assumed to be functional of $f_1, f_2, \cdots f_n$

i.e.,

$$x_i^* = F_i(f_1, f_2, \cdots f_n) \qquad [2.2]$$

for $\qquad i = 1, 2, 3, \cdots n$

x_i^* can be expanded with the origin at X_i by means of a Taylor's series. On retaining the first two terms of the expansion, we have

$$x_i^* = X_i + (\Delta f_1 \cdot \partial F_i/\partial f_1 + \Delta f_2 \cdot \partial F_i/\partial f_2 + \cdots + \Delta f_n \cdot \partial F_i/\partial f_n) + \cdots \qquad [2.3]$$

where $\qquad \Delta f_i = -f_i$

for $\qquad i = 1, 2, \cdots n.$

We assume a linear relationship between x_i^* and $f_1, f_2, \cdots, f_n$. The partial derivatives $\partial F_i/\partial f_j$ in [2.3,] which are to be evaluated at $f_i = 0$ become constants but unknowns as a result of the linear relationship assumed.

Let

$$a^{(1)} = (x_1^*,\ x_2^*,\ \cdots,\ x_n^*) = (x_1^{(1)},\ x_2^{(1)},\ \cdots\ x_n^{(1)})$$

$$a^{(2)} = (x_1^* + h,\ x_2^*,\ \cdots,\ x_n^*) = (x_1^{(2)},\ x_2^{(2)},\ \cdots\ x_n^{(2)})$$

$$a^{(3)} = (x_1^*,\ x_2^* + h,\ \cdots,\ x_n^*) = (x_1^{(3)},\ x_2^{(3)},\ \cdots\ x_n^{(3)})$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$a^{(i+1)} = (x_1^*,\ x_2^*,\ \cdots\ x_{i-1}^*,\ x^*_i + h,\ x_{i+1}^*\ \cdots\ x_n^*)$$
$$= (x_1^{(i+1)},\ x_2^{(i+1)},\ \cdots\ x_n^{(i+1)})$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$a^{(n+1)} = (x_1^*,\ x_2^*,\ \cdots,\ x_{n-1}^*,\ x_n^* + h)$$
$$= (x_1^{(n+1)},\ x_2^{(n+1)},\ \cdots,\ x_n^{(n+1)})$$

where h is some arbitrary small constant.

Then

$$f_i^{(j)} = f_i(a^{(j)}) = f_i(x_1^*,\ x_2^*,\ \cdots,\ x_{j-1}^* + h,\ \cdots,\ x_n^*)$$

If we substitute $a^{(1)}$, $a^{(2)}$, $\cdots$ $a^{(n+1)}$ in [2.3] and taking $i = 1$, we have the $(n+1)$ equations

$$x_1^{(1)} = X_1 + (\Delta f_1^{(1)}\cdot\partial F_1/\partial f_1 + \Delta f_2^{(1)}\cdot\partial F_1/\partial f_2 + \cdots + \Delta f_n^{(1)}\cdot\partial F_1/\partial f_n)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$x_1^{(n+1)} = X_1 + (\Delta f_1^{(n+1)}\cdot\partial F_1/\partial f_1 + \Delta f_2^{(n+1)}\cdot\partial F_1/\partial f_2 + \cdots + \Delta f_n^{(n+1)}\cdot\partial F_1/\partial f_n) \qquad [2.4]$$

By taking $i = j$, we have

$$x_j^{(1)} = X_j + (\Delta f_1^{(1)}\cdot\partial F_j/\partial f_1 + \Delta f_2^{(1)}\cdot\partial F_j/\partial f_2 + \cdots + \Delta f_n^{(1)}\cdot\partial F_j/\partial f_n)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$x_j^{(n+1)} = X_j + (\Delta f_1^{(n+1)}\cdot\partial F_j/\partial f_1 + \Delta f_2^{(n+1)}\cdot\partial F_j/\partial f_2 + \cdots + \Delta f_n^{(n+1)}\cdot\partial F_j/\partial f_n) \qquad [2.5]$$

