SIAM J. Numer. Anal.
Vol. 10, No. 2, April 1973

SOME EFFICIENT ALGORITHMS FOR SOLVING SYSTEMS OF NONLINEAR EQUATIONS*

RICHARD P. BRENT†

Abstract. We compare the Ostrowski efficiency of some methods for solving systems of nonlinear equations without explicitly using derivatives. The methods considered include the discrete Newton method, Shamanskii's method, the two-point secant method, and Brown's methods. We introduce a class of secant methods and a class of methods related to Brown's methods, but using orthogonal rather than stabilized elementary transformations. The idea of these methods is to avoid finding a new approximation to the Jacobian matrix of the system at each step, and thus increase the efficiency. Local convergence theorems are proved, and the efficiencies of the methods are calculated. Numerical results are given, and some possible extensions are mentioned.

1. Introduction. We are interested in comparing iterative processes for approximating a solution $\mathbf{x}^*$ of a system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ of nonlinear equations. If $\mathbf{x}_0, \mathbf{x}_1, \cdots$ is a convergent sequence of vectors with limit $\mathbf{x}^* \in R^n$, then the *order of convergence* ρ is defined by

$$\rho = \liminf_{i\to\infty}(-\log \|\mathbf{x}_i - \mathbf{x}^*\|)^{1/i}. \tag{1}$$

It does not matter which of the usual vector norms is used in (1). Other definitions of order may be given (see Ortega and Rheinboldt (1970, Chap. 9), Voigt (1971), and Brent (1972b, § 3.2)), but (1) is adequate for our purposes. We only consider processes for which $\rho > 1$, and in this case ρ is the same as the R-order of Ortega and Rheinboldt (1970).

If w_i is the amount of work required to compute $\mathbf{x}_i$ from $\mathbf{x}_{i-1}$ and other results which may have been saved from previous iterations, then the *efficiency* E of the process is defined by

$$E = \liminf_{i\to\infty}\left(\frac{\log(-\log \|\mathbf{x}_i - \mathbf{x}^*\|)}{\sum_{j=1}^{i} w_j}\right). \tag{2}$$

In particular, if there exists $w = \lim_{i\to\infty} w_i > 0$, then $E = (\log \rho)/w$ is the logarithm of the "efficiency index" of Ostrowski (1960, § 3.11). The w_i may be measured in any appropriate units: we mainly use function evaluations, i.e., evaluations of $\mathbf{f}$.

Consider iterative methods M and M' with orders ρ, ρ' and efficiencies E, E'. For simplicity, suppose that the w_i are bounded and the lower limits in (1) and (2) may be replaced by limits. Our justification for the term "efficiency" is that method M requires E'/E times as much work as method M' to reduce $\|\mathbf{x}_i - \mathbf{x}^*\|$ to a very small positive tolerance. Thus, if factors such as the domains of convergence, ease of implementation, and storage space required are comparable, the method with the higher efficiency is to be preferred, and this is not always the method with the higher order. (As a trivial illustration, consider taking every second iterate of M as an iterate of M', so $\mathbf{x}'_i = \mathbf{x}_{2i}$ and $w'_i = w_{2i-1} + w_{2i}$. Then $\rho' = \rho^2 > \rho$, but

* Received by the editors January 13, 1972.

† This author received his Ph.D. in Computer Science in 1971 from Stanford University under the direction of Professors Forsythe and G. Golub. He is now a Research Fellow in the Computer Centre at the Australian National University, Canberra, Australia. Most of the work in this paper was performed while the author was visiting the IBM Thomas J. Watson Research Center.

$E' = E$.) Various authors have studied the efficiency of algorithms for finding zeros of functions of one variable, e.g., Traub (1964), (1971), Feldstein and Firestone (1969). Apart from the work of Shamanskii (1967), which is summarized in § 2, we do not know of other studies of the efficiency (as distinct from order) of algorithms for functions of several variables.

In § 3 we describe a class $\{S_k | k = 1, 2, \cdots \}$ of "Newton-like" methods (Dennis (1968)) for solving systems of nonlinear equations. S_1 is the "two-point secant" method of Ortega and Rheinboldt (1970, §§ 7.2 and 11.2), Korganoff (1961), Robinson (1966), Schmidt (1966), and Voigt (1971). The idea of methods S_2, S_3 etc. is to use the same approximation to the Jacobian of the system for several Newton steps in an attempt to increase the efficiency.

In § 4 we describe an interesting class $\{T_k | k = 1, 2, \cdots\}$ of methods based on orthogonal triangularization of an approximation to the Jacobian. T_1 is similar to Brown's method (Brown and Conte (1967)), the main difference being our use of orthogonal transformations instead of elementary stabilized transformations (Wilkinson (1965, § 3.47)). T_2, T_3 etc. use the same approximate factorization of the Jacobian for several Newton steps, in much the same way as the methods suggested by Brown (1968) (we call these Brown's *modified* methods to avoid confusion with his earlier method).

Local convergence theorems, proved in § 5, show that method S_k gives convergence with order at least $\rho_S(k) = \frac{1}{2}(k + \sqrt{k^2 + 4})$, and T_k gives convergence with order at least $\rho_T(k) = k + 1$, under fairly weak conditions on $\mathbf{f}$. Using these results, we choose k (depending on n) to give methods of optimal efficiency.

In § 6 we compare the efficiencies of several methods, including the discrete Newton method, Brown's methods, and the methods S_k and T_k. In computing the efficiencies of the various methods we count only function evaluations and ignore overhead. Since the methods considered may be implemented with an overhead of $O(n)$ operations per evaluation of each component $f_i(x_1, \cdots, x_n)$ of $\mathbf{f}(\mathbf{x})$, this approximation is reasonable if the Jacobian J of $\mathbf{f}$ is dense. Our conclusions may be invalid if J is sparse and the components f_i are easy to evaluate.

It is shown that methods S_k and T_k are more efficient than the discrete Newton method or Brown's (unmodified) method, provided k is suitably chosen. This conclusion is supported by some numerical results given in § 7. Finally, in § 8, we mention how the idea of maximizing efficiency can also be applied to classes of methods for minimizing functions and finding eigenvalues.

2. An illustration: Shamanskii's method. Before becoming too involved in technical details, we consider a simple example. For $k \geqq 1$, let N_k be the Newton-like method for which $\mathbf{x}_{i+1}$ is generated from $\mathbf{x}_i$ in the following way:

1. For a sufficiently small step size h_i (of order $\|\mathbf{f}(\mathbf{x}_i)\|$), compute the matrix J_i whose jth column is $J_i\mathbf{e}_j = (\mathbf{f}(\mathbf{x}_i + h_i\mathbf{e}_j) - \mathbf{f}(\mathbf{x}_i))/h_i$ for $j = 1, 2, \cdots, n$. (Here $\mathbf{e}_j$ is the jth column of the identity matrix.)

2. Perform k "Newton iterations" with the approximate Jacobian J_i (assumed nonsingular), i.e., define $\mathbf{x}_{i+1} = \mathbf{y}_{i,k}$, where $\mathbf{y}_{i,0} = \mathbf{x}_i$, and $\mathbf{y}_{i,j} = \mathbf{y}_{i,j-1} - J_i^{-1}\mathbf{f}(\mathbf{y}_{i,j-1})$ for $j = 1, 2, \cdots, k$.

Traub (1964, § 11.3) and Shamanskii (1967) have shown that, under certain conditions, method N_k gives a sequence which converges to a zero of $\mathbf{f}$ with order at least $k + 1$. If the h_i are chosen sufficiently small, conditions similar to those given in Ortega and Rheinboldt (1970, § 10.1.7) ensure that the order is exactly $k + 1$. Each iteration after the first requires $n + k$ evaluations of $\mathbf{f}$, so, neglecting computations other than evaluations of $\mathbf{f}$, the efficiency of the method is

$$E(N_k) = \frac{\log(k + 1)}{n + k}.$$

In particular, the usual discrete Newton method (N_1) requires $n + 1$ evaluations of $\mathbf{f}$ per iteration, and has efficiency $E(N_1) = \log 2/(n + 1)$.

