Journal of Engineering and Applied Sciences 14 (Special Issue 7): 10080-10086, 2019
ISSN: 1816-949X

A New Line Search Method to Solve the Nonlinear Systems of Monotone Equations

Karrar Habeeb Hashim, Nabiha Kahtan Dreeb, Hasan Hadi Dwail, Mohammed Maad Mahdi, H.A. Wasi, Mushtak A.K. Shiker and Hussein Ali Hussein
Department of Mathematics, College for Pure Science, University of Babylon, Hillah, Iraq

Abstract: In this study, we suggest a new line search algorithm for solving nonlinear systems of equations such that we combine a monotone technique into a modified line search rule. The new proposed algorithm can decrease the CPU time, the number of iterations and the function evaluations and can increase the efficiency of the approach. Under some standard conditions, the global convergence of the algorithm is proved. Preliminary numerical results shows that the new algorithm is promised for solving nonlinear systems of equations monotone equations.

Key words: Nonlinear system of equations, line search method, Monotone strategy, global convergence, numerical results, iterative method

INTRODUCTION

The nonlinear systems are one of the problems that arise in different fields of science and computational geometry, especially in the interpretation of nonlinear partial differential equations, the problem of specified value, etc. There are situations in which thousands of nonlinear equations can be solved in some independent variables effectively. Thus, finding the roots of nonlinear systems of equations has many applications in numerical and applied mathematics.

Therefore, the focus of many researchers is to find and provide appropriate ways and means to solve these non-linear systems and thus some common algorithms are suggested to solve these problem.

Nonlinear equations are one of the most important problems of multiple scientific uses such as computer science tremolo systems (Ortega and Rheinboldt, 1970; Zeidler, 2013), the first-order necessary condition for the problem of unconstrained convex optimization and also some sub-problems in generalization (Iusem and Solodov, 1997; Shiker and Sahib, 2018).

Since, the fixed points that can be found from the problem of improvement are equal to find the answer of a non-linear system of equations and the systems of nonlinear equations can be converted into problem of the lower squares this indicates a close relationship between the problems of unconstrained optimization and systems of nonlinear equations, so, it is appropriate to use unconstrained optimization algorithms to solve this problem.

One of the two important iterative methods that is used to solve nonlinear system of equations is the line search strategy, the other method is trust region. Here, we focus on the line search method and its framework. This method is fairly simple, so, its understanding and application is easy. However, they are ineffective and have some disadvantage, for example, if the array being searched for contains 30.000 items, to find the value of the last element, the algorithm will have to look at all those 30.000 elements. Typically, if we have a matrix of M elements, the linear search will identify an element in M/2 attempts. For example, if we have a matrix of 40.000 items, the linear search will compare with 20.000 items in a typical case. This is through the possibility to find the search element constantly in the array, so, the number M is always maximum in comparisons. An another disadvantage, on the large scale, the research and convergence of the line search method are slow. So, most of researchers used the monotone strategy to address that problem. Consider the nonlinear system of equations:

$$F(x) = 0 \tag{1}$$

where, $F: R^n \rightarrow R^n$ is continuous and monotone, i.e:

$$\langle F(x)-F(y), x-y \rangle \geq 0, \forall x, y \in R^n$$

By fixed point map or a natural map, some monotone variational inequality can be converted into nonlinear monotonous equations but before that there are some coercive conditions that the basic function has to achieve.

Corresponding Author: Karrar Habeeb Hashim, Department of Mathematics, College for Pure Science, University of Babylon, Hillah, Iraq

Quasi-Newton methods are considered to be one of the most important algorithms for solving problem (Eq. 1), the methods of Quasi-Newton have been a major advance in the theoretical aspect as a result of the development of solutions to many problems and this is especially, reflected in the analysis of local convergence (Broyden *et al*., 1973; Dennis and More, 1977). In addition, researchers have done a lot of work to create a global approximation of Quasi-Newton methods for unconstrained optimization problems see (Byrd *et al*., 1987; Amini *et al*., 2016; Nocedal, 1980 and Shiker and Amini, 2018).

By Griewank (1986) who is considered to be the closest approximation of global convergence, suggested a derivative-free line search. By Li and Fukushima (2000) had another view by constructing and deducing an example showing that the line search by Griewank (1986) contains in some special cases certain difficulties. As a result of their research and by using the non-monotonous line search method, they suggested a Gauss-Newton based BFGS method to solve nonlinear symmetric equations and a Broydens method to solve nonlinear equations also they proved these methods converge globally (Li and Fukushima, 1999, 2000). However, some of the merit functions such as the quadratic merit function are used to ensure the global approximation of Quasi-Newton.

