Computers Math. Applic. Vol. 19, No. 6, pp. 9–14, 1990

0097-4943/90 \$3.00 + 0.00

APPLICATION OF CONTRACTOR DIRECTIONS FOR SOLVING SYSTEMS OF NONLINEAR EQUATIONS

T. ALTMAN
Department of Computer Science, University of Kentucky, Lexington, KY 40506, U.S.A.

(*Received* 22 *February* 1989)

Abstract—Our approach combines a method of an inexact steepest descent with the method of *contractor directions* to obtain a backtracking linear-time convergence algorithm for solving systems of nonlinear equations. Using the fundamental inequality of contractor directions, we prove the convergence and give an error estimate for our method.

The algorithm is well-suited for parallel computation. In fact, for systems with m equations and n unknowns, each iteration may be computed in parallel time $O(\log m + \log n)$, with $O(mn)$ processors.

1. INTRODUCTION

Over the years, a number of iterative and backtracking methods to solve systems of nonlinear equations have been proposed (e.g. see Refs [1–3], and the references within). Our method incorporates some of the ideas proposed by Altman [4] together with the method of *contractor directions* [5]. It is an iterative method with linear-time convergence, however, the advantage it possesses is that the individual iteration steps can be computed approximately. Hence, it also encompasses some nonexact methods and, in particular, the nonexact method of steepest descent. Moreover, the method of contractor directions allows for an additional level of parallelism in that the backtracking computation for individual iterations may be computed simultaneously.

The method may be applied to systems of equations where m, the number of equations, is not equal to n, the number of unknowns, but this case is not investigated here in detail. If the starting point $\mathbf{x}_0$ is bad, the algorithm can be used to find a better one, and then switch to a faster convergence method. In order to simplify the exposition, the steepest descent direction is taken as the direction of descent. However, the method in the general case and the line search in particular, is different from the Goldstein–Armijo approach, see for example Ref. [3].

2. DESCRIPTION OF THE METHOD

Consider the system

$$P(\mathbf{x}) = 0, \tag{1}$$

which can be written as $F(\mathbf{x}) = 0$, with

$$F(\mathbf{x}) = \|P(\mathbf{x})\|^2, \tag{2}$$

where

$$P(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^{\mathrm{T}}, \mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}} \tag{3}$$

and

$$\|P(\mathbf{x})\|^2 = f_1^2(\mathbf{x}) + \cdots + f_m^2(\mathbf{x}). \tag{4}$$

$P(\mathbf{x})$ may be defined on a domain containing a ball $B(\mathbf{x}_0, R) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_0\| \leqslant R\}$ for some $R > 0$. Assume that the following condition is satisfied:

$$\|P'(\mathbf{x})^{\mathrm{T}} P(\mathbf{x})\| \geqslant c\,\|P(\mathbf{x})\|, \tag{5}$$

for some constant c and all $\mathbf{x} \in B(\mathbf{x}_0, R)$.

The following is the iterative method under consideration:

$$\mathbf{x}_{i+1} := \mathbf{x}_i - \epsilon_i h_i, \tag{6}$$

where

$$h = \frac{\|P(\mathbf{x})\|^2}{\|\nabla F(\mathbf{x})\|^2} \nabla F(\mathbf{x}) = \frac{\|P(\mathbf{x})\|^2}{2\|P'(\mathbf{x})^{\mathrm{T}} P(\mathbf{x})\|^2} P'(\mathbf{x})^{\mathrm{T}} P(\mathbf{x}). \tag{7}$$

Here, $\nabla F(\mathbf{x}) = F'(\mathbf{x})$ is the gradient of F and $P'(\mathbf{x})$ is the Jacobian of P at $\mathbf{x}$,

$$P'(\mathbf{x}) = \frac{\mathrm{d}(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))}{\mathrm{d}(x_1, x_2, \ldots, x_n)} = \left(\frac{\partial f_j}{\partial x_k}\right) \quad \begin{array}{l} j = 1, \ldots, m, \\ k = 1, \ldots, n. \end{array} \tag{8}$$

In order to determine the step-size ϵ_i, we put

$$\Phi(\epsilon, h, \mathbf{x}) = \frac{|F(\mathbf{x} - \epsilon h) - (1 - \epsilon) F(\mathbf{x})|}{\epsilon}. \tag{9}$$

Let $0 < q' < q < 1$ be fixed.

