X-732-69-306
PREPRINT
NASA TM X-63622

A NEW ITERATION FOR LOCATING EQUILIBRIUM POINTS IN NONLINEAR SYSTEMS

CLARENCE CANTOR
FAWZI P. EMAD

JULY 1969

GSFC
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-732-69-306

PREPRINT

A NEW ITERATION FOR LOCATING EQUILIBRIUM POINTS IN NONLINEAR SYSTEMS

Clarence Cantor

Systems Division

and

Fawzi P. Emad*

Department of Electrical Engineering

July 1969

GODDARD SPACE FLIGHT CENTER

Greenbelt, Maryland

*University of Maryland.

A NEW ITERATION FOR LOCATING EQUILIBRIUM POINTS IN NONLINEAR SYSTEMS

Clarence Cantor, Systems Division

and

Fawzi P. Emad, Department of Electrical Engineering

Abstract

A discrete nth order nonlinear dynamic system, or an iteration for a nonlinear set of n algebraic equations in n unknowns, can be represented by

$$x_{k+1} = f(x_k)$$

where x_k is an nth order state vector. The problem is to locate the equilibrium points of the system, give some approximation of these points, when the original iteration is divergent or only slowly convergent. Newton's method solves the problem in general, but requires f(x) to be available in analytic form and also requires extensive computations of partial derivatives. Steffensen's iteration, utilizing sets of n + 2 iterates obtained from the original iteration, can also solve the problem while avoiding the difficulties in Newton's method. Steffensen's iteration is based on a linearization of the system about an equilibrium point x_e, in the form

$$x_{k+1} - x_e = A(x_k - x_e) .$$

However, Steffensen's iteration breaks down when the A matrix corresponding to a set of iterates has an eigenvalue equal to one, or when the matrix cannot be determined. The proposed iteration is equivalent to Steffensen's iteration in the general nonsingular case, but it can be readily extended in a meaningful way in a large class of singular cases where Steffensen's iteration breaks down. Also, the proposed method should produce better results in near singular cases where Steffensen's iteration can cause discrepancies due to numerical errors.

CONTENTS

INTRODUCTION

A large class of problems in nonlinear systems involves determining the solutions (equilibrium points) of the vector equation,

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

For example, in a discrete nonlinear dynamic system represented by

$$x_{k+1} = f(x_k) \tag{2}$$

where x_k is an nth order state vector, it is often necessary if not mandatory to determine the equilibrium points x_e (there may be more than one), where x_e is defined by

$$x_e = f(x_e) . \tag{3}$$

Equation 2 can also represent any iteration scheme involving n unknowns to be determined.

If an approximation x_0 of an equilibrium point is known, and if the process defined by (2) is convergent (asymptotically stable) in a region containing x_0, then the equilibrium point x_e can be obtained by repeated application of (2). However, this is not possible if the process is divergent, or impractical if the process is only slowly convergent.

Newton's method [1], [2] is a powerful method that can be used in general to solve equation 1 (or 2), even when the iteration defined by equation 2 is divergent. Let

$$g(x) = f(x) - x . \tag{4}$$

Then the equation to be solved is

$$g(x) = 0 . \tag{5}$$

If x_0 is an approximate solution of (5)' (or (1)), and x_e is the exact solution, then (5) can be linearized about $x = x_0$ to yield

$$g(x_0) + J(x_0)(x_e - x_0) = 0 \tag{6}$$

where $J(x_0)$ is the Jacobian of $g(x)$ evaluated at $x = x_0$.

Assuming $J^{-1}(x_0)$ exists, we can solve (6) for an approximation of x_e,

$$x_e \approx x_0 - J^{-1}(x_0)\, g(x_0) . \tag{7}$$

Since this is only an approximate solution of x_e (although hopefully a better approximation than x_0), we can treat (7) as an iteration, namely

$$x_1 = x_0 - J^{-1}(x_0)\, g(x_0) \quad \text{and} \quad x_{k+1} = x_k - J^{-1}(x_k)\, g(x_k) \tag{8}$$

which is Newton's iteration for a system of equations.

In terms of (1), we have

$$g(x_k) = f(x_k) - x_k \qquad \text{and} \qquad J(x_k) = A(x_k) - I$$

where $A(x_k)$ is the Jacobian of $f(x)$ at $x = x_k$. Hence, (8) can be written as

$$x_{k+1} = x_k - \left(A(x_k) - I\right)^{-1} \left(f(x_k) - x_k\right) . \tag{9}$$

Equation (8) (or (9)) will converge to x_e if x_0 is sufficiently close to x_e. One of the difficulties in Newton's method is the need to calculate $J(x_k)$, with its n^2 partial derivatives, at each x_k, which can require rather extensive calculations. This is alleviated in the modified Newton's method which in general is not as rapidly convergent. Here

$J(x_0)$ is retained throughout the iteration, namely

$$x_{k+1} = x_k - J^{-1}(x_0)\, g(x_k) . \tag{10}$$

Another limitation of Newton's method is that $g(x)$ (or $f(x)$) must be available in analytic form. Thus a computer simulation of a discrete control system $x_{k+1} = f(x_k)$, where $f(x)$ is not available in analytic form, and only sampled output x_k is available, cannot be treated directly by Newton's method.

