

## Comparison Between Steepest Descent Method and Conjugate Gradient Method by Using Matlab

Dana Taha Mohammed Salih<sup>1,\*</sup>, Bawar Mohammed Faraj<sup>2</sup>

<sup>1</sup> Department of Physics, College of Science, University of Halabja, Halabja, 46018, Iraq

<sup>2</sup> Computer Science Department, College of Science, University of Halabja, Halabja, 46018, Iraq

\*Corresponding Author: Dana Taha Mohammed Salih, E-mail: dana.taha@uoh.edu.iq

### Article Info

Article History

Received Jul 8, 2021

Revised Aug 23, 2021

Accepted Aug 26, 2021

### Keywords

Conjugate gradient method

Steepest descent method

Comparison

Matlab

Optimization algorithms

### Abstract

The Steepest descent method and the Conjugate gradient method to minimize nonlinear functions have been studied in this work. Algorithms are presented and implemented in Matlab software for both methods. However, a comparison has been made between the Steepest descent method and the Conjugate gradient method. The obtained results in Matlab software has time and efficiency aspects. It is shown that the Conjugate gradient method needs fewer iterations and has more efficiency than the Steepest descent method. On the other hand, the Steepest descent method converges a function in less time than the Conjugate gradient method.

**Copyright:** © 2021 EngiScience Publisher All rights reserved

## 1. Introduction

Optimization presents an essential tool in decision theory and analysis of physical systems. Optimization can be defined as the process of finding the best solution to a problem in a certain sense and under certain conditions [1]. Optimization theory is a very developed area with its wide applications in science, engineering, business management, military and space technology, location science, statistics, portfolio analysis, and machine computations [1,2,3,4,5,6,7]. Researchers have studied algorithms for minimizing continuously differentiable functions [1-10].

In particular, the Steepest descent algorithms have received much attention because of their simplicity and success in many applications. [8]. The Steepest descent method (also known as Cauchy's or gradient method) is one of the earliest and most fundamental methods of unconstrained scalar optimization. [8,9,10]. The technique is fundamental from a theoretical viewpoint, by using a simple optimization algorithm, the method can find the local minimum of a function. Significant improvements in the functional performance of Steepest descent methods have been made over the last few decades merely by developing innovative approaches for selecting the step size in each iteration [8]. Recently, different works on the Steepest descent method have been done [8,11,12,13,14]. Fliege and Svaiter[15] presented a Pareto descent method for multi-objective optimizations. Drummond and Svaiter present a Cauchy-like method to solve smooth

unconstrained vector optimization problems in [9]. They have proved the global convergence for the weak unconstrained minimizer. Fliege and Svaiter have presented a Steepest descent technique for unconstrained multicriteria optimization and a method called "feasible descent direction" for restricted multicriteria optimization [16].

In the last decades, the interest in Gradient methods has been renewed after the innovative approach of Borwein and Barzilai [17], which stimulated novel choices for  $\alpha_k$  and proved to be largely superior to the Cauchy step length, they have derived two-point step sizes to a Steepest descent method by approximating the secant equations [17]. Fletcher proposed a technique for quadratic and nonquadratic objective functions that takes advantage of the availability of a few more "long" vectors of storage to produce large performance improvements. [11].

The Conjugate gradient (CG) method is a method between the Newtons and Steepest descent method [18]. The Conjugate gradient method diverts the Steepest descent method's direction by multiplying it by a positive multiple of the method's direction. [18]. CG method is beneficial for minimizing functions of very many variables because it does not require the storage of any matrices. However, the algorithm's convergence rate is linear unless the iterative procedure is "restarted" occasionally [19]. Fletcher and Reeves were the first to apply the Conjugate gradient algorithm to the general unconstrained minimization problem [20]. Fletcher and Reeves described a quadratically convergent gradient method for locating an unconstrained local minimum of a function of several variables. As a machine method, the Conjugate gradient method outperforms elimination. Because it gives a solution in  $n$  steps, it is more straightforward in coding and requires less storage space [21,22,23]. Stanimirovic et al. investigated classify and compare main Quasi-Newton and Conjugate gradient methods to present a global overview of scientific advances in this field [24]. Lasdon et al. are extended the Conjugate gradient minimization method of Fletcher and Reeves to optimal control problems. They have presented the Conjugate gradient method and compared the method with the method of Steepest descent. The convergence of the proposed method is much more rapid in all cases [25].

