Publ. RIMS, Kyoto Univ.
19 (1983), 367-395

Hybrid Manipulations for the Solution of Systems of Nonlinear Algebraic Equations

By

Satoshi WATANABE*

§ 1. Introduction

The solution of systems of nonlinear algebraic equations plays an important role in the analysis of practical problems arising in mathematics, physics or engineerings.

Many methods by the numerical manipulation languages such as FORTRAN have been proposed so far for finding the numerical solution of a system of nonlinear algebraic equations given by

$$f(x)=0\,, \tag{1.1}$$

where x and $f(x)$ are real n-dimensional vectors, and $f(x)$ is defined to be twice continuously differentiable on a bounded region D in $\mathbb{R}^n$. However the numerical manipulation languages can not handle mathematical operations such as formal differentiations, substitutions or symbolic calculations of determinants. Only very recently, by practical applications of the symbolic and algebraic manipulation languages such as REDUCE 2 [7], it is possible for the computers to treat these mathematical operations.

If we introduce both numerical and symbolic manipulations, termed here as *hybrid manipulations*, new and efficient algorithms can be developed which greatly improve the feasibility of solving the system of nonlinear algebraic equations. For the purpose the author et al. are developing the package **NAES** (*Nonlinear Algebraic Equation's Solver*) [14, 18, 28]. This paper reports on several implementations which incorporate newly developed algorithms into the package.

Communicated by S. Hitotumatu, March 28, 1981.

* Department of Mathematics, Tohoku Dental University, Koriyama, Fukushima, Japan.

In Section 2, a numerical method termed here as *ε-secant method* which is a numerical realization of the Newton-Raphson method and was treated by S̄amanskii [20, 23, 24] is discussed first and its quadratic convergence property to a simple root are given [11-13, 17, 27, 29, 30]. Secondly we show that under appropriate conditions the inverse matrix and determinant of the Jacobian matrix of the system also have quadratic convergence properties.

The convergence of the Newton-Raphson and present ε-secant methods is proved on the assumption that the inverse of the Jacobian matrix is nonsingular in the neighborhood of the solution x^*. However we often encounter the appearance of singular Jacobian matrices on some iteration processes by the Newton-Raphson or ε-secant methods. A property in the neighborhood of such a point (the *singular point*) is also given in this section.

In Section 3 we shall consider the question of the realization of the ε-secant process where the rank of the Jacobian matrix in the neighborhood of the solution x^* is essentially degenerated. This solution is known as the *multiple root*. Several important properties of the multiple root are presented first. Then an algorithm, termed here as the *deflation algorithm*, is proposed for finding the multiple root with a sufficient accuracy [19, 28]. Lastly four categories for a computational realization of the algorithm are distinguished in which both numerical and symbolic manipulations play important roles respectively.

In the package NAES the system of nonlinear algebraic equations (1. 1) is given in FORTRAN expressions. However the expressions of equations in FORTRAN and REDUCE 2 are different each other, and hence the equations in the FORTRAN form must be converted into the REDUCE 2 expressions when some mathematical operations are necessary, and vice versa. In Section 4 the outlines of these processes are explained by showing examples. In addition several procedures in REDUCE 2 which are used in the NAES are given. To show the effectiveness of the present methods, an illustrative example of three dimensional nonlinear algebraic equations with a quadruple root which was given in S̄amanskii [25] is solved by NAES [19].

In the following, for an n-dimensional vector x and $n\times n$ matrix A, we use the norms defined by

$$\|x\| = \max_{1\leq i\leq n}|x_i|, \quad \|\mathrm{A}\| = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}| \ ,$$

respectively.

§ 2. ε-Secant Algorithm for Simple Roots

2.1. Newton-Raphson Method

We state first of all the algorithm of Newton-Raphson method. Let the system of equations be

$$f(x) = 0 \,, \tag{2.1}$$

and x be an approximation for the root x^*. Substitute the new iteration $\bar{x}$ in (2.1) and expand it in a truncated Taylor series about the old iteration x:

$$f(x) + S(x)\,(\bar{x} - x) = 0 \,, \tag{2.2}$$

where

$$S(x) = \left(\frac{\partial f(x)}{\partial x}\right) \tag{2.3}$$

is the Jacobian matrix of the system. Suppose now that $S(x)$ is non-singular, i.e., det $[S(x)] \neq 0$. Let ${}^0x$ be the starting value for the system (2.1) and ${}^kx$ the k-th iteration. Then the new approximation ${}^{k+1}x$ is given by

$${}^{k+1}x = {}^kx - [S({}^kx)]^{-1}f({}^kx), \quad k = 0, 1, 2, \cdots . \tag{2.4}$$

The sufficient conditions for the convergence of the Newton-Raphson method are given in the following.

Theorem 2.1 (*Kantorovich's theorem*). *Assume that there exist positive constants $\overline{B}_0$, $\overline{\pi}_0$, M and $\overline{K}$ such that*

(i) *for the initial approximation ${}^0x$ in* D, *the Jacobian matrix $S({}^0x)$ has an inverse*

$$\|{}^0\Gamma\| \equiv \|S^{-1}({}^0x)\| \leq \overline{B}_0 \,, \tag{2.5}$$

(ii) *for ${}^0x$, the system* (2.1) *satisfies the following relation*

$$\|f({}^0x)\| \leq \bar{\pi}_0 , \tag{2.6}$$

(iii) *for x and y in the region* D, *the following inequality is satisfied*

$$\|S(x) - S(y)\| \leq M\|x-y\| , \tag{2.7}$$

(iv) *for the constant M and the tensor f_{xx} of the third order with components $\partial^2 f_i(x)/\partial x_j \partial x_k$, $(i, j, k = 1, 2, \cdots, n)$*

$$\bar{K} = \max[M, \|f_{xx}\|] , \tag{2.8}$$

(v) *the constants $\bar{B}_0$, $\bar{\pi}_0$, and $\bar{K}$ introduced above satisfy the inequality*

$$\bar{h}_0 \equiv \bar{K}\bar{B}_0^2\bar{\pi}_0 < \frac{1}{2} , \tag{2.9}$$

and the cube below is contained in D,

$$\|x - {}^0x\| \leq \frac{1-(1-2\bar{h}_0)^{1/2}}{\bar{h}_0}\bar{B}_0\bar{\pi}_0 . \tag{2.10}$$

Then the system of equations (2.1) *has a solution x^* in the cube* (2.10). *Moreover, the successive approximations ${}^{k+1}x$ defined by* (2.3) *exist and converge to x^*, and the rate of convergence can be estimated by the inequality*

$$\|{}^{k+1}x - x^*\| \leq \frac{(2\bar{h}_0)^{2k+1-1}}{2^k}\bar{B}_0\bar{\pi}_0 , \tag{2.11}$$

which shows that the order of convergence for the Newton-Raphson method is quadratic.

As for the *proof* of Theorem 2.1, see the next section, Kantorovich [9], Krasnosel'skii [10] and Henrici [8].

2.2. ε-Secant Method

Newton-Raphson method leads to a system of linear equations involving the Jacobian matrix as the coefficient matrix.

Instead of calculating the Jacobian matrix, we consider a perturbation technique with parameter ε which is assumed to be small enough, and some guess x for the solution x^*, and set

$$ {}_jy = x + \varepsilon e_j, \quad (j=1, 2, \cdots, n), \tag{2.12}$$

where ${}_jy$ denotes an n-dimensional vector and e_j is the j-th unit vector.