Steffensen's iteration [3] is another method for solving (2) which is similar to Newton's method. It has the advantage of not requiring the calculation of partial derivatives. Also, it does not require that $f(x)$ be available in analytic form. It utilizes sets of $n + 2$ iterates obtained either from the system equation (2), or from a computer simulation of the system.

Assume that (2) is approximately linear in a region surrounding the equilibrium point x_e. This linearity can be expressed as

$$x_{k+1} - x_e = A(x_k - x_e) \tag{11}$$

where A represents the Jacobian of $f(x)$ at x_e.

It is easy to show then that

$$\Delta x_{k+1} \triangleq x_{k+2} - x_{k+1} = A(x_{k+1} - x_k)$$

or

$$\Delta x_{k+1} = A \Delta x_k . \tag{12}$$

We define $n \times n$ matrices X_0, ΔX_0, and ΔX_1 as follows.

$$X_0 = \left(x_0\ x_1\ x_2 \cdots x_{n-1}\right)$$

$$\Delta X_0 = \left(\Delta x_0\ \Delta x_1 \cdots \Delta x_{n-1}\right)$$

$$\Delta X_1 = \left(\Delta x_1\ \Delta x_2 \cdots \Delta x_n\right) \quad . \tag{13}$$

Then (12) yields

$$\Delta X_1 = A\,\Delta X_0 \quad \text{or} \quad A = \Delta X_1 \left(\Delta X_0\right)^{-1} \tag{14}$$

assuming $\left(\Delta X_0\right)^{-1}$ exists.

Utilizing (11), with $k = 0$, we can solve for x_e as follows.

$$(A-I)\,x_e = A\,x_0 - x_0 - \left(x_1 - x_0\right) = (A-I)\,x_0 - \Delta x_0$$

or

$$x_e = x_0 - (A-I)^{-1}\,\Delta x_0 \quad . \tag{15}$$

Using the expression for A of (14), we obtain after some manipulation,

$$(A-I)^{-1} = \Delta X_0 \left(\Delta X_1 - \Delta X_0\right)^{-1}$$

and

$$x_e = x_0 - \Delta X_0 \left(\Delta X_1 - \Delta X_0\right)^{-1} \Delta x_0 \quad . \tag{16}$$

Since the system is not really linear, (16) yields only an approximation of x_e, but one which hopefully is closer to x_e than the starting point x_0. Calling this new approximation $x_0^{(1)}$, we have

$$x_0^{(1)} = x_0 - \Delta X_0 \left(\Delta X_1 - \Delta X_0\right)^{-1} \Delta x_0 \quad . \tag{17}$$

We can repeat this process using $x_0^{(1)}$ as the starting point for a new set of n + 2 iterates, and apply (17) again to obtain $x_0^{(2)}$, and so forth. This yields the sequence $x_0^{(0)}$, $x_0^{(1)}$, $x_0^{(2)}$, $\cdots$. This process is Steffensen's iteration. Hopefully, the sequence $x_0^{(0)}$, $x_0^{(1)}$, $x_0^{(2)}$, $\cdots$ will converge to x_e. In general, this convergence will occur if the first starting point $x_0^{(0)}$ is close enough to x_e, even when the original iteration of equation 2 diverges.

One of the difficulties in Steffensen's iteration occurs when the matrix $(\Delta X_1 - \Delta X_0)$ is singular or nearly singular. When it is singular, there is nothing one can do to continue the process except perhaps introduce an arbitrary perturbation in the starting point $x_0^{(k)}$ and hope that the new set of n + 2 iterates yields a non-singular $(\Delta X_1 - \Delta X_0)$. When the matrix $(\Delta X_1 - \Delta X_0)$ is nearly singular, numerical errors can produce a large discrepancy in the resulting new approximation of x_e.

The singularity of $(\Delta X_1 - \Delta X_0)$ can arise in two ways. Since $\Delta X_1 = A\Delta X_0$, we have

$$\Delta X_1 - \Delta X_0 = (A - I)\,\Delta X_0 \quad . \tag{18}$$

Thus $\Delta X_1 - \Delta X_0$ will be singular if either $(A - I)$ or ΔX_0 is singular.

The proposed method to be described next provides a systematic and meaningful extension of the iteration when $(\Delta X_1 - \Delta X_0)$ is singular or nearly singular. It avoids the numerical errors associated with inverting a near singular matrix. In the general case, when $(\Delta X_1 - \Delta X_0)$ is non-singular, the proposed algorithm yields the same results as Steffensen's iteration with an equivalent amount of computations. It has one advantage here also in that after an iteration, the A matrix can be obtained with less additional calculations than would be the case in Steffensen's iteration.

PROPOSED NEW METHOD

First we will consider the general case when $(\Delta X_1 - \Delta X_0)$ is non-singular. This implies that neither $(A-I)$ nor ΔX_0 is singular.