Optimization problems are extremely important when developing technological processes and projecting technical objects, as well as during their implementation and operation [26]. Computer algorithms are effective processing methods. It is necessary to improve their efficiency and decrease their complexity in all implementation without loss of efficiency [27]. Nonlinear programming in Matlab was used to solve optimization problems. [26,27,28,29,30,31].

Additionally, several fundamental algorithms are presented and analyzed to provide context for this critical component of optimization. MATLAB programs demonstrate many topics, and ideally, the interested reader will find satisfaction in the ability of actually solving problems [28]. The reported case study

results from ongoing research investigating the effect of technology and Computer software on solving optimization techniques. Rahim and Hussan are compared Matlab with Mathematica for optimization problems. They observed, that Matlab is faster for numeric calculations[32]. Besides computational efficiency, another advantage of Matlab software is that their implementations for large-scale problems are considerably more straightforward than those for the other algorithms. This is particularly true in a high-level programming environment such as MATLAB, which supports high-level sparse-matrix operations [33,34]. Zhang used MATLAB software for solving Linear programs by the interior point method [33]. MATLAB software was used to compare the Steepest descent method and the Conjugate gradient method in the presented work.

## 2. The Method of Steepest Descent

Let  $f$  have continuous first partial derivatives on  $E^n$ , we will frequently need the gradient vector of  $f$  and therefore, we introduce some simplifying notation. The gradient  $\nabla f(X)$  is, according to our conventions, defined as an n-dimensional row vector. For convenience, we define the n-dimensional column vector  $g(X_k) = \nabla f(X_k)^T$ . When there is no chance for ambiguity, we sometimes suppress the argument  $x$  and, for example, write  $g_k$  for  $g(X_k) = \nabla(f(X_k))^T$  [35,36].

The Steepest descent method is defined by the following iterative algorithm

$$X_{k+1} = X_k - a_k g_k,$$

where  $a_k$  is a nonnegative scalar minimizing  $f(X_k - a g_k)$ . In other words, from the point  $X_k$  we search along the direction of the negative gradient  $-g_k$  to a minimum point on this line; this minimum point is taken to be  $X_{k+1}$ .

---

### Algorithm 1. Steepest Descent Method

---

**Step 1:** Given an initial vector  $X_0$ ,  $g_0 = \nabla f(X_0)$  and a convergence tolerance  $e$

**Step 2:** Set  $Q_k = H(f(X_k))$ , and  $a_k = \frac{g_k^T g_k}{g_k^T Q_k g_k} \quad \forall k \in N$

**Step 3:** Set  $X_{k+1} = X_k - a_k g_k$  and  $g_{k+1} = \nabla(f(X_{k+1}))$ .

**Step 4:** If  $\|g_{k+1}\| < e$ , then stop,  $x_{k+1}$  is minimum point and  $f(x_{k+1})$  is a minimum value, else return to step 2.

---

**Example 1 .** Let  $f(x, y) = x^2 + y^2 + xy - 3x$ , then find minimum point by the Steepest descent method  $X_0 = (0,0)^T$  and  $e = 0.01$

**Solution:** By applying Algorithm 1, the gradient is  $g_0 = \nabla f(X_0) = \begin{bmatrix} 2x + y - 3 \\ 2y + x \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$

The hessian matrix is

$$Q_0 = H(f(X_0)) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

then we obtain

$$a_0 = \frac{g_0^T g_0}{g_0^T Q_0 g_0} = \frac{[-3, 0] \begin{bmatrix} -3 \\ 0 \end{bmatrix}}{[-3, 0] \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -3 \\ 0 \end{bmatrix}} = \frac{1}{2}$$