If $E(N_k)$ attains its maximum value (over positive integers k) of $E_N(n)$ at $k = k_N(n)$, then the optimal method from the class $\{N_1, N_2, \cdots\}$ is $N_{k_N(n)}$. Note that the optimal value of k depends on n. In Table 1 we give $k_N(n)$ and $E_N(n)/E(N_1)$ for various values of n.

The table shows that, for all $n \geqq 1$, the method $N_{k_N(n)}$ is more efficient than the usual discrete Newton method N_1. For $n = 1$ the difference is only slight, as pointed out by Ostrowski (1960, Appendix G), but the difference is appreciable if $n > 1$. For example, if $n = 100$ and a very accurate solution is required, then N_1 uses about 3.9 times more function evaluations than N_{37}. In practice the difference is not so marked, because very high accuracy is seldom required, but the optimal method may be exoected to use less function evaluations than N_1 does.

TABLE 1
The efficiency of Shamanskii's optimal method N_k compared with that of the usual discrete Newton method N_1

n	$k_N(n)$	$E_N(n)/E(N_1)$
1	2	1.06
2	3	1.20
3	3	1.33
4	4	1.45
5	5	1.55
10	7	1.94
20	11	2.43
50	22	3.20
100	37	3.87
1000	225	6.39

3. A class of secant methods. Suppose that $k \geqq 1$ and $\mathbf{x}_0$, $\mathbf{x}'_0$ are distinct approximations to a zero $\mathbf{x}^*$ of the system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ of n nonlinear equations in n unknowns. We shall describe an algorithm S_k which, under certain conditions on $\mathbf{f}$ and the initial approximations $\mathbf{x}_0$ and $\mathbf{x}'_0$, generates a sequence $(\mathbf{x}_i)$ with limit $\mathbf{x}^*$. If $\mathbf{x}_i$ and $\mathbf{x}'_i$ have been generated, then $\mathbf{x}_{i+1}$ and $\mathbf{x}'_{i+1}$ are found in the following way: If $\mathbf{f}(\mathbf{x}_i) = \mathbf{0}$, then $\mathbf{x}_{i+1} = \mathbf{x}'_{i+1} = \mathbf{x}_i$; otherwise:

1. Find an orthogonal matrix Q_i such that

$$\mathbf{x}'_i = \mathbf{x}_i + h_i Q_i \mathbf{e}_1, \tag{3}$$

where $h_i = \|\mathbf{x}_i - \mathbf{x}'_i\|$. (We always use the Euclidean vector norm, and the induced matrix norm, unless otherwise specified.) For example, Q_i may be found as an elementary Hermitian, i.e., a matrix of the form $\pm(I - 2\mathbf{u}_i\mathbf{u}_i^T)$, where $\|\mathbf{u}_i\| = 1$ (see Householder (1964), Parlett (1971), and Wilkinson (1965)).

2. Find the matrix A_i whose jth column is $A_i\mathbf{e}_j = (\mathbf{f}(\mathbf{x}_i + h_iQ_i\mathbf{e}_j) - \mathbf{f}(\mathbf{x}_i))/h_i$ for $j = 1, \cdots, n$. Note that an evaluation of $\mathbf{f}$ can be saved, by using (3), when $j = 1$ and $\mathbf{f}(\mathbf{x}_i')$ is known.

3. Perform k Newton iterations with the approximate Jacobian $J_i = A_iQ_i^T$, i.e., let $\mathbf{y}_{i,0} = \mathbf{x}_i$ and compute

$$\mathbf{y}_{i,j} = \mathbf{y}_{i,j-1} - J_i^{-1}\mathbf{f}(\mathbf{y}_{i,j-1}) \quad \text{for } j = 1, \cdots, k.$$

4. Let $\mathbf{x}_{i+1} = \mathbf{y}_{i,k}$ and $\mathbf{x}_{i+1}' = \mathbf{y}_{i,k-1}$.

The method S_k requires $n + k - 1$ evaluations of $\mathbf{f}$ for each iteration after the first. S_1 is the two-point secant method, and reduces to the usual secant method if $n = 1$. The idea of method S_k for $k > 1$ is to perform several Newton iterations (each requiring only one evaluation of $\mathbf{f}$) for every new approximation to the Jacobian, in an attempt to increase the efficiency of the method. The idea is not original, but our determination of the optimal value of k from the results of § 5 appears to be new.

4. A class of methods based on orthogonal triangularization. In this section we describe a class $\{T_k | k \geqq 1\}$ of methods which depend on the sequential evaluation of individual components $f_j(\mathbf{x})$ of $\mathbf{f}(\mathbf{x})$ at certain points $\mathbf{x}$. Some components are evaluated more often than others, so, for purposes of comparison with methods which evaluate all components equally often (such as the methods N_k and S_k), we assume that n component evaluations are equivalent to one function evaluation. This assumption may be unfair to the methods T_k if some components are easier to evaluate than others, for then it can be arranged that the "easy" components are evaluated more often than the "difficult" components. On the other hand, it may be easier to evaluate $\mathbf{f}(\mathbf{x})$ at one point $\mathbf{x}$ than to evaluate $f_1(\mathbf{x}_1), \cdots, f_n(\mathbf{x}_n)$ at distinct points $\mathbf{x}_1, \cdots, \mathbf{x}_n$. Thus, for particular systems of equations our comparison may be biased either way, and the reader should bear this in mind.

Method T_1 is similar to Brown's method (Brown and Conte (1967), Brown and Dennis (1972), and Brown (1969)). Brown's method reduces an approximate Jacobian of $\mathbf{f}$ to triangular form, using stabilized elementary transformations (i.e., Gaussian elimination with partial pivoting), whereas T_1 uses orthogonal transformations (either plane rotations or elementary Hermitians: see Givens (1958), Householder (1964), and Wilkinson (1965)). The use of orthogonal transformations gives greater numerical stability and simplifies the local convergence proof of § 5, but approximately doubles the overhead.

For $k > 1$, method T_k is the same as T_1, except that each factorization of an approximate Jacobian is used k times before a new approximate Jacobian is factored. Brown (1968) independently suggested a similar modification of his method. In § 5 we show how to choose k to maximize the efficiency of these methods.

Suppose that, after the ith iteration, we have an approximation $\mathbf{x}_i$ to $\mathbf{x}^*$, an orthogonal matrix Q_i, and a positive step size h_i. (Initially $\mathbf{x}_0$ and h_0 are given, and $Q_0 = I$.) Method T_k generates $\mathbf{x}_{i+1}$, Q_{i+1} and h_{i+1} in the following way.

(A geometric interpretation is given after the formal description.)

1. Let $Q_{i,1} = Q_i$ and $\mathbf{y}_{i,1,0} = \mathbf{x}_i$.
2. For $j = 1, \cdots, n$ do steps 3 to 5.
3. Compute

$$\mathbf{a}_{i,j} = \frac{1}{h_i} \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ 0 \\ f_j(\mathbf{y}_{i,j,0} + h_i Q_{i,j} \mathbf{e}_j) - f_j(\mathbf{y}_{i,j,0}) \\ \cdot \\ \cdot \\ \cdot \\ f_j(\mathbf{y}_{i,j,0} + h_i Q_{i,j} \mathbf{e}_n) - f_j(\mathbf{y}_{i,j,0}) \end{pmatrix}.$$

4. Find an orthogonal matrix $P_{i,j}$, of the form

$$P_{i,j} = \left(\begin{array}{c|c} I_{(j-1)\times(j-1)} & 0 \\ \hline 0 & \hat{P}_{i,j} \end{array}\right),$$

such that $P_{i,j}^T \mathbf{a}_{i,j} = s_{i,j} \mathbf{e}_j$, where $s_{i,j} = \pm \|\mathbf{a}_{i,j}\|$. (For example, $\hat{P}_{i,j}$ may be an elementary Hermitian, or a product of $n - j$ plane rotations.)