Remark 2.1

If h is chosen by formula (7), then we get

$$\langle F'(\mathbf{x}), h \rangle - F(\mathbf{x}) = 0. \tag{10}$$

However, one can also choose h such that

$$|\langle F'(\mathbf{x}), h \rangle - F(\mathbf{x})| \leqslant q' F(\mathbf{x}), \tag{11}$$

or equivalently,

$$|\langle 2P'(\mathbf{x})^{\mathrm{T}} P(\mathbf{x}), h \rangle - \|P(\mathbf{x})\|^2| \leqslant q' \|P(\mathbf{x})\|^2, \tag{12}$$

where $\|h\| \leqslant c\|P(\mathbf{x})\|^2$, that is, equation (10) is solved approximately. Then condition (5) is not needed, because this assumption is replaced by $\|h\| \leqslant c\|P(\mathbf{x})\|^2$ in condition (12). In general, h satisfying condition (11) does not have to be the steepest descent direction. The choice of h in condition (11) can also be used in cases where the number of equations and unknowns is not equal. This choice makes the method different from the Goldstein–Armijo method. Also, let us mention that the Lipschitz continuity of $\nabla F(\mathbf{x})$ is not required, see Theorem 6.3.3 [1].

To choose the proper ϵ step-size, we proceed as follows:

$$\text{If } \Phi(1, h_i, \mathbf{x}_i) \leqslant q F(\mathbf{x}_i), \text{ then put } \epsilon_i := 1. \tag{13}$$

If $\epsilon_i = 1$ does not satisfy condition (13), take ϵ_i from the left or right half interval (0, 1) which we keep dividing until ϵ_i from the left or right subinterval satisfies

$$\beta q F(\mathbf{x}_i) \leqslant \Phi(\epsilon_i, h_i, \mathbf{x}_i) \leqslant q F(\mathbf{x}_i), \tag{14}$$

where $0 < \beta < 1$ is fixed independently of i. However, the larger the step-size $0 < \epsilon_i \leqslant 1$, the faster the convergence of method (6). Finally, put

$$\mathbf{x}_{i+1} := \mathbf{x}_i - \epsilon_i h_i. \tag{15}$$

Remark 2.2

If equation (10) is solved exactly, then method (6) becomes the method of steepest descent with a different line search. However, if equation (10) is solved approximately, satisfying conditions (11) or (12), then method (6) may no longer be of steepest descent, since the condition

$$\langle F'(\mathbf{x}), h \rangle < 0$$

may not hold [1, pp. 113–114].

In this way, the following *fundamental inequality for contractor directions* [5], is satisfied for each iteration:

$$|F(\mathbf{x}_i - \epsilon_i h_i) - (1 - \epsilon_i) F(\mathbf{x}_i)| \leqslant \epsilon_i q F(\mathbf{x}_i). \tag{16}$$

Also, consider the sequence

$$t_0 = 0,\ t_{i+1} = t_i + \epsilon_i,\ 0 < \epsilon_i \leqslant 1. \tag{17}$$

It follows inductively from condition (16) that

$$F(\mathbf{x}_i) \leqslant F(\mathbf{x}_0)\exp(-(1-q)t_i). \tag{18}$$

In fact, by induction from condition (16) we get

$$F(\mathbf{x}_{i+1}) \leqslant (1-(1-q)\epsilon_i)F(\mathbf{x}_i) \leqslant \exp(-(1-q)\epsilon_i)F(\mathbf{x}_i) \leqslant F(\mathbf{x}_0)\exp(-(1-q)(t_i+\epsilon_i)).$$

It follows from equations (5) and (7) that

$$\|h\| \leqslant (2c)^{-1}\|P(\mathbf{x})\| = (2c)^{-1}\sqrt{F(\mathbf{x})}. \tag{19}$$

Lemma 2.1

The following estimate holds

$$\sum_{i=0}^{\infty} \epsilon_i \|h_i\| \leqslant (2c)^{-1}[\tfrac{1}{2}(1-q)]^{-1}\sqrt{F(\mathbf{x}_0)}\exp(\tfrac{1}{2}(1-q)). \tag{20}$$

