# A NEW GLOBALLY CONVERGENT COMPUTATIONAL SPECTRAL CONJUGATE GRADIENTALGORITHM FOR SOLVING UNCONSTRAINT NONLINEAR TEST-PROBLEMS

by

Ali Abbas Younis AL-ARABO

A thesis submitted to

the Graduate School of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

January 2016
Istanbul, Turkey

# A NEW GLOBALLY CONVERGENT COMPUTATİONAL SPECTRAL CONJUGATE GRADİENT ALGORİTHM FOR SOLVİNG UNCONSTRAINT NONLİNEAR TEST-PROBLEMS

Ali Abbas Younis AL-ARABO

M. Sc. Thesis - Computer Engineering
January 2016

Thesis Supervisor: Assist. Prof. Aişe Zülal ŞEVKLİ

## ABSTRACT

In this thesis, a new spectral conjugate gradient algorithm (SCG) is proposed for solving unconstrained nonlinear optimization problems. All work can be summarized in two parts: Firstly, a minor modification to the standard Conjugate Descent (CD) algorithm is proposed. The direction generated by the modified approach provides a descent direction for solving the objective functions and the modified algorithm coincide with the standard CD-algorithm if line search is exact. It is well-known that the search direction generated by a CG-algorithm may not be a descent direction of the objective function. This property depends neither on the Wolfe line search used, nor on the convexity of the objective function.
Secondly, we have studied and derived theoretically the stability and the global convergence properties for the new proposed SCG algorithm by introducing some well-known and new mild assumptions, which will be used in the proof of the basic idea's of the new algorithm.

Lastly, the performance of the new SCG algorithm is reported on the selected test problems whose second derivatives are available. Modified FORTRAN codes have been written in double precision arithmetic and all the tests were performed on a PC. In order to assess the reliability of our new proposed SCG algorithm, the results of new SCG are compared with the results of three standard CG algorithms and four recent published SCG algorithms in terms of solution quality and performance.

**Keywords:** Spectral Conjugate Gradient, Global Convergence Property, Unconstrained Nonlinear Optimization, Descent Search Direction, Line Searches.

# KISITSIZ DOĞRUSAL OLMAYAN TEST PROBLEMLERİN ÇÖZÜMÜ İÇİN YENİ KÜRESEL YAKINSAK HESAPLAMALI SPEKTRAL EŞLENİK GRADYAN ALGORİTMASI

Ali Abbas Younis AL-ARABO

Yüksek Lisans Tezi- Bilgisayar Mühendisliği
Ocak 2016

Tez Danışmanı: Yrd. Doç. Dr. Aişe Zülal ŞEVKLİ

## ÖZ

Bu tezde, kısıtsız doğrusal olmayan optimizasyon problemlerinin çözümü için yeni bir spektral eşlenik gradyan (SCG) algoritması önerilmiştir. Yapılan çalışmalar iki kısımda özetlenebilir: İlk olarak, standart Eşlenik İniş (CD) algoritmaları üzerinde küçük bir değişiklik yapılmıştır, şöyle ki, değişiklik yapılan yaklaşıma göre oluşturulan yön, seçili amaç fonksiyonlarını çözmek için iniş yönü göstermekte ve doğrultu belirleme tamamen aynıysa, standart CD algoritmasıyla değiştirilmiş algoritma örtüşmektedir. Eşlenik Gradyan algoritması sayesinde oluşturulan arama yönünün, amaç fonksiyonunun iniş yönü olmayacağı bilinen bir gerçektir. Bu özellik ne kullanılan Wolfe doğrultu arama ne de amaç fonksiyonunun dışbükeyliğine bağlıdır.

İkinci olarak, yeni algoritmanın temel fikirlerini kanıt olarak kullanabileceğimiz, tanınmış ve yeni mutedil varsayımları tanıtarak yeni önerilen SCG algoritmasının küresel yakınsama ve kararlılık özellikleri üzerine çalıştık ve kuramsal olarak türettik.
Son olarak ikinci türevleri mevcut olan seçili test problemleri üzerinde yeni SCG algoritmasının performansı raporlanmıştır. Değiştirilmiş FORTRAN kodları çifte hassasiyetli aritmetik ile yazılmış ve tüm testler PC üzerinde gerçekleştirilmiştir. Yeni önerilen SCG algoritmamızın güvenirliğini değerlendirmek için, algoritmamızın sonuçları üç standart CD-algoritması ve dört yeni yayımlanan SCG algoritması ile çözümün kalitesi ve performans kriterleri açısından karşılaştırılmıştır.

**Anahtar Kelimeler:** Spektral Eşlenik Gradyan, Global Yakınsama Özelliği, Kısıtsız Doğrusal Olmayan Optimizasyon, İniş Arama Yönü, Doğrultu Aramalar

To my parents, my wife and my children

# ACKNOWLEDGEMENT

Firstly, thanks God for giving me the ability to complete this work.

I express sincere appreciation to Assist. Prof. Aişe Zülal ŞEVKLİ for her support, guidance and insight throughout the research until completing my thesis, also special thanks to Assist. Prof. Kadir TUFAN, Head of the Department for his important instructions all the time.

Thanks should also go to all Fatih University staff, especially Mr. Mahir Balik for his patience and response throughout my study.

I express my thanks and appreciation to my family, epically my parents and wife for their understanding, motivation and patience.

Lastly, I express my thanks to all colleagues and friends who made my stay at the University a memorable and valuable experience.

# TABLE OF CONTENTS

# LIST OF TABLES

**TABLE**

# LIST OF SYMBOLS AND ABBREVIATIONS

**SYMBOL/ABBREVIATION**

| | |
|---|---|
| $x^*$ | The local minimum of function |
| $x$ | The approximation of vector $(x_1, x_2, ..., x_n)^{\mathrm{T}}$ to $x^*$ |
| n | Dimension of the Problem |
| f | Objective Function |
| g | $n \times 1$ gradient of f at $x_k$, $\nabla f(x_k)$ |
| d | $n \times 1$ search direction vectors |
| g | $n \times n$ Hessian Matrix |
| y | Difference of two successive gradients |
| s | Difference of two successive points |
| $\alpha, \lambda$ | Optimal step-size parameters |
| $\in$ | Positive scalar |
| $\theta_k$ | Scalar parameter denoted here to Spectral |
| $\beta^{CC}$ | Beta of convex combination |
| $\nabla f$ | Gradient of f |
| SD | Steepest Descent |
| CG | Conjugate Gradient |
| SCG | Spectral Conjugate Gradient |
| CD | Conjugate Descent |
| FR | Fletcher and Reeves |
| PR | Polak and Ribiere |
| LS | Liu and Story |
| DY | Dai and Yuan |
| HS | Hestenes and Stiefel |
| LDW | Liu, Du and Wang |

| | |
|---|---|
| BK | Al-Bayati and Al-Khayat |
| $\| . \|$ | Norm of a vector |
| ELS | Exact line search |
| ILS | Inexact line search |
| NOI | Number of Iterations |
| NOF | Number of Function Evaluations |
| CPU | Executed time |
| N.T. Fs | Number of test functions |
| BH | Al-Bayati and Hassan |
| BA | Al-Bayati and Abdullah |

# CHAPTER 1

# INTRODUCTION

## 1.1 OVERVIEW

Optimization is a word that means, making some times, "optimum"; optimum is Latin, and it means "the best." value. Therefore, optimization refers to bringing whatever we are dealing with towards its optimum. Now, optimization is a mathematical technique that is concerned with finding the maxima and minima of non-linear functions, some times, subject to constraints. (Andreessen et al., 2005).

In other words, optimization is a mathematical technique used for finding the least possible values of a mathematical function where the function to be optimized may be a function of any number of independent variables and may also be subject to certain constraints.

However, in optimization problems we seek those independent variables that do not violate the constraints of the objective function to be optimized. All problems in all areas of mathematical programming or computational mathematics can be posed in terms of optimization. Mathematical models are often developed in order to analyze and understand the objective function phenomena.

Moreover, the word optimization is an important tool in mathematics. Our aim in this thesis is to find the best optimal values of the variables that minimize the nonlinear unconstraint objective functions and the variables are restricted or constrains in some ways.

The mathematical model for the optimization algorithm can be used to find a solution for such problems. For all but the simplest problems, an exact solution cannot be calculated directly. Instead, suitable algorithms must be chosen that will approximate the solution as closely as required to the optimal solution. Often a set of optimality conditions can be applied to the final values returned by the algorithm to check that they yield a solution to the problem, (Baldick, 2009).

## 1.2 NOTATIONS

We use the following notations throughout this thesis : The working space is $R^n$, the set of column n-vectors with real components, we denote elements of $R^n$ by lower case letters e.g.: x the transpose of a given column vector x is a row vector with the corresponding elements and denoted by $x^T$, where the scalar product will be denoted by $x^T y$ (actually it will be the usual dot-product $x^T y = \sum_{i=1}^{n} x_i y_i$ and $\| . \|$ will denote the associated norm. The gradient (vector of partial derivative) of the differentiable function $f : R^n \to R$ will be denoted by $\nabla f$, and the Hessian matrix (matrix of second derivatives) by $\nabla^2 f(x)$. We will also use continually use the notation $g(x) = \nabla f$, $G = \nabla^2 f(x)$, and $H$ represents an approximation of the inverse Hessian. We refer to the current point (vector) as $x_k$, the value of $g(x)$, $f(x)$ at $x_k$ denoted as $g_k$ , $f_k$ respectively for an optimal point denoted by $x^*$. Greek letters such as . $\alpha, \beta, ..$ refers to the scalars.

## 1.3 CLASSIFICATION OF OPTIMIZATION MODELS

The general form of the unconstrained optimization problem to be considered may be expressed in mathematical terms as follows:

$$\min_{x \in R^n} f(x) \qquad (1.1)$$

This problem type depends on the nature of the functions and the constraints. Examples are the follows:

1) Unconstrained Optimization (UO)

2) Constrained Optimization (CO)

3) Parametric Optimization (PO)

4) Differentiable Optimization (DO)

5) Non-Differentiable Optimization (NDO)

6) Linear Programming (LP)

7) Nonlinear Programming (NLP)

8) Integer Programming (IP)

9- Convex Programming (CP)

10- Non-Convex Programming (NCP)

(Andreessen et al., 2005)

## 1.4 GRADIENT VECTOR

The vector of the first partial derivative $g = (\partial f / \partial x_1, \partial f / \partial x_2, ..., \partial f / \partial x_n)^T$, called a gradient vector of an n-variable continuously differentiable function $f(x_1, x_2, \ldots, x_n)$, and may also be written as $\nabla f$ (or sometimes as $f_x$), (Biggs, 2005).