Consider the matrix

$$\Delta X_0' \triangleq \left(\Delta x_0 \, \Delta x_1 \cdots \Delta x_{n-1} \, \Delta x_n \right) . \tag{19}$$

This matrix is obviously singular since it contains $n + 1$ columns and only n rows. Hence, the last column, Δx_n, can be written as a linear combination of the first n columns, namely

$$\Delta x_n = c_0 \Delta x_0 + c_1 \Delta x_1 + \cdots + c_{n-1} \Delta x_{n-1} . \tag{20}$$

Now

$$\Delta x_i \triangleq x_{i+1} - x_i = \left(x_{i+1} - x_e \right) - \left(x_i - x_e \right)$$

or

$$\Delta x_i = (A-I) \left(x_i - x_e \right) \tag{21}$$

utilizing (11). Thus we can write (20) as

$$(A-I) \left(x_n - x_e \right) = (A-I) \left[c_0 \left(x_0 - x_e \right) + c_1 \left(x_1 - x_e \right) + \cdots + c_{n-i} \left(x_{n-1} - x_e \right) \right] . \tag{22}$$

Premultiplying by $(A-I)^{-i}$ (which is assumed to exist in this general case) yields

$$x_n - x_e = c_0 \left(x_0 - x_e \right) + c_1 \left(x_1 - x_e \right) + \cdots + c_{n-1} \left(x_{n-1} - x_e \right) . \tag{23}$$

Solving for x_e yields

$$x_e = \frac{c_0 x_0 + c_1 x_1 + \cdots + c_{n-1} x_{n-1} - x_n}{\left(\sum_{i=0}^{n-1} c_i\right) - 1} \quad . \tag{24}$$

Assuming

$$\sum_{i=0}^{n-1} c_i \neq 1$$

(which we will prove), (24) represents our next estimate of x_e which we will call $x_0^{(1)}$.

Thus

$$x_0^{(1)} = \frac{c_0 x_0 + c_1 x_1 + \cdots + c_{n-1} x_{n-1} - x_n}{\left(\sum_{i=0}^{n-1} c_i\right) - 1} \quad . \tag{25}$$

To solve for the c_i, we go back to (20) which can be written as

$$\Delta x_n = \left(\Delta x_0 \; \Delta x_1 \cdots \Delta x_{n-1}\right) \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} \tag{26}$$

The $n \times n$ matrix is simply ΔX_0 whose inverse is assumed to exist. Hence

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \left(\Delta X_0\right)^{-1} \Delta x_n = \left(\Delta X_0\right)^{-1} \left(x_{n+1} - x_n\right) \quad . \tag{27}$$

The combination of (25) and (27) represents the new algorithm for determining $x_0^{(1)}$, $x_0^{(2)}$, etc. Equation (27) is used first to determine the constants c_i which are then used in (25) to obtain $x_0^{(1)}$. Then $x_0^{(1)}$ is used as the starting point to obtain a new set of $n + 2$ iterates from (2), and the process is repeated. This algorithm for the approximation of x_e has been derived using the same equation (21) that yields Steffensen's iteration, and thus gives identical results in the general non-singular case. The amount of calculations in each algorithm is about the same. The new algorithm has the advantage of yielding the A matrix, when that is desired, with fewer additional calculations than Steffensen's iteration. Since $(\Delta X_0)^{-1}$ is calculated in the new algorithm, $A = (\Delta X_1)\,\Delta X_0^{-1}$ is determined from one additional matrix multiplication. In Steffensen's iteration (equation 17), we determine the matrix $\Delta X_0 (\Delta X_1 - \Delta X_0)^{-1} = (A-I)^{-1}$, so that a matrix inversion is required before extracting A. The greatest advantage of the new algorithm, however, is that the same general form is used to extend the iteration in singular or near singular cases, where Steffensen's iteration breaks down or causes large errors. This will be discussed later.

We still have to prove that

$$\sum_{i=0}^{n-1} c_i \neq 1 .$$

in order for the new algorithm to be valid.

Proof: Assume

$$\sum_{i=0}^{n-1} c_i = 1 .$$

Then (20) can be written as

$$\sum_{i=0}^{n-1} c_i \Delta x_n = c_0 \Delta x_0 + c_1 \Delta x_1 + \cdots\cdots + c_{n-1} \Delta x_{n-1} \quad . \tag{28}$$

Using (12), we can write Δx_n as

$$\Delta x_n = A \Delta x_{n-1} = A^2 \Delta x_{n-2} = \cdots\cdots = A^n \Delta x_0 \quad . \tag{29}$$

Utilizing (29) in (28), we obtain

$$c_0 \left(A^n - I\right) \Delta x_0 + c_1 \left(A^{n-1} - I\right) \Delta x_1 + \cdots\cdots + \cdots\cdots + c_{n-1} (A - I) \Delta x_{n-1} = 0 \quad . \tag{30}$$

Each term in parenthesis in (30) contains the factor $(A - I)$ whose inverse exists in the general case. Then multiplying equation 30 by $(A - I)^{-1}$ yields

$$c_0 \left(A^{n-1} + A^{n-2} + \cdots\cdots + A + I\right) \Delta x_0 + c_1 \left(A^{n-2} + A^{n-3} + \cdots\cdots A + I\right) \Delta x_1$$

$$+ \cdots\cdots + c_{n-2} (A + I) x_{n-2} + c_{n-1} \Delta x_{n-1} = 0 \quad . \tag{31}$$

