RESEARCH NOTE

A TRUST REGION ALGORITHM FOR SOLVING NONLINEAR EQUATIONS

S. J. Sadjadi

Department of Industrial Engineering Iran University of Science and Technology
Tehran, Iran seyedjafar@yahoo.com

(Received: April 27, 2001 – Accepted: October 18, 2001)

Abstract This paper presents a practical and efficient method to solve large-scale nonlinear equations. The global convergence of this new trust region algorithm is verified. The algorithm is then used to solve the nonlinear equations arising in an Expanded Lagrangian Function (ELF). Numerical results for the implementation of some large-scale problems indicate that the algorithm is efficient for these classes of problems.

Key Words Trust Region, Nonlinear Equations, Newton Method

چکیده در این مقاله یک روش کار آمد و عملی برای حل مسائل معادلات غیر خطی در ابعاد بزرگ ارائه می‌شود. ابتدا همگرایی این الگوریتم ناحیه اعتماد اثبات شده و سپس الگوریتم برای حل نامعادلات غیرخطی در توابع بسط لاگرانژ مورد استفاده قرار می‌گیرد. نتایج محاسبات با استفاده از این الگوریتم نشانگر کار آمد بودن آن بر روی بعضی مسائل است.

1.INTRODUCTION

The use of trust region methods for solving systems of nonlinear equations has been popular during the past decade. Much of the interest is due to the strong convergence properties of such methods [1,2]. Duff et al.[3] use linear programming combined with trust region idea to solve nonlinear equations. It is based on minimizing the l_1-norm of the linearized vector within an l_∞ norm trust region, thereby permitting linear programming techniques to be easily applied. Duff et al. [3] show that their approach works better than Levenberg's algorithm [4]. However, the algorithm was not used for large-scale problems. Luksan [5] uses an inexact trust region method to solve large and sparse nonlinear equations. The method does not need to use matrices so it can be also used for large dense nonlinear equations. Martinez [6] uses a two-dimensional search algorithm to solve large and sparse nonlinear equations. Although the algorithm has strong convergence properties, the computation of the two dimensional search is expensive.

Martinez and Santos [7] tried to solve this difficulty by using a curvilinear search algorithm.

This paper presents a modification of a trust region algorithm introduced by Martinez [7]. Our algorithm has a curvilinear search method similar to that proposed by Martinez and Santos [7], and our convergence proof eliminates mistakes in their work. Numerical results of a large-scale problem are presented at the end.

2. PROBLEM STATEMENT

Consider solving a set of nonlinear equations of using the following minimization problem:

$$\min_{x \in R^n} \frac{1}{2}\|F(x)\|^2$$

where $F = (f_1,...,f_m)^T$ is a C^1-function and $\|.\|$ represents the Euclidean norm in R^n. Martinez's method [6] uses a Gauss-Newton strategy to obtain an approximation of the Newton direction. Bi-dimensional search methods are then used to find the best direction d_{k+1}, which is a linear combination of the Newton direction and the Cauchy step at every iteration.

The algorithm developed by Martinez [6] needs to solve a bi-dimensional search direction several times. This makes the algorithm inefficient in some cases, especially for large-scale problems. Martinez and Santos [7] present a modification of their method, and report that the use of a curvilinear search method can reduce the burden of the computation and significantly simplify the algorithm for practical implementation. However, the curvilinear search plane used in the algorithm statement by Martinez and Santos is different from what they actually developed. In the following section, we present a new curvilinear search method.

The contributions of the proposed method in addition include the use of a pure Newton direction near optimal solution.

A convergence proof is given at the end that eliminates the mistakes in the work of Martinez and Santos [7].

3. A NEW ALGORITHM WITH CURVILINEAR SEARCH DIRECTION

In this section we present a proposed algorithm that has similar steps as the algorithm in [7]. We consider that a Newton step tends to provide fast convergence when x is close enough to x^*. Therefore, we switch the bi-dimensional search direction to a pure Newton step when step d_k is close enough to the final step.

Algorithm 1 Let $F: \Omega \subset R^n \to R^m, m \ge n$ $F \in C^1(\Omega), \Omega$ an open set. Let $x^0 \in \Omega$ be an arbitrary initial point,

$\eta_k \in [0,1), \theta_1, \theta_2 \in [0,1), \theta_3 \in [0,\frac{1}{2}), \xi \in [0,.1)$

$\overline{M} > 0, \underline{M} \in [\theta_1 \overline{M}, \overline{M}]$.