Let us define the vector ${}_jf(x)$ by

$$ {}_jf(x) = f({}_jy) = f(x+\varepsilon e_j), \tag{2.13}$$

and expanding it into a truncated Taylor expansion about x, we have

$$\begin{aligned} {}_jf(x) &= f(x+\varepsilon e_j) \\ &= f(x) + \varepsilon f_x(x) e_j + \frac{\varepsilon^2}{2} u_j(x), \end{aligned} \tag{2.14}$$

where

$$ f_x(x) = \frac{\partial f(x)}{\partial x}, $$

and

$$ u_j(x) = f_{xx}(x+\theta\varepsilon e_j)[e_j]^2, \quad (0<\theta<1). \tag{2.15}$$

An $n \times n$ matrix $U(x)$ is constructed from $u_j(x)$ as its j-th column vector. Using (2.14), let us define an $n \times n$ matrix $S(x;\varepsilon)$ whose j-th column vector $s_j(x;\varepsilon)$ is defined by

$$\begin{aligned} s_j(x;\varepsilon) &= (f(x+\varepsilon e_j) - f(x))/\varepsilon = ({}_jf(x) - f(x))/\varepsilon \\ &= s_j(x) + \frac{\varepsilon}{2} u_j(x), \quad (j=1, \cdots, n), \end{aligned} \tag{2.16}$$

where $s_j(x)$ is the j-th column vector of $S(x)$ given by (2.3). The matrix $S(x;\varepsilon)$ is called the *perturbed Jacobian matrix* for the ε-secant method.

Analogous to (2.4) for the Newton-Raphson method, let us now consider the following formula as the *ε-secant method:*

$$ S({}^kx; {}^k\varepsilon)[{}^{k+1}x - {}^kx] = -f({}^kx), \quad (k=0, 1, 2, \cdots). \tag{2.17}$$

For the perturbed Jacobian matrix $S({}^kx;{}^k\varepsilon)$, we have the following theorem.

Theorem 2.2. *Let $S({}^kx)$ and $S({}^kx;{}^k\varepsilon)$ be the $n \times n$ matrices defined by* (2.3) *and* (2.16), *respectively. Then*

$$\lim_{{}^k\varepsilon\to 0} S({}^kx;{}^k\varepsilon)=S({}^kx), \quad (k=0,1,2,\cdots). \tag{2.18}$$

As for the *proof*, see Ojika and Kasue [17; p. 366].

We now have a lemma of quadratic convergence property of the ε-secant method given by (2.17) using Kantorovich's theorem [8–10]. In the following, it is assumed that the hypotheses of Theorem 2.1 for the Newton-Raphson method are satisfied.

Lemma 2.1. *By analogy with Theorem* 2.1, *assume that there exist positive constants* $\bar{B}_0$, π_0, M *and* K *such that*

(i) *for the initial approximation* ${}^0x\in \mathrm{D}$, *the Jacobian matrix* $S({}^0x)$ *and the functions* $f({}^0x)$ *of equations given by* (2.3) *and* (2.1), *respectively, satisfy the following inequalities*:

$$\|S^{-1}({}^0x)\|\leq\bar{B}_0, \tag{2.19}$$

$$\|f({}^0x)\|\leq\bar{\pi}_0\equiv\pi_0, \tag{2.20}$$

(ii) *for* x *and* y *in the region* D, *there exists a positive constant* M *for the Jacobian matrix* (2.3) *such that*

$$\|S(x)-S(y)\|\leq M\|x-y\|, \tag{2.21}$$

(iii) *for the constant* M *and for all* x *in* D, *there exists a constant* K *such that*

$$K=\max[M, \|f_{xx}\|], \tag{2.22}$$

(iv) *the constants introduced above satisfy the following inequality*:

$$K\bar{B}_0^2\pi_0<2, \tag{2.23}$$

(v) *the perturbation parameters* ${}^k\varepsilon$ *satisfy the following conditions*:

$$\begin{aligned} &0<{}^0\varepsilon\leq\bar{B}_0\pi_0, &&\textit{for} \quad k=0, \\ &0<{}^k\varepsilon\leq\min[{}^{k-1}\varepsilon, B_k], &&\textit{for} \quad k=1,2,\cdots, \end{aligned} \tag{2.24}$$

where

$$(2.25) \qquad B_0 = \frac{\overline{B}_0}{1 - {}^0\varepsilon K \overline{B}_0/2}, \quad h_0 = 3KB_0^2\pi_0,$$

and

$$(2.26) \qquad B_k = \frac{B_{k-1}}{1-h_{k-1}}, \quad \pi_k = \frac{1}{2}h_{k-1}\pi_{k-1}, \quad h_k = \frac{1}{2}\frac{h_{k-1}^2}{(1-h_{k-1})^2},$$

(vi) *the constant* h_0 *introduced above satisfies the inequality*

$$(2.27) \qquad h_0 < \frac{1}{2},$$

and the cube below is contained in D,

$$(2.28) \qquad \|x - {}^0x\| < \frac{1-(1-2h_0)^{1/2}}{h_0} B_0\pi_0 .$$

Then the system of nonlinear equations given by (2.1) *has an exact solution* x^* *in the cube* (2.28). *Moreover, the speed of convergence may be estimated by the inequality*

$$(2.29) \qquad \|{}^{k+1}x - x^*\| \leq \frac{(2h_0)^{2^{k+1}-1}}{2^k} B_0\pi_0,$$

which shows that the convergence ratio of the ε*-secant method given by* (2.17) *is quadratic.*

The *proof* of this lemma is shown in Watanabe-Ojika-Mitsui [29].

This lemma implies that the matrix $S({}^kx;{}^k\varepsilon)$ is bounded and invertible for every k. Thus $S(x^*;\ 0) = S(x^*)$ has also the same property.

Let us set

$$(2.30) \qquad \Gamma(x;\varepsilon) = S^{-1}(x;\varepsilon) = \frac{G(x;\varepsilon)}{d(x;\varepsilon)},$$

where $d(x;\varepsilon)$ is the determinant of the perturbed Jacobian matrix $S(x;\varepsilon)$. For the quantities $\Gamma(x;\varepsilon)$, $d(x;\varepsilon)$ and $G(x;\varepsilon)$ we have the following theorem.

Theorem 2.3. *Under the assumptions of Theorem* 2.1 *and*

Lemma 2.1, *there hold the following convergence properties for* $\{\Gamma({}^k x; {}^k\varepsilon)\}$, $\{d({}^k x; {}^k\varepsilon)\}$ *and* $\{G({}^k x; {}^k\varepsilon)\}$ $(k=0, 1, \cdots)$:

$$(2.31) \quad \text{(i)} \qquad \|\Gamma({}^k x; {}^k\varepsilon) - \Gamma(x^*; 0)\| = O(\|{}^k x - x^*\|),$$

$$(2.32) \quad \text{(ii)} \qquad |d({}^k x; {}^k\varepsilon) - d(x^*; 0)| = O(\|{}^k x - x^*\|),$$

and

$$(2.33) \quad \text{(iii)} \qquad \|G({}^k x; {}^k\varepsilon) - G(x^*; 0)\| = O(\|{}^k x - x^*\|).$$

By virtue of (2.29) in Lemma 2.1, $O(\|{}^k x - x^*\|)$ implies the quadratic ratio of convergence as k tends to infinity.

Proof. From the first relation of (2.26) we have

$$(2.34) \qquad B_k = \frac{B_0}{\prod_{i=0}^{k-1}(1-h_i)}.$$

Let $\Lambda_k = \prod_{i=0}^{k} (1-h_i)$. Then the sequence of finite products $\{\Lambda_k\}$ converges to a limit Λ, and there exist constants c_0 and c_1 such that

$$(2.35) \qquad 0 < c_0 \leq \Lambda \leq c_1 < \infty .$$

In fact, from the third relation of (2.26) we have

$$0 < h_k \leq \frac{1}{2}(2h_0)^{2^k} < \frac{1}{2},$$

so that

$$(2.36) \qquad 0 < \sum_{i=0}^{k} h_i \leq \frac{1}{2} \sum_{i=0}^{k} (2h_0)^{2^i} \leq \frac{1}{2} \frac{2h_0}{1-2h_0} < \infty .$$

The convergence of the series $\sum_{i=1}^{k} h_i$ is necessary and sufficient for the convergence of the infinite product

$$\Lambda = \lim_{k\to\infty} \prod_{i=0}^{k} (1-h_i).$$

From relations (2.16) and (2.17) we have constants c_2 and c_3 such that for all $k=0, 1, \cdots$,

$$(2.37) \qquad 0 < c_2 \equiv \frac{B_0}{c_1} \leq B_k \leq \frac{B_0}{c_0} \equiv c_3 < \infty .$$