Performing the indicated matrix multiplications, using (12), yields

$$c_0 \Delta x_0 + \left(c_0 + c_1\right) \Delta x_1 + \cdots\cdots \left(c_0 + c_1 + \cdots\cdots + c_{n-1}\right) \Delta x_{n-1} = 0$$

or

$$\Delta x_{n-1} = - c_0 \Delta x_0 - \left(c_0 + c_1\right) \Delta x_1 - \cdots\cdots - \left(c_0 + c_1 + \cdots\cdots + c_{n-2}\right) \Delta x_{n-2} \quad . \tag{32}$$

Equation (32) implies that the last column of ΔX_0 is a linear combination of the first n - 1 columns. This implies that det $\Delta X_0 = 0$ which contradicts the fact that $\left(\Delta X_0\right)^{-1}$ exists in the general case. Q.E.D.

Thus

$$\sum_{i=0}^{n-1} c_i \neq 1$$

and the algorithm of (25) and (27) is valid.

We will now extend the algorithm to cover the case of $|\Delta X_0| = 0$ and $|A-I| \neq 0$. In all of the subsequent arguments, we assume that $\Delta x_0 \neq 0$, because if it were the problem is solved trivially, i.e. $\Delta x_0 = 0 \Rightarrow \Delta x_0 = \Delta x_1 = \cdots = 0$ which implies that $x_0 = x_1 = x_2 = \cdots = x_{n+1}$ and we are at an equilibrium point already.

Let r be the rank of the matrix (ΔX_0). Then the first r columns of (ΔX_0) are linearly independent.

Proof: Let k be the maximum number of consecutive columns, starting with the first, that are linearly independent. Then $1 \leq k \leq r < n$. Then the $k+1$ column must be expressable as a linear combination of the first k columns, or

$$\Delta x_k = a_0 \Delta x_0 + a_1 \Delta x_1 + \cdots + a_{k-1} \Delta x_{k-1} . \tag{33}$$

We can then express Δx_{k+1} as

$$\Delta x_{k+1} = A \Delta x_k = a_0 \Delta x_1 + a_1 \Delta x_2 + \cdots + a_{k-1} \Delta x_k$$

$$\Delta x_{k+1} = a_0 \Delta x_1 + a_1 \Delta x_2 + \cdots + a_{k-2} \Delta x_{k-1} + a_{k-1} \left(a_0 \Delta x_0 + a_1 \Delta x_1 + \cdots + a_{k-1} \Delta x_{k-1} \right)$$

or

$$\Delta x_{k+1} = b_0 \Delta x_0 + b_1 \Delta x_1 + \cdots + b_{k-1} \Delta x_{k-1} . \tag{34}$$

Thus the k + 2 column is also expressable as a linear combination of the first k columns.

We can express the k + 3 column, Δx_{k+2}, as

$$\Delta x_{k+2} = A \Delta x_{k+1} = b_0 \Delta x_1 + b_1 \Delta x_2 + \cdots + b_{k-2} \Delta x_{k-1} + b_{k-1} \left(a_0 \Delta x_0 + a_1 \Delta x_1 + \cdots + a_{k-1} \Delta x_{k-1} \right) . \quad (35)$$

Thus the k + 3 column is again a linear combination of the first k columns. Continuing the process will show that every column after the first k columns is a linear combination of the first k columns. Hence the rank of ΔX_0 must equal k or k = r.

Since the first r columns are linearly independent, we can express the r + 1 column as a linear combination of the first r columns. Thus

$$\Delta x_r = c_0 \Delta x_0 + c_1 \Delta x_1 + \cdots + c_{r-1} \Delta x_{r-1} . \quad (36)$$

From (21), we have

$$\Delta x_i = (A - I) \left(x_i - x_e \right) .$$

Then (36) can be written as

$$(A - I) \left(x_r - x_e \right) = (A - I) \left[c_0 \left(x_0 - x_e \right) + \cdots + c_{r-1} \left(x_{r-1} - x_e \right) \right]$$

Premultiplying by $(A-I)^{-1}$ and solving for x_e, we obtain

$$x_e = \frac{c_0 x_0 + c_1 x_1 + \cdots + c_{r-1} x_{r-1} - x_r}{\left(\sum_{i=0}^{r-1} c_i \right) - 1} . \quad (37)$$

Since this represents our next estimate of x_e, we denote it by $x_0^{(1)}$, or

$$x_0^{(1)} = \frac{c_0 x_0 + c_1 x_1 + \cdots + c_{r-1} x_{r-1} - x_r}{\left(\sum_{i=0}^{r-1} c_i\right) - 1} \quad . \tag{38}$$

Note that this is exactly the same form as (25) for the general non-singular case, except that n is replaced by r. We can prove that

$$\sum_{i=0}^{r-1} c_i \neq 1$$

in the same way that we proved that

$$\sum_{i=0}^{n-1} c_i \neq 1$$

in the non-singular case. Simply replace n by r in the latter proof.