Let x_k be the k-th approximation to the solution. We denote

$$F_k = F(x_k), J_k = J(x_k), g_k = J_k^T F_k = \frac{1}{2}\nabla\|F(x)\|^2 |x_k, D_k = \text{diag}(\sigma_k^1,...,\sigma_k^n) \tag{1}$$

where

$$\sigma_k^i = \begin{cases} (x_k^i)^2 & \text{if} \quad (x_k^i)^2 \in [\underline{M}, \overline{M}], \\ \underline{M} & \text{if} \quad (x_k^i)^2 < \underline{M}, \\ \overline{M} & \text{if} \quad (x_k^i)^2 > \overline{M}. \end{cases}$$

Step 1 Compute J_k and g_k. If g_k =0, stop.

Step 2 Obtain $w_k \in R^n$ such that

$$\|J_k^T J_k w_k + g_k\| \le \eta_k \|g_k\|. \tag{2}$$

If $\|g_k\| \le \varepsilon$ set $x_{k+1} = x_k + w_k$ and k=k+1, go to Step 1.

Step 3 Obtain $v_k \in R^n$ as the solution of the following bi-dimensional problem:

$$\min \| J_k v + F_k \| \quad \text{s.t.} \quad \|\lambda_1 g_k + \lambda_2 w_k\| \le \| w_k \| \tag{3}$$

Step 4 Set $d_k^1 = -g_k$ and test the following two conditions for v_k:

$$v_k^T g_k \le -\theta_1 \|v_k\| \|g_k\| \tag{4}$$

and

$$\underline{M}\|g_k\| \le \|v_k\| \le \overline{M}\|g_k\| \tag{5}$$

If (4) and (5) are satisfied set $d_2^k = v_k$ otherwise set $d_2^k = d_1^k$.

Step 4 Set t = 1. Perform Step 5.a to 5.d

(5.a): Set d=d (t)= $t^2 d_k^2 + \frac{g_k^T d_k^2}{g_k^T d_k^1} t(1-t^2) d_k^1$ (6)

(5.b): If $\frac{1}{2}\|F(x_k + d)\|^2 \le \frac{1}{2}\|F(x_k)\|^2 + \theta_2 g_k^T d$ (7)

go to Step 5.d

(5.c): Let $\hat{t}$ be such that $\theta_3\|d(t)\| \le \|d(\hat{t})\| \le (1-\theta_3)\|d(t)\|$ (8)

Replace t by $\hat{t}$, go to Step 5.a

(5.d): $d_k = d$, $x_{k+1} = x_k + d_k$.

Algorithm (1) has similar steps as the one introduced by Martinez [7]. In Step 5.a, we are using a new curvilinear search algorithm, which is slightly different from the original algorithm.

Theorem 1. The Algorithm 1 is well defined. The proof is similar to Martinez and Santos [7] and corrects the mistakes in the original paper.

It is an easy task to show that if $g_k \neq 0$, the algorithm can reach Step 5.d in a finite number of iterations. In Step 2 of the algorithm, a system of linear equations is solved. Assuming that the system of equations has full rank, it always has a unique solution. In Step 3, a two-dimensional sub problem is solved and v is in the positive cone determined by g_k and w_k. Step 4 does not create any problems. Finally, Step 5 is verified as follows. Let us write

$$d = d(t) = t^2 d_2 + at(1-t^2)d_1, \quad \text{where} \quad a = \frac{g_k^T d_k^2}{g_k^T d_k^1}.$$

By the definition of d_k^1 and (4) we know that a > 0. By (8), we need to show that (7) is satisfied when t is small enough. By Mean Value Theorem,

$$\frac{1}{2}\|F(x_k + d(t))\|^2 - \frac{1}{2}\|F(x_k)\|^2 = g(g_k + \xi(t)d(t))^T d(t) \quad (10)$$

where g(x) denotes $\frac{1}{2}\nabla\|F(x)\|^2$ and $0 \le \xi(t) \le 1$.

