A QUADRATICALLY CONVERGENT NEWTON-LIKE METHOD BASED UPON GAUSSIAN-ELIMINATION

Kenneth M. Brown

Technical Report

No. 68-23

August 1968

Revised June 1969*

Department of Computer Science
Cornell University
Ithaca, New York 14850

* Accepted for publication in the SIAM Journal of Numerical Analysis

A QUADRATICALLY CONVERGENT NEWTON-LIKE METHOD BASED UPON GAUSSIAN ELIMINATION*

Kenneth M. Brown†

1. Introduction. Let a real-valued twice continuously differentiable system of N nonlinear equations in N unknowns be given as

$$
\begin{aligned}
& f_1(\underset{\sim}{x}) = f_1(x_1, x_2, \ldots, x_N) = 0 \\
& f_2(\underset{\sim}{x}) = f_2(x_1, x_2, \ldots, x_N) = 0 \\
(1.1)\qquad & \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
& f_N(\underset{\sim}{x}) = f_N(x_1, x_2, \ldots, x_N) = 0
\end{aligned}
$$

or in vector notation as

$$(1.2) \qquad \underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0} .$$

In this paper we present an iterative method for the numerical solution of (1.1). The method is a variation of Newton's Method incorporating Gaussian elimination in such a way that the most recent information is always used at

---

* A preliminary report of the method treated in this paper, entitled The Solution of Simultaneous Nonlinear Equations, was presented jointly with Professor Samuel D. Conte to the Association for Computing Machinery National Conference, August 29, 1967. This work constitutes a portion of the author's Doctoral Thesis (Department of Computer Sciences, Division of Mathematical Sciences, Purdue University, Lafayette, Indiana, August, 1966) directed by Professor Samuel D. Conte.

† Department of Computer Science, Cornell University. Ithaca, New York 14350.

each step of the algorithm. After specifying the method in terms of an iteration function, we prove that the iteration converges locally and that the convergence is quadratic in nature. Computer results are given and a comparison is made with Newton's Method; these results illustrate the effectiveness of the method for nonlinear systems containing linear or mildly nonlinear equations.

2. Notation. We shall introduce most of the notation as needed; however some comments concerning the symbols for partial differentiation are in order here. If we are given a function $G(u,v)$ in which

$$u = u(x,y)$$
$$v = v(x,y) \quad ,$$

then we adopt the following conventions:

$$(2.1)\qquad \begin{aligned} G_x &\equiv \frac{\partial G}{\partial x} = G_u \frac{\partial u}{\partial x} + G_v \frac{\partial v}{\partial x} \equiv G_u u_x + G_v v_x \\ G_y &\equiv \frac{\partial G}{\partial y} = G_u \frac{\partial u}{\partial y} + G_v \frac{\partial v}{\partial y} \equiv G_u u_y + G_v v_y \quad , \end{aligned}$$

whereas

$$(2.2)\qquad \begin{aligned} G_1 &\equiv G_u \\ G_2 &\equiv G_v \quad ; \end{aligned}$$

thus if we let f_i denote the $\underline{i}$th function of the system (1.1),

$f_{i,x_j} \equiv f_{ix_j} \equiv \frac{\partial f_i}{\partial x_j}$ is meant in the sense of (2.1)

and $f_{ij} \equiv f_{i,j} \equiv (f_i)_j$ is meant in the sense of (2.2).

3. Newton's Method. Let $\underset{\sim}{x}^0 = (x_1^0,\ldots,x_N^0)$ be an initial approximation to a real solution $\underset{\sim}{r} = (r_1,\ldots,r_N)$ of (1.1) . The iteration for Newton's Method is given by

$$(3.1) \qquad \underset{\sim}{x}^{n+1} = \underset{\sim}{x}^n - [\underset{\sim}{J}(\underset{\sim}{f}^n)]^{-1}\underset{\sim}{f}^n, \qquad n = 0,\ 1,\ 2,\ldots$$

where $\underset{\sim}{J}(\underset{\sim}{f})$ is the Jacobian matrix $[\partial f_i/\partial x_j]$ and the superscript n means that all functions involved are to be evaluated at $\underset{\sim}{x} = \underset{\sim}{x}^n$. The following local convergence theorem is well-known for Newton's Method (see e.g. [3, p.45] and [5, pp. 105-108]).

