Journal of Interdisciplinary Mathematics
ISSN: 0972-0502 (Print), ISSN: 2169-012X (Online)
Vol. 27 (2024), No. 4, pp. 655–665
DOI : 10.47974/JIM-1737

A modified technique of spectral gradient projection method for solving nonlinear equations systems

Doaa Hamzah Allawi *
Mushtak A. K. Shiker †
Department of Mathematics
College of Education for Pure Sciences
University of Babylon
Babylon
Iraq

Abstract

This study suggests a new line search algorithm modified of Spectral gradient projection method (SGPM) called (D.A), combining the modified spectral gradient method and projection method for solving nonlinear equations systems. This approach can be more efficient because it requires less CPU time, less function evaluation, and fewer iterations. Furthermore, if this equation is Lipschitz continuous (LC) and monotone, we proved that this technique converges globally. The numerical results show how effective and beneficial this method is for solving nonlinear equations systems.

Subject Classification: *34G20, 34A34.*
Keywords: *Line search, Spectral gradient method, Monotone, Lipschitz continuous, Projection method.*

1. Introduction

The terms "linear system" and "nonlinear system" refer to two types of system equations. Numerous scientific disciplines can be built as nonlinear equations systems, including economics, engineering, and others. On the other hand, the advancement of science in these areas depends more on the solution of these systems [1].

The study of the gradient projection method and its associated problems has become very popular through the many literature studies presented by researchers related to optimization problems [2- 5].

This article has been corrected with minor changes. These changes do not impact the academic content of the article.

* *E-mail:* **doaahamzah44@gmail.com (Corresponding Author)**
† *E-mail:* **mmttmmhh@yahoo.com**

We go over how to resolve these systems of nonlinear equations in this paper. Numerous researchers have focused on developing effective techniques to solve nonlinear systems, leading to the suggestion that certain well-known algorithms can do so. These include, but are not limited to, optimization problems.

The nonlinear equations system is generally regarded as:

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

The function $F: R^n \to R^n$ is continuous. The following requirement for monotonicity in $F(x)$ must be met for nonlinear equations to be considered monotone.

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

A fixed point map or a natural map can convert certain monotone variational inequality to monotonicity, but the underlying function must first meet several requirements. We shall solve these problems monotonically because of the slow performance and convergence [6, 7]. Most often, monotone equations systems can be found in many problems, such as variations of the popular proximal algorithms and other problems [8– 10].

Additionally, several important methods exist for solving systems with nonlinear equations, including the line search technique, the trust region technique, and others [11, 12]. Newton's plan was at the fore and at the heart of several significant algorithms developed to solve nonlinear equation systems and optimization problems.

For large-scale unconstrained optimization problems, in 1988 [13], the spectral gradient approach is particularly effective and simple to use. In 1986 [14], the non-monotone method of Grippo, Lampariello, and Lucidi is modified to ensure global convergence. In 2003 [15], William and Raydan improved the spectral gradient technique for solving large-scale nonlinear equation systems. In 2006 [16], William, Martnez, and Raydan presented (DF- SANE) using a fully derivative-free SANE algorithm. A class of nonlinear equation systems is successfully handled by DF-SANE, according to numerical tests.

To solve nonlinear monotone equations, Zhang and Zhou [17] merged the projection approach with the spectral gradient method. If the nonlinear equations are Lipschitz continuous, then the approach is demonstrated to be globally convergent.

A lot of research has been done in optimization and operations research to discover the algorithms and techniques that give the best solutions to systems and problems related to optimization problems [18].

The new algorithm combining modified spectral gradient and projection methods is used to solve nonlinear monotone equations systems. We'll prove it's a globally convergent problem without employing merit function with well-defined solutions (1). It will be more competent by comparing the new algorithm presented with the SGPM approach in [19] and the BFGS approach in [20].

Now, we will formally state the new algorithm :

2. The Algorithm (D.A)

Select any initial point $x_0 \in R^n$ and $\in \left(0, \frac{1}{4}\right]$, $\mu \in (0,1)$, $\sigma \in (0,1)$, $\theta \in (0,1), m > 0$.

Let $k := 0$.