Now,

$$X_1 = X_0 - a_0 g_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} -3 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ 0 \end{bmatrix},$$

then

$$g_1 = \nabla f(X_1) = \begin{bmatrix} 3 - 3 \\ 3 \\ \frac{3}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ \frac{3}{2} \end{bmatrix},$$

Hence,  $\|g_1\| = \sqrt{0^2 + \left(\frac{3}{2}\right)^2} = \frac{3}{2} \ll e$ , by repeating the same process, one can obtain a solution in 9<sup>th</sup> iteration, since  $g_9 = \left[0, \frac{3}{512}\right]^T$ , and  $\|g_9\| < \epsilon$ . The results are obtained in the Matlab program. The Steepest descent method minimized this function in 9<sup>th</sup> iterations with an elapsed time of 0.074341 seconds. Results are presented in table 2.

### 3. Conjugate Gradient Method

The Conjugate Gradient(CG) method is a fundamental algorithmic method for solving symmetric, positive definite systems of linear equations and minimizing unconstrained nonlinear functions. The Conjugate gradient method is based on a simple procedure called the *CG-standard* for minimizing a strictly convex quadratic function [37,38]. For solving the nonlinear unconstrained optimization problem

$$\text{Min } f(x),$$

Where  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuously differentiable function, and any nonlinear Conjugate gradient algorithm generates the sequence  $\{x_k\}$  of the form

$$X_{k+1} = X_k + a_k d_k$$

where  $a_k$  is the stepsize obtained by line search and  $d_k$  is the search direction computed by

$$d_{k+1} = -g_{k+1} + \beta_k d_k$$

For  $k \geq 0$ , where  $\beta_k$  is the Conjugate gradient parameter and  $g_{k+1} = \nabla(f(X_{k+1}))$ . In Conjugate gradient methods,  $d_0 = -g_0$ . Nonlinear Conjugate gradient methods have a very nice theory, with a lot of important results on their convergence.

The linear Conjugate gradient algorithm is dedicated to minimizing convex quadratic functions. Hestenes introduced this algorithm. Let us consider the quadratic function

$$f(x) = \frac{1}{2} x^T Q x - x^T b \quad (1)$$

Where  $Q$  is a positive definite symmetric  $n \times n$  matrix and  $b \in \mathbb{R}^n$  is a known vector

$$\nabla f(x) = Qx - b, \quad \text{and} \quad \nabla^2 f(x) = Q \quad (2)$$

Observe that the Hessian of function (1) is independent of  $x$ . Since the Hessian  $Q$  is symmetric and positive definite, from the optimality conditions for a minimum of a differentiable function, it follows that there is a unique minimizer  $x^*$  of (1). From (2), observe that  $x^*$  is the solution of the linear system  $Qx = b$ .

**Theorem(Conjugate direction theorem)** Let  $\{d_i\}_{i=0}^{n-1}$  be a set of nonzero  $Q$ -orthogonal vectors.

For any  $x_0 \in E^n$  the sequence  $\{x_k\}$  generated according to

$$X_{k+1} = X_k + a_k d_k, \quad k \geq 0$$

with

$$a_k = -\frac{g_k^T d_k}{d_k^T Q d_k},$$

and

$$g_k = QX_k - b,$$

Converges to the unique solution,  $x^*$ , of  $Qx = b$  after  $n$  step, that is,  $X_n = X^*$ .

The algorithm of the Conjugate gradient method can be obtained as follow:

---

**Algorithm 2.** Conjugate gradient Method

---

**Step 1:** Given an initial vector  $x_0$ , a convergence tolerance  $e$ ,  $g_0 = \nabla f(x_0)$  and  $d_0 = -g_0$

**Step 2:** Set  $Q_k = H(f(x_k))$ , and  $a_k = -\frac{g_k^T d_k}{d_k^T Q_k d_k}$ ,  $\forall k \in N$

**Step 3:** Set  $x_{k+1} = x_k + a_k d_k$  and  $g_{k+1} = \nabla(f(x_{k+1}))$ .

**Step 4:** If  $\|g_{k+1}\| < e$ , then stop,  $x_{k+1}$  is minimum point and  $f(x_{k+1})$  is a minimum value.

**Step 5:** Find  $\beta_k = \frac{g_{k+1}^T Q_k d_k}{d_k^T Q_k d_k}$  and  $d_{k+1} = -g_{k+1} + \beta_k d_k$ , and return to **step 2**.