THEOREM 3.1. If

(1) in a closed region R whose interior contains a root $\underset{\sim}{x} = \underset{\sim}{r}$ of (1.2), each f_i is twice continuously differentiable for $i = 1,\ldots,N$,

(2) $\underset{\sim}{J}(\underset{\sim}{f})$ is non-singular at $\underset{\sim}{x} = \underset{\sim}{r}$, and

(3) $\underset{\sim}{x}^0$ is chosen in R sufficiently close to $\underset{\sim}{x} = \underset{\sim}{r}$,

then the iteration (3.1) is convergent to $\underset{\sim}{r}$.

L.V. Kantorovich [6, p. 708] has given a non-local convergence theorem for Newton's Method which guarantees the existence of the solution $\underset{\sim}{r}$ from information available in a sphere about $\underset{\sim}{x}^0$.

4. Description of the Method. The algorithm which we develop is essentially a modified Newton's Method based on Gaussian elimination; we approximate the forward triangularization of the full Jacobian matrix by working with one row at a time, eliminating one variable for each row treated. We shall assume that conditions (1), (2) and (3) of Theorem 3.1 hold. Let $\underset{\sim}{x}^n$ denote an nth approximation to the root $\underset{\sim}{x} = \underset{\sim}{r}$ of (1.1). The method consists of applying the following steps:

Step 1. Expand $f_1(\underset{\sim}{x})$ in a Taylor series about the point $\underset{\sim}{x}^n$; retain only first order terms and thus obtain the linear approximation

$F_1(x) \doteq f_1(x^n) -$ [illegible]

$$f_1(\underset{\sim}{x}) \approx f_1(\underset{\sim}{x}^n) + f_{1x_1}(\underset{\sim}{x}^n)\,(x_1 - x_1^n) \tag{4.1}$$
$$+ f_{1x_2}(\underset{\sim}{x}^n)\,(x_2 - x_2^n) + \ldots + f_{1x_N}(\underset{\sim}{x}^n)\,(x_N - x_N^n)\ .$$

Equate the right-hand side of (4.1) to zero and solve for that variable, say x_N , whose corresponding partial derivative is largest in absolute value. If we assume the hypotheses of Theorem 3.1, such an explicit solution can always be carried out provided that $\underset{\sim}{x}^n$ is sufficiently close to $\underset{\sim}{r}$ since, then, $\underset{\sim}{J}(\underset{\sim}{f}^n)$ will be close to $\underset{\sim}{J}(\underset{\sim}{f})]_{\underset{\sim}{x}=\underset{\sim}{r}}$, a nonsingular matrix; this implies that at least one of the $f_{1x_j}(\underset{\sim}{x}^n)$ is different from zero. Thus we have,

$$(4.2)\quad x_N = x_N^n - \sum_{j=1}^{N-1} (f_{1x_j}^n \,/\, f_{1x_N}^n)\,(x_j - x_j^n) - f_1^n \,/\, f_{1x_N}^n \ ,$$

where $f_{1x_j}^n \equiv f_{1x_j}(\underset{\sim}{x}^n)$ and the constants $f_{1x_j}(\underset{\sim}{x}^n)/f_{1x_N}(x^n)$,

$j = 1,\ldots, N-1$, and $f_1(\underset{\sim}{x}^n)/f_{1x_N}(\underset{\sim}{x}^n)$ are saved for future use. For purposes of clarity later, we rename the left-hand side of (4.2) as b_N :

$$(4.3)\quad b_N(x_1,x_2,\ldots, x_{N-1}) = \text{right-hand side of (4.2)},$$

and define $b_N^n = b_N(x_1^n,x_2^n,\ldots, x_{N-1}^n)$. We note that

$$b_N^n = x_N^n - f_1^n \,/\, f_{1x_N}^n \ .$$

Step 2. Define the function g_2 of the $n - 1$ variables $x_1,\ldots, x_{N-1}$ by

$$(4.4)\quad g_2(x_1,\ldots, x_{N-1}) = f_2(x_1,\ldots, x_{N-1}, b_N(x_1,\ldots, x_{N-1})\ ) \ ,$$

and g_2^n by

$$g_2^n = f_2(x_1^n,\ldots, x_{N-1}^n, b_N^n) \ .$$

Expand g_2 in a Taylor series, this time about the point $(x_1^n,\ldots, x_{N-1}^n)$, linearize and solve for that variable, say x_{N-1} , whose corresponding partial derivative is largest in magnitude obtaining