To complete the algorithm of (38), we must solve for c_i, $i = 0, 1, \cdots r-1$. If we take the dot product of (36) with Δx_0, Δx_1, $\cdots$, and Δx_{r-1}, we obtain

$$G_r \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{r-1} \end{bmatrix} = \begin{bmatrix} \Delta x_0 \cdot \Delta x_r \\ \Delta x_1 \cdot \Delta x_r \\ \vdots \\ \Delta x_{r-1} \cdot \Delta x_r \end{bmatrix} \tag{39}$$

where

$$G_r \triangleq \begin{bmatrix} \Delta x_0 \cdot \Delta x_0 & \Delta x_0 \cdot \Delta x_1 \cdots & \Delta x_0 \cdot \Delta x_{r-1} \\ \Delta x_1 \cdot \Delta x_0 & \Delta x_1 \cdot \Delta x_1 \cdots & \Delta x_1 \cdot \Delta x_{r-1} \\ \vdots & & \\ \Delta x_{r-1} \cdot \Delta x_0 & \Delta x_{r-1} \cdot \Delta x_1 \cdots & \Delta x_{r-1} \cdot \Delta x_{r-1} \end{bmatrix}$$

is the Gramian corresponding to the first r columns of ΔX_0. Since these columns are linearly independent, $|G_r| \neq 0$ and G_r^{-1} exists. Then,

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{r-1} \end{bmatrix} = G_r^{-1} \begin{bmatrix} \Delta x_0 \cdot \Delta x_r \\ \Delta x_1 \cdot \Delta x_r \\ \vdots \\ \Delta x_{r-1} \cdot \Delta x_r \end{bmatrix} \tag{40}$$

Equations (38) and (40) constitute the algorithm in the case where ΔX_0 is singular with rank r and (A-I) is non-singular. Note that these equations can also serve as the algorithm for the general non-singular case by letting $r = n$. Then (38) becomes the same as (25) and (40) reduces to (27) since

$$G_n = (\Delta X_0)^T \Delta X_0 \tag{41}$$

and

$$\begin{bmatrix} \Delta x_0 \cdot \Delta x_n \\ \Delta x_1 \cdot \Delta x_n \\ \vdots \\ \Delta x_{n-1} \cdot \Delta x_n \end{bmatrix} = (\Delta X_0)^T \Delta x_n \, . \tag{42}$$

Then

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_{n-1} \end{bmatrix} = G_n^{-1} \left(\Delta X_0\right)^T \Delta x_n = \Delta X_0^{-1} \left(\Delta X_0^T\right)^{-1} \Delta X_0^T \Delta x_n = X_0^{-1} \Delta x_n \tag{43}$$

which is the same as (27).

We will now show that the algorithm of (38) and (40) can be used meaningfully in many cases when $(A-I)$ is singular. The only restriction on its use is that the null spaces of $(A-I)$ and $(A-I)^2$ be equal. This is equivalent to the condition that $(A-I)$ is fully degenerate i.e., the number of independent eigenvectors V_i such that $(A-I)V_i = 0$ equals the multiplicity of the zero eigenvalue of $(A-I)$. The proof of this equivalence has been omitted for the sake of brevity.

Again we assume that $\Delta x_0 \neq 0$ for if it were, we would already be at an equilibrium point. We have

$$\Delta X_0 = \left(\Delta x_0 \, \Delta x_1 \cdots \Delta x_{n-1}\right)$$

$$= (A-I)\left[\left(x_0 - x_e\right)\left(x_1 - x_e\right) \cdots \left(x_{n-1} - x_e\right)\right] . \tag{44}$$

Then

$$|A-I| = 0 \Rightarrow |\Delta X_0| = 0 .$$

Let $V_1, V_2, \cdots V_k$ be k independent eigenvectors of $(A-I)$ associated with the zero eigenvalue of $(A-I)$, where k is the maximum number of independent eigenvectors. Then

$$(A-I)V_i = 0 \qquad i = 1, 2, \cdots k . \tag{45}$$

These k eigenvectors thus span the null space of $(A-I)$.

Let r equal the rank of ΔX_0 where $r < n$. Then the first r columns of ΔX_0 are linearly independent as proved previously. The $r + 1$ column can then be expressed as a linear combination of the first r columns, or

$$\Delta x_r = c_0 \Delta x_0 + c_1 \Delta x_1 + \cdots + c_{r-1} \Delta x_{r-1} . \tag{46}$$

Using equation 21, we obtain

$$(A-I)\left(x_r - x_e\right) = (A-I)\left[c_0\left(x_0 - x_e\right) + c_1\left(x_1 - x_e\right) + \cdots + c_{r-1}\left(x_{r-1} - x_e\right)\right] . \tag{47}$$