This inequality shows that the operator $\Gamma({}^kx;{}^k\varepsilon)$ is bounded for all $k=0, 1, \cdots$, and we obtain for some c_4

$$(2.38)\qquad \begin{aligned} &\|\Gamma({}^kx;{}^k\varepsilon)-\Gamma(x^*;0)\| \\ &\leq\|\Gamma({}^kx;{}^k\varepsilon)-\Gamma({}^kx;0)\|+\|\Gamma({}^kx;0)-\Gamma(x^*;0)\| \\ &\leq\|\Gamma({}^kx;0)\{S({}^kx)-S({}^kx;{}^k\varepsilon)\}\Gamma({}^kx;{}^k\varepsilon)\| \\ &\quad+\|\Gamma(x^*;0)\{S(x^*)-S({}^kx)\}\Gamma({}^kx;0)\| \\ &\leq c_4\|{}^kx-x^*\|. \end{aligned}$$

By virtue of the continuity of $d(x;\varepsilon)$ with respect to x and ε we have the following inequality for some constant c_5:

$$(2.39)\qquad |d({}^kx;{}^k\varepsilon)-d(x^*;0)|\leq c_5\|{}^kx-x^*\|,$$

which is equivalent to (2.32).

From the existence of the inverse matrix $\Gamma(x;\varepsilon)$ of the perturbed Jacobian matrix $S(x;\varepsilon)$ there exist positive constants c_6 and c_7 such that for all x satisfying (2.28),

$$(2.40)\qquad 0<c_6\leq|d(x;\varepsilon)|\leq c_7.$$

Then the numerator matrix $G(x;\varepsilon)$ of the inverse matrix $S^{-1}(x;\varepsilon)$ can be written in the form

$$(2.41)\qquad G(x;\varepsilon)=d(x;\varepsilon)S^{-1}(x;\varepsilon).$$

We can easily show (2.33) from relations (i), (ii) and the following:

$$(2.42)\qquad \begin{aligned} G({}^kx;{}^k\varepsilon)-G(x^*;0)&=d({}^kx;{}^k\varepsilon)S^{-1}({}^kx;{}^k\varepsilon)-d(x^*;0)S^{-1}(x^*;0) \\ &=\{d({}^kx;{}^k\varepsilon)-d(x^*;0)\}S^{-1}({}^kx;{}^k\varepsilon) \\ &\quad+d(x^*;0)\{S^{-1}({}^kx;{}^k\varepsilon)-S^{-1}(x^*;0)\}, \end{aligned}$$

Thus the *proof* of Theorem 2.2 terminates.

On the tendencies of $f({}^kx)$ and $d({}^kx;\ {}^k\varepsilon)$ to zero as $k\rightarrow\infty$, we have the following.

Theorem 2.4. *Under the assumptions and notations for Lemma* 2.1, *there exists an integer k^* such that for $k\geq k^*$ the inequality*

$$\|f({}^k x)\| \ll |d({}^k x;\ {}^k\varepsilon)| \tag{2.43}$$

holds.

The above statement is a direct consequence of Theorem 2.2. In fact, we can easily see that for some constants c_8 and c_9,

$$\|f({}^k x)-f(x^*)\| = \|f({}^k x)\| \leq c_8\|{}^k x - x^*\| \tag{2.44}$$

and

$$\big|\,|d({}^k x;{}^k\varepsilon)| - |d(x^*;0)|\,\big| \leq c_9\|{}^k x - x^*\|\ . \tag{2.45}$$

These estimates give (2.43) for sufficiently large k.

Let us now define T as the set of points in $\mathbf{R}^n$ as follows:

$$\mathrm{T} = \{x|f(x)\neq 0,\quad \det[S(x)]=0\}. \tag{2.46}$$

A point $\bar{x}$ of T is called here the *singular point.* Then the following corollary holds.

Corollary 2.1. *If the sequence $\{{}^k x\}$ approaches to a singular point $\bar{x}$, then there exists an integer $\bar{k}$ such that for $\forall k\geq\bar{k}$ the inequality*

$$\|f({}^k x)\| \geq |d({}^k x;{}^k\varepsilon)| \tag{2.47}$$

holds.

In practical computations, the properties (2.43) and (2.47) are often useful for identifying that the sequence $\{{}^k x\}$ is converging to a simple root or singular point. Some properties of the multiple roots and computational algorithms are discussed in the next section.

§ 3. Deflation Algorithm for Multiple Roots

In this section we present a method, termed as the *deflation algorithm* [14, 19, 28, 29], for finding multiple roots of a system of nonlinear equations

$$F^{[0]}(x) = (f_1^{[0]}(x),\ f_2^{[0]}(x),\ \cdots,\ f_n^{[0]}(x))'=0\ , \tag{3.1}$$

$$x = (x_1,\ x_2,\ \cdots,\ x_n)'\ ,$$

where the Jacobian matrix $F_x^{[0]}$ of F is singular at the root x^*, i.e.,

$$d^{[0]}(x^*) = \det[F_x^{[0]}(x^*)] = 0 . \tag{3.2}$$

Here $[\cdot]^{[l]}$ denotes the value of $[\cdot]$ at the root x^* after the l-th deflation process.

If the Jacobian matrix F_x at $x^* \in F^{-1}(0)$ is nonsingular, the Newton-Raphson iteration or ε-secant algorithm can be successfully used to get the solution in good accuracy. However, in the case of a singular Jacobian matrix $F_x^{[0]}(x^*)$, the classical theory is not applicable and except for the one dimensional problems only a few results are available [2-6, 21, 22].

We give here several properties of the multiple roots of a system of nonlinear equations and a practical method, termed here as the deflation algorithm, to determine the multiple roots.

3.1. Properties of Multiple Roots

It is instructive to consider first the one-dimensional case of a real-valued function f of a real variable x, i.e.,

$$f(x) = 0 . \tag{3.3}$$

In general, a root x^* of the nonlinear equation (3.3) is said to have multiplicity m if

$$f(x) = (x - x^*)^m \bar{f}(x), \quad 0 \neq |\bar{f}(x^*)| < \infty , \tag{3.4}$$

where $m \geq 1$ and $\bar{f}(x)$ is twice continuously differentiable at the root x^*.

Analogous to (2.4), starting from an initial guess ${}^0x$ in a neighborhood D in $\mathbf{R}$ of x^*, the Newton method defines the sequence of approximations

$${}^{k+1}x = {}^kx + \Delta^k x , \quad \Delta^k x = -[f_x({}^kx)]^{-1} f({}^kx), \quad k = 0, 1, 2, \cdots . \tag{3.5}$$

Let ${}^ky$ be the error of ${}^kx$ from x^*, i.e.,

$${}^ky = {}^kx - x^* . \tag{3.6}$$

Then, from (3.4) and (3.5), we have

$${}^{k+1}y = {}^ky - \frac{({}^ky)^m \bar{f}({}^kx)}{m({}^ky)^{m-1} \bar{f}({}^kx) \left\{ 1 + \dfrac{{}^ky}{m \bar{f}({}^kx)} \bar{f}_x({}^kx) \right\}} \tag{3.7}$$

From (3. 7) and Taylor's theorem, it follows that

$$ {}^{k+1}y=\left(\frac{m-1}{m}\right){}^{k}y+O({}^{k}y^{2}). \tag{3.8} $$

Consequently, if the sequence $\{{}^{k}x\}$ is convergent to x^{*}, the sequence $\{{}^{k}y\}$ will converge to 0 with the speed of a geometric progression with ratio $(m-1)/m$.

From the above discussion, we now have the following theorem [14, 28].

