Numerical Solution of Systems of Nonlinear Equations*

FERDINAND FREUDENSTEIN† AND BERNARD ROTH†

Columbia University and Stanford University

Abstract. For simultaneous nonlinear equations, the convergence of all functional iteration procedures is dependent upon a good initial approximation to the desired root. In this paper, the restriction on the choice of the initial approximation has been circumvented by dividing each problem into a number of subsidiary problems in accordance with a procedure which is essentially a numerical adaptation of the implicit function theorem. Additional procedures are developed for dealing with sets of equations with singularities in the domain of the iteration.

Introduction

Newton's method, because of its quadratic convergence, is mathematically the most preferable of the several known methods for the solution of systems of nonlinear equations [1]. Practically, however, a very important limitation (on Newton's method, and in fact, all of the so-called functional iteration methods) is that it does not generally converge to some solution from an arbitrary starting point. Thus Newton's method may fail to converge if the initial estimate is not sufficiently close to the root.

The size of the domain of convergence depends upon the system of equations. For real algebraic equations, generally, the size of the domain of convergence is inversely related to the degree and number of equations. Therefore, one finds that for two simultaneous second degree equations almost any initial estimate will lead to one of the roots, while for eight simultaneous tenth degree equations the domain becomes much smaller, and it may be very difficult to obtain an initial estimate from which the iteration converges.

The authors and others [2, 3, 4] have found that, even in physically oriented problems, this need for a "good" initial estimate may become a severe restriction which, in effect, renders the functional iteration techniques useless.

This paper describes several algorithms which greatly relax the restrictions on the domain of the initial estimate. Although the discussion is limited to Newton's method, the reader should note that the algorithms are general and can be used in conjunction with any of the functional iteration techniques.

Algorithm —Parameter-Perturbation Procedure

The parameter-perturbation procedure is an algorithm for determining a root of a set of simultaneous nonlinear equations. It is a method essentially independent of the need of a "good" initial approximation.

* Received March, 1963.

† Department of Mechanical Engineering.

To determine a solution of the set of equations

$$f_i(\mathbf{x}) = 0, \qquad i = 1, 2, 3, \cdots, n, \quad (1)$$

where $\mathbf{x}$ is an n-dimensional vector with components $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$, and $f_i(\mathbf{x})$ is of the form

$$f_i(\mathbf{x}) = \sum_{k=0}^{m_i} p_{ik}\Phi_{ik}(\mathbf{x}), \quad (2)$$

where the p_{ik} are parameters (or numbers),[1] first consider another set of equations—the derived equations—

$$g_i^{(0)}(\mathbf{x}) = 0, \qquad i = 1, 2, 3, \cdots, n. \quad (3)$$

These derived equations may be any set, with *one known root*, which belongs to the same family as $f_i(\mathbf{x})$, that is,

$$g_i^{(0)}(\mathbf{x}) = \sum_{k=0}^{m_i} q_{ik}^{(0)}\Phi_{ik}(\mathbf{x}). \quad (4)$$

Generally, $q_{ik}^{(0)} \neq p_{ik}$.

The derived equations, $g_i^{(0)}(\mathbf{x}) = 0$, are "deformed" into the equations $f_i(\mathbf{x}) = 0$ by means of a finite number, N, of successive small increments in the parameters. Formally, define N sets of equations[2]

$$g_i^{(j)}(\mathbf{x}) = \sum_{k=0}^{m_i} q_{ik}^{(j)}\Phi_{ik}(\mathbf{x}), \quad j = 1, 2, 3, \cdots, N \quad (5)$$

such that

$$g_i^{(N)}(\mathbf{x}) = f_i(\mathbf{x}), \quad \text{and} \quad q_{ik}^{(j)} = q_{ik}^{(0)} + (p_{ik} - q_{ik}^{(0)})\frac{j}{N}. \quad (6)$$

Although the expressions are in terms of N equal steps, the method does not require this—the steps may be varied in any desired manner.