## 1.5 HESSIAN MATRIX

The $n \times n$ matrix H of second partial derivatives of an n-variables continuously differentiable function $f(x_1, x_2, \ldots, x_n)$ is given by the following (Biggs, 2005):

$$H(x)=\begin{bmatrix} \partial^2 f/\partial x_1^2 & . \ . \ . & \partial^2 f/\partial x_1 x_n \\ . & . \ . \ . & . \\ . & . \ . \ . & . \\ . & . \ . \ . & . \\ \partial^2 f/\partial x_n x_1 & . \ . \ . & \partial^2 f/\partial x_n^2 \end{bmatrix}$$

**(1.2)**

is known as the Hessian matrix, which is always symmetric and f ($x_1$,$x_2$,…,$x_n$) is a twice continuously differentiable function because of the properties of the second derivatives of continuous functions:

$$\frac{\partial^2 f}{\partial x_i \partial x_j}=\frac{\partial^2 f}{\partial x_j \partial x_i} \quad i=1,2,...,n, \quad j=1,2,...,n \qquad (1.2)$$

## 1.6 POSITIVE DEFINITE/ POSITIVE SEMI-DEFINITE MATRIX

A positive definite symmetric matrix A is one that has all positive eigenvalues. Equivalently, a matrix A is positive definite if and only if:

$$x^T Gx > 0, \quad for \quad any \quad x \neq 0. \qquad (1.3)$$

For a positive definite symmetric matrix the condition (1.4) becomes:

$$x^T Gx \geq 0, \quad for \quad any \quad x \neq 0. \qquad (1.4)$$

(Biggs, 2005).

## 1.7 OPTIMALITY IN UNCONSTRAINED OPTIMIZATION

What are the necessary and sufficient conditions for a vector $x^*$ to be a local optimum? This is an important question, because the algorithms that we will investigate for solving important classes of optimization problems are always devised based on those conditions that we would like to fulfill.

This is a statement that seems to be universally true: efficient, locally or globally convergent iterative algorithms for an optimization problem are directly based on its necessary and/or sufficient local optimality conditions, (Andreessen et al., 2005). Different definitions for necessary and sufficient optimality conditions may be found in (Fletcher, 1978).

### 1.7.1 The First-Order Necessary Optimality Condition

Suppose that $f:\Re^n \to \Re$ is in $C^1$ on $\Re^n$. Then, $x^*$ is a local minimum of f

over $\Re^n \quad \Rightarrow \nabla f(x^*) = 0$. Note that $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_j}\right)_{j=1}^{n}$ .

### 1.7.2 The Second-Order Necessary Optimality Condition

Suppose that $f:\Re^n \to \Re$ is in $C^2$ on $\Re^n$. Then, $x^*$ is a local minimum of f $(x_1,x_2,\ldots,x_n)$

$$\Rightarrow \begin{cases} \nabla f(x^*) = 0^n. \\ \nabla^2 f(x^*) \quad is \quad positive \quad semidefinite \Leftrightarrow Hissian \quad matrix \quad satisfy \quad conditoin \end{cases} \quad (1.5)$$

### 1.7.3 The Second-Order Sufficient Optimality Condition

Suppose that $f:\Re^n \to \Re$ is in $C^2$ on $\Re^n$. Then, $x^*$ is a strict local minimum of f $(x_1,x_2,\ldots,x_n)$

$$\Rightarrow \begin{cases} \nabla f(x^*) = 0^n. \\ \nabla^2 f(x^*) \quad is \quad positive \quad definite \Leftrightarrow Hissian \quad matrix \quad satisfy \quad conditoin \end{cases} (1.4)$$

## 1.8 LOCAL MINIMUM

Analytically, a local minimizer is a point $x^*$ that satisfies the condition $f(x^*) \le f(x)$ for all x-values such that $\|x - x^*\| < \varepsilon$, where $\varepsilon$ is some (typically small) positive number whose value may depend on $x^*$. By optimality conditions, if f(x) is an n-variable function

whose gradient and Hessian satisfy g(x*) =0 and H(x*) is positive semi-definite or positive definite as defined in (1.5) then the point x* is a local minimum of f(x).

It is the second of the optimality conditions (1.5) that distinguishes a minimum from a maximum (or any other stationary point). It is the second derivative condition in equation (1.5) that ensures that f(x*) < f(x) , for all x in some, possibly small, region around x* (Griva et .al., 2009).

## 1.9 GLOBAL MINIMUM

Analytically, a local minimizer is a point $x^*$ that satisfies the condition $f(x^*) < f(x)$ for all x-values such that $0 < \|x - x^*\| < \varepsilon$. It is possible for a function to have a local minimizer and yet have no global minimizer. It is also possible to have neither global nor local minimizers, to have both global and local minimizers, to have multiple global minimizers, and various other combinations (Biggs, 2005). Under optimality conditions, for some functions f(x) there may be several points $x^*$ that satisfy equation (1.5). These are all local minima; and the one that gives the least value of f will be called the global minimum

## 1.10 A DESCENT (DOWNHILL) DIRECTION

From the current position we wish to find a direction that brings us downhill, a descent direction. This means that if we take a small step in that direction we get to a position with a smaller function value, (Frandsen, 2004) and (Fletcher, 1987). Let $x_k \in R^n$ and $f : R^n \to R.$ Then the vector $d \in R^n$ is a descent direction with respect to the function f at $x_k$ if:

“ $$d^T \nabla f(x_k) < 0 \Rightarrow d^T g_k < 0. \qquad (1.5)$$

## 1.11 RATE OF CONVERGENCE OF OPTIMIZATION METHODS

For many optimization methods, the number of operations or steps required to find an exact solution will be infinite, so some other measure of efficiency must be used. The rate of convergence is one of such measures; it describes how quickly the estimates of the solution approach the exact solution. That is the rate of convergence concerned with the speed at which various algorithms converge.

Let us assume that we have a sequence of points $x_k$ converging to a solution $x^*$. We define the sequence of errors to be $e_k = x_k - x^*$. where $x_k$ is the point reached at the k-th iteration and $x^*$ is the minimum note that $\lim_{k\to\infty} e_k = 0$. We say that the sequence $\{x_k\}$ converges to $x^*$ with rate of convergence p and rate constant C if:

$$\lim_{k\to\infty} \frac{\|e_k\|}{\|e_{k-1}\|^p} = C. \qquad (1.6)$$

If P = 1 and C = 0, the convergence is called superlinear. When P=2, the convergence is called quadratic. For optimization algorithms there is one other important case, and that is when 1< P < 2. This is another special case of superlinear convergence (Griva et. al., 2009). This case is important because:

(a) it is qualitatively similar to quadratic convergence for the precision of common computer calculations and

(b) it can be achieved by algorithms that only compute first derivatives, whereas to achieve quadratic convergence it is often necessary to compute second derivatives as well.

## 1.12 UNCONSTRAINED OPTIMIZATION ALGORITHM

The outlines of this algorithm can be found in (Andreessen et. al., 2005).

Step1. Given $x_0 \in R^n$, $0 < \varepsilon < 1$.

Step2. Compute the descent direction $d_k$ , say $d_k = -g_k$

Step3. Compute the size $\alpha_k$, such that $f(x_{k-1}+\alpha_k d_{k-1}) = \min_{\alpha \geq 0} f(x_{k-1}+\alpha_k d_{k-1})$;

Step4. Set $x_k = x_{k-1}+\alpha_k d_{k-1}$ and hence compute $\|\nabla f(x_k)\|$.

Step5. If $\|\nabla f(x_k)\| \leq \varepsilon$, stop ; otherwise, repeat the above steps.

## 1.13 STEEPEST DESCENT METHOD

It is one of the first and well known method for unconstrained optimization, which designed by Cauchy early in 1847, in which the negative gradient direction is used to find a local minimizer of a differentiable function.

The steepest descent method is one of the simplest and most fundamental minimization methods for unconstrained optimization. Since it uses the negative gradient as its descent direction, it is also called the gradient method (Wenyu and Yuan, 2006). It's direction $d_k = -g_k$; suppose that f(x) is continuously differentiable near $x_k$, and for Taylor expansion:

$$f(x) = f(x_k) + (x - x_k)^T g_k + o(\|x - x_k\|), \tag{1. 8}$$

We know that, if we write $x - x_k = \alpha d_k$, then the direction $d_k$ satisfying $d_k^T g_k < 0$ is called a descent direction that is such that $f(x)<f(x_k)$. By the Cauchy-Schwartz inequality:

$$\left| d_k^T g_k \right| \leq \|d_k\| \cdot \|g_k\|, \tag{1. 9}$$

We have that the value $d_k^T g_k$ is the smallest if and only if $d_k = -g_k$. The iterative scheme of the steepest descent method is as follows:

$$x_k = x_{k-1} - \alpha_k g_{k-1}. \tag{1. 10}$$

### 1.13.1 Outlines of Steepest Descent Algorithm

The outlines of this algorithm can be found in (Wenyu and Yuan, 2006).

Step1. Let $0 < \varepsilon < 1$ be the termination tolerance. Given an initial point $x_0 \in R^n$. Set k=0.

Step2. If $\|g_k\| \le \varepsilon$, stop ; otherwise let $d_k = -g_k$ .

Step3. Find the step length factor $\alpha_k$ , such that

$$f(x_{k-1} + \alpha_k d_{k-1}) = \min_{\alpha \ge 0} f(x_{k-1} + \alpha_k d_{k-1});$$

Step4. Compute $x_k = x_{k-1} + \alpha_k d_{k-1}$

Step5. Set k=k+1, return to Step 2.

### 1.13.2 Basic Properties of Steepest Descent Method

1) The quantity $g_k^T d_k$ is called the **directional derivative** of f at $x_k$ in the direction $d_k$ .

2) Analytically, the condition $g_k^T d_k < 0$ requires that the directional derivative in the direction $d_k$ be negative.

3) Geometrically, this condition requires that the angle between $d_k$ and -$g_k$ be less than $90^{0,}$ for $d_k$ to be a descent direction, (Wenyu and Yuan, 2006).

### 1.13.3 The Initial Step Length of Steepest Descent Direction

Steepest descent and CG- methods do not produce well-scaled search directions, it is important to use current information about the problem and the algorithm to make the initial guess. A popular strategy is to assume that the first-order change in the function at iterate $x_k$ will be the same as that obtained in the previous step. We therefore have:

$$\alpha_k = \alpha_{k-1} \frac{\|d_{k-1}\|_2}{\|d_k\|_2}, \qquad (1.11)$$

Next to interpolate a quadratic to the data $f(x_{k-1})$, $f(x_k)$, and $\phi'(0) = \nabla f_k^T d_k$ to define $\alpha_k$ to its minimizer (Nocedal and Wright, 2006). This strategy yields the following:

$$\alpha_k = \frac{2(f_k - f_{k-1})}{\phi'(0)}, \qquad (1.12)$$

It can be shown that if $x_k \to x^*$ is super linear, then the ratio in this expression converges to one. We find that the unit step length $\alpha_k = 1$ will eventually always be tried and accepted, and the superlinear convergence properties of Newton and quasi-Newton methods will be observed.