Theorem 3. 1. *If the sequence* $\{{}^{k}x\}$ *defined by* (3. 5) *converges to* x^{*}, *and* $\bar{f}(x)$ *has a Taylor series expansion at* x^{*} *which converges in some neighborhood* D *of* x^{*}, *then the following asymptotic relations hold:*

(i)

$$ \lim_{k\to\infty}\frac{f({}^{k+1}x)}{f({}^{k}x)}=\left(\frac{m-1}{m}\right)^{m}, \tag{3.9a} $$

(ii)

$$ \lim_{k\to\infty}\frac{f_{x}({}^{k+1}x)}{f_{x}({}^{k}x)}=\begin{cases}1, & if\quad m=1,\\ \left(\dfrac{m-1}{m}\right)^{m-1}, & if\quad m\geqq 2,\end{cases} \tag{3.9b} $$

and (iii)

$$ \lim_{k\to\infty}\frac{\Delta^{k+1}x}{\Delta^{k}x}=\frac{m-1}{m}. \tag{3.9c} $$

As for the proof, see [14, 19, 28].

In Section 2, we discussed the convergence properties of the Newton and ε-secant method to the simple roots. On the other hand, the properties in Theorem 3. 1 are discussed from the standpoint of the multiplicity m and they play important roles in the subsequent discussions.

3. 2. Deflation Algorithm

Let us now return to the problem of finding multiple roots of the system of nonlinear equations (3. 1). The Newton-Raphson iteration for old x and new $\bar{x}$ is now given by

$$(3.10) \qquad F_x^{[l]}(x)\,(\bar{x}-x) = -F^{[l]}(x)\,, \quad l=0,1,2,\cdots.$$

Assume now that the rank of Jacobian matrix at ${}^k x$ in the neighborhood of the root x^* is given by

$$(3.11\text{a}) \qquad r^* = \operatorname{rank} F_x^{[0]}(x^*)\,, \quad 0 \leq r^* \leq n-1\,,$$

$$(3.11\text{b}) \qquad r = r^{[l]} = \operatorname{rank} F_x^{[l]}({}^k x)\,, \quad r^* \leq r^{[l]} \leq n\,,$$

and that the system of linear equations (3.10) is solved by the familiar Gaussian elimination method [1, 26].

For simplicity, suppose x_1 has been eliminated from equations $2, \cdots, n$ of (3.10) ; hence x_1 remains only in the first equation, and x_1, x_2 from equations $3, \cdots, n$ and so on up to $x_1, \cdots, x_{r^*}$ from equations $r^*+1, \cdots, n$. Then we have the *pivot matrix* P defined by

$$(3.12) \qquad P_r^{[0]}(x^*) = \begin{array}{c} \begin{matrix} f_i^{[0]} & x_j^* \end{matrix} \\ \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ \vdots & \vdots \\ r^* & r^* \end{bmatrix} \end{array}, \quad P_r^{[l]}(x^*) = \begin{array}{c} \begin{matrix} f_i^{[l]} & x_j^* \end{matrix} \\ \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ \vdots & \vdots \\ r^{[l]} & r^{[l]} \end{bmatrix} \end{array}, \quad l=1,2,\cdots,$$

and the equation $f_i^{[l]}$ and variable x_j in (3.12) are called the *pivot equation* and *variable*, respectively.

Taking (3.12) into account, let us define an $(r^{[l]}+1) \times (r^{[l]}+1)$. Jacobian submatrix $D_s^{[l]}$, termed as the *deflation matrix*, by

$$(3.13) \qquad D_s^{[l]}(x) = \begin{bmatrix} d_{11} \cdots\cdots d_{1r} & d_{1s} \\ \vdots \qquad\quad \vdots & \vdots \\ d_{r1} \cdots\cdots d_{rr} & d_{rs} \\ d_{s1} \cdots\cdots d_{sr} & d_{ss} \end{bmatrix}, \quad s=r+1, \cdots, n\,,$$

where

$$(3.14) \qquad d_{ij} = \partial_{x_j} f_i^{[l]}, \quad i, j=1, \cdots, r, s\,,$$

$$r^* = r^{[0]}, \quad r = r^{[l]}, \quad l=1,2,\cdots,$$

where $\partial_x f$ denotes the partial derivative of f with respect to x. Note that d_{ij} is given in the analytic form.

At the $(r^{[l]}+1)$st formal elimination stage, in the l-th deflation process, the deflation matrix $D_s^{[l]}$ is transformed into the form:

$$(3.15)\qquad E_s^{[l]}(x)=\begin{bmatrix} e_{11} & c_{12}\cdots e_{1r} & e_{1s} \\ & c_{22} \ddots & \vdots \\ 0 & e_{rr} & e_{rs} \\ & & e_{ss}\end{bmatrix}^{[l]},\quad \begin{array}{l} r=r^{[l]}, \\ s=r^{[l]}+1,\cdots,n\,. \end{array}$$

Here the $(r^{[l]}+1)\times(r^{[l]}+1)$ upper triangular matrix $E_s^{[l]}(x)$ is termed as the *eliminated matrix* of $D_s^{[l]}$. We now have the following [19].

Theorem 3.2. *Suppose that* $r^*=\mathrm{rank}\,F_x^{[0]}(x^*)$, $0\leq r^*\leq n-1$ *and the pivot matrix* $P_r^{[0]}(x^*)$ *is given by* (3.12). *Let* $e_{ii}^{[l]}(x)$ *be the diagonal element of the eliminated matrix* $E_s^{[l]}(x)$ *obtained from the* $(r^{[l]}+1)\times(r^{[l]}+1)$ *deflation matrix* $D_s^{[l]}(x)$. *If the approximate solution* ${}^kx$ *of* (3.10) *is sufficiently close to the root* x^*, *then the following properties hold:*

$$(3.16\text{a})\quad \text{(i)}\qquad |e_{ii}^{[l]}({}^{k+1}x)/e_{ii}^{[l]}({}^kx)|\approx 1,\quad i=1,2,\cdots,r^{[l]},$$

$$(3.16\text{b})\quad \text{(ii)}\qquad |e_{ss}^{[l]}({}^kx)|=|\det D_s^{[l]}({}^kx)|\ll 1,\quad s=r^{[l]}+1,\cdots,n,$$

$$(3.16\text{c})\quad \text{(iii)}\qquad \det D_s^{[l]}(x^*)=0,\quad \text{if}\quad r^{[l]}\neq n,$$

and

$$(3.16\text{d})\quad \text{(iv)}\qquad 1/e<|e_{ss}^{[l]}({}^{k+1}x)/e_{ss}^{[l]}({}^kx)|\leq 1/2,\quad k,l=0,1,\cdots.$$

Proofs of the theorem are obvious from Theorem 3.1. The upper and lower bounds in condition (iv) can be easily derived from (3.16b) with $m=2$ and $m=\infty$, respectively.

At the root x^*, the deflation matrix $D_s^{[l]}$ satisfies the relation (3.16c). Taking this fact into account, replace $f_s^{[l]}$ in (3.1) by $\det D_s^{[l]}$, $s=r^{[l]}+1,\cdots,n$ and define a new set of equations in the next deflation stage by

$$(3.17)\qquad F^{[l+1]}(x)=\begin{bmatrix} f_1^{[l+1]}(x) \\ \vdots \\ f_r^{[l+1]}(x) \\ f_{r+1}^{[l+1]}(x) \\ \vdots \\ f_n^{[l+1]}(x)\end{bmatrix}=\begin{bmatrix} f_1^{[l]}(x) \\ \vdots \\ f_r^{[l]}(x) \\ \det D_{r+1}^{[l]}(x) \\ \vdots \\ \det D_n^{[l]}(x)\end{bmatrix}=0,\quad r=r^{[l]}.$$

Here $F^{[l+1]}(x)$ is termed as the $(l+1)$st *deflated equations.*

It is easily seen from the above discussion that, compared with $F^{[l]}$, the convergence of $F^{[l+1]}$ will be improved. In fact, for a typical system of equations with multiple roots we have the following theorem [19].