The desired root is obtained by solving the N sets of equations as follows. The known root of $g_i^{(0)}(\mathbf{x}) = 0$ is used as an initial approximation for the iterative solution of $g_i^{(1)}(\mathbf{x}) = 0$. Then the root of $g_i^{(1)}(\mathbf{x}) = 0$ is used as an initial estimate of the root of $g_i^{(2)}(\mathbf{x}) = 0$, and so forth, until the root of $g_i^{(N)}(\mathbf{x}) = f_i(\mathbf{x}) = 0$ is obtained. In order to obtain a root of $g_i^{(j)}(\mathbf{x}) = 0$, we start with a root of $g_i^{(j-1)}(\mathbf{x}) = 0$ as an initial approximation, and compute successive values from

$$\frac{\partial g_i^{(j)}(\mathbf{x}_l)}{\partial \mathbf{x}_r}(\mathbf{x}_{l+1} - \mathbf{x}_l) = -g_i^{(j)}(\mathbf{x}_l). \quad (7)$$

[1] m_i may take on any value provided the functions $f_i(\mathbf{x}) = 0$ are compatible, i.e. it is assumed that the set $f_i(\mathbf{x}) = 0$ is not inconsistent.

[2] It is possible to define a more general algorithm by replacing equation (5) with $g_i^{(j)}(\mathbf{x}) = g_i^{(0)}(\mathbf{x}) + (f_i(\mathbf{x}) - g_i^{(0)}(\mathbf{x}))(j/N)$. Here, $f_i(\mathbf{x})$ and $g_i^{(0)}(\mathbf{x})$ can be of any form, and are not restricted by equations (2) and (4).

(Equation (7) is, of course, Newton's method in matrix form, and subscripts l and $l + 1$ indicate successive approximations to the root of $g_i^{(j)}(\mathbf{x}) = 0$.)

Thus, we have, in effect, two sets of changes for each increment:

(a) a change in $\mathbf{x}$—this is the usual type of increment,

(b) a change in the coefficients $q_{ik}^{(j)}$—the parameters of the equation.

That is to say, instead of starting with the equation we wish to solve and incrementing $\mathbf{x}$, start with a *different equation* (same functions of $\mathbf{x}$, but different parameters), such that the solution of this derived equation is known. Then proceed to increment both $\mathbf{x}$ and the parameters until the correct equation and its solution are found. In this way, proceed in small increments all the time, and in *any single step the first approximation is likely to be a close one.*

Conditions for Convergence

In discussing the convergence of the algorithm it is convenient to regard the discrete index j as a continuous variable t. With this substitution, (5) can be rewritten as

$$g_i(t, \mathbf{x}) = \sum_{k=0}^{m_i} q_{ik}^{(t)} \Phi_{ik}(\mathbf{x}), \qquad 0 < t \leqq N,$$

and after substituting (6) becomes

$$g_i(t, \mathbf{x}) = \sum_{k=0}^{m_i} \left(q_{ik}^{(0)} + (p_{ik} - q_{ik}^{(0)}) \frac{t}{N} \right) \Phi_{ik}(\mathbf{x}). \tag{8}$$

It follows that the roots $\mathbf{X}_f$ of $g_i(t, \mathbf{x}) = 0$ are also functions of the variable t, and we write $\mathbf{X}_f = \mathbf{X}_f(t)$.

A necessary and sufficient condition for the convergence of the algorithm is, clearly, the convergence of each step, i.e. the size of the increment in the $q_{ik}^{(j)}$'s in each step must lie within the domain of convergence of Newton's method. The domain for a Newtonian iteration—and in fact, all functional iteration methods—is given by a Lipschitz condition with constant less than one [5].

The algorithm converges to a root of $f_i(\mathbf{x}) = 0$ if the functions $f_i(\mathbf{x})$ and $g_i^{(0)}(\mathbf{x})$ are such that:

(1) $g_i(t, \mathbf{x}) = 0$ is continuous for $0 \leqq t \leqq N$

(2) $\mathbf{X}_f(t)$ is continuous for $0 \leqq t \leqq N$.

These conditions follow directly from the description of the parameter perturbation procedure; they are sufficient to assure convergence of the entire process.[3]

We now illustrate the sufficienty of the above conditions geometrically. For simplicity, consider finding a root of a single equation, $f(\mathbf{x}) = 0$. For $f(\mathbf{x}) = 0$ as shown in Figure 1, it is obvious that Newton's method would almost surely fail to converge for any initial estimate of the root $\mathbf{X}_N$ for which $\mathbf{x} \geqq A$. How-

[3] We assume the step size is within the domain of convergence of the Newtonian iteration.

