Proceedings of the Ninth International Colloquium on Differential Equations

Plovdiv, Bulgaria, 18 - 23 August, 1998

Editor: D. Bainov

///VSP///
Utrecht, The Netherlands, 1999

9th Int. Coll. on Differential Equations, pp. 113–118
D. Bainov (Ed)

Studying the Numerical Properties of Solvers for Systems of Nonlinear Equations

DEBORAH DENT AND MARCIN PAPRZYCKI
School of Mathematical Sciences, University of Southern Mississippi, Hattiesburg, MS 39406-5106, USA
{m.paprzycki, deborah.dent}@usm.edu
ANNA KUCABA-PIĘTAL
Department of Aeronautical Engineering, Technical University of Rzeszów, Poland
anpietal@ewa.prz.rzeszow.pl

Abstract: Solution of systems of nonlinear equations is a relatively complicated problem for which a number of different approaches have been proposed. The aim of this note is to attempt at a comparison between some of them when applied to a number of test problems. A discussion of the properties of the solvers and the interrelationship between the solvers and the test problems will be presented.
Key Words and Phrases: *Systems of linear algebraic equations, solution methods, performance comparison.*

1. Introduction

The solution of systems of algebraic equations has a well-developed mathematical and computational theory when the equations are linear, or when a single nonlinear equation is to be solved [1]. The situation is much more complicated for systems of nonlinear algebraic equations. Both the mathematical theory and computational practice are far from a complete understanding of the solution process. When systems of linear equations are considered, libraries of subprograms solving standard problems have been developed. These libraries may not provide the most efficient way of solving certain special problems, but they should be robust enough to effectively solve most problems. In addition, sets of test cases have been developed which can be used as benchmarks against which the newly developed software is to be tested. These benchmarks allow for establishing the quality of the new approach and a comparison with existing software targeting the same problem. No such development can be observed when the solution of nonlinear systems is concerned.

The aim of this note is to report on the initial steps in the direction of building a library of nonlinear solvers and compiling a set of test problems. These test problems are used to compare the performance of existing nonlinear solvers. In section 2 we briefly describe the software that we have performed experiments. Section 3 presents and discusses the results of our numerical experiments. Directions for future research are discussed.

2. Algorithms for the Solution of the System of Nonlinear Algebraic Equations

This section contains a brief summary of algorithms behind nonlinear solvers that we have performed experiments (in all cases the references cited should be consulted for the details). We assume that a system of n nonlinear algebraic equations $F(x) = 0$ is to be solved.

Newton's Method

The Newton's Method for a system of equations is a natural extension of the Newton's Method for a single equation [2]. Let us assume that the function G is defined by

$$G(\underline{x}) = \underline{x} - J(\underline{x})^{-1}F(\underline{x}),$$

and the functional iteration procedure is: select starting vector x_0 and generate a series of vectors

$$\underline{x}_k = G(\underline{x}_{k-1}) = \underline{x}_{k-1} - J(\underline{x}_{k-1})^{-1}F(\underline{x}_{k-1})$$

where $J(x)$ is the Jacobian matrix.

The convergence rate for this method is fast, but the success of the method depends on a good starting vector x_0.

Brown's Method

Brown's Method is a modification of Newton's Method [3]. Here, we replace the Jacobian matrix with its difference quotient approximation. For each iteration, only one function from the system at a time is evaluated (function f_i for use with f_{i+1}). A successive substitution scheme is being used for treatment of f_i. As in case of Newton's Method, the convergence is fast, but it requires a good starting vector x_0.

Secant (Broyden's Method)

Broyden's Method belongs to the general category of quasi-Newton methods [4]. Here, matrix B_k is updated at each iterate so that the new approximation B_{k+1} satisfies the general quasi-Newton equation:

$$B_{k+1}(\underline{x}_{k+1} - \underline{x}_k) = F(\underline{x}_{k+1}) - F(\underline{x}_k).$$

Thus given an initial matrix B_0 (e.g. the finite-difference approximation to the Jacobian matrix), the subsequent matrices are generated by:

$$B_{k+1} = B_k + \frac{(\underline{y}_k - B\underline{s}_k)\underline{s}_k^T}{\|\underline{s}_k\|_{21}^2}$$

where $s_k = x_k + 1 - x_k$, $y_k = F(x_{k+1}) - F(x_k)$.

The convergence of Broyden's Method is fast, but as in the case of Newton's Method it requires a good choice of the initial vector x_0.