Theorem 3.3. *Assume that a system of equations is given by*

$$f_i(x) = (a_{i1}x_1 + \cdots + a_{i,i-1}x_{i-1} + x_i + \cdots + a_{in}x_n)^{m_i}\bar{f}_i(x) = 0\,, \tag{3.18}$$

where

$$f_i(x^*) = 0\,, \quad \bar{f}_i(x^*) \neq 0\,, \quad i = 1, 2, \cdots, n\,, \tag{3.19a}$$

$$m_i \geq 1\,, \quad \overline{m} = \max_i [m_i] \geq 2\,, \tag{3.19b}$$

and

$$\begin{vmatrix} 1 & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & 1 \end{vmatrix} \neq 0\,. \tag{3.19c}$$

Then at the l-th deflation process, the multiplicity $m^{[l]}$ of $F^{[l]}(x)$ is given by

$$m^{[l]} = \prod_{i=1}^{n} \{\max[1, m_i - l]\}\,, \quad 0 \leq l \leq \overline{m} - 1\,, \tag{3.20a}$$

and

$$m^{[\overline{m}]} = 1\,. \tag{3.20b}$$

As for the *proof* see [19].

Let $m = m^{[0]}$ be the multiplicity of (3.18). Then this theorem shows that, for the system (3.18), $(\overline{m} - 1)$ deflation processes are necessary to obtain the m-ple root of (3.18) in the same accuracy as usual simple roots.

3.3. Computational Realization

As we have seen, the determinant of the deflation matrix (3.13) must be calculated in analytical form. In the package NAES, a symbolic and algebraic manipulation language, REDUCE 2 [7], is adopted for this purpose. However from a practical stand point, it is often possible to simplify or skip computations of the determinants by using some properties of the deflation matrix [14, 18, 28].

Consider the l-th deflation process at k-th iteration given by

$$F^{[l]}({}^{k}x)=0\,, \tag{3.21}$$

and suppose that the pivot matrix $P_r^{[l]}({}^{k}x)$ is given by

$$P_r^{[l]}({}^{k}x)=\begin{array}{c} \begin{matrix} f_i^{[l]} & {}^{k}x_j \end{matrix} \\ \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ \vdots & \vdots \\ r^{[l]} & r^{[l]} \end{bmatrix} \end{array}, \quad k, l=0, 1, \cdots . \tag{3.22}$$

It is noteworthy that while solving the linear equations (3. 10) by the Gaussian elimination method, the pivot matrix can easily be obtained by checking the properties (3. 16) in Theorem 3. 2.

If (i) the i-th equation $f_i^{[l]}$ of (3. 21) contains x_j explicitly and its partial derivative at the root x^* is zero and (ii) the equation $f_i^{[l]}$ does not contain x_s explicitly, then we have

$$\begin{aligned} F_x^{[l]}(x)\,|_{x=x^*} &= i[\cdots, \partial_{x_j} f_i^{[l]}, \cdots, \partial_{x_s} f_i^{[l]}, \cdots]_{x=x^*} \\ &= [\cdots, 0, \cdots, \bullet, \cdots]. \end{aligned} \tag{3.23}$$

Here 0 and ● in (3. 23) are called *numerical and algebraic zeros,* respectively. As for the numerical zero, we have the following [19].

Theorem 3. 4. *Suppose that the sequence* $\{{}^{k}x\}$ *defined by* (3. 10) *converges to the root* x^*. *If the* (i, j)*-th element of the Jacobian matrix* $F_x^{[l]}$ *at the root is a numerical zero, then the following estimate at the l-th deflation process holds:*

$$\lim_{k\to\infty} |\partial_{x_j} f_i^{[l]}({}^{k+1}x)/\partial_{x_j} f_i^{[l]}({}^{k}x)| \leq 1/2\,. \tag{3.24}$$

Applying (3. 9b) in Theorem 3. 1, this theorem can be easily proved.

The estimate (3. 24) is useful to identify the elements with numerical zeros in the Jacobian matrix $F^{[l]}$. In the following assume that the deflated equations and the pivot matrix at the l-th deflation process in the k-th iteration are given by (3. 17) and (3. 22), respectively. From the computational standpoint, we first provide the following category.

Category 1 (*Singularity with numerical zeros*); Suppose that the (i_u, j_v) element of the Jacobian matrix $F_x^{[l]}$ $(u=1, 2, \cdots, u^{[l]}, v=1, 2, \cdots, v^{[l]}; 0 \leq u^{[l]}, v^{[l]} \leq n)$ has a numerical zero. If the $(i_{\bar{u}}, j_{\bar{v}})$ element has the least total degree of variables which are not in the pivot variables in (3. 22), it is called the *minimum zero element.* In this category, the pivot matrix is updated as follows: (i) the analytical partial derivative $(\partial f_{i\bar{u}}^{[l]}/\partial x_{j_{\bar{v}}})$ corresponding to the pivot matrix is updated; and (ii) the minimum zero is registered as a new pivot function and a variable is selected from the variables which are not contained in the old pivot variables. (iii) Then r is increased by one. If r is still less than n and there are other numerical zeros, the process (i)-(iii) are repeated. If $r=n$, then replace l by $l+1$ and terminate the l-th deflation process. (iv) Otherwise proceed to the next category.

Category 2 (*Singularity with nontrivially proportional rows*); Suppose that, except for the elements with numerical or algebraic zeros, all the elements in i-th and j-th rows of the Jacobian matrix $F_x^{[l]}$ at the root satisfy the following relations:

$$(3.25) \qquad \partial_{x_s} f_i^{[l]}(x) / \partial_{x_s} f_j^{[l]}(x) \Big|_{x=x^*} = a, \quad 0 < |a| < \infty, \quad 1 \leq {}^{\forall}s \leq n \,,$$

where a is a constant. Then it is easily seen that the rank of $F_x^{[l]}(x^*)$ is degenerated by one. From (3. 25), we form at most $n(n-1)/2$ equations:

$$(3.26) \qquad \partial_{x_u} f_i^{[l]}(x) \cdot \partial_{x_v} f_j^{[l]}(x) - \partial_{x_v} f_i^{[l]}(x) \cdot \partial_{x_u} f_j^{[l]}(x) = 0 \,,$$

$$1 \leq u \leq n-1, \quad v+1 \leq v \leq n \,.$$

It is worth mentioning that (3. 26) is generated by REDUCE 2.

We now provide the procedure for Category 2.

(i) From (3. 26), find the equation with the minimum total degree of variables (degree of the equation ≥ 1) and a new pivot variable which is not in the pivot matrix, and let $u=\bar{u}$ and $v=\bar{v}$.

(ii) Increase r by one and register (3. 26) with $u=\bar{u}$ and $v=\bar{v}$ by $f_r^{[l]}$ and its new variable by x_r.

(iii) If $r=n$, then replace l by $l+1$ and terminate the l-th deflation process.

(iv) Otherwise, delete (3.26) with $u=\bar{u}$ and $v=\bar{v}$, and repeat (i)-(iv) until a new pivot variable cannot be found in (3.26) for all u and v.

Category 3 (*Singularity with nontrivially proportional columns*); Similarly to Category 2, suppose that except for the elements with numerical and algebraic zeros, all the elements in i-th and j-th columns of the Jacobian matrix $F_x^{[l]}$ at the root x^* satisfy the following relation:

$$\partial_{x_i} f_s^{[l]}(x) / \partial_{x_j} f_s^{[l]}(x) \,|_{x=x^*} = b, \quad 0<|b|<\infty, \qquad 1\leq {}^\forall s \leq n, \tag{3.27}$$

where b is a constant. From (3.27), form at most $n(n-1)/2$ equations:

$$\partial_{x_i} f_u^{[l]}(x) \cdot \partial_{x_j} f_v^{[l]}(x) - \partial_{x_i} f_v^{[l]}(x) \cdot \partial_{x_j} f_u^{[l]}(x) = 0, \qquad 1\leq u \leq n-1, \quad u+1\leq v \leq n. \tag{3.28}$$

Then the same procedures (i)-(iv) in Category 2 also hold for Category 3.

Applying Categories 1-3, computations of the deflation matrix (3.13) can greatly be reduced. However, if n pivot variables were not obtained by these categories, it is then necessary to compute some of the matrices by the following procedure.