---

Here, for Example 1. we will use the Conjugate gradient method for minimizing  $f(x, y) = x^2 + y^2 + xy - 3x$ , where  $X_0 = (0,0)^T$  and  $e = 0.01$ .

**Solution:** Firstly, we have

$$g_0 = \nabla f(X_0) = \begin{bmatrix} -3 \\ 0 \end{bmatrix} \quad \text{and} \quad Q_0 = H(f(X_0)) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

then

$$d_0 = -g_0 = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \quad \text{and} \quad a_0 = -\frac{g_0^T d_0}{d_0^T Q_0 d_0} = -\frac{[-3,0] \begin{bmatrix} 3 \\ 0 \end{bmatrix}}{[3,0] \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix}} = \frac{1}{2}.$$

From it

$$X_1 = X_0 + a_0 d_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ 0 \end{bmatrix}.$$

Now,  $g_1 = \nabla f(X_1) = \begin{bmatrix} 0 \\ \frac{3}{2} \end{bmatrix}$ , since  $\|g_1\| = \sqrt{0^2 + (\frac{3}{2})^2} = \frac{3}{2} \not\leq e$ .

In the second iteration, we obtain that,

$$g_2 = \nabla f(X_2) = \begin{bmatrix} 2.2 + (-1) - 3 \\ 2.(-1) + 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Since  $\|g_2\| = \sqrt{0^2 + 0^2} = 0 < \varepsilon$ , stop. Hence  $X_2 = [2, -1]^T$  is a minimum point and point  $f(X_2) = -3$  minimum value.

#### 4. MATLAB COMPUTATIONS

We implemented the Matlab software for minimizing different functions. Algorithm 2 (which includes Algorithm 1 as a particular case) was implemented in Matlab software. This section describes the implementations of these algorithms and input parameter settings used in experiments [8]. In this work, Matlab 2018R has been used. Matlab codes for Algorithm 1, and Algorithm 2 are presented in the appendices of this work.

Firstly, we will test the Conjugate gradient method and Steepest descent method for function presented in **Example 1**. To compare both methods. That is,  $f(x) = x^2 + y^2 + xy - 3x$ , with initial  $X_0 = [0,0]^T$ , and tolerance 0.01. By applying Matlab software for Conjugate gradient methods algorithm. The results obtained that the min f solution needs only two iterations with an elapsed time of 0.077635 seconds, as shown in the Table 1.

**Table 1** Results obtained for the problem in Example 1. by using the Conjugate gradient method

Iteration	$X_k$	$g_k$	$F(x, y)$
1	3/2	0	-9/4
	0	3/2	
2	2	0	-3
	-1	0	

For the same example, the Steepest descent method was applied. It is minimized in 9 iterations with an elapsed time of 0.074341 seconds. Obtained results are presented in Table 2.

**Table 2** Results obtained for the problem in Example 1. by using the Steepest descent method

Iteration	$X_k$	$g_k$	$F(x, y)$
1	3	0	-9/4
	0	3/2	
2	3/2	-3/4	45/16
	-3/4	0	
3	15/8	0	-189/64
	-3/4	3/8	
4	15/8	-3/16	-765/256
	-15/16	0	
5	63/32	0	3069/1024
	-15/16	3/32	
6	63/32	-3/64	12285/4096
	-63/64	0	
7	255/128	0	-49149/16384
	-63/64,	3/128	
8	255/128,	-3/256	-196605/65536
	-255/256	0	
9	1023/512	0	-786429/262144
	255/256	3/512	

**Example 2.** Minimize  $f(x, y) = (\exp(x + 3y - 0.1) + \exp(x - 3y - 0.1) + \exp(-x - 0.1))$ , within error  $10^{-1}$  and initial guess  $[1,1]^T$

By applying for the Matlab program, we will obtain the result. Using the Steepest descent method, (10) iterations are needed to minimize the given function, with an elapsed time of 0.094797 seconds. In comparison, the Conjugate gradient method requires (9) iterations with an elapsed time of 0.137276 seconds.