Let V be some vector, as yet undefined, in the null space of $(A-I)$. Then V can be written as a linear combination of the k eigenvectors V_i, or

$$V = b_1 V_1 + b_2 V_2 + \cdots + b_k V_k . \tag{48}$$

Then

$$(A-I)\,V = 0 . \tag{49}$$

Using (49) in (47), we obtain

$$(A-I)\left[c_0\left(x_0 - x_e - V\right) + c_1\left(x_1 - x_e - V\right) + \cdots + c_{r-1}\left(x_{r-1} - x_e - V\right) - \left(x_r - x_e - V\right)\right] = 0 . \tag{50}$$

Each vector $\left(x_i - x_e\right)$ can be divided into two components, one orthogonal to the null space of $(A-I)$ and the other in the null space of $(A-I)$. Denoting these components

by x'_{ia} and x'_{ib} respectively, we have

$$x_i - x_e \triangleq x'_{ia} + x'_{ib} \qquad i = 0, 1, \cdots r \ . \tag{51}$$

By definition,

$$(A - I)\, x'_{ib} = 0 \ . \tag{52}$$

Substituting equation 51 in equation 50, we obtain

$$(A - I) \Big[c_0 x'_{0a} + c_1 x'_{1a} + \cdots + c_{r-1} x'_{r-1,a} - x'_{ra} + c_0 \left(x'_{0b} - V \right)$$

$$+ \cdots + c_{r-1} \left(x'_{r-1,b} - V \right) - \left(x'_{rb} - V \right) \Big] = 0 \ . \tag{53}$$

We can select V by proper choice of the constants b_i of equation 48, so that

$$V = \frac{c_0 x'_{0b} + c_1 x'_{1b} + \cdots + c_{r-1} x'_{r-1,b} - x'_{rb}}{\left(\sum_{i=0}^{r-1} c_i \right) - 1} \ . \tag{54}$$

It can be shown that

$$\left(\sum_{i=0}^{r-1} c_i \right) \neq 1 \ ,$$

provided that the null spaces of $(A - I)$ and $(A - I)^2$ are equal. The proof has been omitted here for the sake of brevity.

Using (54) in (53) we obtain

$$(A - I)\left[c_0 x'_{0a} + c_1 x'_{1a} + \cdots + c_{r-1} x'_{r-1,a} - x'_{ra}\right] = 0 . \tag{55}$$

Let

$$z \triangleq c_0 x'_{0a} + c_1 x'_{1a} + \cdots + c_{r-1} x'_{r-1,a} - x'_{ra} . \tag{56}$$

Then $(A-I)\, z = 0$.

We can easily prove that $z = 0$.

Proof: Assume $z \neq 0$. Then z is orthogonal to the null space of $(A-I)$ since each of its components is orthogonal to this null space. This implies $(A-\bar{1})\, z \neq 0$ which is a contradiction. Thus

$$z = c_0 x'_{0a} + c_1 x'_{1a} + c_{r-1} x'_{r-1,a} - x'_{ra} = 0 . \tag{57}$$

Equation 54 implies that

$$c_0 \left(x'_{0b} - V\right) + c_1 \left(x'_{1b} - V\right) + \cdots + c_{r-1} \left(x'_{r-1,b} - V\right) - \left(x'_{rb} - V\right) = 0 . \tag{58}$$

Adding (57) and (58), and using the identity of (51), we obtain

$$c_0 \left(x_0 - x_e - V\right) + c_1 \left(x_1 - x_e - V\right) + \cdots + c_{r-1} \left(x_{r-1} - x_e - V\right) - \left(x_r - x_e - V\right) = 0 . \tag{59}$$

We note that the point $\left(x_e + V\right)$ is an equilibrium point, which we will denote by x_e', since

$$A\left(x_e' - x_e\right) = AV = V = \left(x_e' - x_e\right) . \tag{60}$$

We can solve for $x_e' \triangleq x_e + V$ from equation 59 to obtain

$$x_e' = \frac{c_0 x_0 + c_1 x_1 + \cdots + c_{r-1} x_{r-1} - x_r}{\left(\sum_{i=0}^{r-1} c_i\right) - 1} \quad . \tag{61}$$

Equation 61 indicates that we can obtain an approximation of an equilibrium point x_e', which in general is not the same as x_e, when $(A-I)$ is singular. Of course this is the best that any iteration scheme can do. If the system were truly linear with $|A-I| = 0$, then the next set of iterates starting at x_e' would remain stationary at x_e'. The question arises, in a nonlinear system with an isolated equilibrium point, can the A matrix corresponding to a set of iterates be such that $|A-I| = 0$? We know that this can be true at the isolated equilibrium point itself, but of course this is a trivial case since this would mean we were already at the equilibrium point. Without answering the question (which may be impossible to answer), we can at least say that the matrix $(A-I)$ can become near singular, as we get closer to an equilibrium point whose $(A-I)$ matrix is singular. In practical computations the difference between a singular matrix and near-singular matrix can be academic since the numerical errors associated with inverting a near singular matrix can cause large discrepancies. It is likely that treating the matrix as singular, when $-\epsilon < |A-I| < \epsilon$, will yield more convergent results in many cases. Thus equation 61 can serve as the algorithm for the next estimate of the equilibrium point, when $(A-I)$ is singular or near singular, or

$$x_0^{(1)} = \frac{c_0 x_0 + c_1 x_1 + \cdots + c_{r-1} x_{r-1} - x_r}{\left(\sum_{i=0}^{r-1} c_i\right) - 1} \quad . \tag{62}$$