Category 4 (*Singularity in general*); Suppose that, from Categories 1-3, the pivot matrix $P_{\bar{r}}^{[l]}$ is given by

$$P_{\bar{r}}^{[l]} = \begin{array}{c} \begin{array}{cc} f_i^{[l]} & x_j \end{array} \\ \begin{bmatrix} 1 & 1 \\ \vdots & \vdots \\ \bar{r} & \bar{r} \end{bmatrix} \end{array}, \quad r^{[l]} \leq \bar{r} < n. \tag{3.29}$$

Then the procedure is executed as follows:

(i) From the Jacobian matrix $F_x^{[l]}$, find the element with a new pivot variable, say, $x_{\bar{r}+1}$ which is not in (3.29).

(ii) Construct the $(r^{[l]}+1)\times(r^{[l]}+1)$ deflation matrix $D_s^{[l]}$, $s=\bar{r}+1, \cdots, n$, given by (3.13) so that the element is included.

(iii) If $\bar{r}+1=n$, then replace l by $l+1$ and terminate the l-th deflation process.

(iv) Otherwise, increase $\bar{r}$ by one and repeat the procedures (i) - (iii).

§ 4. Hybrid Manipulations in NAES and an Illustrative Example

4.1. Outline of the Communication-Flows in Hybrid Manipulations

In the package NAES the system of nonlinear algebraic equations is given in the FORTRAN expression. The numerical manipulations are executed by FORTRAN and on the other hand the symbolic manipulations are treated by REDUCE 2 [7]. The schematic diagram of the communications between numerical and symbolic manipulations in the package is shown in Fig. 4.1., where R- and F- files stand for the data files for REDUCE 2 and FORTRAN programs, respectively.

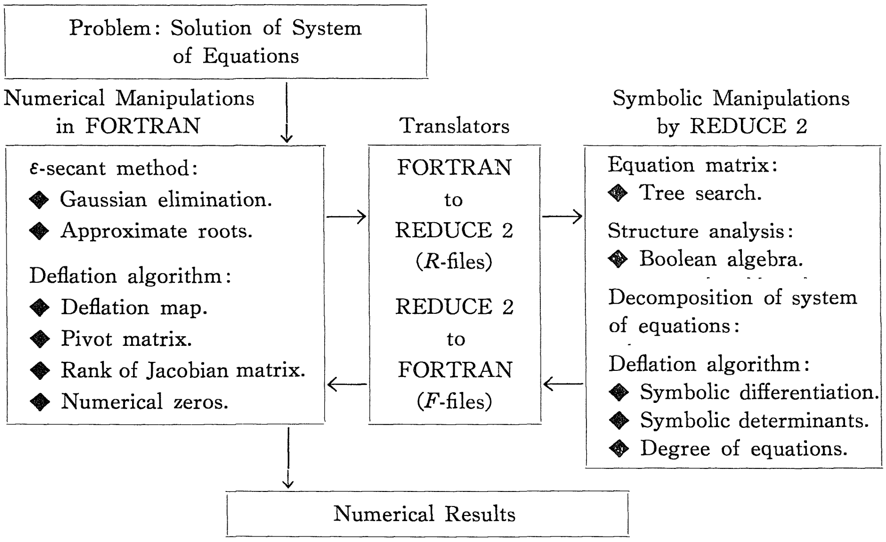

Fig. 4.1. The hybrid manipulation system.

4.1.1. *Translators between FORTRAN and REDUCE 2*

For the REDUCE 2 system, it is easy to convert the REDUCE 2 expressions into FORTRAN forms. However the reverse is in genral difficult. Thus we need translators to attain smooth communications between these two different languages. A user provides his problem in two programs by FORTRAN: (i) the main program for the starting of calculations, and (ii) a function subprogram for the system of equations. Then the system of equations in the subprogram is translated into REDUCE 2 expressions by a character-wise translation program written in FORTRAN.

4.1.2. *Symbolic Manipulations in REDUCE* 2

Using the translated expressions of the system of equations, structure analyses of the equations are examined. For the purpose a Boolean matrix which shows explicit dependency of a function f_i on an unknown variable x_j is generated by the REDUCE 2 procedure TREESCH (see (iv) in this subsection). Then some Boolean transformations are operated on the matrix, and the system is decomposed into independent subsystems and also ordered into hierarchical structures. The details of the analyses are omitted here [15, 16].

For the analysis and decompositions in REDUCE 2 programs, the built-in procedures used mainly in NAES are formal differentiations, determinants of matrices, the degree and coefficients of a polynomial of a variable [7].

(i) The operator DF(F, X) is used to represent partial differentiation of the function F with respect to the variable X.

(ii) The operator DET(M) is used to represent the determinant of the square matrix M.

(iii) COEFF(P, X, CO) is an operator which assigns coefficients of the various powers of a kernel. If the CO has been previously declared a single dimensioned array, the i-th array element is assigned to the coefficient of the i-th power of X in P, up to the maximum dimension of the array. Note that the value of COEFF shows the degree of P.

In addition, two special procedures which are written in REDUCE 2 are developed in the NAES.

(iv) TREESCH(A, B) is a procedure to search tree structures. The argument A is an atom in LISP language [7], and B is a list structure. TREESCH gives "1" if A is contained in B and "0" otherwise. The list of TREESCH is given in Fig. 4. 2. This procedure is used for the structure analysis of the nonlinear algebraic equations [15, 16] and the deflation procedure in Section 3.

(v) DGF (F, N) is a procedure to find the total degree of an algebraic expression. Let F be a polynomial of N variables X(1), ···, X(N), then this procedure finds the total degree of F for all variables. By this procedure, we can see whether a function is linear or not. The procedure is used also for reconstructions of the system of equations and the deflation procedures. The list of DGF is shown in Fig. 4. 3.

```
COMMENT TREE SEARCH;
SYMBOLIC;
EXPR PROCEDURE TREESCH(A, B);
  IF NULL(B) THEN NIL
    ELSE IF A=B THEN '1
    ELSE IF ATOM(B) THEN NIL
    ELSE IF TREESCH(A, CAR(B))='1 THEN '1
    ELSE TREESCH(A, CDR(B));
OPERATOR TREESCH;
ALGEBRAIC;
```

Fig. 4. 2. The list of procedure TREESCH(A, B).

```
COMMENT TO FIND DEGREE OF FN IN XX(I), I=1, ···, N;
PROCEDURE DGF(FN, N);
BEGIN
  MAXDEG:=0;
  OPE:=FN;
  FOR L:=1: N DO
  BEGIN
    DEG:=COEFF(OPE, XX(L), CO);
    FOR ALL Z LET OP(Z)=FOR J:=0: DEG SUM CO(J)*Z**J;
    OPE:=OP(Y);
    MAXDEG:=IF DEG > MAXDEG THEN DEG ELSE MAXDEG;
  END OF L;
  DEG:=COEFF(OPE, Y, CO);
  MAXDEG:=IF DEG > MAXDEG THEN DEG ELSE MAXDEG;
  RETURN MAXDEG;
END OF DGF;
```

Fig. 4. 3. The list of Procedure DGF(FN, N).

4. 1. 3. *Numerical Manipulations in FORTRAN*

The ε-secant method in Subsection 2. 2 is used for the generation of the perturbed Jacobian matrix (2. 16) by numerical differentiations.

Then the Gaussian elimination method is applied to solve the system of linear equations (2.17). In the elimination process, two matrices of integers, the *deflation map* (see the next Subsection) and the *pivot matrix* (3.12), and the *rank* r of the perturbed Jacobian matrix (2.16) are generated.

4.2. An Illustrative Example

In the following the present hybrid manipulations are explained through the problem of Šamanskii [19, 27] given by

$$(4.1) \qquad f^{[0]}(x)=\begin{bmatrix} x_1+x_2+x_3-1.0 \\ 0.2x_1^3+0.5x_2^2-x_3+0.5x_3^2+0.5 \\ x_1+x_2+0.5x_3^2-0.5 \end{bmatrix}=0\,,$$

which is also solved by the NAES. The problem (4.1) has a quadruple root $x=x^{*1}=(0.0,\ 0.0,\ 1.0)$ and a double $x^{*2}=(-2.5,\ 2.5,\ 1.0)$.