Step 1: Determine the search direction d_k Using:

$$d_k = \begin{cases} -F(x_k) & if\ k = 0, \\ -\sigma_k F(x_k) & otherwise, \end{cases} \tag{2}$$

Where $\sigma_k = \frac{r_k^T r_k}{y_k^T r_k}$, $r_k = x_{k+1} - x_k$, with $y_k = F(x_{k+1}) - F(x_k) + \mathrm{s} r_k$

Where $\mathrm{s} = \frac{1}{\| d_k \|}$

If $d_k = 0$, Stop.

Step 2: Select step length $\alpha_k = \mu\, \beta^{h_k}$ s. t. $\boldsymbol{h_k}$ is the most minor non-negative integer h satisfying :

$$-\langle F(x_k + \alpha_k d_k), d_k \rangle \geq \sigma \theta_k \alpha_k \|d_k\|^2 \tag{3}$$

Where $\theta_k = \frac{1}{\theta + \| d_k \|^2}$, find trail point $z_k = x_k + \alpha_k d_k$

Step 3: Compute

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

Step 4: Set $k := k + 1$, then move on to Step 1.

Remark: (i) Suppose that F is a monotone, in **step 1** we have:

$$\begin{aligned} y_k^T r_k &= (F(x_{k+1}) - F(x_k) + \mathrm{s} r_k)^T r_k \\ &= \left(F(x_{k+1}) - F(x_k)\right)^T r_k + \mathrm{s} r_k^T r_k \\ &= \langle F(x_{k+1}) - F(x_k), r_k \rangle + \mathrm{s} r_k^T r_k \geq \mathrm{s} r_k^T r_k > 0 \end{aligned} \tag{5}$$

So, if F is LC, ∃ a positive constant L > 0 s.t.

$$\|F(x) - F(y)\| \geq L\|x - y\| , \forall\, x, y \in R^n. \tag{6}$$

we have:

$$\begin{aligned} y_k^T r_k &= (F(x_{k+1}) - F(x_k) + \mathrm{s} r_k)^T r_k \\ &= \left(F(x_{k+1}) - F(x_k)\right)^T r_k + \mathrm{s} r_k^T r_k \\ &= \langle F(x_{k+1}) - F(x_k), r_k \rangle + \mathrm{s} r_k^T r_k \leq (L + \mathrm{s}) r_k^T r_k \end{aligned} \tag{7}$$

it is clear that $L + s > s$ so we have from (2), (5) and (6) :

$$\frac{\|F(x_k)\|}{L+s} \leq \|d_k\| \leq \frac{\|F(x_k)\|}{s} \tag{8}$$

Now, we'll show how line search (3) is well-defined.

Assume that line search (3) is not achieved, $\boldsymbol{for\ some\ iteration\ index}$ $\boldsymbol{k}$ **and** $\boldsymbol{for\ any\ non-negative\ integer\ h}$, if we taking $\lim_{h\to\infty}$ for two sides of (3) .

$$-\lim_{h\to\infty}\langle F(x_k+\ \mu\,\beta^h d_k),d_k\rangle < \lim_{h\to\infty}\sigma\theta_k\mu\,\beta^h\,\|d_k\|^2$$
$$-\langle F(x_k),d_k\rangle < 0$$

Since $F_k^T\,d_k < 0$, there is a contradiction; as a result, $-F_k^T d_k > 0$.
Hence, the line search (3) is well-defined.

3. Convergence property

We shall employ the next lemma to get the new algorithm's global convergence.

Lemma 1: [20] *Let F be monotone ,* $x,y\ \in R^n$ *achieve* $\langle F(y),x-y\rangle > 0$.

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

for any $\bar{x}\ \in R^n$ s.t. $F(\bar{x})=0$, it holds that

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

We can now determine the convergence result for our Algorithm (D.A) by the next analogous theorem [19] .

Theorem 1: *Let* $\{x_k\}$ *be any sequence created by the Algorithm (* ***D.A*** *), assuming that F is monotone and (LC), as well as solving a set of (1) is non-empty.*

Then, any $\bar{x}$ achieving $(\bar{x})=0$, we obtain

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

Especially, the sequence $\{x_k\}$ is bounded. Additionally, it must be achieved then either $\{x_k\}$ is finite a solution is produced by the final iteration, or $\{x_k\}$ is infinite and

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

As well, $\{x_k\}$ converges to some $\bar{x}$ s.t. $F(\bar{x})=0.$

Proof: To begin, assuming that there is an end to the algorithm at iteration k, hence.

