Special Conference Edition November, 2017

http://dx.doi.org/10.4314/bajopas.v10i1.115S

Bayero Journal of Pure and Applied Sciences: 10(1): 615 - 622
ISSN 2006 – 6996

A GLOBALLY CONVERGENT HYPERPLANE- BFGS FOR SOLVING SYSTEMS OF NONLINEAR EQUATIONS

[1]**Waziri, M. Y. and** [2]**Musa, Y. B.**

[1]Department of Mathematical Sciences, Faculty of Science, Bayero University Kano, Kano, Nigeria,

[2]Department of Mathematics and Computer Sciences, Sule Lamido University, Kafin Hausa, Jigawa, Nigeria

balarabemusa.yau@jsu.edu.ng

ABSTRACT

This paper presents a Globally Convergent Hyper plane-BFGS method for solving nonlinear system of equations. The attractive attributes of our method are due to singularity free requirements and global convergence properties. Numerical performance on some benchmarks problems that demonstrates there liability and efficiency of our approach are reported and shown that the proposed method is very rigorous and efficiently competitive.

Keywords: Hyperplane, Secant, Algorithm, Global convergence

INTRODUCTION

In this paper, we consider the problem of finding the solution of the nonlinear equation

$F(X) = 0$ (1)

Where

$F: \mathbb{R}^n \longrightarrow \mathbb{R}^n$ (2)

is continuously differentiable function. We denote $F = (f_1, f_2, f_3, \ldots, f_n)^T$, and the vector $X = (x_1, x_2, x_3, \ldots, x_n)$. Quasi-Newton's methods are among the numerous efficient algorithms for solving (1). Due to nonlinearity of F,(1) may have no solution. In this work, we assume that the solution set of(1) denoted by X^*, is non-empty.

One special future so far observed is that, all practical algorithms for solving (1) are iterative (Ortega, 1970; Denis *et al.*, 1973;Dennis, 1987; Kelly, 1995; Solodov, 1998; Dai, 2002).Moreover, much effort has been made to establish global convergence of quasi-Newton methods for unconstrained optimization problems, for example (Denis *et al.*, 1973;Dennis, 1983; Dai, 2002;Nocedal *et al.*, 2002).

However, the study of globally convergent quasi-Newton methods for solving nonlinear equations is relatively fewer. The major difficulty is the lack of practical line search strategy (Dennis, 1983; Krejic and Luzanin, 2001; Zhang, 2013; Urroz, 2014).

The BFGS method for solving (1) is to generate a sequence of iterates x_k by letting $x_{k+1} = x_k + a_k d_k$, where a_k is a step length, and d_k is a solution of the system of linear equations.

$B_k d_k + F_k = 0$ (3)

Where $F_k = F(x_k)$, B_k is generated by the following BFGS update formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \quad (4)$$

Where $s_k = x_{k+1} - x_k$, $y_k = F_{k+1} - F_k$.

This paper is organized as follows. In section two, the BFGS preliminaries are stated. Section 3 consists of BFGS-Algorithm. Preliminary numerical results are proposed in Section 4, where the summary and conclusion occupy the last section.

Preliminary Results

The scheme of the Globally Convergent BFGS method for non linear system of equations developed by Wei and Li (2008) requires a lot of assumptions which include invertibility (non-singularity) of the BFGS update at thesolution.In this section, we present our scheme via regularization technique so as to remove the expected singularity of the update matrix. We also modified the parameters r and h in the default algorithm such that they come from abounded interval so that update matrix divergence is prevented. (Refer to the scheme below) The BFGS scheme in (Zhou and Li, 2008) is given by

$d_k = -B^{-1}F_k$

$z_k = x_k + a_k d_k$

$x_{k+1} = x_k - \frac{\langle F(z_k), x_k - z_k\rangle}{||F(z_k)||^2} F(z_k)$

Where B_k is an BFGS -update matrix such that

$s_k = z_k - x_k = a_k d_k, y_k = F(z_k) - F(x_k) + h||F(x_k)||^r s_k, h > 0, r \geq 0$

We propose a new scheme where the update $B_k \approx (B_k + \lambda_k I)$.