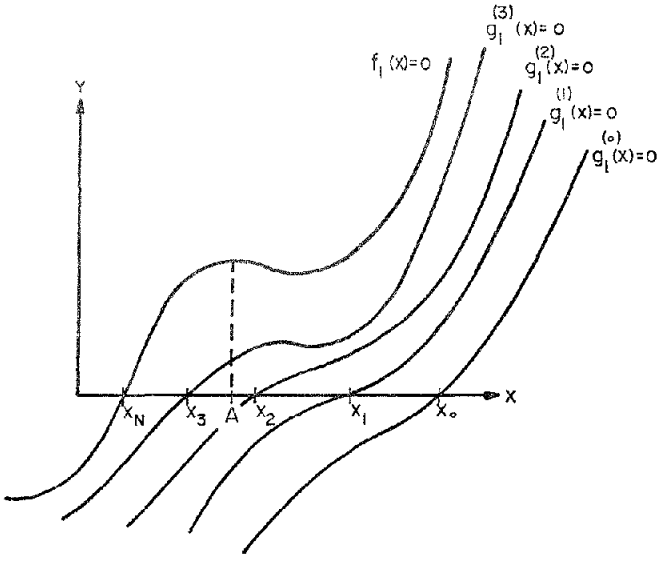

FIG. 1. Parameter perturbation procedure illustrated geometrically for one equation in one unknown. Here $N = 4$, $\mathbf{X}_0$ is the starting point and $\mathbf{X}_N$ the desired root.

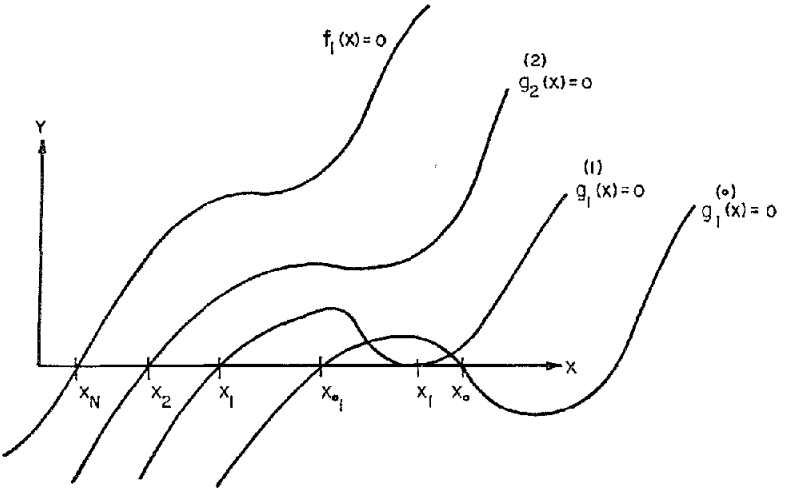

FIG. 2. The parameter perturbation procedure requires continuity of the roots $\mathbf{X}_t$. The violation of this condition causes a start from $\mathbf{X}_0$ to fail. A start from $\mathbf{X}_{0_1}$, however, will converge to $\mathbf{X}_N$.

ever, if for some $\mathbf{x}$, say $\mathbf{X}_0$, we define $g^{(0)}(\mathbf{x}) = 0$ with $\mathbf{X}_0$ as a root, then the desired root, $\mathbf{X}_N$, is obtained by successively deforming $g^{(0)}(\mathbf{x}) = 0$ into $g^{(1)}(\mathbf{x}) = 0$, $g^{(2)}(\mathbf{x}) = 0$, $g^{(3)}(\mathbf{x}) = 0$, and finally $f(\mathbf{x}) = 0$.

Figure 2 illustrates the restriction of a continuous root within the domain defined by $\mathbf{X}_0$ and $\mathbf{X}_N$. Here, since the root $\mathbf{X}_0$ does not continuously approach $\mathbf{X}_N$, the method fails. However, a choice of $\mathbf{X}_{0_1}$ would succeed.