5. Compute $Q_{i,j+1} = Q_{i,j} P_{i,j}$ and $\mathbf{y}_{i,j+1,0} = \mathbf{y}_{i,j,0} - s_{i,j}^{-1} f_j(\mathbf{y}_{i,j,0}) Q_{i,j+1} \mathbf{e}_j$.
6. For $m = 1, \cdots, k - 1$ do step 7.
7. Let $\mathbf{y}_{i,1,m} = \mathbf{y}_{i,n+1,m-1}$ and, for $j = 1, \cdots, n$, compute

$$\mathbf{y}_{i,j+1,m} = \mathbf{y}_{i,j,m} - s_{i,j}^{-1} f_j(\mathbf{y}_{i,j,m}) Q_{i,n+1} \mathbf{e}_j.$$

8. Let $\mathbf{x}_{i+1} = \mathbf{y}_{i,n+1,k-1}$, $Q_{i+1} = Q_{i,n+1}$, and

$$h_{i+1} = \begin{cases} -s_{i,1}^{-1} f_1(\mathbf{x}_{i+1}) & \text{if } f_1(\mathbf{x}_{i+1}) \neq 0, \\ \text{sufficiently small} & \text{otherwise.} \end{cases}$$

(We must ensure that $h_{i+1} \neq 0$. In practice this is not difficult: if the stopping criterion is $\|\mathbf{x}_i - \mathbf{x}_{i+1}\| < t$ for some positive tolerance t, then we may take $h_{i+1} = t$ if $|s_{i,1}^{-1} f_1(\mathbf{x}_{i+1})| < t$.)

To make the formal description more comprehensible, we now give a geometric interpretation. If $\mathbf{x}_i$ is an approximation to a zero of $\mathbf{f}$, take $\mathbf{y}_{i,1,0} = \mathbf{x}_i$ and evaluate f_1 at $\mathbf{y}_{i,1,0}$ and a sufficient number (n) of nearby points to obtain a linear approximation to f_1. If the conditions of Theorem 2 (§ 5) are satisfied, this linear approximation vanishes on a flat (i.e., a translated linear subspace) V_1 of dimension $n - 1$. Let $\mathbf{y}_{i,2,0}$ be the point in V_1 closest in Euclidean distance to $\mathbf{y}_{i,1,0}$. (In Brown's method $\mathbf{y}_{i,2,0} - \mathbf{y}_{i,1,0}$ must also be parallel to a coordinate vector.) Now evaluate f_2 at $\mathbf{y}_{i,2,0}$ and a sufficient number ($n - 1$) of nearby points in V_1 to obtain a linear approximation to f_2 on V_1. This linear approximation vanishes on a flat V_2 of dimension $n - 2$. Let $\mathbf{y}_{i,3,0}$ be the point in V_2 closest to $\mathbf{y}_{i,2,0}$, etc.

For method T_1 (or Brown's unmodified method) the next approximation to a zero is $\mathbf{x}_{i+1} = \mathbf{y}_{i,n+1,0}$. For method T_k ($k \geqq 2$) an "iterative refinement" process is used to improve the approximation $\mathbf{y}_{i,1,1} = \mathbf{y}_{i,n+1,0}$. First $f_1(\mathbf{y}_{i,1,1})$ is evaluated, and a new linear approximation to f_1 is found from the value $f_1(\mathbf{y}_{i,1,1})$ and the previously computed approximation to the gradient of f_1. The new linear approximation to f_1 vanishes on a flat V'_1 parallel to V_1. Let $\mathbf{y}_{i,2,1}$ be the point in V'_1 closest in Euclidean distance to $\mathbf{y}_{i,1,1}$ (and such that the displacement is along a coordinate vector for Brown's modified method). Evaluate $f_2(\mathbf{y}_{i,2,1})$ to find a flat V'_2 parallel to V_2 and a point $\mathbf{y}_{i,3,1}$ in V'_2, etc. After finding $\mathbf{y}_{i,n+1,1}$, take $\mathbf{y}_{i,1,2} = \mathbf{y}_{i,n+1,1}$ and repeat the refinement process. Thus, approximations $\mathbf{y}_{i,n+1,0}$, $\mathbf{y}_{i,n+1,1}, \cdots, \mathbf{y}_{i,n+1,k-1}$ are generated, and finally $\mathbf{x}_{i+1} = \mathbf{y}_{i,n+1,k-1}$.

It is instructive to consider the case when $\mathbf{f}(\mathbf{x}) = J(\mathbf{x} - \mathbf{x}^*)$ is linear. After j iterations of steps 3 to 5 we have an orthogonal matrix $Q_{i,j+1}$ and a vector $\mathbf{y}_{i,j+1,0}$ such that $f_1(\mathbf{y}_{i,j+1,0}) = \cdots = f_j(\mathbf{y}_{i,j+1,0}) = 0$, and $JQ_{i,j+1}$ is a matrix of the form

$$\begin{pmatrix} s_{i,1} & 0 & \cdots & 0 & \cdots & 0 \\ x & s_{i,2} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ x & x & \cdots & s_{i,j} & \cdots & 0 \\ x & & \cdots & & & x \\ \vdots & & & & & \vdots \\ x & & \cdots & & & x \end{pmatrix}$$

where elements marked x are not determined. Thus, after n iterations of steps 3 to 5 we have a zero $\mathbf{y}_{i,n+1,0}$ of $\mathbf{f}$ and a factorization

$$J = LQ_{i,n+1}^T$$

of the Jacobian J, where $Q_{i,n+1}$ is orthogonal, and L is lower triangular with diagonal elements $s_{i,1}, \cdots, s_{i,n}$. Note that the strict lower triangle of L is not determined, so the method does not reduce to any of the usual methods for solving linear equations (but is related to Householder's triangularization).

The idea of the "iterative refinement" phase of methods T_2, T_3 etc. may be seen by supposing that $\mathbf{f}$ is linear but $\mathbf{y}_{i,n+1,0}$ is perturbed slightly from a zero of $\mathbf{f}$. Step 7 retrieves the zero with only one evaluation of each component of $\mathbf{f}$, by making use of the factorization $J = LQ_{i,n+1}^T$ and the known diagonal elements of L.

Our choice of h_{i+1} at step 8 ensures that the methods T_k and S_{k+1} (§ 3) generate sequences with the same order of convergence if $n = 1$. More precisely, suppose that $n = 1$ and T_k produces a sequence $\mathbf{x}_0^{(T)}, \mathbf{x}_1^{(T)}, \cdots$ with associated Q_i and h_i, and S_{k+1} produces sequences $\mathbf{x}_0^{(S)}, \mathbf{x}_1^{(S)}, \cdots$ and $\mathbf{x}'_0, \mathbf{x}'_1, \cdots$. If the initial conditions are such that

$$\begin{aligned} \mathbf{x}_i^{(S)} &= \mathbf{x}_i^{(T)} + h_i Q_i \mathbf{e}_1, \\ \mathbf{x}'_i &= \mathbf{x}_i^{(T)} \end{aligned} \tag{4}$$

for $i = 0$, then (4) holds for all $i \geqq 0$.

A complete iteration requires $n + k + 1 - j$ evaluations of the component f_j, for $j = 1, \cdots, n$. Thus, an iteration requires $\frac{1}{2}n(n + 2k + 1)$ component evaluations, or $\frac{1}{2}(n + 2k + 1)$ equivalent function evaluations, counting n component evaluations as one function evaluation. In § 5 we determine the order of convergence and use this to find the optimal value of k.

If the method T_k is implemented in an efficient way, using elementary Hermitians for the matrices $\hat{P}_{i,j}$, then the overhead is about $n^2(n + k)$ multiplications per iteration. For simplicity, we assume that a component evaluation requires considerably more than n multiplications, so the overhead is negligible. This is often true in practical problems when, for example, the evaluation of $f_j(\mathbf{x})$ may involve the solution of a system of differential equations. If the components f_j are easy to evaluate, then the overhead must be taken into account in comparing different methods. Note that, to avoid excessive overhead in computing the last $n + 1 - j$ columns of $Q_{i,j}$ at step 3, it is best to compute the matrix $Q_{i,j}$ explicitly, instead of keeping it as a product of elementary Hermitians. Thus, the storage required is $n^2 + O(n)$ floating-point words. (If $k = 1$, then this may be reduced to $\frac{1}{2}n^2 + O(n)$ words, at the expense of slightly increasing the overhead, by using products of plane rotations instead of elementary Hermitians at step 4.)