$\lambda_k = ||F_k||^{\delta}$, $\delta \in (0, 2]$, $r \in [0,1)$ and $h = \frac{1}{2}$ so

we have,

$d_k = -(B_k + \lambda_k I)^{-1} F_K$

$z_k = x_k + \alpha_k d_k$

$x_{k+1} = x_k - \frac{<F(z_k), x_k - z_k>}{||F(z_k)||^2} F(z_k)$, (6)

, $s_k = z_k - x_k$, $y_k = F(z_k) - F(x_k) + h||F(x_k)||^r s_k$

Hence, by adding $\lambda_k I$ to the update B_k, the update is now symmetric and regularized and thereforeinvertible.

ALGORITHM

We denote the method as Globally Convergent Hyperplane BFGS- Method for solving nonlinear system of equations (**GH-BFGS**). But firstly, we define a Hyperplane as

$H_k = \{x \in \mathbb{R}^n | \langle F(z_k), x_k - z_k\rangle = 0\}$.(7)

We present the stages of implementation for our algorithm as follows

Algorithm (GH-BFGS Method)

Step 0. Given an initial point $x_0 \in \mathbb{R}^n$ and constants $\beta, \sigma \in (0, 1)$, $h = \frac{1}{2}$, $r \in [0, 1)$ and $\delta \in (0, 2]$. Choose $B_0 = I$.

Let $k := 0$

Step 1. Compute d_k by $(B_k + \lambda_k I) d_k = -F_k, \lambda_k = ||F_k||^{\delta}$. (8)

If $d_k = 0$ stop.

Step2. Determine step length $a_k = \beta^{m_k}$ such that m_k is the smallest nonnegative integer m satisfying

$-\langle F(x_k + \beta d_k), d_k\rangle \geq \sigma\beta ||F(x_k + \beta d_k|| \, ||d_k||$ (9)

.Let $z_k = x_k + a_k d_k$

If $||F(z_k)|| = 0$ stop

Step 3.Compute

$$x_{k+1} = x_k - \frac{<F(z_k), x_k - z_k>}{||F(z_k)||^2} F(z_k), \quad (10)$$

Step 4. Compute B_{k+1} by the following BFGS update process

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \quad (11)$$

$s_k = z_k - x_k$, $y_k = F(z_k) - F(x_k) + h||F(x_k)||^r s_k$(12)

set k=k+1 .

Go to Step1.

Remarks (i) If we suppose that F is Lipschitz continuous, i.e., there exists a constant $L > 0$ suchthat

$||F(x) - F(y) \leq L||x - y||, \forall x, y \in \mathbb{R}^n$(13)

hence, from the monotonocity and Lipschitz continuity of the function F,

$y_k = F(z_k) - F(x_k) + h||F(x_k)||^r s_k$, this implies

$y_k^T = (F(z_k) - F(x_k) + h||F(x_k)||^r s_k)^T$ and

$y_k^T s_k = (F(z_k) - F(x_k) + h||F(x_k)||^r s^T s_k$

since $s_k = z_k - x_k$, then $z_k = s_k + x_k$,

we have,

$y_k^T s_k = F(z) + h||F(x_k)||^r s_k^T s_k$, clearly

$$h||F(x_k)||^r s_k^T s_k \leq y_k^T s_k \leq (L + h||F(x_k)||^r) s_k^T s_k \quad (14)$$

The T denotes transpose of the vectors s_k and y_k.

(ii) The update formular in(11) is different from the one used in (Li et al, 1999)

(iii) We used the same line search as used by Wei and li in(Zhou et al, 2008)

(iv) The BFGS update in (11) is both positive definite and symmetric and hence non-singular at thesolution.