The system of equations corresponding to [2.5] for all values of $j = 1, 2, \cdots, n$ can be put in one single matrix equation

$$DU = Y \qquad [2.6]$$

where

$$D = \begin{Bmatrix} 1 & \Delta f_1^{(1)} & \Delta f_2^{(1)} & \cdots & \Delta f_n^{(1)} \\ 1 & \Delta f_1^{(2)} & \Delta f_2^{(2)} & \cdots & \Delta f_n^{(2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \Delta f_1^{(n+1)} & \Delta f_2^{(n+1)} & \cdots & \Delta f_n^{(n+1)} \end{Bmatrix}$$

$$Y = \begin{Bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \cdots & x_n^{(2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_1^{(n+1)} & x_2^{(n+1)} & x_3^{(n+1)} & \cdots & x_n^{(n+1)} \end{Bmatrix}.$$

$$U = \begin{Bmatrix} X_1 & X_2 & X_3 & \cdots & X_n \\ \dfrac{\partial F_1}{\partial f_1} & \dfrac{\partial F_2}{\partial f_1} & \dfrac{\partial F_3}{\partial f_1} & \cdots & \dfrac{\partial F_n}{\partial f_1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \dfrac{\partial F_1}{\partial f_n} & \dfrac{\partial F_2}{\partial f_n} & \dfrac{\partial F_3}{\partial f_n} & \cdots & \dfrac{\partial F_n}{\partial f_n} \end{Bmatrix}.$$

From [2.6] we have

$$U = D^{-1} Y \qquad [2.7]$$

provided D is nonsingular.

The first row of $D^{-1} Y$ in [2.7] gives $(X_1, X_2, X_3 \cdots X_n)$.

3. One might be inclined to solve the linear system of equations in [2.6] instead of resorting to find the inverse of D and then finding $D^{-1} Y$. It is our intention to show that finding the inverse of D can be exploited in a better way so as to reduce the further work.

We have $f_i^{(j)}$ in our earlier caloulations

Let
$$\theta_j = \sum_{i=1}^{n} [f_i^{(j)}]^2 \qquad [3.1]$$

$$j = 1, 2, \cdots, (n+1).$$

Let the value of $(X_1, X_2, \cdots X_n)$ obtained in [2.7] be $(x_1^{(p)}, x_2^{(p)}, \cdots x_n^{(p)})$ and the values of f_i's corresponding to these values of x_i's be $f_i^{(p)}$. Then let θ_k be the minimum of θ_j's found in [3.1]. Then replace the k-th row of D and Y by

$$\left.\begin{array}{l}(1,\ \Delta f_1^{(p)},\ \Delta f_2^{(p)},\ \cdots\ \Delta f_n^{(p)}) \\ \text{and }\ (x_1^{(p)},\ x_2^{(p)},\ \cdots,\ x_n^{(p)}).\end{array}\right\} \begin{array}{l}\text{where } p \neq i \\ j = 1,\ 2,\ 3,\ \cdots,\ (n+1).\end{array}$$

We call the matrix D in which the k-th row has been replaced by an another row as D_1 and the corresponding Y as Y_1.

Then we have[2]

$$D_1^{-1} = D^{-1} - [1/(1 + \nu d_k)]\, d_k (\nu D^{-1}) \qquad [3.2]$$

where ν is the k-th row of D_1, d_k is the k-th column of D^{-1}. So, the next approximation is obtained from $D_1^{-1} Y_1$ as in [2.7].

The procedure of Section 3 can be repeated as many times as required to obtain the required accuracy.

4. When $n = 1$, that is, when $f_1(x_1) = 0$ has to be solved, the method described reduces to the well known "method of chords"[1] or the "method of linear iterpolation".