Meanwhile, d (t) is a positive combination of d_1 and d_2 in (6), $g_k^T d_k^1 < 0$ and $g_k^T d_k^2 < 0$. Therefore $g_k^T d(t) < 0$ for $t \in [0,1]$. So, by (10)

$$\lim_{t\to 0} \frac{\frac{1}{2}\|F(x_k + d(t))\|^2 - \frac{1}{2}\|F(x_k)\|^2}{g_k^T d(t)} = \frac{g(x_k + \xi(t)d(t))^T d(t)}{g_k^T d(t)} = \frac{g(x_k)^T d_k^1}{g(x_k)^T d_k^1} \quad (11)$$

Taking limits on both sides of (11), we have

$$\frac{\frac{1}{2}\|F(x_k + d_k^1)\|^2 - \frac{1}{2}\|F(x_k)\|^2}{g_k^T d_k^1} = 1 \quad (12)$$

Therefore, given $\theta_2 \in (0,1)$, there exists $\hat{t} > 0$ such that

$$\frac{\frac{1}{2}\|F(x_k + d(\hat{t}))\|^2 - \frac{1}{2}\|F(x_k)\|^2}{g_k^T d(\hat{t})} \ge \theta_2 \quad (13)$$

for $t \in (0, \hat{t})$. Thus, using $g_k^T d(\hat{t}) < 0$, we obtain (7). This completes the proof.

Theorem 2. Assume that (x_k) is generated by algorithm 1, Then:

If there exists c > 0 such that $\|g_k\| \le c$ for all k=0,1,2,... and $x^* \in \Omega$ is a limit point of (x_k), then $J(x^*)^T F(x^*) = 0$.

(a) Let $\varepsilon > 0$. If $x \in \Omega : \|F(x)\|^2 \le \|F(x^0)\|^2$ is compact, then there exists $k \in N$ such that $\|J(x_k)^T F(x_k)\| \le \varepsilon$.

(b) Let x^* be a strict local minimizer of f in $\Omega, \varepsilon \geq 0$. Then there exists $\varepsilon_1 > 0$ such that $\|x_k - x^*\| \leq \varepsilon_1$.

(c) If x^* is a local minimizer of $\|F(x_k)\|^2$ and an isolated stationary point in Ω, then there exists $\varepsilon > 0$ such that $\lim x_k = x^*$, whenever $\|x^0 - x^*\| \leq \varepsilon$.

Proof 2 We prove that if the inequality $\|d_k\| \leq \|d_k^2\|$ in (8) is changed to $\|d_k\| \leq K\|d_k^2\|$ for some constant K, it will become a simple case of the algorithm 3.1 of [6]. It is clear that the Equations 4 and 6 and the definition of d_k^1 implies that $d_k \in C(d_k^1, d_k^2)$. It can be also verified that d_k^1 satisfies:

$$g_k^T d_k^1 \leq -\theta_1 \|d_k^1\| \|g_k\| \tag{14}$$

In fact,

$$\frac{\|g_k^T d_k^1\|}{\|g_k\| \, \|d_k^1\|} = \frac{\|g_k^T D_k g_k\|}{\|g_k\| \, \|D_k g_k\|} \geq \frac{\underline{M}}{\overline{M}} \frac{\|g_k\|^2}{\|g_k\| \, \|g_k\|} = \underline{M}/\overline{M} \geq \theta_1 \tag{15}$$

Therefore, (14) is proved. Now, by (4) and (5) and the choice of d_k^2 we have:

$$g_k^T d_k^2 \leq -\theta_1 \|d_k^2\| \|g_k\| \tag{16}$$

Hence the axiom (2) of [6] is satisfied. By definition of d_k^1, we have:

$$\underline{M}\|g_k\| \leq \|d_k^1\| \leq \overline{M}\|g_k\| \tag{17}$$

Hence, by (4), (5), the axiom (9) of [6] is also satisfied. On the other hand, by (7), the axiom (9) of [6] holds. Finally, we prove the inequality of $\|d_k\| \leq K\|d_k^2\|$. From the Expression 9 for d(t) it implies that $d'(t) = 2td_k^2 + a(1-3t^2)d_k^1$