(4.5) $x_{N-1} = x_{N-1}^n - \sum_{j=1}^{N-2} (g_{2x_j}^n / g_{2x_{N-1}}^n)(x_j - x_j^n) - g_2^n / g_{2x_{N-1}}^n$.

Here $g_{2x_j}^n$ is obtained by differentiation of (4.4) using the chain-rule and is given by

$$f_{2j}^n + f_{2N}^n \cdot \frac{\partial b_N}{\partial x_j}\Bigg]_{(x_1^n,\dots,\, x_{N-1}^n)} = f_{2j}^n + f_{2N}^n \cdot (-f_{1x_j}^n / f_{1x_N}^n)$$

Renaming the left-hand side of (4.5) as b_{N-1}, a function of the $N - 2$ remaining variables, we have

(4.6) $b_{N-1}(x_1, x_2, \dots, x_{N-2})$ = right-hand side of (4.5) .

Again the ratios formed,

$$g_{2x_j}^n / g_{2x_{N-1}}^n \qquad j = 1, \dots, N-2$$

and

$g_2^n / g_{2x_{N-1}}^n$, are saved for future use.

We shall show in Theorem 4.1 that under the hypotheses for Newton's Method this process is well-defined, i.e., that there actually exists at least one non-vanishing partial derivative at each stage of the process.

Step 3. Define

$$g_3 = f_3(x_1, \ldots, x_{N-2}, b_{N-1}, b_N)$$

where b_{N-1} and b_N are obtained by back-substitution in the linear system (4.3) and (4.6) and repeat the process of expansion, linearization and elimination of one variable, saving the ratios formed. We note that $g^n_{3x_j}$ is obtained by differentiating

$$f_3(\text{arg}, b_{N-1}(\text{arg}), b_N(\text{arg}, b_{N-1}(\text{arg})))$$

with respect to x_j at the point $(x^n_1, \ldots, x^n_{N-2})$; where

$$\text{arg} \equiv (x_1, \ldots, x_{N-2}) \ ; \text{ thus}$$

$$g^n_{3x_j} = f^n_{3j} + f^n_{3,N-1} \cdot (-g^n_{2x_j} / g^n_{2x_{N-1}})$$

$$+ \ f^n_{3N} \cdot [(-f^n_{1j}/f^n_{1N}) + (f^n_{1,N-1}/f^n_{1N}) \cdot (g^n_{2x_j}/g^n_{2x_{N-1}})] \ .$$

We continue in this manner replacing one variable at a time, each g_k being expanded about the point $(x^n_1, \ldots, x^n_{N-k+1})$.

Step N. At this stage we have

$$g_N = f_N(x_1, b_2, b_3, \ldots, b_N) \ ,$$

where the b_j's are obtained by back-substitution in the $N - 1$ rowed triangularized linear system built up (i.e., the extension of the system begun in (4.3) and (4.6)) which now has the form (omitting arguments);

$$(4.7)\quad b_i = x_i^n - \sum_{j=1}^{i-1} (g_{N-i+1,x_j} / g_{N-i+1,x_i})(b_j - x_j^n)$$

$$- g_{N-i+1} / g_{N-i+1,x_i},\quad i = N, N-1, \ldots, 2,$$

(with $g_1 = f_1$ and $b_1 = x_1$) so that g_N is a function of just x_1. Now expanding, linearizing and solving for x_1; we obtain

$$x_1 = x_1^n - g_N^n / g_{Nx_1}^n \quad .$$

We use the point x_1 thus obtained as the improved approximation x_1^{n+1} to the first component r_1 of the root $\underset{\sim}{r}$, call it b_1, and back-solve the b_j system (4.7) to get improved approximations to the other components r_j. Here x_j^{n+1} will equal the corresponding b_j obtained when back-solving. We note that the most recent information available is immediately used in the construction of the next function argument, similar to what is done in the Gauss-Seidel process for linear [3, pp. 194-203] and nonlinear [8, p. 503] systems of equations. <u>The algorithm outlined in section (4) will be called</u> "Algorithm (4)" <u>hereafter</u>.