(v) The algorithm has the same convergence properties as that in (Zhou et al, 2008)

Numerical results1

In this section, we report some numerical results of our proposed method and that of Globally Hyper plane BFGS method(GH-BFGS), the regularized (RBFGS) and the BFGS in (Zhou et al, 2008).We have tested our algorithms extensively on exactly 9 number of non- linear systems. Here, we report the results for the 9 problems, whose statements are given in Appendix A. We run the algorithm on the 9 test problems with dimensions n = 10, n=20, n=50,...,n=1000 as shown in our table. Different starting points havebeen used. Since these initial points are independent of the optimal solution x, we can view them as arbitrary initial points. The results are summarized in Table 1 and 2. For each test we report, the dimension(n), the number of iterations (NI) and the cpu-time (CPUTime). The numerical computations were carried out using MATLAB 2010a on a PC with intel COREi5 processor with 4 GB of RAM and CPU 1.70 GHZ. As stated, We used 9 test problems with dimension between 10 to 1000 in order to test the advantages of the proposed method in terms of less number of iterations (NI) and the CPU time (in seconds) . The iteration stops for $||J_k F_k|| \leq 10^{-6}$(Yuan G, et al, 2008)

However, we declare that the algorithm fails if the followings occur during iteration.

1. Insufficient memory to execute thecode.
2. Attainmentofsingularitybythematrixunderconsideration.Weusethesymbol

—- if the algorithm fails to find a solution.

Appendix A

Problem F1 Spare function of Beyong (Beyong et. al,2010)

$F_i(x) = (x_i^2 + x_i - 3)\log^{x_i+3} - 9, i = 1, 2, 3, \ldots, n$

and

$x_0 = (2, 2, 2\ldots, 2)$

Problem F2 (System of nonlinear equations)

$F_i(x) = (x_i^2 - 1)^2 - 2, i = 1, 2, 3, \ldots, n$

and

$x_0 = (-1.2, -1.2, -1.2\ldots, -1.2)$

Problem F3 (System of nonlinear equations)

$f_i(x) = (0.5 - x_i)^2 + x_i^2 - 1, i = 1,2,3,\ldots,n$

$x_0 = (0.5, 0.5, 0.5, \ldots, 0.5)^T$

Problem F4 (Trigonometric/Exponential System of nonlinear equations)

$f_i(x) = sinx_i - 4e^{2-x_i} + 2x_i, i = 1, 2, 3, \ldots, n$

and

$x_0 = (0.05, 0.05, 0.05, \ldots, 0.05)^T$

Problem F5 (Extended System of Byoeng, 2010)

$f_i(x) = cos(x_i^2 - 1) - 1, i = 1, 2, 3, \ldots, n$

and

$x_0 = (0.5, 0.5, 0.5, \ldots, 0.5)^T$

Problem F6 (System of nonlinear equations)

$f_i(x) = (\sum_{i=1} x_i + i)(x_i - 1) + e^{x_i} - 1, i = 1,2,3,\ldots,n$

and

$x_0 = (3, 3, 3\ldots, 3)^T$

Problem F7 (Roose et.al, 1990)

$f_i(x) = x_i - 1/n^2 (\sum^n x_i)^2) + (\sum^n x_i) - n, i = 1,2,3,\ldots,n$

and

$x_0 = (4,4,4\ldots,4)^T$

Problem F8 (System of nonlinear equations)

$f_i(x) = sin(1 - x_i) \sum_{i=1}^{n} x_i^2 + 2x_{n-1} - 3x_{n-2} - 0.5x_{n-4} + 0.5x_{n-5} - x_i log(9 + x_i) - 4.5e^{1-x_n} + 2, i = 1, 2, 3, \ldots, n$

and

$x_0 = (7, 7, 7\ldots, 7)^T$

Problem F9 (System of nonlinear equations)

$f_i(x) = 5x_i^2 - 2x_i - 3, i = 1, 2, 3, \ldots, n$

and

$x_0 = (0.5, 0.5, 0.5, \ldots, 0.5)^T$

Computational Experiments

The Tables below, present comparison of the three methods, (RBFGS),GC- BFGS) and GH-BFGS. The meanings of the columns in Tables 4.1 and 4.2 are stated as follows: n:the dimension of the problem; NI: the total number of iterations; CPUtime: the CPUtime in seconds; $i = i = (1,2,3,\ldots,n)$

2.1 Performance Profile

Below are the figures indicating the performances of the new methods in comparison to the existing methods. The comparison was conducted in terms of number of iterations and CPU- time.