Therefore, if $\gamma(t) = \|d(t)\|^2$, for $t \in [0,1]$, there is

$$\gamma'(t) = 2d'(t)^T d(t) =$$
$$2(2td_k^2 + a(1-3t^2)d_k^1)^T (t^2 d_k^2 + at(1-t^2)d_k^1) =$$
$$2\,[2t^3 d_k^{2\,T} d_k^2 + 2at^2(1-t^2)d_k^{1\,T} d_k^2 + at^2(1-3t^2)d_k^{1\,T} d_k^2$$
$$+ a^2 t(1-3t^2)(1-t^2)d_k^{1\,T} d_k^1] =$$
$$4t^3\|d_k^2\|^2 + [4a^2(1-t^2) + 2at^2(1-3t^2)]d_k^{1\,T} d_k^2$$
$$+ 2a^2 t(1-3t^2)(1-t^2)\|d_k^1\|^2$$
$$\leq 4\|d_k^2\|^2 + 6ad_k^{1\,T} d_k^2 + 2a^2\|d_k^1\|^2$$
$$\leq 4\overline{M}^2\|g_k\|^2 + \frac{6\|g_k\|\|d_k^2\|\|d_k^1\|\|d_k^1\|}{\theta_1\|g_k\|\|d_k^1\|} +$$
$$\frac{2\|g_k\|\|d_k^2\|\|d_k^1\|^2}{\theta_1\|g_k\|\|d_k^1\|}$$
$$\leq 4\overline{M}^2\|g_k\|^2 + \frac{6\overline{M}^2\|g_k\|^2}{\theta_1} + \frac{2\overline{M}^2\|g_k\|^2}{\theta_1}$$
$$\leq (4\overline{M}^2 + \frac{6\overline{M}^2}{\theta_1} + \frac{2\overline{M}^2}{\theta_1})\|g_k\|^2 = C_1\|g_k\|^2 \tag{19}$$

Therefore, for $t \in [0,1]$,

$$\|d(t)\|^2 = \gamma(t) \leq \gamma(1) + \max_{t\in[0,1]}|\gamma'(t)| \leq \|d_k^2\|^2 +$$
$$C_1\|g_k\|^2 \leq \|d_k^2\|^2 + \frac{C_1\|d_k^2\|^2}{\underline{M}^2} = (1 + \frac{C_1}{\underline{M}^2})\|d_k^2\|^2 .$$

Thus (18) is satisfied with $K = \sqrt{1 + \frac{C_1}{\underline{M}^2}}$ and the proof is complete.

4. MOTIVATION

In this section, the motivation of using a curvilinear search direction similar to the one introduced in [7]

TABLE 1. The Summary of the Use of Algorithm 1.

n	q	Curvilinear +Newton(CM)			Curvilinear		
		Iter.	$\|h\|_\infty$	CPU(Sec.)	Iter.	$\|h\|_\infty$	CPU(Sec.)
120	96	28	6.46E-09	21.63	41	1.01E-10	38.98
240	192	34	8.55E-09	60.05	39	1.78E-10	85.49
360	288	48	1.36E-08	139.05	35	5.79E-10	123.06
480	384	40	3.99E-11	162.29	37	9.18E-11	200.16
600	480	29	2.53E-11	147.33	28	4.89E-11	163.08
720	576	31	2.54E-11	185.38	30	4.09E-11	215.10
1200	960	35	3.16E-11	370.29	30	8.39E-11	416.28
2400	1920	44	3.37E-10	820.70	29	6.08E-11	923.42
3600	2880	45	4.63E-07	1639.93	48	2.66E-09	3852.65
4800	3840	53	8.58E-07	2808.88	55	4.54E-09	7567.43
7800	6240	55	9.83E-07	4961.97	55	4.54E-09	10433.95
10800	8640	50	2.88E-06	5983.10	53	1.00E-09	19845.60

is explained. Since d(t) lies in the positive cone generated by d_k^1 and d_k^2 for all $t \in [0,1]$ and it is desired to have a negative gradient direction when the step is infinitesimal, the search direction is assumed to be tangent to d_k^1 for small step t. (Note that $d_k(1) = d_k^2, d_k(0) = 0, a > 0$). Let h be the orthogonal projection of d_k^2 on the orthogonal complement of the line generated by d_k^1, related to the norm $\|.\|_{D_k^{-1}}$ ($\|z\|_{D_k^{-1}}^2 = z^T D_k^{-1} z$ for all $z \in R^n$), Therefore,

$$h = d_k^2 - \frac{d_k^{2\,T} D_k^{-1} d_k^1}{d_k^{1\,T} D_k^{-1} d_k^1} d_k^1,$$