$d_k =\ 0$, we obtain $F(\mathrm{x_k})=0$ from (8), then x_k is a solution of (1)**.**

Now, assume that. $d_k\ \neq\ 0\ \ \forall k$, then an infinite sequence, $\{x_k\}$ is generated. From (3) :

$$\langle F(z_k),x_k-z_k\rangle = \langle F(z_k),x_k-(x_k+\alpha_k d_k)\rangle$$

$since\ z_k = x_k +\ \alpha_k d_k$

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

$$= -\alpha_k \langle F(z_k), d_k \rangle$$
$$= -\alpha_k \langle F(x_k + \alpha_k d_k), d_k \rangle$$
$$\geq \sigma\, \theta_k {\alpha_k}^2 \|d_k\|^2 > 0$$

then, $\langle F(z_k), x_k - z_k \rangle = -\alpha_k \langle F(z_k), d_k \rangle \geq \sigma \theta_k {\alpha_k}^2 \|d_k\|^2 > 0.$ (9)

Let $\bar{x}$ be any solution of $F(x) = 0$ and $F(\bar{x}) = 0$. From lemma 1, (4) and (9)

we have $\|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - x_k\|^2.$ (10)

Especially, $\{\|x_k - \bar{x}\|\}$ is non-increasing and convergent.

Consequently, $\{x_k\}$ will have a bound, we obtain

$$lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{11}$$

By (8) It is obvious that $\{d_k\}$ holds to be bounded, and so is $\{z_k\}$.

Since F is continuous, a positive constant $v > 0$ must exist s.t.

$$\|F(z_k)\| \leq v$$

From (4) and (9) we get :

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

Hence, we have $\|x_{k+1} - x_k\| \geq \frac{\sigma}{v} {\alpha_k}^2 \|d_k\|^2.$ (12)

From (11) and (12) we have

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0\,, \tag{13}$$

From (8), we get

$$\liminf_{k\to\infty} \|F(x_k)\| = 0, when \liminf_{k\to\infty} \|d_k\| = 0.$$

Since $\{x_k\}$ is bounded, and F is continuous, $\{x_k\}$ has an accumulation point at which $\hat{x}\; s.t.\; F(\hat{x}) = 0$. We get $\{\|x_k - \hat{x}\|\}$ converges from equation (8). As a result, $\{x_k\}$ converges to $\hat{x}$.

$$\liminf_{k\to\infty} \|F(x_k)\| > 0, when \liminf_{k\to\infty} \|d_k\| > 0.$$

By (13) we get

$$\lim_{k\to\infty} \alpha_k = 0 \tag{14}$$

equation (3) gives us

$$-\langle F(x_k + \mu\beta^{h_{k-1}} d_k), d_k \rangle < \sigma \theta_k \mu \beta^{h_{k-1}} \|d_k\|^2 \tag{15}$$

Since $\{x_k\}$, $\{d_k\}$ are bounded, to select a subsidiary sequence, let $k \to \infty$ in (15), we get

$$-\langle F(\hat{x}), \hat{d} \rangle \leq 0 \tag{16}$$

s.t. $\hat{x}$ and $\hat{d}$ are subsidiary sequences limiting that chosen. Moreover, by (8) and already known argument,

$$-\langle F(\hat{x}), \hat{d}\rangle > 0 \tag{17}$$

(16) and (17) are a contradiction. Hence it is impossible to get that

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

This proof is finished.

4. Numerical Results

We will compare the effectiveness of the new approach (**D.A**) presented earlier in this paper with the following algorithms.

SGPM : From [19].

BFGS : From [20-22], with the direction used in those papers.