This is identical with (38), the algorithm for obtaining an estimate of x_e when $|\Delta X_0| = 0$ and $|A-I| \neq 0$. Thus without knowing whether or not $|A-I| = 0$ we can use the same algorithm for determining $x_0^{(1)}$ when ΔX_0 is singular or near singular. The c_i in (62) are derived from the same equation that yielded the c_i for (38). Hence

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{r-1} \end{bmatrix} = G_r^{-1} \begin{bmatrix} \Delta x_0 \cdots \Delta x_r \\ \Delta x_1 \cdots \Delta x_{r_1} \\ \vdots \\ \Delta x_{r-1} \cdots \Delta x_r \end{bmatrix} \tag{63}$$

We have thus established that the algorithm of (38) and (40) (or (62) and (63)) is valid for the case of $|\Delta X_0| = 0$ regardless of whether or not $|A-I| = 0$, provided that if $|A-I| = 0$ the null spaces of $(A-I)$ and $(A-I)^2$ are equal. We will show that the algorithm can also be used effectively in the case of near singularity of ΔX_0 in the sense that the c_i obtained from (40) (or (63)) will minimize the norm of y, where

$$y \triangleq c_0 \Delta x_0 + c_1 \Delta x_1 + \cdots + c_{r-1} \Delta x_{r-1} - \Delta x_r \ . \tag{64}$$

If ΔX_0 were truly singular and of rank r, then the selection of c_i as per equation 40 (or 63) would make y identically zero. However assume that ΔX_0 is nearly singular and that it is desired to treat ΔX_0 as though it were of rank r where r is chosen to be the smallest integer satisfying a predetermined condition,

$$-\epsilon < \left|G_{r+1}\right| < \epsilon \qquad \epsilon > 0 \ .$$

In other words, r is chosen as the smallest integer such that the r + 1 column can "almost" be expressed as a linear combination of the first r columns. We would like the difference between the selected linear combination and the r + 1 column, Δx_r, to

be as "small" as possible. This difference vector is y. We will consider the choice of c_i to be optimum if $\|y\|$ is minimized where

$$\|y\| \triangleq \sqrt{\sum_{i=1}^{n} y_i^2} \,. \tag{65}$$

We will now derive a formula for the c_i that minimizes $\|y\|$ (or $\|y\|^2$).

$$\|y\|^2 = y \cdot y = \left(c_0 \Delta x_0 + c_1 \Delta x_1 + \cdots + c_{r-1} \Delta x_{r-1} - \Delta x_r\right)$$

$$\cdot \left(c_0 \Delta x_0 + c_1 \Delta x_1 + \cdots + c_{r-1} \Delta x_{r-1} - \Delta x_r\right) .$$

Setting

$$\frac{\partial \|y\|^2}{\partial c_i} = 0 ,$$

$i = 0, 1, \cdots, r - 1$, we obtain r equations of the form,

$$0 = 2\left(c_0 \Delta x_0 + c_1 \Delta x_1 + \cdots + c_{r-1} \Delta x_{r-1} - \Delta x_r\right) \cdot \Delta x_i \,. \tag{66}$$

This yields the matrix equation

$$\begin{bmatrix} \Delta x_0 \cdot \Delta x_0 & \Delta x_0 \cdot \Delta x_1 & \cdots & \Delta x_0 \cdot \Delta x_{r-1} \\ \Delta x_1 \cdot \Delta x_0 & \Delta x_1 \cdot \Delta x_1 & \cdots & \Delta x_1 \cdot \Delta x_{r-1} \\ & & & \\ \Delta x_{r-1} \cdot \Delta x_0 & \Delta x_{r-1} \cdot \Delta x_1 & \cdots & \Delta x_{r-1} \cdot \Delta x_{r-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_{r-1} \end{bmatrix} = \begin{bmatrix} \Delta x_0 \cdot \Delta x_r \\ \Delta x_1 \cdot \Delta x_r \\ \\ \Delta x_{r-1} \cdot \Delta x_r \end{bmatrix} \tag{67}$$

or

$$\begin{bmatrix} c_0 \\ c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_{r-1} \end{bmatrix} = G^{-1} \begin{bmatrix} \Delta x_0 \cdot \Delta x_r \\ \Delta x_1 \cdot \Delta x_r \\ \\ \\ \Delta x_{r-1} \cdot\cdot \Delta x_r \end{bmatrix} . \tag{68}$$

This is identical to (40) (or (63)) so that selection of the c_i in accordance with the algorithm will minimize $\|y\|$ and thus yield a "best" fit in the case where ΔX_0 is nearly singular. In the above derivation, it was not proven that $\|y\|$ is a minimum rather than a maximum. However the fact that $\|y\| \to 0$ as $|G_{r+1}| \to 0$ indicates that $\|y\|$ must be a minimum for the c_i chosen as per (68).