## 1.14 NEWTON METHOD

A local minimizer $x^*$, is sometimes, approximated well by a quadratic function. Thus, methods based on quadratic function models should have a rapid ultimate rate of convergence. If first and second derivatives of f(x) are available, a quadratic model of the objective function is obtained by taking the first three terms of a truncated Taylor series expansion about the current point $x_k$ , i.e.,

$$f(x_k + s) = f(x_k) + g_k^T s + \frac{1}{2} s^T H(x_k) + .. \qquad (1.\ 13)$$

Minimizing f yields:

$$x_k = x_{k-1} - [\nabla^2 f(x_{k-1})]^{-1} \nabla f(x_{k-1}) \qquad (1.\ 14)$$

Which is Newton's formula. Set $H_k = \nabla^2 f(x_k)$, $g_k = g(x_k)$. , then (1. 14):

$$x_k = x_{k-1} - H_k^{-1} g_k, \qquad (1.\ 15)$$

where $s_k = x_k - x_{k-1} = -H_k^{-1} g_k$ is pure Newton direction (Navon and Legler, 1987). Newton's direction is clearly a descent direction because it satisfies $g_k^T s_k = -g_k^T H_k g_k < 0$ if the matrix $H_k$ is positive definite.

### 1.14.1 Outlines of Pure Newton Algorithm

The outlines of this algorithm can be found in (Wenyu and Yuan, 2006).

Step1. Given $x_0 \in R^n$, $\varepsilon > 0$, $k = 0$;

Step2. If $\|g_k\| < \varepsilon$, stop;

Step3. Solve $H_k s_k = -g_k$ for $s_k$.

Step4. Set $x_k = x_{k-1} + s_k$;

Step5. k=k+1, go to Step2.

## 1.15 EXACT LINE SEARCH

A line search chooses $\alpha^*$ to minimize $\phi(\alpha) = f(x_{k-1} + \alpha d_{k-1})$, that is if we find $\alpha^*$ such that the objective function in the direction $d_k$ is minimized, i.e., $f(x_{k-1} + \alpha_k d_{k-1}) = \min_{\alpha \geq 0} f(x_{k-1} + \alpha_k d_{k-1})$; . Such a line search is called an exact line search or an optimal line search or perfect line search, and $\alpha_k$ is called optimal step size, (Fletcher, 1987). Also, we say that an optimization algorithm has an exact line search (ELS), if and only if:

$$g_i^T d_{i-1} = 0 \text{ , for i=1,2,3,...} \qquad (1.16)$$

An exact line search gives the greatest possible reduction in the objective function (f) along the search direction. However, as we shall see in, it may be computationally expensive to do an accurate minimization of $\phi(\alpha)$ on every iteration.

## 1.16 INEXACT LINE SEARCH

If we choose $\alpha_k$ such that the objective function has an acceptable descent amount i.e., such that the descent $f(x_k) - f(x_{k-1} + \alpha_k d_{k-1}) > 0$ is acceptable to the users, or it is one that accepts any value of $\alpha_k$ such that $f(x_{k-1} + \alpha_k d_{k-1}) - f(x_k)$ is negative and bounded away from zero. Such a line search is called an inexact line search (ILS), an approximate line search, an acceptable line search, or a weak line search. Hence weak (inexact) searches are often preferred in practice. (Biggs, 2005).

## 1.17 A LINE SEARCH METHOD

Another important issue related to the performance of conjugate gradient methods is the line search, which requires sufficient accuracy to ensure that the search directions yield descent. A common criterion for line search accuracy is the Wolfe conditions (Wolfe, 1969):

$$f(x_{k-1}+\alpha_k d_{k-1}) - f(x_k) \le \sigma_1 \alpha_k g_{k-1}^T d_{k-1}, \qquad (1.17)$$

$$g_k^T d_{k-1} \ge \sigma_2 g_{-1}^T d_{k-1}, \qquad (1.18)$$

where $0 < \sigma_1 \le \sigma_1 < 1$. In the "strong Wolfe" conditions, (1.18) is replaced by $\left|g_k^T d_{k-1}\right| \le -\sigma_2 g_{k-1}^T d_{k-1}$. The strong Wolfe conditions may not yield a direction of descent unless $\sigma \le 1/2$. Inequality given in (1.17) is sometimes called the Armijo condition (Hager and Zhang, 2005).

## 1.18 CONVERGENCE OF NEWTON'S METHOD

Let $f \in C^2$ and $x_k$ be close enough to solution $x^*$ of the minimization problem with $g(x^*)=0$. If the Hessian $H(x^*)$ is positive definite and H(x) satisfies Lipschitz condition:

$$\left|H_{ij}(x) - H_{ij}(y)\right| \le \beta \|x - y\|, \text{ for some } \beta \text{ for all i,j} \qquad (1.19)$$

$H_{ij}$ is the (i,j)-element of H(x), then for all k values, Newton's iteration equation (1.14) is well-defined; the generated sequence $\{x_k\}$ converges to $x^*$ with a quadratic rate where Newton's iteration with a line search is as follows:

$$d_k = -H_k^{-1} g_k, \qquad (1.20)$$

**Proof** ( see Fletcher, 1987).

## 1.19 OUTLINES OF NEWTON'S METHOD WITH LINE SEARCH

The outlines of this algorithm can be found in (Wenyu and Yuan, 2006).

Step1. Initial step: Given $x_0 \in R^n$, $\varepsilon > 0$, $k = 0$;

Step2. Compute $g_k$. If $\|g_k\| < \varepsilon$, stop and out put $x_k$; otherwise continue

Step3. Solve $H_k s_k = -g_k$ for $s_k$.

Step4. Line search step: Find $\alpha_k$ such that: $f(x_{k-1} + \alpha_k d_{k-1}) = \min_{\alpha \geq 0} f(x_{k-1} + \alpha_k d_{k-1})$;

Step5. Set $x_k = x_{k-1} + \alpha_k d_{k-1}$, k=k+1 , go to Step2.

## 1.20 LAYOUT OF THE THESIS

The rest of the thesis is organized as follows: In the second chapter a literature review on CG-methods sheds light on the different types of CG-algorithms that have been used to solve our selected set of 55-test problems. In the third chapter, we have described different types of spectral CG-methods with their outlines which are used in our thesis, especially those that are used in our comparisons theoretically and numerically. In chapter four, we have produced a new spectral CG algorithms with both theoretical and experimental implementation and properties. The fifth chapter, provides our practical implementation for solving our complicated set of nonlinear test problems. This chapter, is very important because it compares our numerical results with the previously published works, taking into consideration the different data sets used by the previous researches and also partial results in some recent research papers and it could be used as reference for later papers in the future. The last chapter - chapter six - discusses the concluding results of this thesis, some limitations and future works; what are the positive results we have achieved, what could be better, and all the weakness that may be found in our work and all possible improvements in the future which will be good starts for new researches.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 INTRODUCTION

The development of modern numerical descent methods for unconstrained nonlinear optimization has been taking place over the last fifty years ago and considerable progress has been made. We now have good methods for many classes of problems and many insights into why these methods are successful. There is a wealth of literature available, nevertheless there is a number of open questions of interest and new ideas continue to enrich the field. It is turns out that most, if not all optimization algorithms are based on the idea of iterative descent. The method generates a sequence of points each of which is calculated based on the one preceding it and as each new point is generated by the method, the corresponding value of some function decreases as illustrated in the gradient descent method. Of course we would like the sequence of points generated by the algorithm to converge to either a global or a local minimum of $f(x)$. An even more desirable feature is that such a converge occurs in a finite number of steps. One of the most natural and widely, some times, classes of iterative descent algorithm is the class of the so called gradient descent methods.

Let $f : R^n \to R$ be a continuously differentiable function and try to find the following:

$$\min_{x \in R^n} f(x) \tag{2.1}$$

Consider the unconstrained optimization problem given in equation 2.1. In general it may be too ambitious to find a global minimum of $f$. Hence we will just look for stationary points of $f$ i.e. a point $x^* \in R^n$ that satisfies $g(x^*) = 0$. To begin, let $x_0 \in R^n$ be an

initial estimate with $g(x_0) \neq 0$. In order to achieve progress we need to proceed in some search direction $d_k \in R^n$. For instance we can update the iterates according to:

$$x_k = x_{k-1} + \alpha_k \, d_{k-1} \qquad k \geq 1 \qquad (2.2)$$

We know that $\alpha_k > 0,$ controls how far we proceed in the direction $d_k$. Note that equation 2.2 actually defines a family of update rules that are parameterized by the search direction $d_k$ and step-sizes $\alpha_k$. There are many possibilities in choosing search directions and step-sizes in gradient descent methods and it must be said that there is no single choice that is superior to others in most situations.

## 2.2 HISTORY OF CONJUGATE GRADIENT METHODS

The work of (Hestenes and Stiefel, 1952) was the beginning of CG-methods. They presented an algorithm for solving symmetric, positive definite linear algebraic systems. In 1964 (Fletcher and Reeves, 1964) extended the domain of the application of CG methods to non-linear problems, thus starting the non-linear CG-directions. The main advantages of the CG methods are their low memory requirements, its convergence speed and it satisfaction of a quadratic termination property in which the method is able to locate the minimizer of quadratic function in a finite number of iterations, yet which can be applied iteratively to minimizing non-quadratic functions.

A set of non-zero vectors $d_k \quad (\mathrm{k} = 1,2,....,\mathrm{n})$ are said to be conjugate relative to a given positive definite matrix G if:

$$d_i^T \, G \, d_j = 0 \qquad \forall \, i \neq j \qquad (2.3)$$

Hence, CG-method is Conjugate Direction (CD) method, which generates such directions when applied to a quadratic function with Hessian G. It's readily shown in this case that the CD methods terminates in at most $n$ iterations if exact line search are used (Fletcher, 1987).

An excellent survey of development of different versions of non-linear CG methods, is presented by (Hager and Zhang, 2006). This family of algorithms includes a lot of variants with important convergence properties and numerical efficiency. For solving the non-linear unconstrained optimization problem (2.2), a CG-method generates a sequence according to (2.2) and the directions $d_k$ are generated as:

$$d_k = -g_k + \beta_k d_{k-1} \qquad k \geq 1 \tag{2.4}$$

Therefore, various choices of the scalar $\beta_k$ (known as the conjugacy parameter) exists which give different performances on non-quadratic functions, yet are equivalent for quadratic functions. The line search in the CG algorithms often based on the standard or strong Wolfe conditions that are needed to ensure convergence and to enhance stability (Andrei, 2007a). According to the formula for $\beta_k$ computation the conjugate gradient algorithms can be classified as follows:

## 2.3 CLASSICAL CONJUGATE GRADIENT METHODS

These algorithms are defined by (2.2) and (2.4) where the parameter $\beta_k$ is computed, as in one of the following manners.

$$\beta^{FR} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{2.5}$$

$$\beta^{DY} = \frac{g_k^T g_k}{d_{k-1}^T y_{k-1}} \tag{2.6}$$

$$\beta^{CD} = \frac{-g_k^T g_k}{g_{k-1}^T d_{k-1}} \tag{2.7}$$

$$\beta^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} \tag{2.8}$$

$$\beta^{PR} = \frac{g_1^T y_{k-1}}{g_{k-1}^T g_{k-1}} \tag{2.9}$$

$$\beta^{LS} = -\frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}} \tag{2.10}$$

Observe that these algorithms can be classified as algorithms with $g_k^T g_k$ in the numerator of $\beta_k$ and algorithms with $g_k^T y_{k-1}$ in the numerator of parameter $\beta_k$ where $y_k = g_k - g_{k-1}$.