5. Local convergence theorems. In this section we give local convergence theorems for the algorithms described in §§ 3 and 4. Since our object is to deduce the relative efficiency of the different algorithms from results on their orders of convergence, we are willing to assume the existence of a solution $\mathbf{x}^*$. Convergence theorems without this assumption are also possible, in the style of Brown and Dennis (1968), Dennis (1967), (1968), Kantorovich and Akilov (1964), and Ortega and Rheinboldt (1970, § 12.6).

THEOREM 1. *Suppose that* $\varepsilon > 0, n \geqq 1, S = \{\mathbf{x} \in R^n \mid \|\mathbf{x} - \mathbf{x}^*\| < 3\varepsilon\}, \mathbf{f}: S \to R^n$ *is Fréchet differentiable,* $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$, *the Jacobian* $J(\mathbf{x}) = (\partial f_i/\partial x_j)$ *satisfies the Lipschitz condition* $\|J(\mathbf{x}) - J(\mathbf{y})\| \leqq M\|\mathbf{x} - \mathbf{y}\|$ *for all* $\mathbf{x}, \mathbf{y} \in S$, *and*

$$c\varepsilon < 1, \tag{5}$$

where

$$c = M\|J^{-1}(\mathbf{x}^*)\|(\tfrac{5}{2} + 2\sqrt{n}). \tag{6}$$

If $\mathbf{x}_0 \neq \mathbf{x}'_0$, $\|\mathbf{x}_0 - \mathbf{x}^*\| < \varepsilon$, $\|\mathbf{x}'_0 - \mathbf{x}^*\| < \varepsilon$, *and* $k \geqq 1$, *then the algorithm* S_k *of* § 3 *is well-defined, generates a sequence* $(\mathbf{x}_i)$, *and* $\mathbf{x}_i \to \mathbf{x}^*$ *with order at least* $\rho_S(k) = (k + \sqrt{k^2 + 4})/2$.

THEOREM 2. *Suppose that* k, n, ε, S, $\mathbf{f}$, *and* $\mathbf{x}^*$ *are as in Theorem* 1, $J(\mathbf{x}^*)$ *nonsingular,* $h_0 < \varepsilon$, *and* $\|\mathbf{x}_0 - \mathbf{x}^*\| < \varepsilon$. *If* ε *is sufficiently small, then the algorithm* T_k *of* § 4 *generates a sequence* $(\mathbf{x}_i)$, *and* $\mathbf{x}_i \to \mathbf{x}^*$ *with order at least* $\rho_T(k) = k + 1$. (This result is not sharp if $n = 1$, when the order is at least $\rho_S(k + 1) > k + 1$: see our comment about the choice of h_{i+1} in § 4.)

Theorems 1 and 2 give lower bounds $\rho_S(k)$ and $\rho_T(k)$ on the orders of convergence of the algorithms S_k and T_k, and in practice the orders are usually equal to these lower bounds. (Perhaps results similar to those of Voigt (1971) could be established.) Thus, the efficiency of algorithm S_k is

$$E(S_k) = \frac{\log \rho_S(k)}{n + k - 1},$$

and the efficiency of T_k for $n > 1$ is

$$E(T_k) = \frac{2 \log \rho_T(k)}{n + 2k + 1}.$$

We may choose k to maximize $E(S_k)$ or $E(T_k)$. In § 6 we compare the efficiencies of various algorithms, including S_k and T_k with the optimal choice of k. The remainder of this section is devoted to a proof of Theorems 1 and 2, and may be omitted without loss of continuity.

LEMMA 1 (Ortega and Rheinboldt (1970, Thm. 3.2.12)). *With the assumptions of Theorem* 1,

$$\|\mathbf{f}(\mathbf{y}) - \mathbf{f}(\mathbf{x}) - J(\mathbf{x})(\mathbf{y} - \mathbf{x})\| \leqq \frac{M}{2}\|\mathbf{y} - \mathbf{x}\|^2 \quad \textit{for all } \mathbf{x}, \mathbf{y} \in S.$$

LEMMA 2. *With the assumptions of Theorem* 1, *if* A_i, J_i, *and* Q_i *are defined by the algorithm* S_k *of* § 3, $\mathbf{x}_i \in S$, *and* $\mathbf{x}_i + h_i Q_i \mathbf{e}_j \in S$ *for* $j = 1, \cdots, n$, *then*

$$\|J_i - J(\mathbf{x}_i)\| \leqq Mh_i\sqrt{n}/2.$$

Proof. For $j = 1, \cdots, n$ we have, from Lemma 1,

$$\|\mathbf{f}(\mathbf{x}_i + h_i Q_i \mathbf{e}_j) - \mathbf{f}(\mathbf{x}_i) - J(\mathbf{x}_i)h_i Q_i \mathbf{e}_j\| \leqq Mh_i^2/2,$$

so, by the definition of A_i, $\|(A_i - J(\mathbf{x}_i)Q_i)\mathbf{e}_j\| \leqq Mh_i/2$. Thus $\|A_i - J(\mathbf{x}_i)Q_i\| \leqq Mh_i\sqrt{n}/2$, and the result follows from the definition of J_i.

LEMMA 3. *With the assumptions of Theorem* 1, *suppose that* $\mathbf{x}_i \neq \mathbf{x}'_i$,

$$\varepsilon_i = \|\mathbf{x}_i - \mathbf{x}^*\|,$$

and

$$\varepsilon'_i = \max\{\|\mathbf{x}_i - \mathbf{x}^*\|, \|\mathbf{x}'_i - \mathbf{x}^*\|\} \leqq \varepsilon.$$

For $j = 0, \cdots, k$, *let* $\mathbf{y}_{i,j}$ *be defined by the algorithm* S_k *of* § 3. *Then, for* $j = 1, \cdots, k$,

(7) $$\|\mathbf{y}_{i,j-1} - \mathbf{x}^*\| \leqq \varepsilon_i$$

and

(8) $$\|\mathbf{y}_{i,j} - \mathbf{x}^*\| \leqq c\varepsilon'_i\|\mathbf{y}_{i,j-1} - \mathbf{x}^*\|.$$

Also,

(9) $$\varepsilon_{i+1} \leqq (c\varepsilon'_i)^k \varepsilon_i$$

and

(10) $$\varepsilon'_{i+1} \leqq (c\varepsilon'_i)^{k-1}\varepsilon_i.$$

Proof. $\|J(\mathbf{x}_i) - J(\mathbf{x}^*)\| \leqq M\varepsilon_i$, and $h_i \leqq 2\varepsilon'_i$, so Lemma 2 gives

(11) $$\|J_i - J(\mathbf{x}^*)\| \leqq (1 + \sqrt{n})M\varepsilon'_i.$$

Thus, from (5) and the Banach lemma, J_i is nonsingular and

$$\|J_i^{-1}\| \leqq \left(\frac{5 + 4\sqrt{n}}{3 + 2\sqrt{n}}\right)\|J^{-1}(\mathbf{x}^*)\|. \tag{12}$$

Assume for the moment that (7) holds. By (4) and the triangle inequality,

$$\|J_i(\mathbf{y}_{i,j} - \mathbf{x}^*)\| \leqq \|J(\mathbf{x}^*)(\mathbf{y}_{i,j-1} - \mathbf{x}^*) - \mathbf{f}(\mathbf{y}_{i,j-1})\| + \|J_i - J(\mathbf{x}^*)\| \cdot \|\mathbf{y}_{i,j-1} - \mathbf{x}^*\|.$$

Since $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$, Lemma 1 and (11) give

$$\|J_i(\mathbf{y}_{i,j} - \mathbf{x}^*)\| \leqq (\tfrac{3}{2} + \sqrt{n})M\varepsilon_i'\|\mathbf{y}_{i,j-1} - \mathbf{x}^*\|.$$

Thus, from (12) and (6),

$$\|\mathbf{y}_{i,j} - \mathbf{x}^*\| \leqq c\varepsilon_i'\|\mathbf{y}_{i,j-1} - \mathbf{x}^*\|.$$

This shows that (7) implies (8), but (7) holds for $j = 1$ (by the definition of $\mathbf{y}_{i,0}$), and $c\varepsilon_i' < 1$, so (7) and (8) hold for $j = 1, \cdots, k$, by induction. Now (9) and (10) are immediate from the definition of $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i+1}'$.