MATERIALS AND METHODS

In this study, the new algorithm is used to solve the nonlinear monotone equations and we proved that it has a global convergence without using merit function. In comparison with BFGS method by Zhou and Li (2008) and PRP method by Cheng (2009), the new method well be more efficient. Now, we will give our algorithm.

The new algorithm (K)

Step 0. Choose an initial point $x_0 \varepsilon R^n$ and constants $\mu\in(0, 1)$, $\rho\in(0, 1)$, $\beta\in[1/2, 1)$, $\sigma\in(0, 1/2]$, $m>0$, $r>0$. Let $k: = 0$

Step 1. Compute the search direction d_k by:

$$d_k = -F(x_k) \tag{2}$$

Stop if $d_k = 0$

Step 2. Determine step length $\alpha_k = \mu^{hk}\beta$ such that h_k is the smallest nonnegative integer h satisfies:

$$-\left\langle F(x_k+\mu^h\beta d_k),\, d_k\right\rangle \geq \rho\sigma_k\mu^h\beta\left\|F(x_k+\beta\mu^h d_k)\right\|\,\|d_k\|^2 \tag{3}$$

Where $\sigma_k = \dfrac{\sigma}{1+\|F(z_k)\|}$

Let $z_k = x_k+\alpha_k d_k$

Stop if $\|F(z_k)\| = 0$

Step 3. Calculate

$$x_{k+1} = x_k - \frac{\left\langle F(z_k),\, x_k-z_k\right\rangle}{\|F(z_k)\|^2}F(z_k) \tag{4}$$

Set k: = k+1 Go to Step 1.

Remark: The mapping F is Lipschitz Continuous (LC), satisfies for a positive constant L>0 that:

$$\|F(x)-F(y)\| \geq L\|x-y\|,\ \forall\, x, y\in R^n \tag{5}$$

It is clear that L+m>m, so:

$$\frac{\|F(x_k)\|}{L+m} \leq \|d_k\| \leq \frac{\|F(x_k)\|}{m} \tag{6}$$

Now, we will show that the line search (3) is well-define in a similar way to Solodov and Svaiter (1998). Suppose that for some iteration index k and for any nonnegative integer h, the line search (3) is not satisfied, i.e.:

$$-\left\langle F\left(x_k+\mu^h\beta d_k\right),\, d_k\right\rangle < \rho\sigma_k\mu^h\beta\,\|F\left(x_k+\beta\,\mu^h d_k\right)\|\,\|d_k\|^2\ (*)$$

Now if, we take lim⊤(h→∞) for two side to (*):

$$-\lim_{h\to\infty}\left\langle F\left(x_k+\mu^h\beta d_k\right), d_k\right\rangle < \lim_{h\to\infty}\rho\sigma_k\mu^h\beta$$
$$\left\|F\left(x_k+\beta\,\mu^h d_k\right)\right\|\,\|d_k\|^2$$
$$\Rightarrow\quad -\left\langle F(x_k),\, d_k\right\rangle<0$$
$$\Rightarrow\quad -\left\langle F(z_k-\alpha_k d_k),\, d_k\right\rangle<0\ \ (\text{since, } x_k = z_k - \alpha_k d_k)$$
$$\Rightarrow\quad -(-\alpha_k)\left\langle F(z_k+d_k),\, d_k\right\rangle<0$$
$$\Rightarrow\quad \alpha_k\left\langle F(z_k+d_k),\, d_k\right\rangle<0$$
$$\Rightarrow\quad \alpha_k\,\|F(z_k)\|\|\|d_k\|^2<0$$

Then, we have a contradiction, since, it is not possible to have each of α_k, $\|F(z_k)\|$ and $\|d_k\|^2$ less than zero, so, the line search is well-defined.

Convergence property: In this study, to obtain the global convergence of our algorithm then, we need the following lemma.