(i) The main and subroutine programs for the problem (4.1) are shown in Fig. 4.4. The FORTRAN subroutine is transformed into the REDUCE 2 expressions as shown in Fig. 4.5. Note that colons and semicolons are added in the equations.

(ii) The result after the structure analyses and the translations from REDUCE 2 to FORTRAN is given in Fig. 4.6. From the figure it is easily seen that (a) the equations can not be decomposed into subsystems, (b) the equation f_1 is linear, and (c) the total degree of f_2, for example, is three.

(iii) Then numerical methods such as ε-secant method, Gaussian eliminations and deflation algorithms are applied to the newly generated subroutine.

By way of a numerical example, let the initial guess ${}^0x=(0.2,\ 0.2,\ 0.5)$ and define the convergence condition E by

$$(4.2) \qquad {}^kE^{[l]}=\left[\frac{1}{n}f^{[l]}({}^kx)'f^{[l]}({}^kx)\right]^{1/2}\leq 10^{-14}\,.$$

At the 6-th iteration by the ε-secant with $\varepsilon=10^{-8}$, the condition ${}^6E^{[0]}\leq 10^{-4}$ was satisfied and the singularity of the system was indicated by the following informations.

(A) The rank of the perturbed Jacobian matrix $=1$.

```
C     MAIN PROGRAM FOR THE SAMANSKII PROBLEM
      DIMENSION JX(20), JF(20), KX(20), KF(20), M(20, 20), KFX(20, 4)
      REAL*8 XX(20), FF(20), X(20), Y(20), F(20), G(20), A(20, 21),
     .  B(20, 21), CRIT
C
      COMMON / ALG / C
      REAL C(20)
C
      DATA ICH, II, MAX, ID, MULTI, CRIT/1, 2, 20, 20, 5, 1.D-13/
C
      XX(1)=0.2D0
      XX(2)=0.5D0
      XX(3)=0.3D0
C
C
      CALL ALGO(NT, XX, FF, JX, JF, KX, KF, IND, MAXIND, II, MD, NX,
     .  MAX, M, KRANK, KFX, X, Y, F, G, A, B, CRIT, INDEX, MULTI,
     .  ILL, ICH)
C
      WRITE(6, 1100)  (I, XX(I), I=1, NT)
      STOP
C
 1100 FORMAT('     X(', I2, ')=', D22.13)
      END
C
      BLOCK DATA
      COMMON / ALG / C
      REAL*8 C(20)
      DATA / 1.D0,  2.D0,  3.D0,  4.D0,  5.D0,
     .       6.D0,  7.D0,  8.D0,  9.D0,  0.5D0,
     .      18.D0, 5.5D0, 18.5D0, 6.5D0, 0.2D0,
     .       0.D0,  0.D0,  0.D0,  0.D0,  0.D0/
      END
```

Fig. 4.4a. Main program by FORTRAN.

```
C     EXAMPLE FOR FORTRAN TO REDUCE TRANSLATIONS
      SUBROUTINE ALG0(N, X, F)
      REAL*8 X(1), C(20)
      COMMON / ALG / C
      N=3
      M=0
      F(1)=X(1)+X(2)+X(3)-C(1)
      F(2)=C(15)*X(1)**3+C(10)*X(2)**2-X(3)+C(10)*X(3)**2+C(10)
      F(3)=X(1)+X(2)+C(10)*X(3)**2-C(10)
      RETURN
      END
```

Fig. 4.4b. The FORTRAN expression of system of equations (4.1).

```
N:=3;
M:=0;
F(1):=X(1)+X(2)+X(3)-C(1);
F(2):=C(15)*X(1)**3+C(10)*C(2)**2-X(3)+C(10)*X(3)**3+C(10);
F(3):=X(1)+X(2)+C(10)*X(3)**2-C(10);
```

Fig. 4.5. The REDUCE 2 expressions of system of equations (4.1).

```
C     STRUCTURE ANALYSIS OF ALG EQ'S
      SUBROUTINE ALGE(NT, NX, X, F, JX, JF, ID, MAXID)
      DIMENSION JX(1), JF(1)
      REAL*8 F(1), X(1), E, C(20)
      COMMON ALG / C
      E=DEXP(1.D0)
      NT=3
      DO 5 I=1, NT
      JX(I)=0
      JF(I)=0
    5 CONTINUE
      MAXIND=1
      IF(ID.EQ.1) GO TO 10
C......(ID=1)......
   10 CONTINUE
      NX=3
      F(1)=X(1)+X(2)+X(3)-C(1)
      F(2)=C(15)*X(1)**3+C(10)*X(2)**2-X(3)+C(10)*X(3)**2+C(10)
      F(3)=X(1)+X(2)+C(10)*X(3)**2-C(10)
      JX(1)=1
      JX(2)=1
      JX(3)=1
      JF(1)=1
      JF(2)=3
      JF(3)=2
      RETURN
C
      END
```

Fig. 4.6. Subroutine of system of equations decomposed after structure analysis.

(B) The deflation category $=1$.

(C) The deflation map $M^{[0]}$ was given by

$$M^{[0]}=\left[\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \tag{4.3}$$

where the m_{ij} element of $M^{[0]}$ is defined by

$$(4.4) \qquad m_{ij}=\begin{cases} 0, & \text{if } \left|{}^6d_{ij}/{}^5d_{ij}\right| \leq 0.5\,, \\ 1, & \text{otherwise,} \end{cases}$$

and d_{ij} is given by (3.14). The map shows that (a) the (4, 4)-th element is the number of the category, (b) the ones in the fourth column mean that the first and third rows are proportional, and (c) a zero in the column has numerical zeros in the corresponding row.

(D) The diagonal elements $e_{jj}({}^kx)$ in (3.15) were computed to be

$$(4.5) \qquad [|e_{jj}({}^6x)/e_{jj}({}^5x)|]=[1.000,\ 0.505,\ 0.500].$$

From the properties (3.16) in Theorem 3.2 and (4.5), the rank of the Jacobian matrix at the solution is expected to be one.

(E) The pivot matrix was given by

$$(4.6) \qquad P_1^{[0]}({}^6x)=\begin{array}{c c} f_i^{[0]} & {}^6x_j \\ [\,1 & 1\,] \end{array}.$$

(iv) Decoding these informations (A)-(E) by the translator, the Jacobian matrix

$$(4.7) \qquad F_x=\begin{bmatrix} 1.0 & 1.0 & 1.0 \\ 0.6x_1^2 & x_2 & -1.0+x_3 \\ 1.0 & 1.0 & x_3 \end{bmatrix},$$

is generated by using the symbolic differentiation of the REDUCE 2. From (4.7), we have

$$(4.8) \qquad \det F_x^{[0]}(x)\Big|_{x=x^{*1}}=\det\begin{bmatrix} 1.0 & 1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 1.0 \end{bmatrix}=0\,.$$

It is easily seen from (4.8) that, as would be expected, the rank of the Jacobian matrix (4.7) is one.

(v) Since there are numerical zeros in (4.3), Category 1 can be applied to (4.7). In fact, from the (2, 2)- and (2, 3)- elements of (4.7), new pivot variables x_2 and x_3 which are not in the pivot matrix (4.6) can be obtained. Thus the following deflated equations are generated by the REDUCE 2:

$$(4.9) \qquad f^{[1]}(x) = \begin{bmatrix} x_1 + x_2 + x_3 - 1.0 \\ x_2 \\ x_3 - 1.0 \end{bmatrix} = 0.0 .$$

It is easily seen that the rank of the Jacobian matrix of (4.9) at the solution x^{*1} is three. Thus the first deflated equations (4.9) have a set of simple roots.