Proof. From equations (18) and (19) we have

$$\|h_i\| \leqslant (2c)^{-1}\sqrt{F(\mathbf{x}_0)}\exp(-\tfrac{1}{2}(1-q)t_i). \tag{21}$$

Furthermore,

$$\sum_{i=0}^{\infty} \epsilon_i \exp(-\tfrac{1}{2}(1-q)t_i) = \sum_{i=0}^{\infty} (t_{i+1}-t_i)\exp(-\tfrac{1}{2}(1-q)t_i)$$

$$= \sum_{i=0}^{\infty} (t_{i+1}-t_i)\exp(-\tfrac{1}{2}(1-q)t_{i+1})\exp(\tfrac{1}{2}(1-q)\epsilon_i)$$

$$\leqslant \exp(\tfrac{1}{2}(1-q)) \sum_{i=0}^{\infty} \int_{t_i}^{t_{i+1}} \exp(-\tfrac{1}{2}(1-q)\theta)\,\mathrm{d}\theta$$

$$= \exp(\tfrac{1}{2}(1-q)) \int_0^{\infty} \exp(-\tfrac{1}{2}(1-q)\theta)\,\mathrm{d}\theta = [\tfrac{1}{2}(1-q)]^{-1}\exp(\tfrac{1}{2}(1-q)).$$

Hence, relation (20) results from condition (21). □

Lemma 2.2

Suppose that the mapping $P(\mathbf{x})$ in equation (3) is continuous in $B(\mathbf{x}_0, R)$ and that $P'(\mathbf{x})$ is also continuous. If $\{\mathbf{x}_i\}$ converges to some $\mathbf{x}$ and $\{h_i\}$ is bounded, then

$$\epsilon_i^{-1}|F(\mathbf{x}_i - \epsilon_i h_i) - F(\mathbf{x}_i) + \epsilon_i\langle F'(\mathbf{x}_i), h_i\rangle| \to 0 \text{ as } \epsilon_i \to 0. \tag{22}$$

Proof. We have, by equation (2)

$$\langle F'(\mathbf{x}), h\rangle = \langle 2P'(\mathbf{x})^{\mathrm{T}}P(\mathbf{x}), h\rangle. \tag{23}$$

Hence, it follows that $F'(\mathbf{x})$ is uniformly continuous in $B(\mathbf{x}_0, R)$. But

$$\epsilon_i^{-1}[F(\mathbf{x}_i - \epsilon_i h_i) - F(\mathbf{x}_i) + \langle F'(\mathbf{x}_i), \epsilon_i h_i\rangle] = \epsilon_i^{-1}\int_0^1 \langle F'(\mathbf{x}_i - \theta\epsilon_i h_i) - F'(\mathbf{x}_i), -\epsilon_i h_i\rangle\,\mathrm{d}\theta$$

$$\leqslant \|h_i\| \int_0^1 \|F'(\mathbf{x}_i - \theta\epsilon_i h_i) - F'(\mathbf{x}_i)\|\,\mathrm{d}\theta$$

$$\leqslant \|h_i\| \int_0^1 [\|F'(\mathbf{x}_i - \theta\epsilon_i h_i) - F'(\mathbf{x})\| + \|F'(\mathbf{x}) - F'(\mathbf{x}_i)\|]\,\mathrm{d}\theta,$$

by Taylor's formula, and $\|F'(\mathbf{x}) - F'(\mathbf{x}_i)\| \to 0$ and $\|F'(\mathbf{x}_i - \theta\epsilon_i h_i) - F'(\mathbf{x})\| \to 0$ uniformly in $0 \leqslant \theta \leqslant 1$ as $i \to \infty$ by estimate (20) and $\|\mathbf{x}_i - \mathbf{x}\| \to 0$ as $i \to \infty$, by assumption. Hence, relation (22) follows. □

Lemma 2.3

Under the assumption of Lemma 2.2, we get for $i \to \infty$,

$$\Phi(\epsilon_i, h_i, \mathbf{x}_i) \to 0 \quad \text{if} \quad \epsilon_i \to 0. \tag{24}$$