Lemma 1: Solodov and Svaiter (1998) let, F be monotone and x, $y\in R^n$ satisfy $\langle F(y), x-y\rangle>0$. Let:

$$x^+ = x - \frac{\left\langle F(y),\, x-y\right\rangle}{\|F(y)\|^2}F(y)$$

Then for any $\bar{x} \in R^n$ such that $F(\bar{x}) = 0$ it holds that:

$$\| x^+ - \bar{x} \|^2 \leq \| x - \bar{x} \|^2 - \| x^+ - x \|^2$$

Now, we can state our convergence result by the following theorem similar to Solodov and Svaiter (1998).

Theorem 1: Suppose that F is LC and monotone and let $\{x_k\}$ be any sequence generated by algorithm (K). Also, we suppose that the solution set of 1 is nonempty. Then for any $\bar{x}$ satisfying, $F(\bar{x}) = 0$, we have:

$$\| x_{k+1} - \bar{x} \|^2 \leq \| x_k - \bar{x} \|^2 - \| x_{k+1} - x_k \|^2$$

In particular, the sequence $\{x_k\}$ is bounded. Also, its satisfy that either $\{x_k\}$ is finite and the last iterate is a solution or the sequence is infinite and:

$$\lim_{k \to \infty} \| x_{k+1} - x_k \| = 0$$

Furthermore, the sequence $\{x_k\}$ converges to some $\bar{x}$ such that $F(\bar{x}) = 0$.

Proof: First, if the algorithm finishes at some iteration k then: either $d_k = 0$, so by the positive definiteness of B_k, we get $F(x_k) = 0$ or $\|F(z_k)\| = 0$ in this case x_k or z_k will be a solution of 1. Now suppose that $d_k \neq 0$ and $F(x_k) \neq 0$ for all k, then:

$$\begin{aligned} \langle F(z_k), x_k - z_k \rangle &= \langle F(z_k), x_k - x_k - \alpha_k d_k \rangle \\ &= \langle F(z_k), -\alpha_k d_k \rangle \\ &= -\alpha_k \langle F(z_k), d_k \rangle \\ &= -\alpha_k \langle F(x_k + \alpha_k d_k), d_k \rangle \\ &\geq \rho\sigma_{\downarrow} k \| F(z_{\downarrow} k) \alpha_{\downarrow} k^{\uparrow} 2 \| d_{\downarrow} k \|^{\uparrow} 2 > 0 \end{aligned}$$

Then:

$$\langle F(z_{\downarrow} k), \dashv\vdash x_{\downarrow} k - z_{\downarrow} k \rangle = -\alpha_{\downarrow} k \langle F(z_{\downarrow} k), \dashv\vdash d_{\downarrow} k \rangle \geq \rho\sigma_{\downarrow} k \| F(z_{\downarrow} k) \| \alpha_{\downarrow} k^{\uparrow} 2 \| d_{\downarrow} k \|^{\uparrow} 2 > 0 \quad (7)$$

Let $\bar{x}$ be any solution of 1 and $F(\bar{x}) = 0$. From lemma 1, (4) and (12), we obtain:

$$\| x_{k+1} - \bar{x} \|^2 \leq \| x_k - \bar{x} \|^2 - \| x_{k+1} - x_k \|^2 \quad (8)$$

In particular, the sequence $\{\| x_k - \bar{x} \|\}$ is decreasing and hence convergent. Consequently, the sequence $\{x_k\}$ will be bounded and also we have:

$$\lim_{x \to \infty} \| x_{k+1} - x_k \| = 0 \quad (9)$$

By Eq. 6, it is clear that $\{d_k\}$ holds to be bounded and so is $\{z_k\}$. From Eq. 4:

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

Since, $\langle F(z_k), x_k - z_k \rangle = \alpha_k \langle F(z_k, d_k \rangle$ then:

$$\begin{aligned} &x_{\downarrow}(k+1) - x_{\downarrow} k = (\alpha_{\downarrow} k \langle F(z_{\downarrow} k), \dashv\vdash d_{\downarrow} k \rangle) / \\ &\| F(z_{\downarrow} k) \|^{\uparrow} 2 F(z_{\downarrow} k) \geq (\rho \| F(z_{\downarrow} k) \| \alpha_{\downarrow} k^{\uparrow} 2 \| d_{\downarrow} k \|^{\uparrow} 2) / \\ &F(z_{\downarrow} k) \| = \rho\alpha_{\downarrow} k^{\uparrow} 2 \| d_{\downarrow} k \|^{\uparrow} 2 \end{aligned}$$