Steepest Descent Method

In the Steepest Descent Method the problem of solving a system of nonlinear algebraic equations is transformed into a minimization problem [5]. A relative minimum x^* of the function $f(x, y)$ with known partial derivatives $g = \frac{\partial f}{\partial x}$, $h = \frac{\partial f}{\partial x}$ is located starting from a given initial guess (x_0, y_0). Using a given step size c a sequence of steps is generated according to:

$$\underline{x}_n = \underline{x}_n - \frac{c\underline{g}}{\sqrt{\underline{g}^2 + \underline{h}^2}},$$

$$\underline{y}_n = \underline{y}_n - \frac{c\underline{h}}{\sqrt{\underline{g}^2 + \underline{h}^2}}.$$

The stopping criteria is met when the relative minimum x^* is located and

$$\sqrt{\underline{g}^2 + \underline{h}^2} < \varepsilon.$$

The convergence rate of the steepest descent method is slow, but it works for any choice of starting vectors.

Trust Region Method

In the Trust Region Method we replace the Jacobian matrix with its approximation [6]. Then we calculate

$$\min\{\|f(\underline{x}_k) + B_k\delta\| : \quad \|D_k\delta\|_2 <= \Delta_k\},$$

where D_k is a scaling matrix and Δ_k is the trust region radius. The solution to this minimization problem is an approximate solution to the original problem. Stopping criteria is met when

$$p_k = \frac{\|f(\underline{x}_k)\| - \|f(\underline{x}_k + \delta_k)\|}{\|f(\underline{x}_k)\| - \|f(\underline{x}_k + B_k\delta_k)\|}$$

is smaller than some constant σ_0 (typically .0001). Otherwise, we decrease the radius of the trust region and re-solve the minimization problem.

The convergence rate of this method is slow, but it can use an arbitrary starting solution vector.

Experimental Results

We have experimented with the algorithms described above. We used two implementations of the Brown's method, our own implementation (based on [7]) and the SOS algorithm from [8]. Two hybrid methods, an algorithm by Kucaba-Piętal [9, 10] and the HYBRDI algorithm (a combination of Trust Region, Steepest Descent, and Newton's methods) from [8]. All algorithms were Fortran-based implementations and the experiments were run in double precision on a PC with a Pentium Pro 200 MHz processor. For the numerical tests we have used 17 problems found at [11, 12, 13]. These problems come from the three collections of test problems (with complete source code) for the solution of systems of nonlinear algebraic equations that we were able to locate. While some of the problems come from the real applications, others are artificially generated with properties not typical for real life applications. The problems used were (see the references for the complete formulation):

1. Rosenbrock's function [11]	2. Powell singular function [11]
3. Powell badly scaled function [11]	4. Wood function [11]
5. Helical valley function [11]	6. Watson function [11]
7. Chebyquad function [11]	8. Brown almost-linear function [11]
9. Discrete boundary value function [11]	10. Discrete integral equation function [11]
11. Trigonometric function [11]	12. Variably dimensioned function [11]
13. Broyden tridiagonal function [11]	14. Broyden banded function [11]
15. Exponential/Sine Function [12]	16. The Freudenstein-Roth function [13]
17. Semiconductor Boundary Condition [12]	

Table 1: Comparison of performance of nonlinear solvers for the test problems.

		QUASI_A		HYBRD		SOS		BROWN	
#	n	IT	FC	IT	FC	IT	FC	IT	FC
1	2	8	33	8	22	1	10	5	20
2	4	*85*	164	*83*	108	*11*	168	*18*	238
3	2	87	266	87	181	13	55	14	65
4	4	52	142	52	94	*16*	188	*820*	11466
5	3	14	38	14	27	19	174	8	63
6	6	38	127	38	96	23	648	36	987
6	9	57	180	57	182	8	486	30	1629
7	5	11	24	11	17	3	50	10	180
7	6	10	34	10	25	7	111	12	297
7	7	11	27	11	20	3	84	20	665
7	8	*30*	153	*30*	120	*14*	660	*88*	3884
7	9	19	57	19	43	8	486	18	918
8	10	8	37	8	31	5	280	9	520
8	30	*1*	110	*1*	85	*8*	2130	*10*	4455
8	40	*1*	130	*1*	105	*9*	4500	*10*	7740
9	10	6	20	6	16	2	140	6	325
10	1	6	11	6	7	2	5	5	8
10	10	6	20	6	16	2	140	5	260
11	10	*30*	160	*30*	130	*4*	270	8	455
12	10	27	75	27	60	*4*	585	*9999*	9999
13	10	11	27	11	21	3	205	8	455
14	10	20	41	20	30	4	270	7	390
15	2	12	22	12	15	*1*	32	4	29
16	6	3	11	3	9	1	54	5	108
17	2	1	7	1	5	*1*	126	5	36