(vi) The original equations (4.1) with $x={}^kx$ $(k=0, 1, \cdots, 6)$ and the deflated equations (4.9) with $x={}^kx$ $(k=7, 8, 9)$ were solved by the ε-secant method in Subsection 2.2. The convergence tendencies of ${}^kE^{[l]}$ with and without the deflation are given in Table 4.1. As would be expected, the deflation algorithm resulted in faster convergence as well as higher accuracy as shown in Table 4.2.

Table 4.1. Convergence tendencies

no. of iteration	no. of deflation	with deflation	without deflation
0	0	0.103×10 (0)	0.103×10 (0)
1	0	0.569×10 (−1)	0.569×10 (−1)
2	0	0.157×10 (−1)	0.157×10 (−1)
3	0	0.394×10 (−2)	0.349×10 (−2)
4	0	0.986×10 (−3)	0.986×10 (−2)
5	0	0.247×10 (−3)	0.247×10 (−3)
6	0	0.617×10 (−4)	0.617×10 (−4)
7	1	0.739×10 (−2)	0.154×10 (−4)
8	2	0.758×10(−13)	0.386×10 (−5)
9	3	0.0	0.964×10 (−6)
⋮			⋮
22			0.149×10(−13)
23			0.368×10(−14)

Table 4.2. Numerical solutions

x	with deflation	without deflation
x_1	0.0	0.32895108875546×10 (−7)
x_2	−0.16787888226717×10(−18)	0.59028952604004×10 (−7)
x_3	1.0	0.99999990807594×10 (0)

§ 5. Concluding Remarks

In this paper, the hybrid manipulation with both numerical and symbolic manipulations for the solution of a system of nonlinear algebraic equations has been presented. The ε-secant method which is a numerical realization of the Newton-Raphson method for the simple roots was given and several properties of the method were discussed first. According to the method, it is not necessary to provide analytically the Jacobian matrix in advance and under the appropriate conditions a quadratic convergence can be obtained.

Several properties of the multiple root were then discussed and the deflation algorithm for finding the root has been developed, in which four types of categories were given and to realize these categories both the numerical manipulation language FORTRAN and symbolic manipulation language REDUCE 2 are incorporated efficiently.

In order to attain smooth communications between FORTRAN and REDUCE 2, we need some procedures. The outlines of them were given by showing examples.

Lastly the example of S̄amanskii with quadruple roots was solved by the present methods and the roots have been obtained in sufficient accuracies. This fact shows that the present methods can be used practically.

The methods in this paper have been put into the package NAES for the solution of a system of nonlinear algebraic equations. In the package, the algorithms for the structure analysis of the system [15, 16] and for finding multi-roots are also included. However these are omitted here.

All the computations were done on the DEC-System 2020 in the Computer Programming Laboratory of the Research Institute for the Mathematical Sciences, Kyoto University, Kyoto.

Acknowledgements

The author wishes to appreciate Professor T. Ojika of Osaka Kyoiku University and Professor T. Mitsui of Fukui University for many useful and valuable disccussions in the course of this work. He also would like to acknowledge to the reference for the remarks about S̄amanskii's works.

References

[1] Blum, E. K., *Numerical Analysis and Computation Theory and Practice*, Addison-Wesley Publ. Comp., Readings, Mass. 1972.

[2] Branin, F. H., Jr., Widely Convergent Method for Finding Multiple Solutions of Simultaneous Nonlinear Equations, *IBM J. Research and Development*, **16** (1972), 504-521.

[3] Decker, D. W. and Kelley, C. T., Newton's Method at Singular Point. I, *SIAM J. Numer. Anal.*, **17** (1980), 66-70.

[4] ———, Newton's Method at Singular Point. II, *SIAM J. Numer. Anal.*, **17** (1980), 465-471.

[5] Griewank, A. O., Starlike Domains of Convergence for Newton's Method at Singularities, *Numer. Math.* **35** (1980), 95-111.

[6] Griewank, A. O. and Osborne, M. R., Newton's Method for Singular Problems When the Dimension of the Null Space is >1, *SIAM J. Numer. Anal.* **18** (1981), 145-149.

[7] Hearn, A. C., *REDUCE 2 User's Manual*, Univ. of Utah, 1973.

[8] Henrici, P., *Discrete Variable Methods in Ordinary Differential Equations*, Wiley, New York, 1962.

[9] Kantorovich, L., On Newton's Method for Functional Equations, *Dokl. Akad. Nauk USSR*, **59** (1948), 1237-1240.

[10] Krasnosel'skii et al., *Approximate Solution of Operator Equations*, Wolters-Noordhoff Publ., 1972.

[11] Mitsui, T., On the Convergence of the Initial-Value Adjusting Method for Nonlinear Boundary Value Problems, *Publ. RIMS Kyoto Univ.*, **16** (1980), 513-529.

[12] ———, The Initial-Value Adjusting Method for Problems for the Least Square Type of Ordinary Differential Equations, *Publ. RIMS Kyoto Univ.*, **16** (1980), 785-810.

[13] Ojika, T., On a Quadratic Convergence of the Initial-Value Adjusting Method for Nonlinear Multipoint Boundary Value Problems, *J. Math. Anal. Appl.*, **73** (1980), 192-203.

[14] ———, Deflation Algorithm for Multiple Roots of Simultaneous Nonlinear Equations, *Memo. Osaka Kyoiku Univ., Ser.* III, **30** (1982), 197-209.

[15] ———, Structure Analysis for Large Scale Nonlinear Multipoint Boundary Value Problems, *to appear in J. Math. Anal. Appl.*

[16] ———, Structure Analysis for Large Scale Nonlineare Equations, to appear in *Memo. Osaka Kyoiku Univ.*

[17] Ojika, T. and Kasue, Y., Initial-Value Adjusting Method for the Solution of Nonlinear Multipoint Boundary Value Problems, *J. Math. Anal. Appl.*, **69** (1979), 359-371.

[18] Ojika, T., Mitsui, T. and Watanabe, S., *Manual for Nonlinear Algebraic Equations Solver NAES, in preparation.*

[19] Ojika, T., Watanabe, S. and Mitsui, T., The Deflation Algorithm for the Multiple Roots of a System of Equations, *to appear in J. Math. Anal. Appl.*

[20] Ortega, J. M. and Rheinboldt, C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York or London, 1970.

[21] Rall, L. B., Convergence of the Newton Process to Multiple Solutions, *Numer. Math.*, **9** (1966), 23-37.

[22] Reddien, G. W.. On Newton's Method for Singular Problems, *SIAM J. Numer. Anal.*, **15** (1978), 993-997.

[23] $\bar{\text{S}}$amanskii, V., A Realization of the Newton Method on a Computer (Russian), *Ukrain. Math. $\bar{Z}$.*, **18** (1966), 135-140.

[24] ————, On a Modification of the Newton Method (Russian), *Ukrain. Math. Ž.*, **19** (1967), 133-138.

[25] ————, The Application of Newton's Method in the Singular Case, *USSR J. Comp. Math. Phy.*, **7** (1967), 774-783.

[26] Stoer, J. and Bulirsch, R., *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1980.

[27] Urabe, M., The Newton Method and its Application to Boundary Value Problems with Nonlinear Boundary Conditions, *Proceedings U.S.-Japan Seminar on Differential and Functional Equations*, Benjamin, New York, (1967), 383-410.

[28] Watanabe, S., On the Deflation Algorithm for Multiple Roots of Systems of Nonlinear Algebraic Eqations and the Order of Convergence, *Bull, Yamagata Univ.*, **10** (1982), 245-263.

[29] Watanabe, S., Ojika, T. and Mitsui, T., On the Quadratic Convergence Properties of the ε-secant Method for the Solution of Systems of Nonlinear Equations and its Application to a Chemical Reaction Problem, *J. Math. Anal. Appl.*, **94** (1983).

[30] Welsh, W. and Ojika, T., Multipoint Boundary Value Problems with Discontinuities II. Convergence of the Initial-Value Adjusting Method, *J. Comput. Appl. Math.*, **6** (1980), 183-187.