So:

$$\begin{aligned} &\| x_{\downarrow}(k+1) - x_{\downarrow} k \| = \langle F(z_{\downarrow} k), \dashv\vdash x_{\downarrow} k - z_{\downarrow} k \rangle / \\ &\| F(z_{\downarrow} k) \| \geq \rho\alpha_{\downarrow} k^{\uparrow} 2 \| d_{\downarrow} k \|^{\uparrow} 2 \end{aligned} \quad (10)$$

From Eq. 9 and 10, we get:

$$\lim_{x \to \infty} \alpha_{\downarrow} k \| d_{\downarrow} k \| = 0, \lim_{k \in K, x \to \infty} \alpha_{\downarrow} k \| d_{\downarrow} k \| = 0 \quad (11)$$

From Eq. 6, we get lim in $f_{x \to \infty} \|F(x_k)\| = 0$, if lim in $f_{x \to \infty} \|d_k\| = 0$ then by Eq. 11, we get:

$$\lim_{x \to \infty} \alpha_k = 0 \quad (12)$$

Now, since, $\{x_k\}$ is bounded and by continuity of F, it is clear that $\{x_k\}$ has some accumulation point $\hat{x}$ with $F(\hat{x}) = 0$. We also have from Eq. 8 that the sequence $\{\| x_k - \hat{x} \|\}$ converges. Therefore $\{x_k\}$ converges to $\hat{x}$ Eq. 3 gives us:

$$\begin{aligned} &-\langle F(x_k + \mu^h_{k-1} \beta d_k), d_k \rangle < \rho\sigma_k \mu^h_{k-1} \beta \\ &\| F(x_k + \beta\mu^h_{k-1} d_k) \| \| d_k \|^2 \end{aligned} \quad (13)$$

Since, $\{x_k\}$, $\{d_k\}$ are bounded, so, we can choose a subsequence, let $k \to \infty$ in Eq. 13, we obtain:

$$-\langle F(\hat{x}), \hat{d} \rangle \leq 0 \quad (14)$$

Such that $\hat{x}$ and $\hat{d}$ are limits of subsequences that chosen. Otherwise by Eq. 6 and already familiar argument:

$$-\langle F(\hat{x}), \hat{d} \rangle > 0 \quad (15)$$

Equation 14 and 15 are a contradiction. Hence, it is not possible to get that:

$$\lim\inf_{k\to\infty} \|F(x_k)\| > 0$$

This finishes the proof.

RESULTS AND DISCUSSION

Numerical results: In this study, we compare the performance of the new method (K) discussed earlier with the following algorithms.

PRP: It is coming from Cheng (2009).

BFGS: It is coming from using the line search by Zhou and Li (2008) with the direction of this study. We wrote all the codes in MATLAB with version R2014a, also the experiments are running on a computer with 4 GB of RAM and CPU 2.30 GHz. The purpose of running the codes is to compare the results of the new algorithm (K) with the algorithms mentioned above.

When $\|F_k\| \leq 10^{-8}$ or $\|F(z_k)\| \leq 10^{-8}$ or the total number of iterates exceeds 500000 then all the algorithms will be end. In all of the algorithms, the parameters are specified as follows $\mu = 0.4$, $\rho = 0.3$, $\sigma = 0.25$, $\varepsilon = 10^{-8}$.

The comparison of these methods is based on three things: N_i (Number of iterations), N_f (Number of functions evaluations) and the CPU time. Also, the special dimensions to compare these algorithms are limited to 5000|50000 for the following initial points:

$$x_0 = (10, 10, ..., 10)^T, x_1 = (-10, -10, ..., -10)^T, x_2 = (1, 1, ..., 1)^T$$
$$x_3 = (-1, -1, ..., -1)^T$$

$$x_4 = \left(1, \frac{1}{2}, \frac{2}{3}, ..., \frac{1}{n}\right)^T, x_5 = (0.1, 0.1, ..., 0.1)^T,$$
$$x_6 = \left(\frac{1}{n}, \frac{2}{n}, ..., 1\right)^T, x_7 = \left(1-\frac{1}{n}, 1-\frac{2}{n}, ..., 0\right)^T$$

Numerical results are displayed in Table 1 and 2 the first table contains both of N_i and N_f for all algorithms while the second table contains CPU times of these algorithms.

In order to obtain a comprehensive comparison of the results obtained by our proposed algorithm and the two other algorithms used in the comparison, we use the performance profile provided by Dolan and More (2002) as a tool to evaluate these algorithms and compare them through durability and efficiency (Fig. 1-3).