In this section, we report the performance of sour proposed method i.e GH- BFGS and that of the RBFGS and GC-BFGS. In Table 1 and 2, we can observe that the algorithm for GC-BFGS failed in Problems 2,5,6,8 and 9 due to singularity attained by the BFGS- update. Moreover, the numerical results show that the GH-BFGS method solve some nonlinear problems where other methods failed, e. ginproblems6, 8 and 9.Similarly, from the table, our proposed method is a fully derivative free approach which makes it capable of handling large-scale nonlinear systems of algebraic equations without failing and it can also solve some problems which encountered singularity e.g. in problems 6, 8 and 9. Hence, these show the reliability of our proposed method, in term of solving singular problems, minimum number of iterations and cputime.

The Figures (1-4) show the performance of these methods relative to CPU time and number of iteration, which were evaluated using the profile of Dolan and More. That is, for each method, we plot the fraction $p(\tau)$of the problems for which the method is within a factor of the best time. Clearly, the proposed method is more efficient in all aspects i.e. less CPUtime and number of iterations.

Table1:Problem F1—F4

problem	Dimension	RBFGS		GC-BFGS		GH-BFGS	
	N	NI	CPU time	NI	CPU Time	NI	CPU Time
F1	10	10	0.078391	18	0.11457	6	0.012106
	20	6	0.007214	21	0.347894	6	0.011766
	50	8	0.019124	25	0.4422557	6	0.019091
	100	10	0.069882	63	0.892091	12	0.186724
	200	12	0.51919	8	0.45673	5	0.302722
	500	15	3.312338	4	1.12126	7	52.249362
	1000	17	22.622404	5	6.774975	7	293.594706
F2	10	8	0.007231	13	0.323835	5	0.010492
	20	8	0.013302	94	0.336452	4	0.009595
	50	8	0.016868	6	0.01434	4	0.014416
	100	8	0.0435224	15	0.221797	9	143.08678
	200	9	0.3606033	14	0.388576	7	606.414219
	500	15	3.285598	-	-	5	0.15231
	1000	14	18.577114	-	-	9	0.3.4352
F3	10	8	0.008356	4	0.010457	3	0.008102
	20	8	0.008274	4	0.011311	3	0.005993
	50	9	0.01886	4	0.01639	4	0.012843
	100	10	0.06556	4	0.034664	6	.041764
	200	8	0.422803	4	0.160336	3	0.339954
	500	13	2.949401	15	3.285598	6	0.01234
	1000			-	-	-	-
F4	10	39	0.029919	11	0.207654	11	0.010292
	20	38	0.034947	11	0.240135	12	0.026409
	50	39	0.077549	12	0.110443	12	0.039135
	100	40	0.376645	12	0.171902	13	0.090552
	200	42	1.190021	12	0.550433	14	0.592209
	500	38	3.3327	16	3.502367	8	32.115323
	1000	43	18.69678	-	-	9	196.847685