All the codes for the tests were written in $MATLAB$ version $R2017a$ on a computer with 8 GB of RAM and CPU 2.40 GHZ. The codes aim to compare the outcomes of the new algorithm (**D.A**) with the algorithm mentioned above. All algorithms are completed when $\|F_k\| \leq 10^{-8}$ or $\|F(z_k)\| \leq 10^{-8}$ or when 500000 iterations have been performed overall. For our method, the parameters are determined as follows: $\mu = 0.2,\ \beta = 0.25,\ \theta = 0.3,\ \sigma = 0.2$. For **SGPM** method, we set $\beta = 0.4, \sigma = 0.01, r = 0.001$. For **BFGS** method, we set $\beta = 0.3, \sigma = 0.3$.

Three sides: The number of iterations, the number of functions evaluations, and CPU time are used to compare various algorithms. The specific dimensions for these algorithm comparisons are set at $10000 \_ 50000$, for the following starting points:

$$x_0 = (10,10,\dots,10)^T,$$

$$x_1 = (-10,-10,\dots,-10)^T,$$

$$x_2 = (1,1,\dots,1)^T,$$

$$x_3 = (-1,-1,\dots,-1)^T,$$

$$x_4 = \left(1,\tfrac{1}{2},\tfrac{2}{3},\dots,\tfrac{1}{n}\right)^T,$$

$$x_5 = (0.1,0.1,\dots,0.1)^T,$$

$$x_6 = \left(\tfrac{1}{n},\tfrac{2}{n},\dots,1\right)^T,$$

$$x_7 = (1-\tfrac{1}{n}, 1-\tfrac{2}{n},\dots,0)^T.$$

Numerical results are viewed in schedules (1, 2); the first schedule consists of N_i (number of iterations), N_f (number of functions evaluations) of these algorithms as shown in Table 1, and the second schedule consists of CPU times of these algorithms as listed in Table 2.

Table 1
Numerical results(N_i, N_f)

$P.$	$Dim.$	$S.P$	$D.A$		$SGPM$		$BFGS$	
			Ni	Nf	Ni	Nf	Ni	Nf
$\boldsymbol{P_1}$	50000	x_0	48	249	108	1169	102	1099
	50000	x_1	48	249	108	1169	102	1099
	50000	x_2	46	202	108	1169	124	1321
	50000	x_3	46	202	108	1169	124	1321
	50000	x_4	132	133	110	1191	140	150
	50000	x_5	36	53	108	1169	152	1576
	50000	x_6	46	60	108	1169	162	881
	50000	x_7	40	53	108	1169	160	879
$\boldsymbol{P_2}$	50000	x_0	48	249	108	1169	120	1297
	50000	x_1	72	664	108	1169	118	1275
	50000	x_2	46	202	108	1169	98	1038
	50000	x_3	44	212	108	1189	122	1310
	50000	x_4	132	133	110	1191	76	77
	50000	x_5	36	53	108	1169	32	244
	50000	x_6	150	182	108	1169	166	962
	50000	x_7	146	178	108	1169	164	960
$\boldsymbol{P_3}$	10000	x_0	2660	2662	64378	64400	2678	2859
	10000	x_1	4950	4952	71498	71540	5100	6777
	10000	x_2	766	767	52476	52498	738	751
	10000	x_3	858	859	62064	62086	828	841
	10000	x_4	134	135	226	247	186	187
	10000	x_5	180	181	51326	51348	138	139
	10000	x_6	546	547	22544	22566	536	539
	10000	x_7	546	547	22540	22562	536	539
$\boldsymbol{P_4}$	10000	x_0	46	867	96	1057	340	2724
	10000	x_1	68	661	96	1057	1162	10942
	10000	x_2	52	852	96	1057	114	714
	10000	x_3	170	936	96	1057	170	1102
	10000	x_4	48	852	96	1057	58	170
	10000	x_5	46	852	96	1057	46	101
	10000	x_6	56	852	96	1057	68	292
	10000	x_7	56	837	96	1057	72	299

Contd...