REMARK. As the following example shows, Algorithm (4) is <u>not</u> mathematically equivalent to Newton's Method.

EXAMPLE. Consider the nonlinear system

$$f(x,y) = x^2 - 2y + 1 = 0$$

$$g(x,y) = x + 2y^2 - 3 = 0$$

which has a solution at $(1,1)^T$. For $\underset{\sim}{x}^o = (0,0)^T$ we obtain

$\underset{\sim}{x}^1 = (3,\ 0.5)^T$ from Newton's Method, whereas

$\underset{\sim}{x}^1 = (2.5,\ 0.5)^T$ from Algorithm (4).

MATRIX FORMULATION. Recall that for the sake of definiteness we eliminated the variables in the order $x_N, x_{N-1}, \ldots, x_2$. If we use the chain rule for differentiation to expand each derivative g_{ix_j} which appears in Algorithm (4), we obtain the following matrix representation for the forward part of the method:

$$\underset{\sim}{H} \cdot (\underset{\sim}{x}^{n+1} - \underset{\sim}{x}^n) = -\underset{\sim}{g}$$

where $H = (h_{ij})$ is given by

$$
(4.8)\quad h_{ij} = \begin{cases} f_{1j}\,, \quad j = 1, \ \ldots, \ N\,, \\[2ex] (-1)^{i+1} \dfrac{\begin{vmatrix} f_{1j} & f_{1,N-i+2} & \cdots & f_{1,N-1} & f_{1N} \\ f_{2j} & f_{2,N-i+2} & \cdots & f_{2,N-1} & f_{2N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f_{ij} & f_{i,N-i+2} & \cdots & f_{i,N-1} & f_{iN} \end{vmatrix}}{\begin{vmatrix} f_{1,N-i+2} & \cdots & f_{1N} \\ \cdots & \cdots & \cdots \\ f_{i-1,N-i+2} & \cdots & f_{i-1,N} \end{vmatrix}}\,, \quad \begin{matrix} i=2,\ldots,N, \\ j=1,\ldots,N, \end{matrix} \end{cases}
$$

and where the argument of each f_{ij} is the progressive argument generated successively in Algorithm (4).

We observe that $h_{ij} = 0$ for $j > N - i + 1$; i.e., $\underset{\sim}{H}$ is transverse upper triangular. More important, the h_{ij} have precisely the same form as the corresponding elements obtained by transverse triangularization of the Jacobian matrix $\underset{\sim}{J}$ using Gaussian elimination with partial pivoting. The argument of each f_{ij} in the triangularized form of $\underset{\sim}{J}$ is simply $\underset{\sim}{x}^n$; however, at the root, the two types of arguments coincide (see Lemma 6.1).

Let us illustrate the matrix formulation explicitly for the 3×3 case. Here

$$
H' = \begin{bmatrix}
f_{11} & f_{12} & f_{13} \\[2ex]
-\dfrac{\begin{vmatrix} f_{11} & f_{13} \\ f_{21} & f_{23} \end{vmatrix}}{f_{13}} & -\dfrac{\begin{vmatrix} f_{12} & f_{13} \\ f_{22} & f_{23} \end{vmatrix}}{f_{13}} & 0 \\[4ex]
\dfrac{\begin{vmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{vmatrix}}{\begin{vmatrix} f_{12} & f_{13} \\ f_{22} & f_{23} \end{vmatrix}} & 0 & 0
\end{bmatrix} ,
$$

and the arguments are given by

$$\underline{x}^n \quad \text{for} \quad f_{1j} \quad ,$$

$(x_1^n, x_2^n, b_3(x_1^n, x_2^n))$ for f_{2j} , and

$(x_1^n, b_2(x_1^n), b_3(x_1^n, b_2(x_1^n)))$ for f_{3j} where $j = 1,2,3$

THEOREM 4.1. Under the hypotheses for Newton's Method given in Theorem 3.1, there exists a non-vanishing partial derivative g_{ix_j} at the $\underline{i}$th step of the elimination process defined by Algorithm (4).