From the comparisons of the results we can see the superiority of the new approach compared to other methods for solving the nonlinear systems of monotone equations. Figure 1 shows the performance for the total of

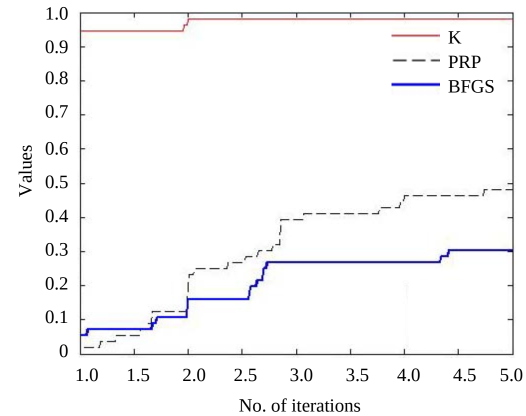

Fig. 1: Performance profile for the total number of iterations

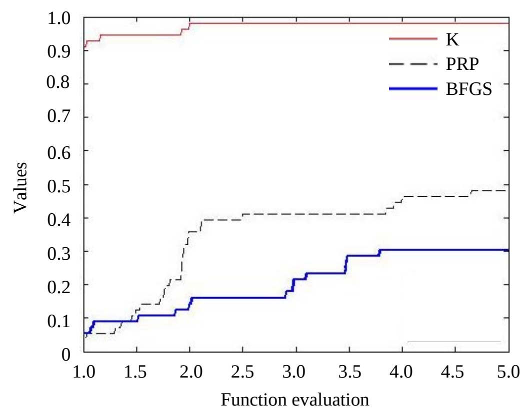

Fig. 2: Performance profile for the total number of function evaluation

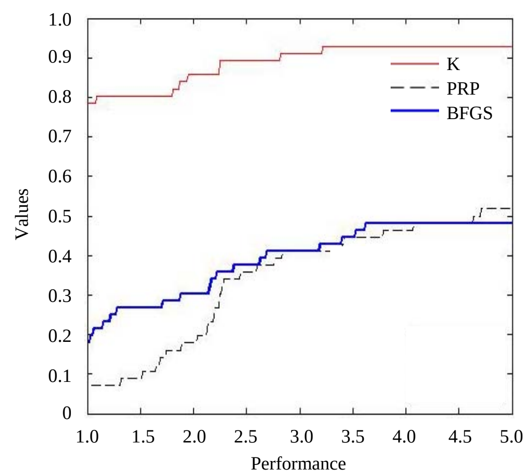

Fig. 3: Performance profile for the CPU time

Table 1: Numerical results

		New		PRP		BFGS	
P/Dim.	SP	Ni	Nf	Ni	Nf	Ni	Nf
P_1							
50000	x_0	16	145	188	994	1255	8837
50000	x_1	16	145	188	994	1255	8837
50000	x_2	14	101	40	194	148	798
50000	x_3	14	101	40	194	148	798
50000	x_4	15	81	25	115	15	79
50000	x_5	10	46	20	97	27	160
50000	x_6	19	136	45	202	49	254
50000	x_7	19	134	48	230	50	267
P_2							
50000	x_0	16	145	188	994	1255	8837
50000	x_1	14	109	196	1016	1327	9350
50000	x_2	14	101	40	194	148	798
50000	x_3	15	121	46	235	65	375
50000	x_4	15	81	71	376	16	89
50000	x_5	10	46	20	97	27	160
50000	x_6	18	126	50	221	49	254
50000	x_7	30	250	50	256	50	267
P_3							
10000	x_0	22534	135605	149031	911135	409228	2866839
10000	x_1	13699	87210	36187	217508	149424	1057285
10000	x_2	61248	385783	99331	521291	446334	3081533
10000	x_3	29703	149950	111908	587876	241250	1466562
10000	x_4	60325	386954	94078	592519	102701	586236
10000	x_5	11159	38646	14865	49971	28606	133867
10000	x_6	9873	57261	20338	104094	51190	307639
10000	x_7	10121	59001	20326	104018	51133	307162
P_4							
10000	x_0	26	263	8567	106957	12677	163074
10000	x_1	26	272	14175	204841	19577	256721
10000	x_2	23	219	421	5346	5072	58466
10000	x_3	146	1471	5282	60829	7269	83804
10000	x_4	21	185	3599	35949	4134	41272
10000	x_5	1365	14992	257	1809	2328	20899
10000	x_6	536	4801	6679	73228	4428	49192
10000	x_7	539	4828	7036	77151	4525	50314
P_5							
5000	x_0	88	973	62668	609170	228454	2427634
5000	x_1	67	675	61618	597442	225811	2396330
5000	x_2	88	973	62578	608175	228235	2425065
5000	x_3	86	933	62398	606160	227774	2419561
5000	x_4	92	1041	62491	607212	228010	2422394
5000	x_5	91	1024	62497	607267	228023	2422517
5000	x_6	91	1024	62516	607479	228068	2423049
5000	x_7	88	973	62549	607843	228167	2424258
P_6							
50000	x_0	16	91	376	1916	659	4044
50000	x_1	16	115	378	1944	2560	17934
50000	x_2	14	87	40	169	181	1010
50000	x_3	16	91	117	510	659	4044
50000	x_4	15	101	117	510	182	1023
50000	x_5	15	101	117	510	182	1023
50000	x_6	14	87	117	510	181	1010
50000	x_7	14	87	117	510	181	1010
P_7							
50000	x_0	10	41	350	1755	606	3643
50000	x_1	12	63	401	2064	684	4145
50000	x_2	31	65	62	126	62	189
50000	x_3	22	164	26	142	97	538
50000	x_4	49	100	99	200	25	52
50000	x_5	2584	5170	5168	10338	1292	2586
50000	x_6	55	112	110	222	110	333
50000	x_7	55	112	110	222	110	333