LEMMA 4. *Suppose that* $k \geqq 1$, $0 \leqq \gamma_0 < 1$, $0 \leqq \delta_0 < 1$, *and, for* $i \geqq 0$,

$$0 \leqq \gamma_{i+1} \leqq \gamma_i\delta_i^k \tag{13}$$

and

$$0 \leqq \delta_{i+1} \leqq \gamma_i\delta_i^{k-1}. \tag{14}$$

Then $\gamma_i \to 0$ *with order at least* $(k + \sqrt{k^2 + 4})/2$.

Proof. We may suppose that all $\gamma_i > 0$, for otherwise the result is trivial. Let

$$\mathbf{u}_i = \begin{pmatrix} -\log \gamma_i \\ -\log \delta_i \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} 1 & k \\ 1 & k - 1 \end{pmatrix}.$$

From (13) and (14), $\mathbf{u}_{i+1} \geqq U\mathbf{u}_i$, so

$$\mathbf{u}_i \geqq U^i\mathbf{u}_0.$$

Now

$$U = Q\Lambda Q^T,$$

where

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix},$$

$$\lambda_1 = \tfrac{1}{2}(k + \sqrt{k^2 + 4}),$$

$$\lambda_2 = \tfrac{1}{2}(k - \sqrt{k^2 + 4}),$$

$$Q = \begin{pmatrix} c & -s \\ s & c \end{pmatrix},$$

$c > 0, s > 0$, and $c^2 + s^2 = 1$. Thus

$$\mathbf{u}_i \geqq Q\Lambda^i Q^T \mathbf{u}_0,$$

giving

$$\log \gamma_i \leqq (c^2 \log \gamma_0 + sc \cdot \log \delta_0)\lambda_1^i + O(\lambda_2^i) \quad \text{as } i \to \infty.$$

Since $\gamma_0 < 1, \delta_0 < 1, c > 0, s > 0$, and $\lambda_1 > |\lambda_2|$, it follows that

$$\liminf_{i \to \infty} (-\log \gamma_i)^{1/i} \geqq \lambda_1,$$

which is the desired result.

Proof of Theorem 1. Suppose that $\mathbf{x}_i$ and $\mathbf{x}'_i$ are defined and distinct, with ε_i and ε'_i as in Lemma 3. If $\mathbf{x}_i = \mathbf{x}^*$, then the result is trivial ($\mathbf{x}_{i+1} = \mathbf{x}_{i+2} = \cdots = \mathbf{x}^*$), so suppose that $\mathbf{x}_i \neq \mathbf{x}^*$. By Lemma 3, $\varepsilon_{i+1} \leqq \varepsilon'_{i+1} \leqq \varepsilon_i$. Also, from (8) with $j = k$, $\mathbf{x}'_{i+1} \neq \mathbf{x}_{i+1}$ unless $\mathbf{x}_{i+1} = \mathbf{x}^*$ (when the result is trivial). It follows, by induction on i, that the algorithm is well-defined and $\varepsilon'_i \leqq \varepsilon$ for all $i \geqq 0$.

Let $\gamma_i = c\varepsilon_i$ and $\delta_i = c\varepsilon'_i$. From Lemmas 3 and 4, $\gamma_i \to 0$ with order at least $(k + \sqrt{k^2 + 4})/2$, so the proof is complete. (It can also be shown that the sequence of function norms $\|\mathbf{f}(\mathbf{x}_i)\|$ is eventually monotonic decreasing, and tends to 0 with order at least $(k + \sqrt{k^2 + 4})/2$.)

LEMMA 5. *If $U = (u_{i,j})$ is nonsingular and triangular, then $|u_{i,i}| \geqq 1/\|U^{-1}\|$ for $i = 1, \cdots, n$.*

Proof. Let $U^{-1} = V = (v_{i,j})$. Since U and V are triangular, the (i, i)th element of UV is $u_{i,i}v_{i,i} = 1$, but $|v_{i,i}| \leqq \|V\|$, so the result follows.

LEMMA 6. *With the assumptions of Theorem* 2 *and the notation of* § 4, *suppose that $1 \leqq j \leqq n, h_i < \varepsilon, \|\mathbf{y}_{i,1,0} - \mathbf{x}^*\| \leqq \varepsilon, \cdots, \|\mathbf{y}_{i,j,0} - \mathbf{x}^*\| \leqq \varepsilon$. There is a constant c_1 such that, for all sufficiently small ε,*

$$\|\mathbf{y}_{i,j+1,0} - \mathbf{x}^*\| \leqq c_1 \|\mathbf{y}_{i,j,0} - \mathbf{x}^*\|$$

and

$$|s_{i,j}| \geqq 1/(2\|J^{-1}(\mathbf{x}^*)\|). \tag{15}$$

Proof. Write $J = J(\mathbf{x}^*)$ for brevity, keep i and j fixed, and let $L = (m_{p,q})$, where

$$m_{p,q} = \begin{cases} \mathbf{e}_p^T J Q_{i,j+1} \mathbf{e}_q & \text{if } 1 \leqq q < p \leqq n \text{ or } j < p \leqq q \leqq n, \\ s_{i,p} & \text{if } 1 \leqq p = q \leqq j, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose $p \leqq j$. By Lemma 1 and the construction of $\mathbf{a}_{i,p}$ (step 3 of the algorithm),

$$|(\mathbf{a}_{i,p}^T - \mathbf{e}_p^T J(\mathbf{y}_{i,p,0})Q_{i,p})\mathbf{e}_q| \leqq \tfrac{1}{2}Mh_i \quad \text{for } q = p, p+1, \cdots, n.$$

Let $\mathbf{x}'$ denote the $(n - p + 1)$-vector formed from components $p, p + 1, \cdots, n$ of an n-vector $\mathbf{x}$. Then

$$\|[\mathbf{a}_{i,p} - Q_{i,p}^T J^T(\mathbf{y}_{i,p,0})\mathbf{e}_p]'\|^2 \leqq \left(\frac{Mh_i}{2}\right)^2 n.$$

However, by step 4 and the definition of L,

$$(P_{i,j}^T P_{i,j-1}^T \cdots P_{i,p}^T \mathbf{a}_{i,p})' = (s_{i,p}\mathbf{e}_p)' = (L^T\mathbf{e}_p)'.$$

Also, by step 5,

$$Q_{i,p}P_{i,p}P_{i,p+1} \cdots P_{i,j} = Q_{i,j+1},$$

so

$$\|[(L^T - Q_{i,j+1}^T J^T(\mathbf{y}_{i,p,0}))\mathbf{e}_p]'\|^2 \leqq \left(\frac{Mh_i}{2}\right)^2 n,$$

i.e.,

$$\sum_{q=p}^{n} [\mathbf{e}_p^T(L - J(\mathbf{y}_{i,p,0})Q_{i,j+1})\mathbf{e}_q]^2 \leqq \left(\frac{Mh_i}{2}\right)^2 n.$$

Now $\|\mathbf{y}_{i,p,0} - \mathbf{x}^*\| \leqq \varepsilon$, so

$$\|J(\mathbf{y}_{i,p,0}) - J(\mathbf{x}^*)\| \leqq M\varepsilon,$$

and thus

$$\|(L - JQ_{i,j+1})^T\mathbf{e}_p\| \leqq \tfrac{1}{2}Mh_i\sqrt{n} + M\varepsilon \leqq \tfrac{3}{2}M\varepsilon\sqrt{n}.$$

Since the last $n - j$ rows of L are the same as those of $JQ_{i,j+1}$, it follows that

$$\|L - JQ_{i,j+1}\| \leqq \tfrac{3}{2}nM\varepsilon.$$

Thus $\|L^{-1}\| \leqq 2\|J^{-1}\|$, provided ε is so small that $3nM\varepsilon\|J^{-1}\| \leqq 1$.