**Table 3** Results obtained for the problem in Example 2. by using the Conjugate gradient method

Iteration	$X_k$	$g_k$	$F(x, y)$
1	0.900590397827910	18.1565591814171	18.8918838015203
	0.701237942980636	53.9425731538766	
2	0.509282268870050	17.6807451432240	18.7682271210071
	0.828862167258138	53.9218537005105	
3	0.643984521222065	6.73115204703044	7.68158530183416
	0.456063114530497	18.9876043890114	
4	-0.0230162774053545	5.27982507220765	7.13163492183488
	0.642512789818808	17.8452088652112	
5	0.463923957743200	2.71518751398399	3.85313162489058
	0.175088821374367	4.74647541989943	
6	-0.251693430220347	0.494520097267355	2.82212688683971
	0.196385254928735	2.63319293440899	
7	-0.109033530844227	0.615917281082720	2.63406619373276
	0.0175734944502638	0.256773197547600	
8	-0.329768907591008	0.0436758028150615	2.56029418406624
	-0.0106649193990685	-0.124927460095907	
9	-0.336134980189466	0.0267230029782450	2.55941346327887
	0.00112241921744625	0.0130622323406557	



**Table 4** Results obtained for the problem in Example 2. by using the Steepest descent method

Iteration	$X_k$	$g_k$	$F(x, y)$
1	0.900590397827910	18.1565591814171	18.8918838015203
	0.701237942980636	53.9425731538766	
2	0.802069988655053	7.06023955584850	7.87169742705322
	0.408536868627521	18.8433896723855	
3	0.704381764578780	3.57868065837855	4.47340949360141
	0.147812390638997	5.03041592127231	
4	0.586327497814813	2.75405695837841	3.76089994811188
	-0.0181320229015673	-0.531058379938799	
5	0.0213404117118031	1.03201063752364	2.80347542669767
	0.0908131191803420	1.52974449006004	
6	-0.0515591309406691	0.768321217482770	2.67374856247466
	-0.0172455283236301	-0.266883326043632	
7	-0.233668375137115	0.303238574846252	2.58926598489225
	0.0460117652256389	0.595126579543410	
8	-0.260388751788208	0.221102567197339	2.56903689545340
	-0.00642881351187573	-0.0807077816254727	
9	-0.318646956182128	0.0727847048655619	2.56156845340500
	0.0148368393374650	0.175768574123117	
10	-0.325501416348148	0.0539506230847806	2.55985224054417
	-0.00171607094760603	-0.0201844418835506	

**Example 3.** Minimize  $f(x, y) = 10(\exp(x + 3y) + \exp(x - 3y) + \exp(-x))$ , within error  $10^{-5}$  and initial guess  $[2, 1]^T$

Using the Matlab program for the given function, we will obtain minimizing points that are different for both methods. The results are explained in the following table.

**Table 5** In this table, the results of both methods are presented for Example 3.

	Conjugate gradient method	Steepest descent method
<b>Last Iteration</b>	14	32
<b>Elapsed time(Seconds)</b>	0.090517	0.083051
$X_k$	-0.34657 $7.3227 \times 10^{-10}$	-0.346571 $-1.5758 \times 10^{-8}$
$g_k$	$9.9914 \times 10^{-9}$ $9.3203 \times 10^{-8}$	$4.6409 \times 10^{-6}$ $-2.0057 \times 10^{-6}$
$F(x, y)$	28.284	28.284

#### 4. Results and Discussions

Generally, both methods are frequently used for minimizing functions. However, like other methods, there are many different advantages and disadvantages for techniques. We will compare both methods using the results of Example 2 and Example 3. the comparison has two aspects: time and efficiency.