Proof. From equations (7) and (23), we get

$$\langle F'(\mathbf{x}_i), h_i \rangle = F(\mathbf{x}_i).$$

Hence, relation (24) results from equations (9) and (22). □

Theorem 2.1

Suppose that condition (5) is satisfied and both $P(\mathbf{x})$ and $P'(\mathbf{x})$ are uniformly continuous and bounded in $B(\mathbf{x}_0, R)$, where R is such that

$$(2c)^{-1}[\tfrac{1}{2}(1-q)]^{-1}\sqrt{F(\mathbf{x}_0)}\exp(-\tfrac{1}{2}(1-q)) \leqslant R \tag{25}$$

holds. Then the equation $P(\mathbf{x}) = 0$ has a solution $\mathbf{x}$ such that

$$\|\mathbf{x}_i - \mathbf{x}\| \to 0 \quad \text{as} \quad i \to \infty, \tag{26}$$

where $\{\mathbf{x}_i\}$ is determined by equation (6), and $\mathbf{x}_i, \mathbf{x} \in B(\mathbf{x}_0, R)$. The error estimate is given by the formula

$$\|\mathbf{x}_i - \mathbf{x}\| \leqslant (2c)^{-1}\sqrt{F(\mathbf{x}_0)}\exp(-\tfrac{1}{2}(1-q)) \int_{t_i}^{\infty} \exp(-\tfrac{1}{2}(1-q)\theta)\, d\theta$$
$$= (2c)^{-1}\sqrt{F(\mathbf{x}_0)}\exp(-\tfrac{1}{2}(1-q))[\tfrac{1}{2}(1-q)]^{-1}\exp(-\tfrac{1}{2}(1-q)t_i). \tag{27}$$

Proof. Relations (6) and (20) imply that the sequence $\{\mathbf{x}_i\}$ converges to some $\mathbf{x}$, that is, condition (26) holds. Since $t_0 = 0$ and $t_i = \Sigma_{j=0}^{i-1} \epsilon_j$, we consider two cases:

(a) $t_i \to \infty$ as $i \to \infty$. Then estimation (18) implies

$$\|P(\mathbf{x}_i)\| \to 0 \quad \text{as} \quad i \to \infty.$$

Since $P(\mathbf{x})$ is continuous and $\mathbf{x}_i \to \mathbf{x}$ as $i \to \infty$, it follows that

$$P(\mathbf{x}_i) \to P(\mathbf{x}) = 0 \quad \text{as} \quad i \to \infty.$$

(b) Suppose that $\lim_{i\to\infty} t_i < \infty$. Then $\epsilon_i \to 0$ as $i \to \infty$. But it results from conditions (24) and (14) that $F(\mathbf{x}_i) \to 0$ as $i \to \infty$, or equivalently,

$$\|P(\mathbf{x}_i)\| \to 0 \quad \text{as} \quad i \to \infty.$$

Hence, $P(\mathbf{x}) = 0$ as in case (a). This completes the proof. □

Remark 2.3

In addition to hypotheses of Theorem 2.1, suppose that the Jacobian $P'(\mathbf{x})$ in equation (7) is Lipschitz continuous in $B(\mathbf{x}_0, R)$ with Lipschitz constant c_1. Then ϵ_i satisfies condition (18) if

$$\epsilon_i \leqslant q(2c)^2 c_1^{-1} \tag{28}$$

holds.

Proof. Using the same Taylor formula as in Lemma 2.2, we get

$$|F(\mathbf{x}_i - \epsilon_i h_i) - (1-\epsilon_i)F(\mathbf{x}_i)| \leqslant c_1 \epsilon_i^2 \|h_i\|^2 \leqslant \epsilon_i q F(\mathbf{x}_i).$$

by condition (21), if relation (28) holds.

Based on relation (28) and the line search algorithm described above, relation (14) will be satisfied after a finite number of trials.

3. PARALLEL COMPUTATION

Although it is theoretically possible to perform a Newton-type iteration in parallel time $O((\log n)^2)$, there is no known practical algorithm that achieves this bound.† However, it would not be constructive to compare the parallel computation times per iteration for quadratic versus linear convergence methods.