There is an orthogonal matrix H, of the form $\left(\begin{array}{c|c} I_{j\times j} & 0 \\ \hline 0 & \hat{H} \end{array}\right)$, such that LH is lower triangular. Thus, from Lemma 5,

$$|s_{i,j}| = |m_{j,j}| \geqq 1/\|L^{-1}\| \geqq 1/(2\|J^{-1}\|),$$

proving (15).

By the definition of $\mathbf{y}_{i,j+1,0}$ (see § 4),

$$\|\mathbf{y}_{i,j+1,0} - \mathbf{x}^*\| \leqq \|\mathbf{y}_{i,j,0} - \mathbf{x}^*\| + |s_{i,j}^{-1}| \cdot |f_j(\mathbf{y}_{i,j,0})|,$$

so the result follows from (15) and Lemma 1 if

$$c_1 = 1 + 2\|J^{-1}\|(1 + \|J\|).$$

LEMMA 7. *With the assumptions of Theorem* 2 *and the notation of* § 4, *suppose that* $0 \leqq m < k$, $h_i < \varepsilon$, *and* $\|\mathbf{x}_i - \mathbf{x}^*\| < \varepsilon$. *There is a constant* c_2 *such that, for all sufficiently small* ε,

(16) $$\|\mathbf{y}_{i,j,m} - \mathbf{x}^*\| \leqq c_2\|\mathbf{y}_{i,1,m} - \mathbf{x}^*\| \quad \textit{for } j = 1, \cdots, n+1,$$

and

(17) $$|s_{i,j}| \geqq 1/(2\|J^{-1}(\mathbf{x}^*)\|) \quad \textit{for } j = 1, \cdots, n.$$

Proof. Since $\mathbf{y}_{i,1,0} = \mathbf{x}_i$, Lemma 6 with $j = 1$ gives

$$\|\mathbf{y}_{i,2,0} - \mathbf{x}^*\| \leqq c_1\|\mathbf{y}_{i,1,0} - \mathbf{x}^*\|$$

and

$$|s_{i,1}| \geqq 1/(2\|J^{-1}\|).$$

Hence, if $\|\mathbf{y}_{i,1,0} - \mathbf{x}^*\|$ is sufficiently small, Lemma 6 with $j = 2$ gives

$$\|\mathbf{y}_{i,3,0} - \mathbf{x}^*\| \leqq c_1^2\|\mathbf{y}_{i,1,0} - \mathbf{x}^*\|$$

and

$$|s_{i,2}| \geqq 1/(2\|J^{-1}\|).$$

Proceeding in this way, we find that (16) with $m = 0$ holds if $c_2 \geqq c_1^n$. Also, (17) holds. The proof of (16) with $m = 1, 2, \cdots, k - 1$ follows from (17) as in the last part of the proof of Lemma 6. Finally, we note that from (16), (17) and the relations $\mathbf{y}_{i,1,m} = \mathbf{y}_{i,n+1,m-1}$, it follows that iteration i of algorithm T_k is well-defined, provided $\|\mathbf{x}_i - \mathbf{x}^*\|$ is sufficiently small.

LEMMA 8. *With the assumptions of Lemma 7, suppose that* $\|\mathbf{y}_{i,1,m} - \mathbf{x}^*\| < \varepsilon^{m+1}$. *There is a constant* c_3 *such that, for all sufficiently small* ε,

$$\|\mathbf{f}(\mathbf{y}_{i,n+1,m})\| \leqq c_3\varepsilon^{m+2}.$$

Proof. Fix j in the range $1 \leqq j \leqq n$. From Lemma 7, there is a constant c_5 such that

$$|f_j(\mathbf{y}_{i,j,m})| \leqq c_5\varepsilon^{m+1}.$$

As in the proof of Lemma 6, there is a constant c_6 such that

$$|s_{i,j} - \mathbf{e}_j^T J(\mathbf{x}^*)Q_{i,n+1}\mathbf{e}_j| \leqq c_6\varepsilon,$$

so there is a constant c_7 such that

$$|s_{i,j} - \mathbf{e}_j^T J(\mathbf{y}_{i,j,m})Q_{i,n+1}\mathbf{e}_j| \leqq c_7\varepsilon.$$

Hence, from (17),

$$\begin{aligned}|f_j(\mathbf{y}_{i,j,m}) - \mathbf{e}_j^T J(\mathbf{y}_{i,j,m})s_{i,j}^{-1}f_j(\mathbf{y}_{i,j,m})Q_{i,n+1}\mathbf{e}_j| &\leqq 2c_5c_7\|J^{-1}(\mathbf{x}^*)\|\varepsilon^{m+2}\\ &\leqq c_8\varepsilon^{m+2}, \quad \text{say.}\end{aligned}$$

By the definition of $\mathbf{y}_{i,j+1,m}$, this gives

$$|f_j(\mathbf{y}_{i,j,m}) - \mathbf{e}_j^T J(\mathbf{y}_{i,j,m})(\mathbf{y}_{i,j,m} - \mathbf{y}_{i,j+1,m})| \leqq c_8\varepsilon^{m+2}.$$

Thus, from Lemma 1,

$$|f_j(\mathbf{y}_{i,j+1,m})| \leqq c_8\varepsilon^{m+2} + \tfrac{1}{2}M\|\mathbf{y}_{i,j+1,m} - \mathbf{y}_{i,j,m}\|^2.$$

Since $\|\mathbf{y}_{i,j+1,m} - \mathbf{y}_{i,j,m}\|$ is of order ε^{m+1}, this gives

$$|f_j(\mathbf{y}_{i,j+1,m})| \leqq c_9\varepsilon^{m+2}.$$

Similarly, there is a constant c_{10} such that

$$|f_j(\mathbf{y}_{i,n+1,m}) - f_j(\mathbf{y}_{i,j+1,m})| \leqq c_{10}\varepsilon^{m+2},$$

so

$$|f_j(\mathbf{y}_{i,n+1,m})| \leqq (c_9 + c_{10})\varepsilon^{m+2}.$$

Thus, the lemma follows.

LEMMA 9. *With assumptions of Lemma 7, there is a constant c_4 such that, for all sufficiently small ε,*

$$\|\mathbf{y}_{i,n+1,m} - \mathbf{x}^*\| \leqq c_4\varepsilon^{m+2} \quad \textit{for } m = 0, \cdots, k-1.$$

Proof. This follows easily from (16), Lemma 1 and Lemma 8, by induction on m.

Proof of Theorem 2. If $\|\mathbf{x}_i - \mathbf{x}^*\|$ and h_i are of order ε_i, then, from Lemma 9, $\|\mathbf{x}_{i+1} - \mathbf{x}^*\|$ is of order ε_i^{k+1}. From Lemmas 7 and 9, h_{i+1} is also of order ε_i^{k+1}. Hence, the order of convergence is at least $k + 1$.

6. A comparison of the efficiencies of various methods. In this section we compare the efficiency (as defined in § 1) of the discrete Newton method, the two-point secant method, Brown's methods, and the methods of §§ 3 and 4 (with the optimal choice of k), for different values of n. In Table 2, the symbols are:

$n \geqq 2$, the number of variables;

$E(N_1) = (\log 2)/(n + 1)$, the efficiency of the discrete Newton method;

$E(S_1) = (\log [(1 + \sqrt{5})/2])/n$, the efficiency of the two-point secant method;

$E(T_1) = (2 \log 2)/(n + 3)$, the efficiency of Brown's method and of our orthogonal method T_1;

$E_S(n) = \max_{k \geqq 1} E(S_k)$, the efficiency of the optimal method S_k, attained at $k = k_S(n)$ (see § 5);

$E_T(n) = \max_{k \geqq 1} E(T_k)$, the efficiency of the optimal method T_k (or the corresponding modified Brown's method), attained at $k = k_T(n)$ (see § 5).

Table 2 shows that, for $n \geqq 2$, the optimal method T_k and the corresponding modified Brown's method are the most efficient of those considered. However, the reader should recall our comment at the beginning of § 4.

When n is large the discrete Newton method, the two-point secant method, and Brown's unmodified method are much less efficient than the optimal methods T_k and S_k. Shamanskii's optimal method N_k is slightly less efficient than the optimal method S_k, but the difference is only appreciable when n is small (see Table 1).