The first CG algorithm 2.5 was introduced by (Fletcher and Reeves, 1964), for non-linear functions. Equation 2.6 was proposed by (Dai and Yuan, 1999) and equation 2.7 was introduced by (Fletcher, 1987). On the other hand equation 2.8 suggested by (Hestenes and Stiefel, 1952) and equation 2.9 was developed by (Polak and Ribiere, 1969) while equation 2.10 was derived by (Liu and Story, 1991). However, HS, PR and LS methods automatically adjust $\beta_k$ to avoid jamming and their performances are better than the performance of CG methods with $g_k^T g_k$ in the numerator of $\beta_k$ (Dai and Yuan, 2001).

## 2.4 HYBRID CONJUGATE GRADIENT METHODS

There are two classes of hybrid algorithms. The first class computes $\beta_k$ in (2.4), by one of the following manners:

$$\beta^{HDY} = \max \{ C\beta_k^{Dy}, \min \{ \beta_k^{HS}, \beta_k^{Dy} \} \} \tag{2.11}$$

or

$$\beta_k^{HTS} = \begin{cases} \beta_k^{PR} & 0 \le \beta_k^{PR} \le \beta_k^{FR} \\ \beta_k^{FR} & other\ wise \end{cases} \tag{2.12}$$

where $\beta_k^{HDY}$ is due to (Dai and Yuan, 2001) and $\beta_k^{HTS}$ was proposed by (Touti-Ahmed and Story, 1990).

Another class of hybrid algorithms, involves with more recently established convex combinations of CG-algorithms with $g_k^T g_k$ in the numerator of $\beta_k$ and algorithms having $g_k^T y_{k-1}$ in the numerator of $\beta_k$. For reference we list some of this type of $\beta_k$:

$$\beta_k^{CC} = (1-\theta_k)\ \frac{g_k^T\ y_{k-1}}{g_{k-1}^T\ g_{k-1}} + \theta_k\ \frac{g_k^T\ g_k}{s_{k-1}^T\ y_{k-1}} \tag{2.13}$$

$$\theta_k = \frac{(y_k^T\ g_k)(s_{k-1}^T y_{k-1}) - (y_k^T g_k\ )(g_{k-1}^T g_{k-1})}{(y_{k-1}^T\ g_k)(s_{k-1}^T g_{k-1}) - (g_k^T g_k)(g_{k-1}^T g_{k-1})} \tag{2.14}$$

where $\beta^{CC}$ with $\theta_k$ is defined in equation 2.14 and introduced by (Andrei, 2007b; 2009b).

## 2.5 SCALED CONJUGATE GRADIENT METHODS

This class involves CG-algorithms that generates the sequence $x_k$ according to (2.2) and defines the new search directions as follows:

$$d_k = -\theta_k g_k + \beta_k\ d_{k-1} \tag{2.15}$$

where $\theta_k$ is a parameter. The first iteration is initialized with an initial point $x_1$ and $d_1 = -g_1$. Observe that if $\theta_k = 1$, then we get the classical CG-algorithms according to the value of $\beta_k$ . On the other hand, if $\beta_k = 0$ then we get another class of algorithms according to the selection of the parameter $\theta_k$ . considering $\beta_k = 0$ there are two possibilities for $\theta_k$ a positive scalar or positive definite matrix. If $\theta_k = 1$ then we have SD-algorithms. If $\theta_k = G_k^{-1},$ or an approximation of it, then we get Newton-algorithm.

Therefore, we see that in the general case, when $\theta_k \neq 0$ and $\beta_k \neq 0,$ in equation 2.15, it represents a combination between the QN and CG methods. However if $\theta_k$ is a matrix , we are better of using $d_k = -\theta_k\ g_k$ since the addition of the term $\beta_k d_k$ in equation 2.15 may prevent the direction $d_k$ from being a descent direction unless the line search is sufficiently accurate. Therefore in the scaled CG, $\theta_k$ is considered as a positive scalar.

The first SCG method computes $\beta_k$ in (2.15) as follows :

$$\beta_k = \frac{\left(\theta_k\ y_{k-1} - s_{k-1}\right)^{\mathrm{T}} g_k}{d_{k-1}^T\ y_{k-1}} \tag{2.16}$$

which is called a scaled Perry CG introduced by (Birgin and Martinez, 2001). Other scaled CG methods are introduced by (Andrei, 2007c).

$$\beta_k^{SFR} = \frac{\theta_k g_{\mathrm{k}}^T g_k}{\theta_{k-1} g_{k-1}^T g_{k-1}} \tag{2.17}$$

$$\beta_k^{SPR} = \frac{\theta_k g_{\mathrm{k}}^T y_{k-1}}{\theta_{k-1} g_{k-1}^T g_{k-1}} \tag{2.18}$$

where

$$\theta_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}} \tag{2.19}$$

## 2.6 MODIFIED CONJUGATE GRADIENT METHODS

All modified CG-methods are designed to improve the performance of the classical CG-methods using the idea of modification of classical CG-methods in order to satisfy the sufficient descent condition. The most famous algorithms in this class are given by equations 2.2 and 2.4, where $\beta_k$ is computed (for example) as one of the following methods:

$$\beta_k^{A1}\ \frac{1}{\mathrm{s}_{\mathrm{k\text{-}1}}^{\mathrm{T}}\ y_{k-1}}\left(y_{k-1} - \frac{y_{k-1}^T\ y_{k-1}}{g_{k-1}^T\ g_{k-1}}\ s_{k-1}\right)^T\ g_k \tag{2.20}$$

$$\beta_k^{A2} = \frac{1}{\mathrm{s}_{\mathrm{k\text{-}1}}^{\mathrm{T}}\ y_{k-1}}\left(y_{k-1} - \frac{g_k^T\ y_{k-1}}{y_{k-1}^T\ s_{k-1}}\ s_k\right)^T\ g_k \tag{2.21}$$

or

$$d_k = \frac{s_{k-1}^T g_k}{s_{k-1}^T y_{k-1}}\ y_{k-1} + [\frac{y_{k-1}^T g_k}{s_{k-1}^T y_{k-1}} - (1 + \frac{y_{k-1}^T y_{k-1}}{s_{k-1}^T y_{k-1}})\frac{s_{k-1}^T g_k}{s_{k-1}^T y_{k-1}}]\ s_{k-1} \tag{2.22}$$

where $\beta^{A1}$ *and* $\beta^{A2}$ are introduced by (Andrei, 2009a) and equation 2.22 is suggested by (Shanno, 1978). For other methods in this class see (Hager and Zhang, 2005).

## 2.7 PARAMETRIC CONJUGATE GRADIENT METHODS

The parametric CG methods have been introduced in the same way that the QN methods have been combined to get the Broyden or Huang families. These algorithms are defined by equations 2.2 and 2.4 where the parameter $\beta_k$ is computed (for example) as follows:

$$\beta_k = \frac{g_k^T \left(y_{k-1} - t\, s_{k-1}\right)}{s_{k-1}^T\, y_{k-1}}, \quad t > 0 \text{ constant} \tag{2.23}$$

introduced by (Dai and Liao, 2001).

or

$$\beta_k = \frac{\left\| g_k \right\|^2}{\lambda \left\| g_k \right\| + (1-\lambda) d_{k-1}^T y_{k-1}}, \quad \lambda \in [0,1] \tag{2.24}$$

(Dai and Yuan, 2001). Other forms of this type of methods can be found in (Zhang et. al., 1999) and (Nazareth, 1999). Usually CG methods are implemented with restart since the rate of convergence of the algorithm is only linear unless the iterative procedure is occasionally restarted occasionally. It is typical to restart at every $n$ *or* $(n+1)$ iterations, but this is not satisfactory since $n$ is large, therefore other restarts are used such as the Powell restart (Powell, 1977) defined as follows:

$$\left| g_k^T g_{k-1} \right| \geq 0.2 \left| g_k^T g_k \right| \tag{2.25}$$

## 2.8 CONVERGENCE OF LINE SEARCH DESCENT METHODS

To ensure that an algorithm converges to a point $x$ where $g(x) < 0$ , we need not only well-chosen step lengths but also well-chosen search directions $d_k$. We focus in this

section, on a key parameter. The angle $\theta_k$ between $d_k$ and the SD direction $-g_k$ is defined by:

$$d_{k-1}^T g_{k-1} = -\cos\theta_k \|g_{k-1}\| \|d_{k-1}\| \quad (2.26)$$

Zoutendijk theorem analyses the convergence properties of the various descent methods.

## 2.9 ZOUTENDIJK THEOREM

Consider any iteration of the from equation 2.2 where:

(i) $d_k$ is descent direction.

(ii) $\alpha_k$ satisfies Wolfe conditions 1.17 and 1.18

(iii) $f$ is bounded below in $R^n$

(iv) The gradient $g$ is Lipschitz continuous in an open set $N$ containing the level set $\delta = \{x : f(x) \leq f(x_1)\}$ where $x_1$ is the starting point i.e. there exists a constant $L$ such that:

$$\|g(x) - g(y)\| \leq L \|x - y\| \qquad \forall \quad x, y \in N \quad (2.27)$$

Then

$$\sum_{k\geq 1} \cos\theta_k \|g_k\|^2 < \infty \quad (2.28)$$

**Proof.** (see Zoutendijk, 1970**)**

## 2.10 APPLICATION AREA

The word optimization means selecting the best value (the optimal solution) from a range of choices (local solutions). Optimization is an important tool in many fields of the life, such as; management and engineering sciences, mathematics of finance and in physical mathematics. Hence, it is central to any problem involving engineering,

mathematics and computer science, and any subject related to the computational mathematics. (Andreessen et al., 2005). The procedure consists of first identifying some non-linear models for the a non-linear objective or a quantitative measure of the performance of the system under study for some examples of the profit from an investment or the potential energy of a physical system and the objective depends on certain characteristics of the system, the variables or the unknown (which is under control). Our aim in this thesis is to find values of the variables that minimize some complicated non-linear objective functions, the variables are restricted or constrained in some ways. This must be done after designing the non-linear model for the selected model. Once the physical or financial description of the non-linear optimization problem has been translated into the mathematical notation of a mathematical model, then the optimization algorithm can be used to find the optimal solution for this complicated test problem. In general, for all but the simplest problems, an exact solution cannot be calculated directly. Instead, a suitable CG- or SCG-algorithm must be chosen which will approximate the solution as closely as required. Often a set of optimality conditions can be applied to the final values returned by the algorithm to check that they yield a solution to the problem.

# CHAPTER 3

# SPECTRAL CONJUGATE GRADIENT METHODS

## 3.1 INTRODUCTION

Our aim in this chapter is to study the performance of different recent spectral CG-methods (SCG) proposed by different authors with the outlines of their algorithms suitable for solving nonlinear unconstrained optimization problems with or without different restarting criteria and with appropriate conditions.