A geometrical interpretation of equation (7) leads to the conclusion that these considerations should be equally valid for more complicated problems. For n equations in n unknowns, the $(l+1)$-th approximation to a root of the system $g_i^{(j)}(\mathbf{x}) = 0$ is obtained at the point of intersection of n hyper-planes. These planes are parallel to the tangent planes of the n-dimensional surfaces $g_i^{(j)}(\mathbf{x}) =$ constant at the point $\mathbf{x}_l$, and pass through the point whose distance from $\mathbf{x}_l$ parallel to, say, the r-th coordinate axis is $-g_i^{(j)}(\mathbf{x}_l)/(\partial g_i^{(j)}(\mathbf{x}_l)/\partial \mathbf{x}_r)$.

Singularities

As a practical matter it is difficult to know a priori if a given set of equations satisfies the sufficiency conditions, since knowledge of these conditions implies, in effect, knowledge of the solution. (This limitation is common to all functional iteration techniques.) To avoid making a restart if a nonconvergent sequence occurs in the iteration of any member of the family $g_i^{(0)}(\mathbf{x}) = 0 \rightarrow f_i(\mathbf{x}) = 0$, two additional algorithms have been developed. Although we only discuss the case of a singularity which is manifested in a vanishing Jacobian determinant, i.e. $| \partial g^{(j)}/\partial \mathbf{x}_r | \rightarrow 0$, the methods are equally applicable to other computational difficulties which lead to nonconvergence of the Newtonian iteration.

The more general of the two schemes follows. Whenever the Jacobian determinant falls below a predetermined value, the increments in the parameters $q_{ik}^{(j)}$ are made selectively unequal. At this point, the parameters are perturbed one at a time and the effect on the Jacobian of the change in each parameter is noted. The variables are then selectively incremented so as to increase the value of the Jacobian to above a predetermined minimum. Thereafter, the regular variation may be re-introduced in a suitably "continuous" manner.

A simpler scheme can be applied to problems where the system of simultaneous equations can be easily transformed into a different set of equations which describe the same physical system but whose roots are numerically different from those of the original set. For such problems the singularities can be eliminated by making the (above) transformation simultaneously in $g_i^{(j)}(\mathbf{x}) = 0$ and $f_i(\mathbf{x}) = 0$. In this way the j-value in $g_i^{(j)}(\mathbf{x})$ remains the same, and the total number of "deformed" equations is not increased by the presence of a singularity.

In mechanism synthesis the $\Phi_{ik}(\mathbf{x})$'s are functions of vectors which represent moving links (or bars) in a given reference position. $\Phi_{ik}(\mathbf{x})$ is the same for all i, and the p_{ij}'s are functions of known displacements from the reference position. In such problems an effective transformation is obtained by simply choosing a different reference position.

If the singularity is due to a multiple root, or several nearly coincident roots, the difficulty could be resolved either by the above or by a process described by Milne [6]. This latter is especially useful if the multiplicity occurs in the root of $f_i(\mathbf{x}) = 0$, the final set.

While there is no absolute guarantee that these routines will assure convergence, it has been found in practice that in a large majority of cases they suffice to eliminate the singularities.

TABLE 1

Step number	j	$q_{10}^{(j)}$	$q_{11}^{(j)}$	$q_{12}^{(j)}$	$q_{13}^{(j)}$	$q_{14}^{(j)}$	$q_{20}^{(j)}$	$q_{21}^{(j)}$	$q_{22}^{(j)}$	$q_{23}^{(j)}$	$q_{24}^{(j)}$	Root
	0	−71	1	−50	−13	−1	129	1	106	19	1	15.0, −2.00
1	1	−59.4	1	−40.4	−9.4	−1	97.4	1	82	15.4	1	9.9, −1.91
2	2	−47.8	1	−30.8	−5.8	−1	65.8	1	58	11.8	1	5.68, −1.78
3	3	−36.2	1	−21.2	−2.2	−1	34.2	1	34	8.2	1	3.70, −1.61
4	4	−24.6	1	−11.6	1.4	−1	2.6	1	10	4.6	1	4.85, −1.31
5	$4\frac{1}{2}$	−18.8	1	−6.8	3.2	−1	−13.2	1	−2	2.8	1	10.1, −0.85
6	$4\frac{2}{3}$	−17.35	1	−5.7	3.65	−1	−17.2	1	−5	2.35	1	16.0, −0.21
7	$4\frac{11}{12}$	−16.99	1	−5.3	3.76	−1	−18.1	1	−5.7	2.24	1	19.7, 1.29
8	$4\frac{15}{16}$	−16.63	1	−5	3.88	−1	−19.1	1	−6.5	2.13	1	18.9, 1.73
9	$4\frac{3}{4}$	−15.9	1	−4.4	4.1	−1	−21.1	1	−8	1.9	1	16.6, 2.35
10	5	−13	1	−2	5	−1	−29	1	−14	1	1	5.00, 4.00