For large n the following asymptotic formulas hold:

$$k_N(n) \sim k_S(n) \sim k_T(2n) \sim n/\log(n/\log n),$$

$$E(S_1)/E(N_1) \sim \log_2((1 + \sqrt{5})/2) \doteqdot 0.69,$$

$$E(T_1)/E(N_1) \sim 2,$$

$$E_N(n)/E(N_1) \sim E_S(n)/E(N_1) \sim \log_2(n/\log n),$$

$$E_T(n)/E(N_1) \sim 2 \log_2(n/\log n).$$

TABLE 2
Comparison of the efficiencies of various methods

n	$k_S(n)$	$k_T(n)$	$E(S_1)/E(N_1)$	$E(T_1)/E(N_1)$	$E_S(n)/E(N_1)$	$E_T(n)/E(N_1)$
2	3	2	1.04	1.20	1.29	1.36
3	4	3	0.93	1.33	1.39	1.60
4	4	3	0.87	1.43	1.49	1.82
5	5	3	0.83	1.50	1.58	2.00
6	6	4	0.81	1.56	1.67	2.17
7	6	4	0.79	1.60	1.75	2.32
8	7	4	0.78	1.64	1.82	2.46
9	7	5	0.77	1.67	1.89	2.58
10	8	5	0.76	1.69	1.96	2.71
20	12	7	0.73	1.83	2.44	3.60
50	23	14	0.71	1.92	3.21	5.04
100	38	22	0.70	1.96	3.87	6.30
1000	226	128	0.69	2.00	6.39	11.17
10000	1572	866	0.69	2.00	9.18	16.64

In the comparison of different methods we have not mentioned some methods based on rank-one or rank-two updating of an approximation to the Jacobian (Broyden (1967), (1970), Dixon (1971)) or inverse Jacobian (Broyden (1965), (1970), Dennis (1971)). It is not known whether any of these methods are appreciably more efficient than the discrete Newton method, although numerical experience suggests that they may be. We have also omitted a comparison with Wolfe's secant method (Wolfe (1959), Bittner (1959), Tornheim (1964), and Barnes (1965)), for it seems unlikely that convergence of this method can be established without making assumptions about a certain determinant of normalized directions (see Brent (1972a)).

7. Some numerical results. In this section we give some numerical results to illustrate the behavior of the methods S_k and T_k described above. Most of the results are for the theoretically optimal k, but we also give some results with nonoptimal k for purposes of comparison. All results were obtained on an IBM 360/91 computer with 14-digit hexadecimal floating-point arithmetic.

Rosenbrock's function. This is a well-known function of two variables, defined by $f_1 = 10(x_2 - x_1^2)$ and $f_2 = 1 - x_1$. (The problem originally stated by Rosenbrock (1960) was to minimize $f_1^2 + f_2^2$.) The initial guess is $(-1.2, 1.0)^T$. Method T_1 (with $h_0 = 0.1$) reduces $\|\mathbf{x} - \mathbf{x}^*\|$ to 10^{-12} in 15 function evaluations (i.e., 30 component evaluations), and S_3 (with $h_0 = 10^{-6}$) requires 8 function evaluations. Because of the special form of the function, Brown's method finds $\mathbf{x}^*$ exactly (except for the effect of rounding errors) after 10 component evaluations. Brown's modified methods and the methods T_k ($k \geqq 2$) require only 7 component evaluations to find $\mathbf{x}^*$ exactly. If f_1 and f_2 are interchanged, then only 5 component evaluations are required. Thus, it may be difficult to establish exact order of convergence theorems for Brown's methods or the methods T_k.

Powell's (1962) *singular function.*

$$f_1 = x_1 + 10x_2, \quad f_2 = \sqrt{5}(x_3 - x_4), \quad f_3 = (x_2 - 2x_3)^2,$$

and

$$f_4 = \sqrt{10}(x_1 - x_4)^2,$$

with starting point $(3, -1, 0, 1)^T$. The Jacobian of this function is singular at the zero $\mathbf{x}^* = \mathbf{0}$, so the theorems of § 5 are not applicable. Method S_4 (with $h_0 = 10^{-6}$) requires 72 function evaluations to reduce $\|\mathbf{f}\|$ to 10^{-10} (then $\|\mathbf{x} - \mathbf{x}^*\| \simeq 6.6 \times 10^{-6}$), and method T_3 requires 66 function evaluations ($\|\mathbf{x} - \mathbf{x}^*\| \simeq 5.4 \times 10^{-6}$). Convergence appears to be linear for both methods.

Brown and Conte's function.

$$f_1 = \frac{1}{2}\sin(x_1x_2) - \frac{x_2}{4\pi} - \frac{x_1}{2}, \quad \text{and} \quad f_2 = \left(1 - \frac{1}{4\pi}\right)(\exp(2x_1) - e) + \frac{ex_2}{\pi} - 2ex_1,$$

starting from $(0.6, 3.0)^T$, near the zero $\mathbf{x}^* = (0.5, \pi)^T$. Method T_1 (with $h_0 = 10^{-6}$) reduces $\|\mathbf{x} - \mathbf{x}^*\|$ to 2.2×10^{-16} in 10 function evaluations, and method T_2 reduces $\|\mathbf{x} - \mathbf{x}^*\|$ to 4.8×10^{-13} in 9.5 function evaluations. According to Brown and Conte (1967), Brown's method reduces $\|\mathbf{x} - \mathbf{x}^*\|$ to 2×10^{-8} in 10 function evaluations, and the discrete Newton method is slower.

A trigonometric function, $n = 5$.

$$(18) \qquad f_i = E_i - \sum_{j=1}^{n} (A_{i,j}\sin x_j + B_{i,j}\cos x_j) \quad \text{for } i = 1, \cdots, n.$$

The coefficients E_i, $A_{i,j}$ and $B_{i,j}$ are randomly generated as suggested by Fletcher and Powell (1963), and the components of $\mathbf{x}_0 - \mathbf{x}^*$ are uniformly distributed in $[-\pi/10, +\pi/10]$. In tests with two different randomly generated sets of coefficients, method T_3 (with $h_0 = 10^{-3}$) required 11 or 12 function evaluations to reduce $\|\mathbf{x} - \mathbf{x}^*\|_\infty$ to 10^{-4}, and method S_5 required 16 (both times). Box (1966) reported that Powell's method for sums of squares (Powell (1965)) required 21 or 22, and the method of Barnes (1965) required 37 or 42.

A trigonometric function, $n = 20$. This example illustrates the effect of varying the parameter k in methods S_k and T_k. The function is defined by equation (18) with $n = 20$. The components of $\mathbf{x}_0 - \mathbf{x}^*$ are uniformly distributed in $[-\pi/40, +\pi/40]$. (With a larger interval the methods sometimes converge to different zeros.) Table 3 gives the number of equivalent function evaluations required to reduce $\|\mathbf{x} - \mathbf{x}^*\|$ to 10^{-12} with methods S_k and T_k ($h_0 = 10^{-6}$), for various values of k. The table shows that the predicted optimal values of $k = 12$ (for S_k) and $k = 7$ (for T_k) are nearly best possible, and much better than $k = 1$. Numerical results for $n = 5$ and 10 lead to similar conclusions.

8. Some extensions and analogies. The numerical results given in § 7 show that the idea of using the same approximation to the Jacobian for a (theoretically) optimal number of steps does give practical algorithms. We have concentrated on algorithms with good local properties. Algorithms with better global convergence properties may be obtained by modifying our algorithms appropriately: see, for example, Brent (1972b, c), Broyden (1969), Davidenko (1953a, b), Deist and Sefor (1967), Freudenstein and Roth (1963), and Kizner (1964).