Proof. Let $\underset{\sim}{r}$ again denote a solution of (1.2). When the g_{ix_j} obtained at each step of Algorithm (4) are evalua at the point $(r_1, r_2, \ldots, r_{N-i+1})$, the resulting values are equal to the elements, say $T_{ij}(\underset{\sim}{r})$, which appear when triangularizing $\underset{\sim}{J}(\underset{\sim}{f})]_{\underset{\sim}{r}}$ relative to its minor diagonal by Gaussian elimination with pivoting. (This follows from our mode of construction of the g_{ix_j} , or can be proven formally using induction when comparing the $T_{ij}(\underset{\sim}{r})$ with (4.8) above. We assumed, however, $\underset{\sim}{J}(\underset{\sim}{f})]_{\underset{\sim}{r}}$ to be non-singular; hence no row of $[T_{ij}]$ may contain all zeros. The conclusion now follows from our assumptions of twice continuous differentiability and sufficient closeness to $\underset{\sim}{r}$, since a matrix which is sufficiently close to an invertible matrix is itself invertible.

5. The Iteration Function. We may formalize the process by writing down the method in terms of the iteration function $\underset{\sim}{F} = (F_1, \ldots, F_N)$ used, beginning with a starting guess $\underset{\sim}{x}^0$ to form successively

(5.1) $$\underset{\sim}{x}^{n+1} = \underset{\sim}{F}(\underset{\sim}{x}^n), \quad n = 0, 1, 2, \ldots \quad .$$

The iteration function, $\underset{\sim}{F}$, for Algorithm (4) is given by

(5.2) $$F_i(x_1, \ldots, x_N) = x_i - \sum_{j=1}^{i-1} (g_{N-i+1,x_j} / g_{N-i+1,x_i})(F_j - x_j) - g_{N-i+1} / g_{N-i+1,x_i}, \quad i = 1, 2, \ldots, N.$$

where as usual we define

$$\sum_{j=p}^{q} = 0 \text{ whenever } p > q \text{ and where}$$

$$g_1 = f_1(x_1, \ldots, x_N)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$g_i = f_i(x_1, x_2, \ldots, x_{N-i+1}, b_{N-i+2}, \ldots, b_N)$$

(5.3) $$\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$g_{N-i+1} = f_{N-i+1}(x_1, \ldots, x_i, b_{i+1}, b_{i+2}, \ldots, b_N)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$g_N = f_N(x_1, b_2, b_3, \ldots, b_N) .$$

The $b_{i+1}, b_{i+2}, \ldots, b_N$, $i = 1, \ldots, N-1$, are themselves functions of the x_j and are obtained recursively by successive substitution in the system

$$(5.4) \quad b_{k+1} = x_{k+1} - \sum_{j=i+1}^{k} (g_{N-k,x_j}/g_{N-k,x_{k+1}})(b_j - x_j) - g_{N-k} / g_{N-k,x_{k+1}}, \quad k = i, i+1, \ldots, N-1$$

For purposes of completeness, define $b_1 = x_1$.

THEOREM 5.1. Any fixed point $\underset{\sim}{x} = \underset{\sim}{r}$ of the iteration function $\underset{\sim}{F}$ defined by (5.2) - (5.4) is a solution of the original system $\underset{\sim}{f}(\underset{\sim}{x}) = 0$.