Table 2: Numerical results (CPU time)

		CPU time		
P/Dim.	SP	New	PRP	BFGS
P_1				
50000	x_0	0.5148	4.9296	40.3730
50000	x_1	0.5148	4.9608	41.4182
50000	x_2	0.2652	0.7332	2.5584
50000	x_3	0.3120	0.7020	2.5896
50000	x_4	0.2652	0.4368	0.2652
50000	x_5	0.1716	0.3744	0.5460
50000	x_6	0.4212	0.7956	0.7176
50000	x_7	0.3744	0.8424	0.8892
P_2				
50000	x_0	0.5148	4.9452	42.6974
50000	x_1	0.4056	5.1012	44.7410
50000	x_2	0.2964	0.6708	2.8548
50000	x_3	0.3588	0.8736	1.2636
50000	x_4	0.2652	1.2480	0.2808
50000	x_5	0.1560	0.3432	0.5304
50000	x_6	0.3900	0.7956	0.7332
50000	x_7	0.7488	0.9828	0.8580
P_3				
10000	x_0	0.5265	3.6298	1.1398
10000	x_1	0.3429	0.8903	0.4389
10000	x_2	1.4979	2.0869	1.3758
10000	x_3	0.5859	2.3832	0.5998
10000	x_4	1.5049	2.3747	0.2346
10000	x_5	0.1506	0.2027	0.0535
10000	x_6	0.2228	0.4189	0.1233
10000	x_7	0.2290	0.4198	0.1229
P_4				
10000	x_0	0.1716	0.7960	1.0721
10000	x_1	0.1716	1.5104	1.7052
10000	x_2	0.1248	0.0388	0.4040
10000	x_3	1.0140	0.4524	0.5508
10000	x_4	0.1248	0.2664	0.2676
10000	x_5	10.3116	0.0143	0.1396
10000	x_6	3.2292	0.5561	0.3291
10000	x_7	3.3384	0.5779	0.3325
P_5				
5000	x_0	0.3900	2.6088	9.1360
5000	x_1	0.2808	2.5382	8.9972
5000	x_2	0.4056	2.6067	9.0797
5000	x_3	0.3744	2.5844	9.0674
5000	x_4	0.3744	2.5744	9.0594
5000	x_5	0.3744	2.5384	9.0630
5000	x_6	0.3744	2.5518	9.1413
5000	x_7	0.3744	2.5476	9.2811
P_6				
50000	x_0	0.5616	13.3224	0.2694
50000	x_1	0.8580	13.3380	1.1963
50000	x_2	0.5616	1.2012	0.0670
50000	x_3	0.5616	3.6660	0.2751
50000	x_4	0.7020	3.6972	0.0680
50000	x_5	0.6552	3.6660	0.0656
50000	x_6	0.5304	3.5256	0.0641
50000	x_7	0.5772	3.6348	0.0658
P_7				
50000	x_0	0.1560	8.4396	16.9261
50000	x_1	0.2340	9.9060	18.7045
50000	x_2	0.2496	0.5304	0.6708
50000	x_3	0.4836	0.4836	1.7472
50000	x_4	0.2808	0.9204	0.1248
50000	x_5	16.8325	43.9922	8.6424
50000	x_6	0.4368	0.9984	0.9672
50000	x_7	0.3744	1.0608	0.9828