Firstly, in the case of efficiency analysis, by noticing table 5. the Conjugate gradient method needs fewer iterations than the Steepest descent method. By looking at table 5 of example 3, one can see the last iteration for the Conjugate gradient method is  $g_{14} = [9.9914 \times 10^{-9}, 9.3203 \times 10^{-8}]^T$  and the last iteration of the Steepest descent method is  $g_{32} = [4.6409 \times 10^{-6}, -2.0057 \times 10^{-6}]^T$ . The results obtained from the Conjugate gradient method have more accuracy than the Steepest descent method, according to  $g_k$  values. Similarly, the results of Example 2, presented in table 3 and table 4, show that the Conjugate gradient method has more accuracy than the Steepest descent method. Hence, we can conclude that the Conjugate gradient method is more accurate than the Steepest descent method.

Secondly, in the time aspects, by looking at Table 5, one can conclude that the Steepest descent method needs 0.083051seconds to do 32 iterations for minimizing a function  $f(x, y)$ . On the other hand, the Conjugate gradient method needs 0.090517 seconds to do 14 iterations for finding min f. Similarly, for Example 2, the Steepest descent method minimized the function in 0.094797 seconds, while the Conjugate gradient method required 0.137276 seconds. We obtain that the Steepest descent method requires less time than the Conjugate gradient method to minimize the function. Finally, we conclude that the Conjugate gradient method is slower but more productive because it converges after fewer iterations.

## 5. Conclusions

In the present work, the Conjugate gradient method and the Steepest descent method have been studied. These methods are used for minimizing nonlinear functions. Algorithms for both methods have been presented. Matlab program has been implemented for solving difficult examples. Matlab codes have been presented for minimizing three examples by both methods. Moreover, a comparison has been made between the Conjugate gradient method and the Steepest descent method. A discussion has been carried out based on the results. The provided results demonstrates that, the Conjugate gradient method requires fewer iterations and is more efficient than the Steepest descent method. On the other hand, The Steepest descent approach takes less time to converge a function than the Conjugate gradient method.

**Declaration of Competing Interest,** The authors declare no conflict of interest.

## References

- [1] Djordjevic, S.S., 2019. Unconstrained Optimization Methods: Conjugate Gradient Methods and Trust-Region Methods. In Applied Mathematics. IntechOpen.
- [2] E.Carrizosa, E.Conde, M.MuñozMárquez, J.Puerto, Planar point objective location problems with nonconvex constraints: a geometrical construction, *J. Global Optim.* 6 (1) (1995) 77–86.
- [3] E. Carrizosa, J.B.G. Frenk, Dominating sets for convex functions with some applications, *J. Optim. Theory Appl.* 96 (2)(1998) 281–29
- [4] Das, I. and Dennis, J.E., 1998. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, 8(3), pp.631-657.
- [5] H. Eschenauer, J. Koski, A. Osyczka, *Multicriteria Design Optimization*, Springer, Berlin, 1990.
- [6] G.W. Evans, An overview of techniques for solving multi-objective mathematical programs, *Management Sci.* 30 (11)(1984) 1268–1282
- [7] D.J. White, Epsilon-dominating solutions in mean-variance portfolio analysis, *European J. Oper. Res.* 105 (1998)457–466.
- [8] Curtis, F.E. and Guo, W., 2016. Handling nonpositive curvature in a limited memory Steepest descent method. *IMA Journal of Numerical Analysis*, 36(2), pp.717-742.
- [9] Drummond, L.G. and Svaiter, B.F., 2005. A Steepest descent method for vector optimization. *Journal of computational and applied mathematics*, 175(2), pp.395-414.
- [10] Shewchuk, J.R., 1994. An introduction to the Conjugate gradient method without the agonizing pain.
- [11] Fletcher, R., 2012. A limited memory Steepest descent method. *Mathematical programming*, 135(1), pp.413-436.
- [12] Gonzaga, C.C. and Schneider, R.M., 2016. On the Steepest descent algorithm for quadratic functions. *Computational Optimization and Applications*, 63(2), pp.523-542.
- [13] Abrudan, T.E., Eriksson, J. and Koivunen, V., 2008. Steepest descent algorithms for optimization under unitary matrix constraint. *IEEE Transactions on Signal Processing*, 56(3), pp.1134-1147.
- [14] Pradhan, R. and Subudhi, B., 2012, December. A Steepest-descent based maximum power point tracking technique for a photovoltaic power system. In 2012 2nd International Conference on Power, Control and Embedded Systems (pp.1-6). IEEE.