Table2:problem F5—F9

Problem	Dimension	BFGS		GC-BFGS		GH-BFGS	
	N	NI	CPU time	NI	CPU Time	NI	CPU Time
	10	10	0.192894	9	0.016364	4	0.025726
	20	10	0.136975	12	0.199982	10	0.053257
	50	17	0.035768	12	0.045732	14	0.054418
F5	100	20	0.25247	10	7.11197	26	0.255919
	200	20	0.770961	27	0.939748	21	0.892675
	500	20	4.347882	26	5.539537	23	6.074027
	1000	20	27.169179	-	—-	32	44.269853
	20	8	0.014474	-	-	15	0.030286
	50	9	0.019419	-	-	408	1.652439
F6	100	15	0.147855	-	-	143	1.558488
	200	20	0.745675	-	-	201	23.983052
	500	20	4.301953	-	-	407	34.9898
	1000	20	26.631707	-	-	569	54.99999
	10	8	0.007098	5	0.012349	4	0.008726
	20	7	0.008573	5	0.013236	3	0.006239
	50	9	0.018707	5	0.019739	5	0.012424
F7	100	10	0.065893	5	0.042617	7	0.062937
	200	10	0.488916	5	0.042617	8	0.319449
	500	13	3.160063	5	1.183624	10	2.275631
	1000	13	22.690442	5	6.910771	13	17.200156
	10	30	0.04161	-	-	31	0.260028
	20	284	0.705477	-	—	156	0.586409
	50	49	0.425362	-	—	36	0.241306
F8	100	43	0.656513	-	—	708	7.484649
	200	124	3.35449	-	-	34	2.977987
	500	441	98.700629	-	—	4	17.204617
	1000	4	84.497552	-	—	5	107.233772
	10	7	0.006112	10	0.013199	8	0.02583
	20	8	0.012902	10	0.2903	5	0.27221
	50	9	0.021539	63	0.378012	32	0.224946
F9	100	10	0.07193	-	-	173	1.800175
	200	12	0.581812	-	—	230	26.641631
	500	14	3.162133	—	—	4638	127.57515
	1000	19	25.57476	-	-	5	107.233772

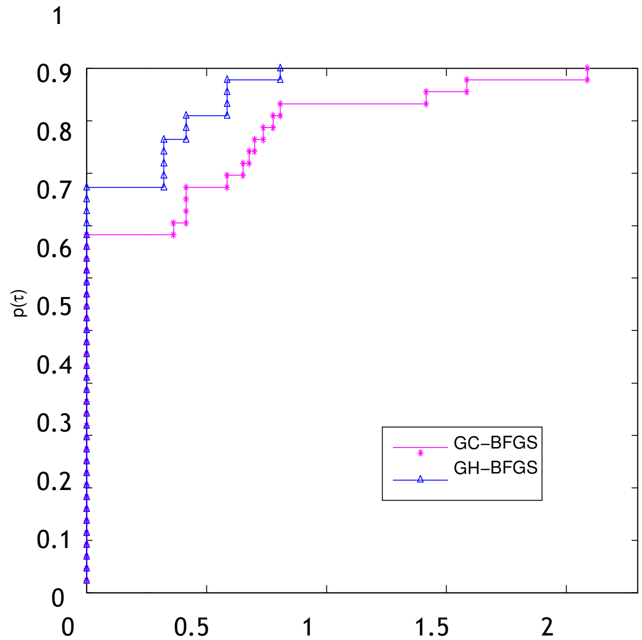

Figure 1: Performance profile of GC-BFGS and GH-BFGS methods with respect to number of iterations for problem 1-9

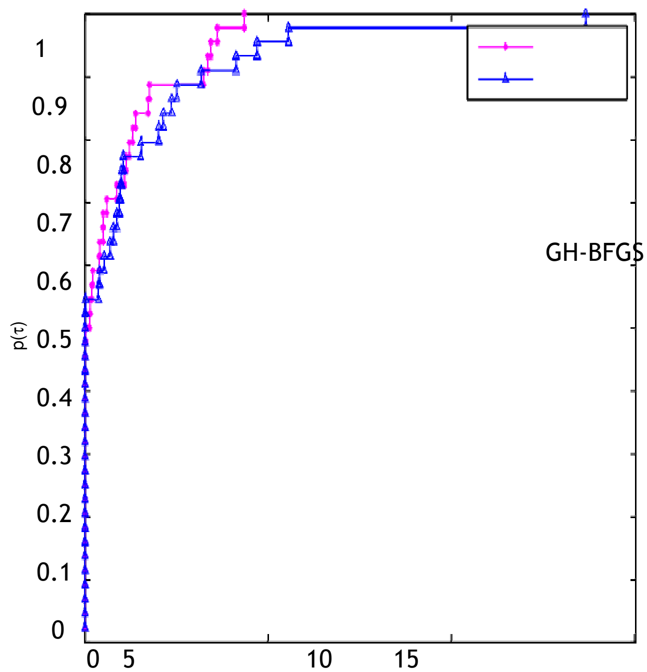

Figure 2: Performance profile of GC-BFGS and GH- BFGS methods with respect to CPU-time for problem 1-9