N_i for the three algorithms, Fig. 2 shows the performance for the total of N_f and Fig. 3 shows the performance for the CPU time. The algorithm K solved about 95, 91 and 79% of the test functions, respectively and has least of N_i, N_f and CPU time among the three methods and will reach to 1 faster than the other algorithms. It means that the new algorithm K is the best algorithm closing to the performance index.

CONCLUSION

From the numerical results obtained through the comparison technique presented in the tables above of different problems with different initial points and dimensions, it is easy to conclude that the performance of the proposed algorithm K is the most efficient and effective in terms of N_i, N_f and the CPU time compared with the two famous algorithms. This can improve the behavior of the new algorithm to solve the nonlinear monotone equations which does not require Jacobian information of the nonlinear equations. The algorithm K is able to calculate the best solution of problem (1), also its global convergence has been created without using any merit functions.

REFERENCES

Amini, K., M.A. Shiker and M. Kimiaei, 2016. A line search trust-region algorithm with nonmonotone adaptive radius for a system of nonlinear equations. 4OR, 14: 133-152.

Broyden, C.G., J.E. Dennis and J.J. More, 1973. On the local and superlinear convergence of quasi-Newton methods. IMA. J. Appl. Math., 12: 223-245.

Byrd, R.H., J. Nocedal and J. Y.X Yuan, 1987. Global convergence of a class of quasi-newton methods on Convex problems. SIAM J. Numer. Anal., 24: 1171-1190.

Cheng, W., 2009. A PRP type method for systems of monotone equations. Math. Comput. Modell., 50: 15-20.

Dennis, J.E. and J.J. More, 1977. Quasi-Newton methods, motivation and theory. SIAM Rev., 19: 46-89.

Dolan, E.D. and J.J. More, 2002. Benchmarking optimization software with performance profiles. Math. Program., 91: 201-213.

Griewank, A., 1986. The global convergence of Broyden-like methods with suitable line search. ANZIAM. J., 28: 75-92.

Iusem, N.A. and V.M. Solodov, 1997. Newton-type methods with generalized distances for constrained optimization. Optim., 41: 257-278.

Li, D. and M. Fukushima, 1999. A globally and superlinearly convergent gauss-newton-based BFGS method for symmetric nonlinear equations. SIAM. J. Numer. Anal., 37: 152-172.

Li, D.H. and M. Fukushima, 2000. A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. Optim. Methods Software, 13: 181-201.

Nocedal, J., 1980. Updating quasi-Newton matrices with limited storage. Math. Comput., 35: 773-782.

Ortega, J.M. and W.C. Rheinboldt, 1970. Iterative Solution of Nonlinear Equations in Several Variables. Vol. 30, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA., ISBN-13:978-0-898714-61-6, Pages: 572.

Shiker, M.A. K. and K. Amini, 2018. A new projection-based algorithm for solving a large-scale nonlinear system of monotone equations. Croatian Oper. Res. Rev., 9: 63-73.

Shiker, M.A.K. and Z. Sahib, 2018. A modified technique for solving unconstrained optimization. J. Eng. Applied Sci., 13: 9667-9671.

Solodov, M.V. and B.F. Svaiter, 1998. A Globally Convergent Inexact Newton Method for Systems of Monotone Equations. In: Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Fukushima, M. and L. Qi (Eds.). Springer, Boston, Massachusetts, USA., ISBN: 978-1-4419-4805-2, pp: 355-369.

Zeidler, E., 2013. Nonlinear Functional Analysis and its Applications: III: Variational Methods and Optimization. Springer, Berlin, Germany, ISBN:978-1-4612-9529-7, Pages: 651.

Zhou, W.J. and D.H. Li, 2008. A globally convergent BFGS method for nonlinear monotone equations without any merit functions. Math. Comput., 77: 2231-2240.