## 3.2 SCG-Method of Raydan (1997)

A new class of SCG-algorithms was established by Raydan (1997), for solving large-scale optimization problems. The main property of this method (Algorithm R) is that only gradient directions are used at each iteration. The first idea was started by Barzilai and Borwein (1988). Birgin and his friends (2000) discussed the effect of SCG-methods based on spectral projected gradient methods on convex sets. In addition, this method outperforms standard CG-method in many problems. Birgin and Martinez (2001) proposed three kinds of spectral CG-methods. The search direction $d_k$ is given by the following way:

$$d_k^R = -\theta_k^R g_k + \beta_k^R d_{k-1} \tag{3.1}$$

where the parameter $\beta_k$ is computed in the following way:

$$\beta_k^{R1} = \frac{(\theta_k y_{k-1} - s_{k-1})^T g_k}{s_{k-1}^T y_{k-1}}, \beta_k^{R2} = \frac{\theta_k y_{k-1}^T g_k}{\alpha_{k-1}\theta_{k-1} g_{k-1}^T g_{k-1}}, \beta_k^{R3} = \frac{\theta_k g_k^T g_k}{\alpha_{k-1}\theta_{k-1} g_{k-1}^T g_{k-1}} \tag{3.2}$$

respectively, and $\theta_k^R$ is taken to be the spectral gradient parameter and computed by the following:

$$\theta_k^R = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}} \tag{3.3}$$

### 3.2.1 Algorithm R

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \le \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$.

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: Compute $g_k$ ; if $\|g_k\| \le \varepsilon$, then stop ; otherwise continue.

Step 4: If the restarting criterion $\left|g_k^T g_{k-1}\right| \ge 0.2\left|g_k^T g_k\right|$ is satisfied then

do a restart step by $-g_0$ direction; otherwise continue.

Step 5: Compute the parameter $\theta_k^R$ from equation 3.3.

Step 6: Compute the parameter $\beta_k$ from one of the forms given in equation 3.2.

Step 7: Compute the new search direction from equation 3.1.

Step 8: Set k=k+1 and go to Step 2.

The search direction $d_k$ computed by Step 6 can fail to be a descent direction. This fact motivated several modifications. One of them suggested by Raydan is to "restart" the algorithm with the spectral gradient direction. For more details and comparisons see (Yu,et.al., 2008).

## 3.3 SCG-METHOD OF AL-BAYATI AND ABDULLAH (2008)

A new family of SCG–methods for solving large-scale unconstrained optimization problems was introduced by Al-Bayati and Abdullah (2008) using both spectral and scaling

properties for their search directions, which is a generalization of Algorithm R. They implemented two modifications in their method (Algorithm BA), one using Raydan line search, and in the other, they modified the standard Wolfe line search subroutine.

### 3.3.1 Algorithm BA

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \le \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$.

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: Compute $g_k$ ; if $\|g_k\| \le \varepsilon$, then stop ; otherwise continue.

Step 4: If the restarting criterion $\left|g_k^T g_{k-1}\right| \ge 0.2\left|g_k^T g_k\right|$ is satisfied then

do a restart step by $-g_0$ direction; otherwise continue.

Step 5: Compute the parameter $\theta_k^{BA}$: $\theta_k^{BA} = \dfrac{s_{k-1}{}^T s_{k-1}}{s_{k-1}^T s_{k-1} + \varepsilon \quad s_{k-1}^T y_{k-1}}, \qquad \varepsilon = 0.0001$

Step 6: Compute the parameter: $\beta_k^{CD} = -\dfrac{g_k^T g_k}{g_{k-1}^T d_{k-1}}$

Step 7: Compute the new search direction: $d_k = -\theta_k^{BA} g_k + \beta_k^{BA} d_{k-1}$

Step 8: Set k=k+1 and go to Step 2.

## 3.4 SCG-METHOD OF AL-BAYATI AND HASSAN (2011)

In this work, another family of SCG–methods for solving large-scale unconstrained optimization problems was introduced by Al-Bayati and Hassan (2011). The search directions of the new SCG method (Algorithm BH) are always of sufficient descent. They have proven the global convergence property of the proposed method. Finally, they have presented some numerical results to examine the efficiency of their SCG- method.

### 3.4.1 Algorithm BH

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \le \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$, $\mu = 1.1$.

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: Compute $g_k$ ; if $\|g_k\| \le \varepsilon$, then stop ; otherwise continue.

Step 4: If the restarting criterion $\left|g_k^T g_{k-1}\right| \ge 0.2\left|g_k^T g_k\right|$ is satisfied then

do a restart step by $-g_0$ direction; otherwise continue

Step 5: Set $\beta_k^{CD} = -\dfrac{g_k^T g_k}{g_{k-1}^T d_{k-1}}$

Step 6: Compute the parameter $\theta_k^{BH} = \dfrac{\mu\left|g_k^T d_{k-1}\right| + y_{k-1}^T d_{k-1}}{\left|g_{k-1}^T d_{k-1}\right|}$ .

Step 7: Compute the new search direction: $d_k = -\theta_k^{BH} g_k + \beta_k^{CD} d_{k-1}$

Step 8: Set k=k+1 and go to Step 2.

## 3.5 SCG-METHOD OF LIU AND JIANG ( 2012)

Liu and Jiang in 2012 did a minor modification to the CD-method in such away that the search directions generated are always descent directions. For any line search, their new method satisfies the sufficient descent condition. For more details see (Liu and Jiang, 2012).

Moreover, the authors proved that their method (Algorithm LJ) was globally convergent under the strong Wolfe line search. Their numerical results show that their spectral method was more effective for their selected test problems taken from the CUTE library. The search direction $d_k$ is defined by:

$$d_k = \begin{cases} -g_k, & if \quad k = 1 \\ -\theta_k g_k + \beta_k d_{k-1}, & if \quad k \geq 2 \end{cases} \tag{3.4}$$

where $\beta_k$ is specified by the following

$$\beta_k^{CD} = -\frac{g_k^T g_k}{g_{k-1}^T d_{k-1}} \tag{3.5}$$

$$\theta_k^{LJ} = 1 - \frac{g_k^T d_{k-1}}{g_{k-1}^T d_{k-1}} \tag{3.6}$$

### 3.5.1 Algorithm LJ

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \leq \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: Compute $g_k$ ; if $\|g_k\| \leq \varepsilon$, then stop ; otherwise continue

Step 4: If the restarting criterion $\left|g_k^T g_{k-1}\right| \geq 0.2\left|g_k^T g_k\right|$ is satisfied then

do a restart step by $-g_0$ direction; otherwise continue.

Step 5: Compute the parameters $\theta_k^{LJ}$ , $\beta_k^{LJ}$ from equations 3.6 and 3.5 respectively

Step 6: Compute the new search direction from equation 3.4

Step 7: Set k=k+1 and go to Step 2

## 3.6 SCG-METHOD OF LIU, DU AND WANG (2012)

Liu, Du and Wang (2012) proposed a mixed spectral CD-DY method for solving unconstrained optimization problems. Their proposed method (Algorithm LDW) generates descent directions and satisfies the global convergence property at each iteration. Their

numerical results show that their SCG-method is efficient when compared against the others. The new search directions are defined as follows:

$$d_k = \begin{cases} -g_k, & if \quad k = 1 \\ -\theta_k^{LDW} g_k + \beta_k^{LDW} d_{k-1}, & if \quad k \geq 2 \end{cases} \tag{3.7}$$

$$\beta_k^{LDW} = \begin{cases} \beta_k^{CD} + \min\,(0, \psi_k \beta_k^{CD}\,), & if \;\; g_k^T d_{k-1} \leq 0, \\ 0, & else, \end{cases} \tag{3.8}$$

$$\theta_k^{LDW} = 1 - \frac{g_k^T d_{k-1}}{g_{k-1}^T d_{k-1}} \tag{3.9}$$

$$\psi_k^{LDW} = -(1 - \frac{g_k^T d_{k-1}}{d_{k-1}(g_k - g_{k-1})}) \tag{3.10}$$

Under certain conditions, they proved the global convergence of their mixed spectral CD-DY, CG-method with the Wolfe line search. For more details see (Liu, et. al., 2014) and (Zhang, et. al., 2009 ).

Recently, Liu and Zeng (2015), proposed a new SCG-method to solve unconstrained optimization problems. Their algorithm has the following properties: (1) Their method satisfies the sufficient descent condition with any line searches condition; (2) their method possesses inherent properties when $\theta_k \geq 0$*;* (3) under the strong Wolfe line search their method was globally convergent. Preliminary numerical results show that their method was very efficient.

### 3.6.1 Algorithm LDW

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \leq \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: Compute $g_k$ ; if $\|g_k\| \leq \varepsilon$, then stop ; otherwise continue

Step 4: If the restarting criterion $\left|g_k^T g_{k-1}\right| \geq 0.2\left|g_k^T g_k\right|$ is satisfied then

do a restart step by $-g_0$ direction; otherwise continue

Step 5: Compute the parameters $\theta_k^{LDW}$ and $\psi_k^{LDW}$ from equations 3.9 and 3.10

Step 6: Compute the parameter $\beta_k^{LDW}$ from equation 3.8

Step 7: Compute the new search direction from equation 3.7

Step 8: Set k=k+1 and go to Step 2

## 3.7 SCG-METHOD OF LIVIERIS AND PINTELAS (2012)

Livieris and Pintelas (2012) proposed a new class of SCG-methods (Algorithm LP) that ensures sufficient descent directions independent of the accuracy of the line search and the global convergence property for general functions is established provided that the line search procedure satisfies the Wolfe conditions. Their numerical trials indicate that their proposed method is superior to the classical CG-methods. The search direction is defined by:

$$d_k = \begin{cases} g_k^T (z_{k-1} - \delta_k s_{k-1}) / \delta_k z_{k-1} d_{k-1}) \\ -\min \{ (g_k^T (z_{k-1} - \delta_k s_{k-1}) / \eta_k) / \delta_k z_{k-1} d_{k-1}) , \\ \qquad (c \| z_{k-1} - \delta_k s_{k-1} \| / \delta_k (z_{k-1}^T d_{k-1})^2 \, g_k^T d_{k-1}) \} \\ \qquad\qquad c \succ 0.25 \end{cases} \tag{3.11}$$

where

$$\begin{aligned} z_{k-1} &= y_{k-1} + (\theta_k / s_{k-1}^T u) u \\ \theta_k &= 6(f_{k-1} - f_k) + 3(g_k - g_{k-1})^T s_{k-1} \\ s_{k-1} &= x_k - x_{k-1} \\ y_{k-1} &= g_k - g_{k-1} \end{aligned} \tag{3.12}$$

*u is arbitrary vector*

### 3.7.1 Algorithm LP

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \leq \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: Compute $g_k$ ; if $\|g_k\| \le \varepsilon$, then stop ; otherwise continue

Step 4: Compute the new search direction from equations 3.11 and 3.12

Step 5: Set k=k+1 and go to Step 2

## 3.8 SCG-METHOD OF AL-BAYATI AND AL-KHAYAT (2013)

Recently, a new spectral CG-methods was proposed by Al-Bayati and Al-Khayat (2013) in this field. They tried to construct a descent direction:

$$d_k = -\theta_k^{BK} g_k + \beta_k^{CD} d_{k-1} \tag{3.13}$$

$$\beta_k^{CD} = -\frac{\|g_k\|^2}{g_{k-1}^T d_{k-1}} \tag{3.14}$$

$$\theta_k^{BK} = -\frac{d_{k-1}^T y_{k-1}}{d_{k-1}^T g_{k-1}} - \frac{d_{k-1}^T g_k g_k^T g_{k-1}}{\|g_k\|^2 d_{k-1}^T g_{k-1}} \tag{3.15}$$

This spectral CG-method (Algorithm BK) is reduced to the standard CD-method if the line search is exact. But generally the authors prefer to use the inexact line search (ILS), i.e., Wolfe line search. They have reported the performance of their SCG methods on a set of selected test problems.