$P.$	$Dim.$	$S.P$	$D.A$		$SGPM$		$BFGS$	
			Ni	Nf	Ni	Nf	Ni	Nf
P_5	50000	x_0	66	444	120	1301	122	1314
	50000	x_1	50	273	118	1279	114	1229
	50000	x_2	24	173	120	1301	390	4253
	50000	x_3	52	328	120	1301	120	1289
	50000	x_4	418	4351	120	1301	126	1352
	50000	x_5	52	309	120	1301	124	1330
	50000	x_6	70	559	120	1301	292	3177
	50000	x_7	48	259	120	1301	328	3573
P_6	50000	x_0	24	47	118	1279	114	1229
	50000	x_1	44	387	118	1279	108	1163
	50000	x_2	28	49	120	1301	122	1273
	50000	x_3	36	119	120	1301	110	1174
	50000	x_4	40	41	122	1323	40	41
	50000	x_5	22	23	64	706	22	23
	50000	x_6	112	113	120	1301	160	422
	50000	x_7	112	113	120	1301	160	422

Table 2
Numerical Results (CPU time)

$P.$	$Dim.$	$S.P$	$CPU\ time$		
			$D.A$	$SGPM$	$BFGS$
P_1	50000	x_0	1.187500	15.375000	15.656250
	50000	x_1	1.000000	14.546875	13.968750
	50000	x_2	0.718750	15.531250	22.671875
	50000	x_3	0.656250	15.656250	23.234375
	50000	x_4	0.390625	16.671875	0.250000
	50000	x_5	0.109375	15.406250	27.171875
	50000	x_6	0.187500	16.234375	2.468750
	50000	x_7	0.171875	16.546875	2.531250
P_2	50000	x_0	1.031250	11.500000	18.734375
	50000	x_1	3.609375	15.265625	20.968750
	50000	x_2	0.625000	22.171875	24.812500
	50000	x_3	0.609375	19.796875	17.000000
	50000	x_4	0.390625	11.359375	0.250000
	50000	x_5	0.093750	9.312500	0.593750
	50000	x_6	0.390625	10.078125	2.265625
	50000	x_7	0.500000	10.125000	2.218750

Contd...

P_3	10000	x_0	6.906250	2.101875	7.500000
	10000	x_1	12.890625	2.341562	17.500000
	10000	x_2	1.890620	1.666875	1.890625
	10000	x_3	2.171875	1.975000	2.328125
	10000	x_4	0.406250	0.006562	0.171875
	10000	x_5	0.468750	1.511718	0.296875
	10000	x_6	1.437500	0.719531	1.500000
	10000	x_7	1.343750	0.714218	1.359375
P_4	10000	x_0	0.031250	0.546875	1.078125
	10000	x_1	0.062500	0.484375	4.078125
	10000	x_2	0.031250	0.500000	0.250000
	10000	x_3	0.109375	0.781250	0.343750
	10000	x_4	0.031250	0.687500	0.062500
	10000	x_5	0.031250	0.687500	0.031250
	10000	x_6	0.062500	0.671875	0.093750
	10000	x_7	0.062500	0.593750	0.093750
P_5	50000	x_0	1.515625	28.296875	18.515625
	50000	x_1	0.718750	24.843750	15.500000
	50000	x_2	0.406250	13.921875	56.250000
	50000	x_3	0.812500	10.328125	16.812500
	50000	x_4	18.359375	11.531250	18.265625
	50000	x_5	0.906250	10.703125	17.343750
	50000	x_6	1.781250	12.421875	30.328125
	50000	x_7	0.671875	11.906250	31.250000
P_6	50000	x_0	0.140625	10.562500	15.859375
	50000	x_1	1.875000	9.437500	13.312500
	50000	x_2	0.109375	10.296875	14.218750
	50000	x_3	0.234375	10.687500	13.562500
	50000	x_4	0.140625	2.000000	0.140625
	50000	x_5	0.078125	1.109375	0.078125
	50000	x_6	0.281250	10.265625	0.703125
	50000	x_7	0.296875	12.062500	0.671875

5. Conclusion

This work presented a new algorithm that employed the line search combining the modified spectral gradient and projection methods for solving nonlinear monotone equations systems. An important property of the presented technique is that it is truly globally convergent without using a merit function. By the numerical results offered in the schedules above, we concluded that the performance of the proposed algorithm (**D.A**) is the most efficient and effective than the other algorithms **BFGS** and **SGPM** from where: N_i, N_f, and the CPU time means that the new technique is promising for solving nonlinear systems of monotone equations.