Numerical Experience

The methods have proved especially useful in the solution of simultaneous-nonlinear-algebraic equations[4] such as are encountered in the synthesis of mechanisms [3]. Of value in this application is the fact that an arbitrary mechanism of the type required can serve as the desired starting approximation $g_i^{(0)}(\mathbf{x}) = 0$. The computational time on the IBM 7090 for sets of eight simultaneous seventh-degree equations was found to be less than five minutes, with $N = 100$.

In connection with the preparation of this paper, the methods were successfully tested on some simpler equations, namely, algebraic sets of second and third-degree equations. The computations involved in one such example follow.

EXAMPLE.

$$f_1(\mathbf{x}) = p_{10} + p_{11}\mathbf{x}_1 + p_{12}\mathbf{x}_2 + p_{13}\mathbf{x}_2^2 + p_{14}\mathbf{x}_2^3$$

$$f_2(\mathbf{x}) = p_{20} + p_{21}\mathbf{x}_1 + p_{22}\mathbf{x}_2 + p_{23}\mathbf{x}_2^2 + p_{24}\mathbf{x}_2^3$$

where $p_{10} = -13$, $p_{11} = 1$, $p_{12} = -2$, $p_{13} = 5$, $p_{14} = -1$; $p_{20} = -29$, $p_{21} = 1$, $p_{22} = -14$, $p_{23} = 1$, $p_{24} = 1$.

To solve the above, start with the "derived" set:

$$g_1^{(0)}(\mathbf{x}) = -71 + \mathbf{x}_1 - 50\mathbf{x}_2 - 13\mathbf{x}_2^2 - \mathbf{x}_2^3$$

$$g_2^{(0)}(\mathbf{x}) = 129 + \mathbf{x}_1 + 106\mathbf{x}_2 + 19\mathbf{x}_2^2 + \mathbf{x}_2^3$$

which has a known solution at $(15, -2)$.[5] A choice of $N = 5$ proves sufficient for $j = 1, 2, 3, 4$ but fails to converge the final set. The solution is obtained by dividing the interval between $j = 4$ and 5 into fractional parts. The computations are summarized in Table 1.

[4] Newton's method and several of the descent methods were found ineffective when applied to these equations.

[5] With this point (i.e. $\mathbf{x}_1 = 15$, $\mathbf{x}_2 = -2$) as an initial approximation, the Newtonian iteration fails to converge to a root of the system $f_1(\mathbf{x}) = 0, f_2(\mathbf{x}) = 0$.

Acknowledgments. The authors are grateful to the National Science Foundation for the support of the early phases of this research through Grant G-11988; to Dr. Reuben Hersh of the Department of Mathematics, Stanford University, for his critical comments; and to Columbia University and Stanford University for the use of computational facilities.

REFERENCES

1. TURNER, L. R. Solution of nonlinear systems. *Ann. New York Acad. Sci. 86* (1960), 817–827.
2. ARONSON, R. S. Analytical synthesis of multiloop linkages. Doctoral dissertation, Purdue University, 1962.
3. ROTH, B., AND FREUDENSTEIN, F. Synthesis of path-generating mechanisms by numerical methods. *Trans. ASME* (Aug. 1963), 298–306; see also discussion by T. P. Goodman.
4. MCLARNAN, C. W. Synthesis of six-link plane mechanisms by numerical analysis. *Trans. ASME* (Feb. 1963), 5–11.
5. HOUSEHOLDER, A. S. *Principles of Numerical Analysis*, pp. 135–136. McGraw-Hill, New York, 1953.
6. MILNE, W. E. *Numerical Calculus*, pp. 44–53. Princeton University Press, Princeton, N. J., 1949.