Proof. Since $\underset{\sim}{r} = \underset{\sim}{F}(\underset{\sim}{r})$, it follows from (5.2) that

$$(5.5) \quad g_{N-i+1}(r_1, \ldots, r_i) = 0 \text{ for } i = 1, 2, \ldots, N,$$

and from (5.3) that

$$(5.6) \quad f_{N-i+1}(r_1, \ldots, r_i, b_{i+1}, b_{i+2}, \ldots, b_N) = 0$$

for $i = 1, 2, \ldots, N$. As a consequence of (5.4), using (5.5), we have

$$b_{k+1} = r_{k+1}, \quad k = i, i+1, \ldots, N-1,$$

but this implies from (5.6) that

$$f_{N-i+1}(\underset{\sim}{r}) = 0, \quad i = 1, \ldots, N. \qquad \text{Q. E. D.}$$

6. Local Quadratic Convergence of the Method. Henrici gives the following result for iteration functions [5, p. 104].

THEOREM 6.1. Let the functions $F_1, \ldots, F_N$ be defined in a region R, and let them satisfy the following conditions:

(1) The first partial derivatives of $F_1, \ldots, F_N$ exist and are continuous in R.

(2) The system

$$\underset{\sim}{x} = \underset{\sim}{F}(\underset{\sim}{x})$$

has a solution $\underset{\sim}{r}$ in the interior of R such that $\underset{\sim}{J}(\underset{\sim}{F})]_{\underset{\sim}{r}} = \underset{\sim}{0}$, the zero matrix.

Then there exists a number $\varepsilon > 0$ such that algorithm (5.1) converges to $\underset{\sim}{r}$ for any choice of the starting point, $\underset{\sim}{x}^0$, which satisfies $||\underset{\sim}{r} - \underset{\sim}{x}^0|| < \varepsilon$ (where $|| \cdot ||$ denotes the Euclidean norm).

We now establish the local quadratic convergence of Algorithm (4) by showing that the Jacobian matrix of the iteration function $\underset{\sim}{F}$ defined by (5.2) - (5.4) is the zero matrix at the root $\underset{\sim}{x} = \underset{\sim}{r}$, and then appealing to Theorem 6.1. We remark that since the order of the elimination of the variables may change from step to step, the coordinatized representation of the iteration function will perhaps change from point to point in the iteration sequence. At the root $\underset{\sim}{x} = \underset{\sim}{r}$, we shall assume the coordinatized representation induced by eliminating the variables in the order $x_N, x_{N-1}, \ldots, x_2, x_1$.

DEFINITION 6.1. $\underset{\sim}{R}_{N-i+1} \equiv (r_1, \ldots, r_{N-i+1})$.

LEMMA 6.1. For $\underset{\sim}{x} = \underset{\sim}{r}$,

$$h_i = r_i , \quad \text{and}$$

$$g_i(\underset{\sim}{R}_{N-i+1}) = 0 , \qquad i = 1, \ldots, N .$$

Proof. The conclusions follow directly from (5.2) and (5.

LEMMA 6.2. If $h_i = x_i - (g_j/g_{jx_i})$ and $g_{jx_i}(\underset{\sim}{R}_{N-j+1}) \neq 0$ then $h_{ix_i}(\underset{\sim}{R}_{N-j+1}) = 0$.

Proof. With all functions evaluated at $\underset{\sim}{R}_{N-j+1}$,

$$h_{ix_i} = 1 - \frac{(g_{jx_i})^2 - g_j g_{jx_i,x_i}}{(g_{jx_i})^2}$$

and since, by Lemma 6.1, $g_j(\underset{\sim}{R}_{N-j+1}) = 0$, we obtain the required result.

LEMMA 6.3. For each $i = N-1, N-2, \ldots, 1$,

$$g_{N-i+1\,x_j}(\underset{\sim}{R}_i) = 0$$

whenever $j = i+1, i+2, \ldots, N$.

Proof. We must avoid the temptation to say that the result is true immediately since g_{N-i+1} seems to be

independent of the variables $x_{i+1}, x_{i+2}, \ldots, x_N$.
In reality, the b_j , $j = i+1, \ldots, N$, which appear as arguments of the function g_{N-i+1} do depend on $x_{i+1}, \ldots, x_N$. The proof is accomplished by an induction within an induction, first on the row subscripts taken in the order $N-1, N-2, \ldots, 1$ and then on the relevant column subscripts. Lemmas 6.1 and 6.2 are invoked repeatedly.

THEOREM 6.2. $\underset{\sim}{J}(\underset{\sim}{F})]_{\underset{\sim}{x}=\underset{\sim}{r}} = \underset{\sim}{0}$, the zero matrix.