But $d_k^1 = -D_k g_k$, hence,

$$h = d_k^2 - \frac{g_k^T d_k^2}{d_k^T d_k^1} d_k^1.$$

For each point z in the plane spanned by (d_k^1, h) may be expressed as $z = y_1 d_k^1 + y_2 h$. d_k^2 corresponds to $y_1 = \frac{g_k^T d_k^2}{g_k^T d_k^1}, y_2 = 1$. A simple curve proposed by Martinez [7] has the following form:

$$P = \{z = y_1 d_k^1 + y_2 h \mid y_2 = (\frac{g_k^T d_k^1}{g_k^T d_k^2} y_1)^2\},$$

and the curve used by Martinez and Santos, in the coordinate (y_1, y_2) has the form:

$$P = \{z = y_1 d_k^1 + y_2 h \mid y_1 = \frac{g_k^T d_k^2}{g_k^T d_k^1}(-y_2^{\frac{3}{2}} + y_2 + \sqrt{y_2})\}.$$

5. NUMERICAL IMPLEMENTATION AND RESULTS

In this section some numerical results for the implementation of the curvilinear search method to solve a large-scale problem from water resources management are presented. The problem has the form of the minimization of a quadratic objective function, f(x), subject to some equality constraints, h(x), and some bound constraints, g(x). The

Expanded Lagrangian Techniques developed in [8] is used to convert the constrained optimization into a set of nonlinear equations in order to use the curvilinear search algorithm developed in this paper.

The implementation uses

$\eta=10^{-4}, \theta_1=10^{-7}, \theta_2=10^{-4}, \xi=10^{-3}, \overline{M}=10^{9}, \underline{M}=10^{-5}$

In Table 1, we report:

(n, q, Iter.): the number of variables, the number of equality constraints, and the total number of iterations needed to reach the convergence criteria corresponding to the application, respectively.

(h_∞ CPU Time (Sec.)): The maximum violation of equality constraints, the value of the objective function, and the running time in seconds, respectively. The algorithm has been coded using FORTRAN77 on a Sparc2 workstation. The performance of the proposed method that uses Newton step at final stage (Algorithm 1) with the implementation of the curvilinear line search is compared. The results are summarized in Table 1. In both algorithms, the number of iterations does not increase significantly with the size of the problem. In most cases, Algorithm 1 converges faster, however, it doesn't reach the accuracy required for large cases.

6. COMNCLUSION

In this paper, a practical and efficient method for solving nonlinear equations has been presented.

Like the algorithm introduced in [7], it does not need to solve the two-dimensional trust-region subproblem several times. Instead, it uses a curvilinear search direction similar to the one used in [7]. The numerical results indicate that using a pure Newton step when the step is close enough to the final solution can provide a fast convergence. The global convergence of the proposed algorithm has been verified.

7. ACKNOWLEDGMENT

The author would like to thank Dr. Martinez for his constructive comments on this work.

8. REFERENCES

1. More, J. J., "Recent Developments in Algorithm and Software for Trust Region Methods", ***Math. Programming***, (1983), 258-287.
2. More, J. J., "The Levenberg-Marquardt Algorithm: Implementation and Theory", (G. A. Watson, Ed.), ***Proc. of Dundee Conference on Numerical Analysis***, Berlin, (Springer 1978), 105-116.
3. Duff, I. S., Nocedal, J. and Reid, J. K., "The Use of Linear Programming for the Solution of Sparse Sets of Nonlinear Equations", ***Siam Sci. Stat. Put.*** ,Vol. 8, No. 2, (1987), 99-108.
4. Levenberg, K. "A Method for the Solution of Certain Nonlinear Problems in Least Squares", ***Quart, Appl. Math.***, Vol. 2, (1944), 164-168.
5. Luksan, L., "Inexact Trust Region Method for Large Sparse Systems of Nonlinear Equations", ***Journal of Optimization Theory and Applications***, Vol. 81, No. 3, (June 1994), 569-590.
6. Martinez, J. M., "An Algorithm for Solving Sparse Nonlinear Least-Square Problem", ***Computing***, Vol. 39, (1987), 307-325.
7. Martinez, J. M. and Santos, R. F., "An Algorithm for Solving Nonlinear Least-Square Problems with a New Curvilinear Search", ***Computing***, Vol. 44, (1990), 83-90.
8. Sadjadi, S. J., "Nonlinear Programming Using an Expanded Lagrangian Function", Ph.D thesis, University of Waterloo, Department of Systems Design Engineering, (1997).