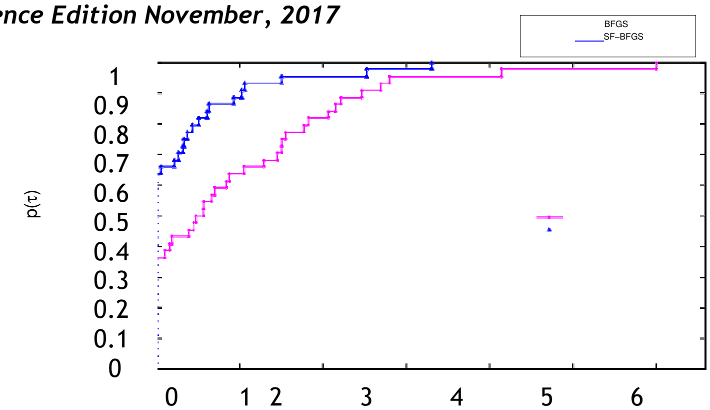

Figure 3: Performance profile of RBFGS and GH- BFGS methods with respect to CPU-time for problem 1-9

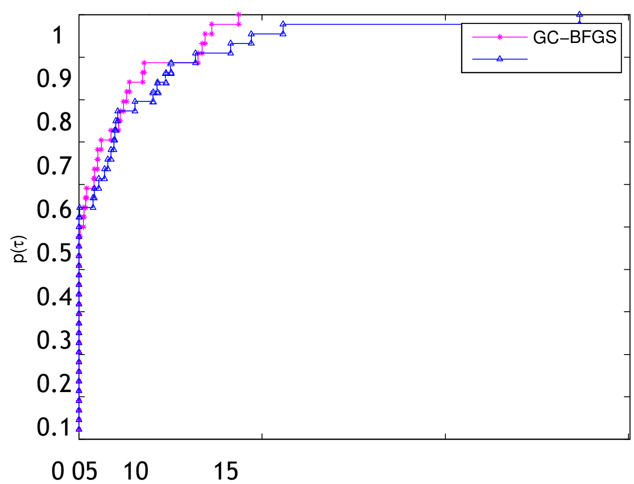

Figure 4: Performance profile of RBFGS and GH- BFGS methods with respect to CPU-time for problem 1-9Finalremarks

REFERENCES

Dai Yuan (2002). Convergence properties of the BFGS algorithm.13:693–701.

Dennis, J.E., Broyden, C.G. and Jr. J.J. Mores (1973). On the local and superlinear convergence of quasi-Newton methods. 12:223–246.

Dennis, J.E. Jr. (1987). A variable Metricvariant of the karmarkaralgorithm for linear programming. 39:1–20.

Dennis J.E. and R.B. Schnabel (1983). Numerical Methods for unconstrained optimization and nonlinear equations. 7:67–78.

Jorne M. J. and Ortega (1970). Iterative solution of nonlinear equations in several variable. 23:1–16.

Kelly. C.T. (1995). Iterative methods for linear and nonlinear equations. 23:1–20.

Krejic N. and Z. Luzanin. (2001). Newton-like method with modification of right-hand vector. 237:237–250.

Li and Fukshima.(1999).A globally and super linearly convergent Gauss-Newton based BFGS symmetric methods for solving non linear systems of equations. 37:152–172.

Moore V. Solodov and B.F. Svaiter. (1998). A globally convergent inexact Newton method for systems of monotone equations. 34:355–369.

Nocedal R. Byrd and Y.X. Yuan.(2002). Global convergence of a class of quasi-Newton methods on convex problems. 24:693–701.

Urroz G. E (2014). Solutions of nonlinear equations. 13:01–18.

WanjieZhang (2013). Method for solving nonlinear system of equations. 113:01

Yuan G. and X. Lu. (2008). A new back tracking inexact BFGS method for symmetric nonlinear equations. 55:116–129.

Zhou W.J. and D.H. Li. (2008). A globally convergent BFGS method for nonlinear system of equation. 43:2231–2240.