In order to assess the reliability of their proposed method, they have tested their algorithm against standard (CD; FR and modified FR) methods using the same test problems.

### 3.8.1 Algorithm BK

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \le \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: If $(\|g_k\|_\infty \le 10^{-5} \text{ or } |\alpha_k g_k^T d_k| \le 10^{-10}|f_k|)$ is satisfied then stop; otherwise continue

Step 4: If the restarting criterion $|g_k^T g_{k-1}| \ge 0.2|g_k^T g_k|$ is satisfied then

do a restart step by $-g_0$ direction; otherwise continue

Step 5: Compute $\theta_k^{BK} = -\frac{d_{k-1}^T y_{k-1}}{d_{k-1}^T g_{k-1}} - \frac{d_{k-1}^T g_k g_k^T g_{k-1}}{\|g_k\|^2 d_{k-1}^T g_{k-1}}$

Step 6: Compute $\beta_k^{CD} = -\frac{\|g_k\|^2}{g_{k-1}^T d_{k-1}}$

Step 7: Compute $d_k = -\theta_k^{BK} g_k + \beta_k^{CD} d_{k-1}$ .

Step 8: Set k=k+1 and go to Step 2

## 3.9 SCG-METHOD OF GHANBARI, AHMAD, ALIAS AND ASKARIPOUR (2013)

A new nonlinear SCG-method for solving optimization problems was proposed by the Ghanbari and his friends (2013). Their method (Algorithm GAAA) is based on a mixed spectral Hestenes and Stiefel CG-method, which combines the advantages of the SHS-method, and the CD-method. The directions generated by the method are descent directions for the objective function. They have proven that their SCG-method with an Armijo-type line search was globally convergent. Numerical results show that their proposed method was promising.

### 3.9.1 Algorithm GAAA

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \le \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: If $(\|g_k\| \le \varepsilon)$ is satisfied then stop; otherwise continue

Step 4: If the restarting criterion $\left|g_k^T g_{k-1}\right| \geq 0.2\left|g_k^T g_k\right|$ is satisfied then

do a restart step by $-g_0$ direction; otherwise continue

Step 5: Compute $\theta_k^{GAAA} = 1 - \frac{g_k^T d_{k-1}}{g_{k-1}^T d_{k-1}}$

Step 6: Compute $\beta_k^{GAAA} = \begin{cases} \beta_k^{HS} & if\ g_k^T d_{k-1} \leq 0, \\ \beta_k^{CD} & if\ g_k^T d_{k-1} \leq 0 \end{cases}$

Step 7: Compute $d_k = -\theta_k^{GAAA} g_k + \beta_k^{GAAA} d_{k-1}$

Step 8: Set k=k+1 and go to Step 2

## 3.10 SCG-METHOD OF SUN AND LIU (2013)

SUN and LIU (2013), proposed a new spectral CG-algorithm combining the advantage of spectral-gradient method, a SCG-method for the global optimization was presented. Their method (Algorithm SL) has the property that the generated search direction is sufficiently descending without utilizing the used line search. Their numerical experiments show that their method was efficient and feasible.

### 3.10.1 Algorithm SL

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \leq \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: If $(\|g_k\| \leq \varepsilon)$ is satisfied then stop; otherwise continue

Step 4: If the restarting criterion $\left|g_k^T g_{k-1}\right| \geq 0.2\left|g_k^T g_k\right|$ is satisfied then

do a restart step by $-g_0$ direction; otherwise continue

Step 5: Compute $\theta_k^{SL} = \dfrac{((g_k^T g_{k-1})^2 g_k - g_{k-1}\|g_k\|^2\|g_{k-1}\|^2)^T d_{k-1}}{\|g_k\|^2\|g_{k-1}\|^4}$

Step 6: Compute $\beta_k^{SL} = (g_k^T g_{k-1})^2 / \|g_{k-1}\|^4$

Step 7: Compute $d_k = -\theta_k^{SL} g_k + \beta_k^{SL} d_{k-1}$

Step 8: Set k=k+1 and go to Step 2

## 3.11 SCG-METHOD OF LIU, ZHANG AND XU (2014)

Recently, a new SCG method (Algorithm LZX) used to prove the global convergence of the nonlinear CG-methods, the spectral method, was proposed by (Liu et al., 2014) and it is applied with sufficiently descending property. Under standard Wolfe line searches, the global convergence of their SCG algorithm has been proven for non-convex functions. They demonstrated their algorithms using 72 unconstrained problems. The search directions of their method are defined by:

$$d_k = \begin{cases} -g_k, & \text{if } k = 1 \\ -g_k + (g_k^T y_{k-1}/\eta_k) d_{k-1} - c_k (g_k^T d_{k-1}/\eta_k) y_{k-1}, & \text{if } k \geq 2 \end{cases} \tag{3.16}$$

C, is a small positive number

$$\eta_k^{LZX} = \begin{cases} y_{k-1}^T d_{k-1} & \text{if } \|g_{k-1}\|^2 \geq \mu \alpha_k \|d_{k-1}\|^2 \\ \alpha_k \|d_{k-1}\|^2, & \text{otherwise} \end{cases} \tag{3.17}$$

### 3.11.1 Algorithm LZX

Step 1: Take $x_0 \in R^n$ and the parameter $0 < \mu \leq \sigma < 1$. Compute $f(x_0)$ and

set $d_0 = -g_0$ for $k = 0$

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18.

Step 3: Compute $g_k$ ; if $\|g_k\| \leq \varepsilon$, then stop ; otherwise continue

Step 4: Compute the new search direction from equations 3.16 and 3.17

Step 5: Set k=k+1 and go to Step 2.

# CHAPTER 4

# A FAST SPECTRAL CG-METHOD

## 4.1 INTRODUCTION

Now, here we are concerned with the CG-methods for solving unconstrained nonlinear optimization problems. We know that the direction generated by a CG-method may not be a descent direction for certain objective function. Therefore, we are going to do some modifications to the standard CD-method such that the direction generated by the proposed Fast SCG-method hopefully provides descent directions for any objective function. In addition, the proposed fast SCG-method reduces to the standard CD-method if the line search is exact. We call the newly proposed SCG method as **Fast SCG** method. Under certain conditions, here, we are going to prove that the proposed Fast SCG-method with a strong Wolfe line search is globally convergent even if the objective function is non convex. We are also going to present some numerical results to prove the efficiency of the proposed Fast SCG-method.

## 4.2 THE FAST SCG-METHOD

In this section we have, first, to investigate how to determine sufficiently decreasing descent directions of any general objective function. Now, let $x_k$ be the current iterate and let $d_k$ be defined by:

$$d_k^{New} = -\theta_k^{New} g_k + \beta_k^{CD} d_{k-1}, \tag{4.3}$$

where, $\beta_k^{CD}$ is defined by equation 2.7 with the following fast spectral parameter defined in equation 4.2 which is a combination of new gradient and old gradient and old search

directions. The main advantages for the this type of search direction are employing **the descent** and **conjugacy properties**, therefore using these two different types of theoretical properties in the above new search direction yields the unknown parameter $\theta_k$ .

$$\theta_k^{New} = 1 - (\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}})(\frac{d_{k-1}^T g_k}{\|g_k\|^2}) - \frac{d_{k-1}^T g_k}{2 g_{k-1}^T g_{k-1}} \tag{4.2}$$

This is the proposed fast SCG-method designed for solving a number of some complicated nonlinear unconstrained optimization problems which is reduced to the standard CD-method if the line search is exact. Here, we use Wolfe's inexact line search to get better results. We have to first prove that $d_k$ is a sufficiently descending direction.

### 4.3 LEMMA (DOWNHILL DIRECTION)

Suppose that $d_k^{New}$ which is defined by equations 4.1 and 4.2 is used with the parameter $\alpha_k$ of the strong Wolfe conditions defined in equations 1.17 and 1.18 respectively, with $\sigma_k < 0.5$. Then:

$$g_k^T d_k \le -c_1 \left| g_k^T g_k \right| \tag{4.3}$$

holds for any $k \ge 0$

**Proof**

For initial $k$, we have the following:

$$d_0^T g_0 = -\|g_0\|^2 \tag{4.4}$$

For the second step, suppose that the condition 4.3 is true for all values of k-1; i.e.

$$d_{k-1}^T g_{k-1} = -\|g_{k-1}\|^2 \tag{4.5}$$

Using induction scheme, condition 4.3 must be true for all values of k, i.e.

$$g_k^T d_k = -\theta_k^{New} \|g_k\|^2 + \beta_k^{CD} g_k^T d_{k-1} \tag{4.6}$$

From equations 4.1 and 4.2, it is follows that:

$$g_k^T d_k = -\left[1-(\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}})(\frac{d_{k-1}^T g_k}{\|g_k\|^2}) - \frac{d_{k-1}^T g_k}{2g_{k-1}^T g_{k-1}}\right]\|g_k\|^2 - \frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}} g_k^T d_{k-1} \tag{4.7}$$

$$g_k^T d_k = -\left[1-(\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}})(\frac{d_{k-1}^T g_k}{\|g_k\|^2}) - \frac{d_{k-1}^T g_k}{2g_{k-1}^T g_{k-1}} + \frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right]\|g_k\|^2$$

$$g_k^T d_k = -\left[1-\frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} - \frac{d_{k-1}^T g_k}{2g_{k-1}^T g_{k-1}} + \frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right]\|g_k\|^2$$

$$g_k^T d_k = -\left[1-\frac{d_{k-1}^T g_k}{2g_{k-1}^T g_{k-1}}\right]\|g_k\|^2 \tag{4.8}$$

Now, from equation 1.18, we have the following:

$$d_{k-1}^T g_k \le -\sigma d_{k-1}^T g_{k-1}; \quad \sigma \text{ is a positive parameter} \tag{4.9}$$

Therefore, using equations 4.9; 4.8 becomes:

$$g_k^T d_k \le -\left[1+\frac{\sigma d_{k-1}^T g_{k-1}}{2g_{k-1}^T g_{k-1}}\right]\|g_k\|^2 \tag{4.10}$$

From equation 4.5 we have:

$$g_k^T d_k \le -\left[1+\frac{\sigma g_{k-1}^T g_{k-1}}{2g_{k-1}^T g_{k-1}}\right]\|g_k\|^2 \tag{4.11}$$

Hence:

$$g_k^T d_k \le -\left[1+\frac{\sigma}{2}\right]\|g_k\|^2 \tag{4.12}$$

this implies:

$$g_k^T d_k \le -c\|g_k\|^2 \quad with \;\; c=[1+\sigma/2]>0 \tag{4.13}$$

Hence, the Lemma is true.