- [15] J. Fliege, B.F. Svaiter, Steepest descent methods for multicriteria optimization, *Math. Methods Oper. Res.* 51 (3) (2000)479–494
- [16] Fliege, J. and Svaiter, B.F., 2000. Steepest descent methods for multicriteria optimization. *Mathematical methods of operations research*, 51(3), pp.479-494.
- [17] Barzilai, J. and Borwein, J.M., 1988. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1), pp.141-148.
- [18] Djordjevic, S.S., 2019. Some unconstrained optimization methods. In *Applied Mathematics*. IntechOpen.
- [19] Powell, M.D., 1977. Restart procedures for the Conjugate gradient method. *Mathematical programming*, 12(1), pp.241-254.
- [20] Fletcher, R. and Reeves, M., 1964. Function minimization by Conjugate gradients. *The computer journal*, 7(2), pp.149-154.
- [21] Hestenes, M.R. and Stiefel, E., 1952. *Methods of Conjugate gradients for solving linear systems* (Vol. 49, No. 1). Washington, DC: NBS.
- [22] Nazareth, L., 2009. Conjugate gradient method. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3), pp.348-353.
- [23] Faber, V. and Manteuffel, T., 1984. Necessary and sufficient conditions for the existence of a Conjugate gradient method. *SIAM Journal on numerical analysis*, 21(2), pp.352-362.
- [24] Stanimirović, P.S., Ivanov, B., Ma, H. and Mosić, D., 2020. A survey of gradient methods for solving nonlinear optimization. *Electronic Research Archive*, 28(4), p.1573.
- [25] Lasdon, L., Mitter, S. and Waren, A., 1967. The Conjugate gradient method for optimal control problems. *IEEE Transactions on Automatic Control*, 12(2), pp.132-138.
- [26] Vagaská, A. and Gombár, M., 2021. *Mathematical Optimization and Application of Nonlinear Programming*. In *Algorithms as a Basis of Modern Applied Mathematics* (pp. 461-486). Springer, Cham.
- [27] Osadcha, O. and Marszaek, Z., 2017. Comparison of Steepest descent method and Conjugate gradient method. In *CEUR Workshop Proceedings, SYSTEM*.
- [28] Beck, A., 2014. *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*. Society for Industrial and Applied Mathematics.
- [29] Baldick, R., 2006. *Applied optimization: formulation and algorithms for engineering systems*. Cambridge University Press.
- [30] Antoniou, A. and Lu, W.S., 2007. *Practical optimization: algorithms and engineering applications* (Vol. 19, p. 669). New York: Springer.
- [31] Venkataraman, P., 2009. *Applied optimization with MATLAB programming*. John Wiley & Sons.
- [32] Rahim, I. and Hussan, M.A., 2012. Teaching Optimization Techniques using Matlab and Mathematica: A Comparative Study. *Journal of Information & Communication Technology (JICT)*, 6(1), p.9.
- [33] Zhang, Y., 1998. Solving large-scale linear programs by interior-point methods under the MATLAB environment. *Optimization Methods and Software*, 10(1), pp.1-31.
- [34] Arora, J.S., 2006. Jan A. Snyman, *Practical Mathematical Optimization: An introduction to basic optimization theory and classical and new gradient-based algorithms*. *Structural and Multidisciplinary Optimization*, 31(3), pp.249-249.
- [35] Allwright, J.C., 1976. Conjugate gradient versus Steepest descent. *Journal of Optimization Theory and Applications*, 20(1), pp.129-134.
- [36] Meza, J.C., 2010. Steepest descent. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(6), pp.719-722.
- [37] Axelsson, O. and Lindskog, G., 1986. On the rate of convergence of the preconditioned Conjugate gradient method. *Numerische Mathematik*, 48(5), pp.499-523.
- [38] Dai, Y.H. and Yuan, Y., 2000. *Nonlinear Conjugate gradient methods*. Shanghai Science and Technology Publisher, Shanghai.