Below, we show that an iteration step in our algorithm can be performed in parallel time $O(\log m + \log n)$ on a PRAM with $O(mn)$ processors. Moreover, the optimal step-size ϵ_i can be determined by evaluating the Φ function with a bounded number of different ϵ values simultaneously [see equation (28)], thereby eliminating the sequential nature of condition (14).

Let us examine an iteration step of our method:

$$\mathbf{x}_{i+1} := \mathbf{x}_i - \epsilon_i \frac{\|P(\mathbf{x}_i)\|^2}{2\|P'(\mathbf{x}_i)^{\mathrm{T}}P(\mathbf{x}_i)\|^2} P'(\mathbf{x}_i)^{\mathrm{T}}P(\mathbf{x}_i). \tag{29}$$

It is clear that major computational work, for both the sequential and parallel case, is the evaluation of the Jacobian $P'(\mathbf{x})$.

$$P'(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}. \tag{30}$$

Let σ_{f_j} and $\delta_{\partial_{jk}}$ denote the computation times for evaluating

$$f_j(\mathbf{x}) \quad \text{and} \quad \left(\frac{\partial f_j}{\partial x_k}\right), \tag{31}$$

respectively. Let us assume that the computation times for evaluating the functions and their partial derivatives are independent of i, the number of iterations. There are many families of problems for which σ_{f_j} and $\delta_{\partial_{jk}}$ increase with n, the dimension of the problem, however, for the sake of exposition, let us also assume that these computation times are also independent of n. Let

$$\sigma_{\max} = \max_{j=1,\ldots,m} (\sigma_{f_j}) \quad \text{and} \quad \delta_{\max} = \max_{\substack{j=1,\ldots,m \\ k=1,\ldots,n}} (\delta_{\partial_{jk}}). \tag{32}$$

It follows that we can compute $P(\mathbf{x})$ and $P'(\mathbf{x})$ in parallel time $O(\sigma_{\max} + \delta_{\max})$. The computation of the product $P'(\mathbf{x})^{\mathrm{T}}P(\mathbf{x})$ will take $O(\log m)$ steps, whereas the computation of the norms

$$\|P'(\mathbf{x})^{\mathrm{T}}P(\mathbf{x})\| \quad \text{and} \quad \|P(\mathbf{x})\|^2 \tag{33}$$

will take additional $O(\log n)$ time.

The evaluation of the function Φ, in the determination of an optimal step-size ϵ_i, also takes $O(\sigma_{\max} + \delta_{\max} + \log m + \log n)$ steps, which becomes the computation time for each iteration. If the number of available processors is $p < mn$, then the computation time for each iteration becomes

$$O\left(\sigma_{\max} + \delta_{\max} + \frac{mn}{p} + \log m + \log n\right). \tag{34}$$

4. CONCLUDING REMARKS

The algorithm has been implemented on both sequential and parallel machines (Vax 750 and Sequent BALANCE 21000 multiprocessor). Initial computational results indicate that the method

†This is a consequence of the fact that the matrix inversion problem belongs to the class NC^2, see Ref. [6] for explanation.

is a good candidate to be used as a startup procedure for higher order methods [7]. As expected from the methods under consideration for nonlinear functionals, the performance of this method degrades in the close vicinity of the solution, indicating a transfer to a faster convergence method.

REFERENCES

1. J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, N.J. (1983).
2. L. V. Kantorovitch and G. P. Akilov, *Functional Analysis in Normed Spaces.* Pergamon Press, New York (1964).
3. J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York (1970).
4. M. Altman, Approximation methods in functional analysis. Lecture notes given at California Institute of Technology—advanced topics in numerical analysis (1959).
5. M. Altman, Iterative methods of contractor directions. *Nonlinear Analysis* **4,** 761–772 (1980).
6. S. A. Cook, The classification of problems which have fast parallel algorithms. *Lecture Notes in Computer Science* #158: *Foundations of Computation Theory* (Ed. M. Karpinski), pp. 78–93 (1983).
7. S. Rajendran, Experiences with a parallel algorithm for solving systems of nonlinear equations. M.S. Thesis, Department of Computer Science, University of Kentucky, Lexington, Ky (1989).