From **Lemma 4.3**, and for the new descent directions of the proposed fast CG-method, using exact line searches, we have:

$$\theta_k^{New} = 1-(\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}})(\frac{d_{k-1}^T g_k}{\|g_k\|^2}) - \frac{d_{k-1}^T g_k}{2g_{k-1}^T g_{k-1}} = 1 \tag{4.14}$$

and the proposed Fast SCG-method is reduced to the standard CD-method. Next, for the Fast SCG-algorithm defined by equations 4.1-4.2, and with inexact line search, we are going to study the effect of the modified line search.

## 4.4 A MODIFIED LINE SEARCH PROCESS

This modified type line search scheme was originally established by (Andrei, 2009b). In this accelerating process, the new estimation of the minimum point is computed as follows:

$$x_k = x_{k-1} + \lambda_k \alpha_k d_{k-1} \tag{4.15}$$

Let:

$$z = x_k + \alpha_k d_k,\; y_k = g_k - g_z,\;\; g_z = \nabla f(z).$$

Compute: $a_k = \alpha_k g_k^T d_k$, $b_k = -\alpha_k y_k^T d_k$, (4.16)

If $b_k \neq 0$, then compute, $\lambda_k = -\frac{a_k}{b_k}$ and use (4.15); otherwise update

$$x_k = x_{k-1} + \alpha_k d_{k-1}.$$

Hence, using the above acceleration scheme algorithm defined by equations 4.15 and 4.16 in equation 4.1:

## 4.5 ALGORITHM FAST SCG

Below are the outlines of the proposed fast SCG-method:

Step 1: Take $x_0 \in R^n$ ; set the parameters $0 < \delta \leq \sigma < 1$

Compute $f(x_0)$ and $g_0 = \nabla f(x_0)$; set $d_0 = -g_0$ for $k = 0$

Step 2: Compute $\alpha_k$ satisfying Wolfe conditions 1.17 and 1.18

Compute $z = x_k + \alpha_k d_k$, $y_k = g_k - g_z$, $g_z = \nabla f(z)$.

Compute, $a_k = \alpha_k g_k^T d_k$, $b_k = -\alpha_k y_k^T d_k$, if $b_k \neq 0$ compute, $\lambda_k = -\frac{a_k}{b_k}$

Update $x_k = x_{k-1} + \lambda_k \alpha_k d_{k-1}$; otherwise use $x_k = x_{k-1} + \alpha_k d_{k-1}$

Step 3: If $\|g_k\|_\infty \leq \varepsilon$ or number of iterations (NOI) exceeds 1000, then

the iterations are stopped, $\varepsilon$ is a small positive number, and $\varepsilon = 0.0000001$

Step 4: If the restarting criterion $\left|g_k^T g_{k-1}\right| \geq 0.2\left|g_k^T g_k\right|$ is satisfied then

do a restart step by negative gradient direction; otherwise continue

Step 5: Compute the parameters:

$$\left( \beta_k^{CD} = - \frac{g_k^T g_k}{d_{k-1}^T g_{k-1}} \right)$$

$$\theta_k^{New} = 1 - (\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}})(\frac{d_{k-1}^T g_k}{\|g_k\|^2}) - \frac{d_{k-1}^T g_k}{2 g_{k-1}^T g_{k-1}}$$

Step 6: Compute the new spectral search direction

$$d_k = -\theta_k^{New} g_k + \beta_k^{CD} d_{k-1}$$

Step 7: Set k=k+1 and go to Step 2.

It is well known that, if the nonlinear objective function is bounded along the direction $d_k$, then $\alpha_k$ satisfies the Wolfe line search conditions 1.18.

In our proposed fast SCG-algorithm, when the Powell restarting condition defined by (2.25) is satisfied, then we restart the algorithm with the negative gradient. Hence, conditions 1.18 and 2.25 are sufficient to prove the global convergence of the new proposed SCG-algorithm.

## 4.6 SPEED OF CONVERGENCE OF FAST SCG METHOD

To study the speed of convergence of the proposed fast SCG-method defined by equations 4.1-4.2. we first state the following common and general assumptions, that will be used in the proof of the convergence of any CG-method.

### 4.6.1 Assumptions

(i) The level set $S = \{x : x \in R^n, f(x) \leq f(x_1)\}$, which satisfies the descent property, is bounded below, where $x_1$ is the starting point.

(ii) In a neighborhood $\Omega$ of the level set S, the objective function $f$, is continuously differentiable and its gradient $g$ is Lipschitz continuously; namely, there exists a constant $L \geq 0$ such that:

$$\| g(x) - g(x_k) \| \leq L \| x - x_k \|, \forall x, x_k \in \Omega \qquad (4.17)$$

From Assumption (4.6.1., i) there exists a positive constant D such that:

$$D = \max\{\|x - x_k\|, \forall x, x_k \in S\} \tag{4.18}$$

where D is the diameter of the neighborhood Ω.

From Assumption (4.6.1., ii), there exists a constant $\Gamma \geq 0$, such that:

$$\|g(x)\| \leq \Gamma, \ \forall x \in S \tag{4.19}$$

## 4.7 A NEW THEOREM ON CONVERGENCE OF FAST SCG METHOD

Suppose that the proposed fast SCG-algorithm which is defined by equations 4.1 and 4.2 satisfies both strong Wolfe line search conditions 1.18 and the assumptions of section (4.6.1), for both parts (i) and (ii) , with $\sigma_k < 0.5$ , then it yields:

$$\lim_{k\to\infty} \inf \|g_k\| = 0 \tag{4.20}$$

**Proof**

Assume that there exists a positive constant $\psi \succ o$ such that:

$$\|g_k\| \geq \psi \tag{4.21}$$

then, for all k, from (4.1), it follows that:

$$\|d_k\|^2 = d_k^T d_k$$

$$= (-\theta_k^{New} g_k + \beta_k^{CD} d_{k-1})(-\theta_k^{New} g_k + \beta_k^{CD} d_{k-1})$$

$$= (\theta_k^{New})^2 \|g_k\|^2 - 2\theta_k^{New} \beta_k^{CD} d_{k-1}^T g_k + (\beta_k^{CD})^2 \|d_{k-1}\|^2$$

but the new search direction is defined by

$$d_k = -\theta_k^{New} g_k + \beta_k^{CD} d_{k-1}$$

hence:

$$= (\theta_k^{New})^2 \|g_k\|^2 - 2\theta_k^{New}(d_k^T + \theta_k^{New} g_k^T) g_k + (\beta_k^{CD})^2 \|d_{k-1}\|^2$$

$$= (\theta_k^{New})^2 \|g_k\|^2 - 2\theta_k^{New} g_k^T d_k - 2(\theta_k^{New})^2 \|g_k\|^2 + (\beta_k^{CD})^2 \|d_{k-1}\|^2$$

$$= (\theta_k^{New})^2 \|d_{k-1}\|^2 - 2\theta_k^{New} g_k^T d_k - (\theta_k^{New})^2 \|g_k\|^2 \qquad (4.22)$$

In the last equality 4.22, if we divide both sides of the above by $(g_k^T d_k)^2$, then from equations 4.3, 4.17, and 4.22 we obtain:

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} = \frac{(\beta_k^{CD})^2 \|d_{k-1}\|^2 - 2\theta_k^{New} g_k^T d_k - (\theta_k^{New})^2 \|g_k\|^2}{(g_k^T d_k)^2} \qquad (4.23)$$

$$= \left[ \frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}} \right]^2 \frac{\|d_{k-1}\|^2}{(g_k^T d_k)^2} - (\theta_k^{New})^2 \frac{\|g_k\|^2}{(g_k^T d_k)^2} - 2\theta_k^{New} \frac{1}{(g_k^T d_k)} \qquad (4.24)$$

We have, from equation 4.13

$$g_k^T d_k \le -c \|g_k\|^2$$

Equation 4.24 becomes:

$$\le \left[ \frac{\|g_k\|^2}{c\|g_{k-1}\|^2} \right]^2 \frac{\|d_{k-1}\|^2}{c^2 \|g_k\|^4} - (\theta_k^{New})^2 \frac{\|g_k\|^2}{c^2 \|g_k\|^4} - 2\theta_k^{New} \frac{1}{c\|g_k\|^2} \qquad (4.25)$$

Reformulating the above inequality, yields:

$$\le \frac{\|d_{k-1}\|^2}{c^4 \|g_{k-1}\|^4} - \{(\theta_k^{New} \frac{\|g_k\|}{c \|g_k\|^2})^2 + 2\theta_k^{New} \frac{1}{c\|g_k\|^2}\} + \frac{1}{\|g\|^2} - \frac{1}{\|g\|^2}$$

$$\le \frac{\|d_{k-1}\|^2}{c^4 \|g_{k-1}\|^4} - \{(\theta_k^{New} \frac{\|g_k\|}{c \|g_k\|^2})^2 + 2\theta_k^{New} \frac{1}{c\|g_k\|^2} + \frac{1}{\|g\|^2}\} + \frac{1}{\|g\|^2}$$

$$\leq \frac{\|d_{k-1}\|^2}{c^4 \|g_{k-1}\|^4} - \left[ \theta_k^{New} \frac{\|g_k\|}{c\|g_k\|^2} + \frac{1}{\|g_k\|} \right]^2 + \frac{1}{\|g_k\|^2}$$

$$\leq \frac{\|d_{k-1}\|^2}{c^4 \|g_{k-1}\|^4} + \frac{1}{\|g_k\|^2} \tag{4.26}$$

and the mid term of equation 4.26 is always positive followed by negative sign

Now, from equation 4.17 and the definition of the norm of the search direction d:

$$\|d_1\|^2 = -d_1^T g_1 = \|g_1\|^2 \tag{4.27}$$

We have:

$$\frac{\|d_k\|^2}{(d_k^T g_k)^2} \leq \sum_{i=1}^{k} \frac{1}{\|g_i\|^2} \tag{4.28}$$

Therefore, from equations 4.28 and 4.21 we have:

$$\frac{(d_k^T g_k)^2}{\|d_k\|^2} \geq \frac{\psi}{k} \tag{4.29}$$

which indicates:

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \Rightarrow +\infty \tag{4.30}$$

This is a contradiction to our assumption in equation 4.20.

Hence the proof of this new theorem is complete and the proposed **Fast SCG** has a global convergence property.