References

[1] H. Masoud, et al., "An Adaptive Nonmonotone Line Search Technique for Solving Systems of Nonlinear Equations." *Journal of Mathematics* (2021).

[2] H. H. Dwail and M. A. K. Shiker, ''Using trust region method with BFGS technique for solving nonlinear systems of equations,'' *J. Phys.: Conf. Ser.* 1818 012022 (2021).

[3] L. H. Hashim, et al., ''An application comparison of two Poisson models on zero count data,'' *J. Phys.: Conf. Ser.* 1818 012165 (2021).

[4] L. H. Hashim, et al., ''An application comparison of two negative binomial models on rainfall count data,'' *J. Phys.: Conf. Ser.* 1818 012100 (2021).

[5] K. H. Hashim, et al., "A new line search method to solve the nonlinear systems of monotone equations." *Journal of Engineering and Applied Sciences* 14, 10080-10086 (2019).

[6] A. Bouaricha and R.B. Schnabel, "Tensor methods for large sparse systems of nonlinear equations." *Math. Program*. 82, 377–400 (1998).

[7] D.W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math*. 11, 431–441 (1963).

[8] Y.B. Zhao and D. Li, "Monotonicity of fixed point and normal mappings associated with variational inequality and its application," *SIAM J. Optim.* 11, 962–973 (2001).

[9] C. T. Kelley, "Iterative methods for linear and nonlinear equations", Society for Industrial and Applied Mathematics, Philadelphia (1995).

[10] N. K. Dreeb, et al., ''Using a New Projection Approach to Find the Optimal Solution for Nonlinear Systems of Monotone Equation,'' *J. Phys.: Conf. Ser.* 1818 012101 (2021).

[11] M. M. Mahdi, H. H. Dwail, and M. A. K. Shiker, ''Hybrid spectral algorithm under a convex constrained to solve nonlinear equations,'' *Journal of Interdisciplinary Mathematics*, 25 (5), pp. 1333– 1340 (2022).

[12] H. H. Dwail et al., "A new modified TR algorithm with adaptive radius to solve a nonlinear systems of equations," *J. Phys.: Conf. Ser.* 1804 012108 (2021).

[13] B. Jonathan and J. M. Borwein, "Two-point step size gradient methods." *IMA Journal of Numerical Analysis* 8.1, 141-148 (1988).

[14] L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for Newton's method." *SIAM journal on Numerical Analysis* 23.4, 707-716 (1986).

[15] L. C. William, and M. Raydan, "Nonmonotone spectral methods for large-scale nonlinear systems." *Optimization Methods and software* 18.5, 583-599 (2003).

[16] L. C. William, J. Martinez, and M. Raydan, "Spectral residual method without gradient information for solving large-scale nonlinear systems of equations." *Mathematics of computation* 75.255, 1429-1448 (2006).

[17] L. Zhang and W. Zhou, "Spectral gradient projection method for solving nonlinear monotone equations." *Journal of Computational and Applied Mathematics* 196.2 ,478-484 (2006).

[18] K. H. Hashim, et al., ''Solving the Nonlinear Monotone Equations by Using a New Line Search Technique,'' *J. Phys.: Conf. Ser*. 1818 012099 (2021).

[19] M. V. Solodov, and B. F. Svaiter., "A globally convergent inexact Newton method for systems of monotone equations." Reformulation: Non smooth, piecewise smooth, semi smooth and smoothing methods: 355-369 (1999).

[20] Z. W. Jun and D. H. Li., "A globally convergent BFGS method for nonlinear monotone equations without any merit functions." *Mathematics of Computation* 77.264, 2231-224 (2008).

[21] Shelash, Hader Baqer , Ahmad, Hasanain Hamid & Obaid, Ahmed J. On characterizations some of subgroups of the dihedral group D2n, *Journal of Discrete Mathematical Sciences and Cryptography*, 1-6, DOI: 10.47974/JDMSC-1553.

[22] Shubham Sharma & Ahmed J. Obaid. Mathematical modelling, analysis and design of fuzzy logic controller for the control of ventilation systems using MATLAB fuzzy logic toolbox, *Journal of Interdisciplinary Mathematics*, 23:4, 843-849 (2020), DOI: 10.1080/09720502.2020.1727611.

Received September, 2023