Table 1 summarizes the results obtained for the test problems for the four codes tested. Here, IT denotes the number of iterations, FC the number of times the function was evaluated. *9999* denotes lack of convergence while results ***italicized bold and underlined*** represent the lack of solution due to the lack of significant improvement over the previous five Jacobian evaluations. Some of the problems have been run varying the number of equations. In each case a number of starting points has been tried and the best results are reported. All problems converged for some starting point except #2 (for $n = 2$) and #7 (for $n = 8$). In addition to runs reported in the table, problems 6–14 were run for $n = 10, 20, 40, 80$, and 100 with multiple initial starting points. All converged except #6, #7, and #12 for $n = 40, 80$ and 100. Problems #15 and #17 for SOS ended because the Jacobian matrix was singular.

From the results presented in the table and the discussion above it is clear that the difficulty of the system increases with the number of equations. The system either becomes unsolvable or the solution requires more iterations (and thus function evaluations). The numbers also show that the hybrid methods require much less function evaluation (while they require more irerations) than the implementations of the Brown's method (which require fewer iterations, but substantially more function evaluations).

In all experiments we have observed extreme sensitivity of the solvers to the selection of the starting vector x_0. This becomes more of a problem as the number of equations increases.

At this moment it is difficult to assess which of the test examples should be kept when a library of test cases will be compiled. The tests used here clearly cover a wide spectrum of functions but they clearly do not exhaust the possibilities arising in practical engineering applications. It should also be stressed that in real-life engineering applications systems of 100's of nonlinear algebraic equations have to be solved for the method to have a real predictive power (see for instance [9, 10]). Only few of the examples found can be even extended to this many equations.

Conclusions and Future Work

In this note we have presented an initial report of our comparisons between solvers for systems of nonlinear algebraic equations. We have found that methods based on similar algorithms behave similarly and the implementation details have only a minimal impact on the performance. All methods, regardless of their underlying algorithm, showed high sensitivity to the starting vector and this sensitivity increased as the number of equations in the system increased. We were not able to come to firm conclusions on the quality of the collection of test sets.

In the near future we plan to proceed in three directions. First, we will expand the test set by including additional recently located test problems. We will experiment with the described above codes on these sets. We will also use the Homotopy and Continuation methods on all test problems. Second, we will use the experimental data to attempt at evaluating the test set and to develop a library of test problems. Finally, we plan to investigate various methods for finding the starting vectors.

Acknowledgements

Work of Deborah Dent was sponsored from the Long Term Training Program, US Army Corps of Engineers, Waterways Experiment Station, Vicksburg, Mississippi, USA.

References

1. W.C. Rheinboldt, Methods for Solving Systems of Nonlinear Equations, Society for Industrial and Applied Mathematics, Philadelphia (1998).

2. G.H. Hostetter, M.S. Santina and P.D'Carpio-Montalvo, Analytical, Numerical and Computational Methods for Science and Engineering, Prentice Hall, New Jersey (1991).

3. K.M. Brown, SIAM J. Num. Anal., **6**, 560-569 (1969).

4. http://www.mcs.anl.gov/otc/Guide/OptWeb/continuous/unconstrained/nonlineareq/broydn.html

5. R.L. Burden and J.D. Faires, Numerical Analysis, PWS-Kent Publishing Company, Boston (1993).

6. http://www.mcs.anl.gov/otc/Guide/OptWeb/continuous/unconstrained/nonlineareq/trls.hml

7. W. Prosnak, Introduction to Fluid Mechanics, (in Polish). Ossolineum, Wrocław (1993).

8. http://www.netlib.org/liblist.html

9. A. Kucaba-Piętal and L. Laudański, in: Proceedings of I Symposium on Dynamics of Mechanical Systems, Warsaw (1994).

10. A. Kucaba-Piętal and L. Laudański, Zeszyty Naukowe Politechniki Śląskiej, Seria Mechanika, **121**, 173 (1995).

11. J.J. More, B.S. Garbow and K.E. Hillstrom, ACM Trans. Math. Software, **7**, 17-41 (1981).

12. U.N. Weimann, A Family of Newton Codes for Systems of Highly Nonlinear Equations, ZIB Technical Report TR-91-10, Berlin (1991), http://weyl.zib-berlin.de/Documents/tr91-10/section1.1.4.1.html.

13. J.J. More, A Collection of Nonlinear Model Problems, Preprint MCS-P60-0289, Mathematics and Computer Science Division, Argonne National Laboratory (1989).