EXAMPLE

To illustrate some of the points that have been discussed, a simple example will be presented in which the initial starting point x_0 results in singular matrices ΔX_0 and $(\Delta X_1 - \Delta X_0)$. Obviously, Steffensen's iteration would break down in this instance. Let

$$x = y + x^2 y \quad \text{and} \quad y = x + xy^2 . \tag{69}$$

This obviously has an equilibrium point at (0, 0). However, we will pretend that we only have some approximation which happens to be (0.1, 0.1). Forming n + 2 iterates from (69), we obtain

$$x_0 = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} \quad x_1 = \begin{bmatrix} 0.101 \\ 0.101 \end{bmatrix} \quad x_2 = \begin{bmatrix} .102030 \\ .102030 \end{bmatrix} \quad x_3 = \begin{bmatrix} .103092 \\ .103092 \end{bmatrix} \tag{70}$$

$$\Delta x_0 = \begin{bmatrix} 0.001 \\ 0.001 \end{bmatrix} \quad \Delta x_1 = \begin{bmatrix} .001030 \\ .001030 \end{bmatrix} \quad \Delta x_2 = \begin{bmatrix} .001062 \\ .001062 \end{bmatrix} . \tag{71}$$

We can see that ΔX_0 and $(\Delta X_1 - \Delta X_0)$ are singular. Using the algorithm of (38) and (40), we obtain

$$c_0 = 1.03 \quad \text{and} \quad x_0^{(1)} = \begin{bmatrix} .066667 \\ .066667 \end{bmatrix} . \tag{72}$$

Using $x_0^{(1)}$ as the starting point for a new set of iterates, we obtain

$$x_0 = \begin{bmatrix} .066667 \\ .066667 \end{bmatrix} \quad x_1 = \begin{bmatrix} .066963 \\ .066963 \end{bmatrix} \quad x_2 = \begin{bmatrix} .067263 \\ .067263 \end{bmatrix} \tag{73}$$

$$\Delta x_0 = \begin{bmatrix} .000296 \\ .000295 \end{bmatrix} \quad \Delta x_1 = \begin{bmatrix} .000300 \\ .000300 \end{bmatrix} . \tag{74}$$

It is not necessary to go beyond x_2 since we recognize that ΔX_0 is singular. Using (38) and (40) once more, we obtain

$$c_0 = 1.01351 \quad \text{and} \quad x_0^{(2)} = \begin{bmatrix} .044757 \\ .044757 \end{bmatrix} . \tag{75}$$

Repeating will show that the sequence $x_0^{(1)}, x_0^{(2)}, \cdots$ converges to (0, 0). The relatively slow convergence is due to the fact that in our example, A has an eigenvalue that approaches one as we approach the equilibrium point. Any initial condition of the form (a, a) is along the eigenvector corresponding to this eigenvector, resulting in a singular ΔX_0.

This example also has an eigenvalue that is approximately equal to minus one near the equilibrium point. If we select an initial condition of the form (a, -a), which is along the eigenvector corresponding to this eigenvalue, ΔX_0 will still be singular but the convergence of the algorithm will be much faster. For example, starting at

(0.1, -0.1) yields

$$x_0 = \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix} \quad x_1 = \begin{bmatrix} -.101 \\ .101 \end{bmatrix} \quad x_2 = \begin{bmatrix} .102030 \\ -.102030 \end{bmatrix} \tag{76}$$

$$\Delta x_0 = \begin{bmatrix} -.201 \\ .201 \end{bmatrix} \quad \Delta x_1 = \begin{bmatrix} .203030 \\ -.203030 \end{bmatrix} . \tag{77}$$

The algorithm of (38) and (40) yields

$$c_0 = -1.01001 \quad \text{and} \quad x_0^{(1)} = \begin{bmatrix} .0000005 \\ -.0000005 \end{bmatrix} \tag{78}$$

which already is extremely close to the equilibrium point.

CONCLUSIONS

The algorithm of (38) and (40) is a general method for obtaining the equilibrium points of the vector equation $x_{k+1} = f(x_k)$. When the matrix $(\Delta X_1 - \Delta X_0)$ is non-singular, the algorithm yields results identical to Steffensen's iteration with about the same amount of computations. The algorithm has one advantage in this case in that it enables the determination of the A matrix, when that is desired, with fewer additional calculations than does Steffensen's iteration. When the matrix $(\Delta X_1 - \Delta X_0)$ is singular, the algorithm yields meaningful results whereas Steffensen's iteration breaks down in this case. Finally when the matrix $(\Delta X - \Delta X_0)$ is nearly singular, (or equivalently when ΔX_0 is nearly singular), the algorithm yields good results by treating ΔX_0 as singular. This avoids the numerical errors in inverting a near singular matrix. Thus the algorithm developed in this paper is a more comprehensive method than Steffensen's iteration and includes the latter as a special case.

REFERENCES

1. T. Saaty and J. Bram, Nonlinear Mathematics, New York: McGraw Hill, 1964, pp. 53-70.

2. M. Urabe, Nonlinear Autonomous Oscillations, New York: Academic Press, 1967, pp. 280-302.

3. P. Henrici, Elements of Numerical Analysis, New York: John Wiley, 1964, pp. 115-118.