Proof. Now $F_{1x_1}(\underset{\sim}{r}) = 0$ by Lemma 6.2;
moreover, for $j = 2, 3, \ldots, N$,

$$F_{1x_j}(\underset{\sim}{r}) = 0 - \frac{g_{Nx_j}(r_1)}{g_{Nx_1}(r_1)} = 0 \text{ by Lemma 6.3.}$$

Now assume that for each $i = 1, 2, \ldots, k-1$ $(2 \leq k \leq N)$ we have shown

(6.1) $F_{ix_j}(\underset{\sim}{r}) = 0$

for all $j = 1, \ldots, N$. We now show

$$F_{kx_j}(\underset{\sim}{r}) = 0 \quad \text{for} \quad j = 1, 2, \ldots, N .$$

Once accomplished, repeating the argument $N-1$ times will establish the required result. From (5.3) with $i = k$, we first obtain by using (6.1) and Lemma 6.1

$$F_{kx_m}(\underset{\sim}{r}) = 0 - \frac{g_{N-k+1,x_m}(\underset{\sim}{R}_k)}{g_{N-k+1,x_k}(\underset{\sim}{R}_k)} \cdot (0 - 1)$$

$$- \frac{g_{N-k+1,x_m}(\underset{\sim}{R}_k)}{g_{N-k+1,x_k}(\underset{\sim}{R}_k)} = 0 ,$$

for all $m = 1, 2, \ldots, k-1$; moreover

$$F_{kx_k}(\underset{\sim}{r}) = 1 - 0 - \frac{g^2_{N-k+1,x_k}(\underset{\sim}{R}_k)}{g^2_{N-k+1,\ x_k}(\underset{\sim}{R}_k)} = 0 .$$

Finally for $m = k+1, k+2, \ldots, N$:

$$F_{kx_m}(\underset{\sim}{r}) = 0 - 0 - \frac{g_{N-k+1,x_m}(\underset{\sim}{R}_k)}{g_{N-k+1,x_k}(\underset{\sim}{R}_k)} = 0 \text{ by Lemma 6.3.}$$

Thus the theorem is established.

7. Numerical Results. The structure of Algorithm (4) suggests that it should be best suited to nonlinear systems in which the initial equations are nearly linear; i.e., in Algorithm (4), information generated from the first equations of the system enters into computations performed with the remaining equations. This contrasts sharply with Newton's Method in which all equations are treated simultaneously.

EXAMPLE 7.1. We consider an extreme case of the situation presented in the previous paragraph, namely a system in which all but the last equation are linear:

$$\begin{cases} f_i(\mathbf{x}) = -(N+1) + 2x_i + \sum\limits_{\substack{j=1 \\ j \neq i}}^{N} x_j , & i = 1, \ldots, N-1 , \\ f_N(\mathbf{x}) = -1 + \prod\limits_{j=1}^{N} x_j . & \end{cases}$$

In the computer results given in Table 1 for several values of N, the starting guess used for all cases was $\mathbf{0.5}$ (a vector of length N each component of which is 0.5). Algorithm (4) converged to the root $\mathbf{1}$ in each instance and for $N=5$ Newton's Method converged to the root given approximately by $(-0.579, -0.579, -0.579, -0.579, 8.90)^T$. We observe that the Jacobian matrix of the system is nonsingular at the starting guess and at the two roots. In the table "diverged" means that $\|\mathbf{x}^n\|_\infty \to \infty$ whereas "converged" means that each component of $\mathbf{x}^{n+1}$ agreed with the corresponding component of $\mathbf{x}^n$ to 15 significant digits <u>and</u> $\|\mathbf{f}(\mathbf{x}^{n+1})\|_{\ell_2} < 10^{-15}$.