Let $x_1^{(1)}$ and $x_2^{(1)}$ be the two approximations. Then we have from [2.4]

$$x_1^{(1)} = X_1 - f_1^{(1)} \cdot \partial F_1/\partial f_1$$

$$x_1^{(2)} = X_1 - f_1^{(2)} \cdot \partial F_1/\partial f_1$$

Then

$$X_1 = (x_1^{(2)} f_1^{(1)} - x_1^{(1)} f_1^{(2)})/(f_1^{(1)} - f_1^{(2)})$$

$$= x_1^{(1)} - [(x_1^{(1)} - x_1^{(2)})/(f_1^{(1)} - f_1^{(2)})]\, f_1^{(1)}$$

So, the metod described is a generalised linear interpolation method.

5. CONCLUSION

The method of Steepest descent[1] and Newton's method[1] require the evaluation of partial derivatives during each cycle of iteration. In problems where the expressions involve complicated transcendental functions, the work involved in finding the partial derivatives is so much, the desire to avoid the calculation of partial derivatives often arises. In addition to this, though the Newton's method has a better convergence factor if the initial approximation is close enough to the actual root, the process may diverge if the initial guess is not good. Secondly, we have to solve a set of linear simultaneous equations each time during the iterative cycle. The method which has been described here does not demand the evaluation of partial derivatives. Secondly, the inverse of the matrix D in [2 6] can be obtained easily from the procedure of section (3), without actually resorting to find the inverse, which saves time and involves less arithmetic. It is very unusual that in any method, the first cycle gives good enough accuracy. It will be necessary to repeat

TABLE 6.1

Number of Iterations	SYSTEM I				SYSTEM II			
	Present Method		Newton's Method		Present Method		Newtons Method	
	x	y	x	y	x	y	x	y
0	3 0	5.0	3.0	5.0	3.0	5.0	3.0	5.0
1	1.81167	11.1883	1.60000	11.400000	1 89043	3.68688	1.695274	3.687836
2	1.28088	11.7191	– 0.42758	13.42758	1.30395	3.55565	0.166748	3.919783
3	0 86639	12.1332	– 53.40888	66.40888	0.888815	3.62577	– 118.4137	8.659002
4	0.494560	12.5054	diverges		0.514820	3.71263	– 78.94054	– 264.47
5	0.135214	12.8648			0.129541	3.78305	diverges	
10	– 1.59712	14.5971			– 2.61444	3.08303		
15	– 2.25244	15.2524			– 1.60021	3.28713		
20	– 2.29564	15.2956			– 2.04678	2.77605		
22	– 2.29568	15.2957			– 2.03794	2.98269		
25					– 2.00379	2.99926		
30					– 2.00000	3.00000		

the iterative procedure several times to obtain the required accuracy. As such, we find the method described a very useful one in solving a set of non-linear equations and especially so where the partial derivatives can be calculated with much difficulty.

6. NUMERICAL EXAMPLE

Two systems of equations are considered here.

System I: $f_1 = x + y - 13 = 0,\ f_1 = 3x^3 + y + 21 = 0$

System II: $f_1 = x^2 + y^2 - 13 = 0,\ f_2 = 3x^3 + y + 21 = 0$

The real root of System I is

$$(-2.295679336,\ 15.295679336)$$

and that of System II are

$$(-2,\ 3) \text{ and } (-1.81319376,\ -3.11646122)$$

An approximation for both the systems was taken to be $(3, 5)$ and the results are given in Table 6.1.

ACKNOWLEDGEMENT

The author is very grateful to Prof. P. L. Bhatnagar for his valuable suggestions.

REFERENCES

1. Zaguskin V. L. .. .. Handbook of Numerical methods for the solution of algebraic and transcendental equations. Chapter 6, p. 171-176, [Pergamon Press, 1961.]
2. Faddeev D. K. and V. N. Faddeeva .. Computational methods of linear algebra, Chapter 2, p. 173-178. [W. H. Freeman and Company, 1963].
3. Scarborough J. .. .. *Numerical Mathematical Analysis* Chapter X, p. 192-211. [Oxford Book Company, 1964].