An interesting unsolved problem is to determine a sharp upper bound $E_{\text{opt}}(n)$ on the efficiency of all possible algorithms which converge under conditions like

TABLE 3
Comparison of function evaluations required by S_k and T_k on a trigonometric function of 20 *variables*

k	S_k	T_k
1	162	69
2	106	37.5
3	68	38.5
4	70	28
5	70	28
6	72	28
7	52	28
8	52	30
9	53	31
10	53	32
11	54	
12	54	
13	55	
14	56	
15	57	
16	57	
17	58	
18	59	
19	60	
20	61	

those of Theorem 2. By a result of Winograd and Wolfe (1971), no algorithm for solving one nonlinear equation using only one function evaluation per step can have order greater than two, under mild restrictions on the definition of an algorithm. Applying this result to a system of equations with diagonal Jacobian matrix gives $E_{\mathrm{opt}}(n) \leqq \log 2$, so

$$E_{\mathrm{opt}}(n)/E(N_1) \leqq n + 1. \tag{19}$$

On the other hand, our results for methods N_k, S_k or T_k show that

$$E_{\mathrm{opt}}(n)/E(N_1) \geqq c \cdot \log n \tag{20}$$

for some positive constant c. A large gap remains between the bounds (19) and (20). For further discussion see Brent (1972a).

There is a close connection between methods for finding solutions of systems of nonlinear equations and methods for minimizing functions of several variables. It is theoretically possible to obtain highly efficient methods for minimizing functions of several variables by using an approximation to the Hessian for an optimal number of steps. The efficiency of the Rayleigh quotient and inverse iteration method for finding eigenvalues and eigenvectors of a matrix (Ostrowski (1957–1960), Wilkinson (1965)) may be increased in an analogous way. However, we have not yet performed any numerical experiments to verify these predictions.

Acknowledgment. I wish to thank Dr. P. Wolfe and the referees for many helpful suggestions.

REFERENCES

J. P. G. Barnes (1965), *An algorithm for solving non-linear equations based on the secant method*, Comput. J., 8, pp. 66–72.

L. Bittner (1959), *Eine Verallgemeinerung des Sekantenverfahrens (regula falsi) zur näherungsweisen Breechnung der Nullstellen eines nichtlinearen Gleichungssystems*, Wiss. Z. Techn. Hochsch. Dresden, 9, pp. 325–329.

M. J. Box (1966), *A comparison of several current optimization methods, and the use of transformations in constrained problems*, Comput. J., 9, pp. 67–77.

R. P. Brent (1972a), *The computational complexity of iterative methods for systems of nonlinear equations*, Proc. Symposium on the Complexity of Computation (Yorktown Heights, 1972), Plenum Press, New York.

——— (1972b), *Algorithms for Minimization Without Derivatives*, Prentice-Hall, Englewood Cliffs, N.J.

——— (1972c), *On the Davidenko–Branin method for solving simultaneous nonlinear equations*, IBM J. Res. Develop., 16, pp. 434–436.

K. M. Brown (1968), *Computational results for high order Newton-like methods which do not require derivative evaluations*, SIAM Fall Meeting, Philadelphia, 1968.

——— (1969), *A quadratically convergent Newton-like method based on Gaussian elimination*, this Journal, 6, pp. 560–569.

K. M. Brown and S. D. Conte (1967), *The solution of simultaneous nonlinear equations*, Proc. 22nd National Conference of the ACM, Thompson Book Co., Washington, D.C., pp. 111–114.

K. M. Brown and J. E. Dennis (1968), *On Newton-like iteration functions: General convergence theorems and a specific algorithm*, Numer. Math., 12, pp. 186–191.

——— (1972), *On the second order convergence of Brown's method for solving simultaneous non-linear equations*, to appear.

C. G. Broyden (1965), *A class of methods for solving nonlinear simultaneous equations*, Math. Comp., 19, pp. 577–593.

——— (1967), *Quasi-Newton methods and their application to function minimization*, Ibid., 21, pp. 368–381.

——— (1969), *A new method of solving nonlinear simultaneous equations*, Comput. J., 12, pp. 94–99.

——— (1970), *The convergence of single-rank quasi-Newton methods*, Math. Comp., 24, pp. 365–382.

D. F. Davidenko (1953a), *On a new method of numerical solution of systems of nonlinear equations*, Dokl. Akad. Nauk SSSR, 88, pp. 601–602. (In Russian.)

——— (1953b), *On the approximate solution of systems of nonlinear equations*, Ukrain. Mat. Zh., 5, pp. 196–206. (In Russian.)

F. H. Deist and L. Sefor (1967), *Solution of systems of nonlinear equations by parameter variation*, Comput. J., 10, pp. 78–82.

J. E. Dennis (1967), *On Newton's method and nonlinear simultaneous displacements*, this Journal, 4, pp. 103–108.

——— (1968), *On Newton-like methods*, Numer. Math., 11, pp. 324–330.

——— (1971), *On the convergence of Broyden's method for nonlinear systems of equations*, Math. Comp., 25, pp. 559–567.

L. C. W. Dixon (1971), *Variable metric algorithms: Necessary and sufficient conditions for identical behaviour on non-quadratic functions*, Rep. 26, Numerical Optimization Centre, The Hatfield Polytechnic, England.

A. Feldstein and R. M. Firestone (1969), *A study of Ostrowski efficiency for composite iteration algorithms*, Proc. 24th National Conference of the ACM, pp. 147–155.

R. Fletcher and M. J. D. Powell (1963), *A rapidly convergent descent method for minimization*, Comput. J., 6, pp. 163–168.

F. Freudenstein and B. Roth (1963), *Numerical solution of systems of nonlinear equations*, J. Assoc. Comput. Mach., 10, pp. 550–556.

J. W. Givens (1958), *Computation of plane unitary rotations transforming a general matrix to triangular form*, J. Soc. Indust. Appl. Math., 6, pp. 26–50.

A. S. Householder (1964), *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York.

L. V. Kantorovich and G. P. Akilov (1964), *Functional Analysis in Normed Spaces*, Pergamon Press, New York.

W. Kizner (1964), *A Numerical Method for Finding Solutions of Nonlinear Equations*, J. Soc. Indust. Appl. Math., 12, pp. 424–428.

A. Korganoff (1961), *Méthodes de calcul numérique*, vol. 1, Dunod, Paris.

J. M. Ortega and W. C. Rheinboldt (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.

A. M. Ostrowski (1957–1960), *On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors*, Arch. Rational Mech. Anal., 1, pp. 233–241; 2, pp. 423–428; 3, pp. 325–340, 341–347, 472–481; and 4, pp. 153–165.

——— (1960), *Solution of Equations and Systems of Equations*, Academic Press, New York.

B. N. Parlett (1971), *Analysis of algorithm for reflections in bisectors*, SIAM Rev., 13, pp. 197–208.

M. J. D. Powell (1962), *An iterative method for finding stationary values of a function of several variables*, Comput. J., 5, pp. 147–151.

——— (1965), *A method of minimizing a sum of squares of non-linear functions without calculating derivatives*, Ibid., 7, pp. 303–307.

S. M. Robinson (1966), *Interpolative solution of systems of nonlinear equations*, this Journal, 3, pp. 650–658.

H. Rosenbrock (1960), *An automatic method for finding the greatest or least value of a function*, Comput. J., 3, pp. 175–184.

J. Schmidt (1966), *Rate of convergence of the regula falsi and Steffensen processes in a Banach space*, Z. Angew. Math. Mech., 46, pp. 146–148.

V. E. Shamanskii (1967), *A modification of Newton's method*, Ukrain. Mat. Zh., 19, pp. 133–138. (In Russian.)

L. Tornheim (1964), *Convergence of multipoint iterative methods*, J. Assoc. Comput. Mach., 11, pp. 210–220.

J. F. Traub (1964), *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, N.J.

——— (1971), *Computational complexity of iterative processes*, Rep. CMU-CS-71-105, Dept of Computer Science, Carnegie-Mellon Univ., Pittsburgh, Pa.

R. G. Voigt (1971), *Orders of convergence for iterative processes*, this Journal, 8, pp. 222–243.

J. H. Wilkinson (1965), *The Algebraic Eigenvalue Problem*, Oxford Univ. Press, London.

S. Winograd and P. S. Wolfe (1971), *Optimal iterative processes*, Rep. RC 3511, IBM Research Center, Yorktown Heights, N.Y.

P. S. Wolfe (1959), *The secant method for simultaneous nonlinear equations*, Comm. ACM, 2, pp. 12–13.