TABLE 1

Computer Results for Example 7.1

N	Newton's Method	Algorithm (4)
5	converged in 18 iterations	converged in 6 iterations
10	diverged, $\|\|\underset{\sim}{x}^1\|\|_{\ell_2} \sim 10^3$	converged in 7 iteration
15	diverged, $\|\|\underset{\sim}{x}^1\|\|_{\ell_2} \sim 10^5$	converged in 8 iteration
20	diverged, $\|\|\underset{\sim}{x}^1\|\|_{\ell_2} \sim 10^6$	converged in 8 iteration

EXAMPLE 7.2. In the following system, $f_1(x_1,x_2)$ is approximately linear near the roots:

$$f_1(x_1,x_2) = x_1^2 - x_2 - 1$$

$$f_2(x_1,x_2) = (x_1 - 2)^2 + (x_2 - 0.5)^2 - 1 \ .$$

The system has roots at

$$\underset{\sim}{r}_1 \doteq (1.54634288,\ 1.39117631)^T \ , \text{ and}$$

$$\underset{\sim}{r}_2 \doteq (1.06734609,\ 0.139227667)^T \ .$$

The starting guess used was $(0.1,\ 2.0)^T$.

TABLE 2

Computer Results for Example 7.2

Method	Result
Newton	converged to r_2 in 24 iterations
Algorithm (4)	converged to r_2 in 10 iterations

EXAMPLE 7.3. The following system was studied first by Freudenstein and Roth [4] and later by Broyden [2]:

$$f_1(x_1,x_2) = -13 + x_1 + ((-x_2 + 5)\, x_2 - 2)\, x_2$$

$$f_2(x_1,x_2) = -29 + x_1 + ((x_2 + 1)\, x_2 - 14)\, x_2 \quad .$$

This system has a solution at $(5,4)^T$. In the computer results summarized in Table 3, the starting guess used in each case was $(15, -2)^T$.

TABLE 3

Computer Results for Example 7.3

Method	Result
Newton	converged in 42 iterations
Broyden's I [2,p.591]	diverged
Broyden's II [2,p.591]	diverged
Damped Newton (discrete form) [9]	diverged
Algorithm (4)	converged in 10 iterations

REMARK. In [1] we have given a description of the discretized form of Algorithm (4) in which the analytic partial derivatives are approximated by first difference quotients. This discretized version of the method requires only $(N^2/2 + 3N/2)$ function evaluations per iterative step as compared with $(N^2 + N)$ evaluations for the discretized Newton's Method. Experimental evidence shows a quadratic type of convergence behavior for this discretized form of Algorithm (4), but rigorous convergence results are yet to be obtained. In the latter connection, recent work by Ortega and Rheinboldt [7] appears extremely useful.

Acknowledgements. The author wishes to thank Professor Samuel D. Conte, his thesis advisor, for his abundant encouragement and support, and, in particular, for many helpful discussions relative to this research. The author also wishes to thank Professor Peter Henrici who guided and stimulated the author's early work in numerical analysis and who proposed research along the present lines. Finally, the author wishes to thank Mr. D. Jordan, Mr. R. Tenney, and Mr. R. Weber for assistance with the numerical experiments.

REFERENCES

1. K. M. Brown, "Solution of Simultaneous Non-linear Equations," Comm. Assoc. Comput. Mach., Vol. 10 (1967) pp. 728-729.

2. C.G. Broyden, "A Class of Methods for Solving Nonlinear Simultaneous Equations," Mathematics of Computation, Vol. 19 (1965), pp. 577-593.

3. S. D. Conte, Elementary Numerical Analysis, McGraw-Hill, New York, 1965.

4. F. Freudenstein and B. Roth, "Numerical Solutions of Systems of Nonlinear Equations," J. Assoc. Comput. Mach., Vol. 10 (1963), pp. 550-556.

5. P. K. Henrici, Elements of Numerical Analysis, Wiley, New York, 1964.

6. L. V. Kantorovich and G. P. Akilov, Functional Analysis in Normed Spaces, Pergamon, Oxford, 1964.

7. J. M. Ortega and W. C. Rheinboldt, Iterative Methods for Nonlinear Operator Equations, Blaisdell, Walthem, (to appear).

8. J. M. Ortega and M. L. Rockoff, "Nonlinear Difference Equations and Gauss-Siedel Type Iterative Methods," this Journal, 3 (1966), pp. 497-513.

9. H. Späth, "The Damped Taylor's Series Method for Minimizing a Sum of Squares and for Solving Systems of Nonlinear Equations," Comm. Assoc. Comput. Mach., Vol. 10 (1967), pp. 726-728.