# CHAPTER 5

# EXPERIMENTAL RESULTS

## 5.1 INTRODUCTION

The main work of this section is to report the performance of the proposed **Fast SCG** on a set of 55-complicated nonlinear unconstrained test problems. The original codes were written by (Andrei, 2008) in the FORTRAN language and in double precision arithmetic and modified in this work to make it suitable to evaluate all algorithms considered in this thesis. All the tests were performed on a PC with CPU intel core i7, RAM 8G, HD Graphics 3000 approximate and total memory: 1696 MB. Our experiments were performed on the selected set of complicated nonlinear unconstrained problems that have second derivatives available. These test problems are contributed in CUTE (Bongartz et al., 1995) and their details are given in the Appendix. For each test function we have considered four different numerical experiments with a number of variables n= 100,400,700 and 1000. All these methods terminate when the following stopping criterion is met.

$$\|g_k\|_\infty \leq 10^{-5} \tag{5.1}$$

We also force these routines to be stopped if the number of iterations (NOI) exceed 1000 or the number of functions (NOF) reaches 2000 without achieving the minimum. In all these tables (n) indicates as a dimension of the problem; (NOI) number of iterations; and (NOF) number of function evaluations.

## 5.2 NUMERICAL RESULTS

In **Table 5.1** we assess the reliability of the standard CD-method, against the standard (FR and PR) classical CG-methods using standard Wolfe conditions as a line search subroutine and using the same group of our test problems.

In **Table 5.2** we have compared the percentage performance of the standard CD-method against (FR and PR) CG-methods, using the standard Wolfe conditions as a line search subroutine. Now, taking over all the tools (NOI and NOF) as 100% , in order to summarize our numerical results, we have concentrated only on a total of four different dimensions n= 100, 400,700 and 1000, for all tools used in these comparisons.

In **Tables 5.3** we assess the reliability of **Fast SCG** against some recent spectral CG-algorithms like (BK and LDW) SCG-methods using both standard and modified Wolfe conditions respectively as a line search subroutine and using the same set of test problems.

In **Table 5.4** we have compared the percentage performance of **Fast SCG** against (BK and LDW) SCG algorithms using both standard and modified Wolfe conditions respectively as a line search subroutine. Now, taking over all the tools (NOI and NOF) as 100%, in order to summarize our numerical results, we have concentrated only on the total of four different dimensions n= 100, 400,700 and 1000, for all tools used in these comparisons.

In **Tables 5.5** we assess the reliability of **Fast SCG** against some other spectral CG-algorithms like (BH and BA) SCG-methods using both standard and modified Wolfe conditions respectively as a line search subroutine and using the same set of test problems.

In **Table 5.6** we have compared the percentage performance of **Fast SCG** against (BH and BA) SCG algorithms using both standard and modified Wolfe conditions respectively as a line search subroutine. Now, taking over all the tools (NOI and NOF) as 100%, in order to summarize our numerical results, we have concentrated only on the total of four different dimensions n= 100, 400,700 and 1000, for all tools used in these comparisons.

In **Tables 5.7** we assess the reliability of **Fast SCG** against all other spectral CG-algorithms; namely, (LDW, BK, BH and BA) SCG-methods using both standard and

modified Wolfe condition respectively s as a line search subroutine and using the same set of test problems.

In **Table 5.8,** we have compared the percentage performance of **Fast SCG** against (LDW, BK, BH and BA) SCG algorithms. Now, taking over all the time tool CPU as 100%, in order to summarize our numerical results, we have concentrated only on the total of four different dimensions n= 100, 400,700 and 1000, for all tools used in these comparisons.

In **Table 5.9,** we have compared the percentage performance of **Fast SCG** against both standard and spectral CG-algorithms mentioned in this thesis; namely ( CD, FR, PR, LDW, BK, BH and BA) algorithms. Now, taking over all the tool NOI as 100%, in order to summarize our numerical results, we have concentrated only on the total of four different dimensions n= 100, 400,700 and 1000, for all tools used in these comparisons.

In **Table 5.10,** we have compared the percentage performance of **Fast SCG** against both standard and spectral CG-algorithms mentioned in this thesis; namely ( CD, FR, PR, LDW, BK, BH and BA) algorithms. Now, taking over all the tool NOF as 100%, in order to summarize our numerical results, we have concentrated only on the total of four different dimensions n= 100, 400,700 and 1000, for all tools used in these comparisons.

**Table 5.1 Comparisons between FR, PR & CD Algorithms**

| No. of Test Functions | PR | | PR | | CD | |
|---|---|---|---|---|---|---|
| | NOI | NOF | NOI | NOF | NOI | NOF |
| 1 | 235 | 389 | 268 | 470 | 265 | 426 |
| 2 | 23 | 52 | 23 | 52 | 23 | 52 |
| 3 | 47 | 126 | 54 | 129 | 47 | 126 |
| 4 | 53 | 148 | 53 | 148 | 53 | 148 |
| 5 | 72 | 151 | 80 | 163 | 72 | 149 |
| 6 | 17 | 46 | 17 | 46 | 17 | 46 |
| 7 | 26 | 61 | 24 | 59 | 25 | 60 |
| 8 | 29 | 71 | 30 | 73 | 29 | 71 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | 20 | 55 | 20 | 55 | 20 | 55 |
| 10 | 22 | 58 | 22 | 58 | 22 | 58 |
| 11 | 22 | 50 | 17 | 40 | 22 | 50 |
| 12 | 163 | 329 | 164 | 331 | 163 | 329 |
| 13 | 26 | 95 | 38 | 115 | 26 | 95 |
| 14 | 7 | 25 | 7 | 25 | 7 | 25 |
| 15 | 30 | 90 | 30 | 90 | 30 | 90 |
| 16 | 29 | 59 | 29 | 59 | 29 | 59 |
| 17 | 33 | 73 | 33 | 74 | 33 | 72 |
| 18 | 29 | 78 | 30 | 79 | 29 | 78 |
| 19 | 18 | 33 | 18 | 33 | 18 | 33 |
| 20 | 4 | 12 | 4 | 12 | 4 | 12 |
| 21 | 1438 | 2320 | 1575 | 2122 | 1699 | 2556 |
| 22 | 25 | 69 | 25 | 69 | 25 | 69 |
| 23 | 69 | 200 | 105 | 230 | 68 | 194 |
| 24 | 4 | 12 | 4 | 12 | 4 | 12 |
| 25 | 7 | 16 | 7 | 16 | 7 | 16 |
| 26 | 4 | 8 | 4 | 8 | 4 | 8 |
| 27 | 34 | 59 | 37 | 63 | 34 | 59 |
| 28 | 19 | 65 | 19 | 65 | 19 | 65 |
| 29 | 23 | 54 | 23 | 54 | 23 | 54 |
| 30 | 31 | 52 | 31 | 52 | 31 | 52 |
| 31 | 25 | 54 | 25 | 54 | 25 | 54 |
| 32 | 24 | 48 | 22 | 41 | 24 | 48 |
| 33 | 58 | 135 | 58 | 129 | 58 | 135 |
| 34 | 73 | 151 | 81 | 173 | 75 | 154 |
| 35 | 23 | 59 | 23 | 59 | 23 | 59 |
| 36 | 17 | 46 | 17 | 46 | 17 | 46 |
| 37 | 24 | 36 | 24 | 36 | 24 | 36 |
| 38 | 23 | 62 | 23 | 62 | 23 | 62 |
| 39 | 30 | 60 | 31 | 61 | 30 | 60 |
| 40 | 59 | 138 | 58 | 140 | 55 | 132 |
| 41 | 19 | 55 | 19 | 55 | 19 | 55 |
| 42 | 41 | 95 | 47 | 111 | 41 | 95 |
| 43 | 20 | 44 | 20 | 44 | 20 | 44 |
| 44 | 57 | 134 | 59 | 141 | 57 | 134 |
| 45 | 10 | 35 | 10 | 35 | 10 | 35 |
| 46 | 13 | 43 | 13 | 43 | 13 | 43 |
| 47 | 7 | 27 | 7 | 27 | 7 | 27 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 48 | 16 | 72 | 16 | 72 | 16 | 72 |
| 49 | 33 | 73 | 33 | 74 | 33 | 72 |
| 50 | 24 | 64 | 24 | 64 | 24 | 64 |
| 51 | 117 | 250 | 118 | 236 | 120 | 266 |
| 52 | 4 | 12 | 4 | 12 | 4 | 12 |
| 53 | 77 | 154 | 74 | 154 | 76 | 153 |
| 54 | 8 | 16 | 8 | 16 | 8 | 16 |
| 5 | 21 | 60 | 21 | 60 | 21 | 60 |
| **TOTAL** | **3382** | **6779** | **3626** | **6747** | **3671** | **7053** |

**Table 5.2 Running Percentage Performance of CD against FR and PR**

| **Tools** | **CD** | **FR** | **PR** |
|---|---|---|---|
| **NOI** | 100% | 92% | 98% |
| **NOF** | 100% | 96% | 95% |

Clearly, this table shows that the CD-method is the worse, because it does not generate sufficiently descending search directions in general. For this reason many authors deal with improving the search direction of this method by implementing spectral CG-methods (as we will show in the next table). From this table we can conclude that both (FR and PR) are very close to each other, while CD needs some an additional necessary

conditions to become an effective method. The numerical results of this table indicates that FR saves about 8% NOI and 4% NOF to complete solving this set of selected complicated nonlinear test problems, while PR saves about 2% NOI and 5% NOF to complete solving this set of selected complicated nonlinear test problems.

**Table 5.3 Comparisons between BK, LDW & Fast SCG Algorithms**

| No. of Test Functions | BK | | LDW | | Fast SCG | |
|---|---|---|---|---|---|---|
| | NOI | NOF | NOI | NOF | NOI | NOF |
| 1 | 154 | 332 | 122 | 331 | 48 | 56 |
| 2 | 23 | 52 | 23 | 52 | 24 | 72 |
| 3 | 55 | 148 | 59 | 148 | 39 | 79 |
| 4 | 53 | 148 | 53 | 148 | 26 | 103 |
| 5 | 69 | 135 | 67 | 132 | 69 | 77 |
| 6 | 17 | 46 | 17 | 46 | 14 | 25 |
| 7 | 22 | 53 | 22 | 53 | 24 | 82 |
| 8 | 29 | 71 | 29 | 71 | 32 | 62 |
| 9 | 20 | 55 | 20 | 55 | 18 | 51 |
| 10 | 22 | 58 | 22 | 58 | 9 | 20 |
| 11 | 24 | 51 | 24 | 51 | 15 | 24 |
| 12 | 166 | 333 | 164 | 329 | 72 | 101 |
| 13 | 27 | 105 | 35 | 113 | 8 | 20 |
| 14 | 7 | 25 | 7 | 25 | 7 | 22 |
| 15 | 30 | 90 | 30 | 90 | 29 | 77 |
| 16 | 29 | 59 | 29 | 59 | 27 | 83 |
| 17 | 34 | 76 | 36 | 78 | 20 | 52 |
| 18 | 26 | 72 | 26 | 72 | 24 | 83 |
| 19 | 18 | 33 | 18 | 33 | 14 | 38 |
| 20 | 4 | 12 | 4 | 12 | 8 | 16 |
| 21 | 1113 | 2017 | 1158 | 2058 | 85 | 132 |
| 22 | 24 | 67 | 24 | 67 | 27 | 45 |
| 23 | 59 | 173 | 76 | 205 | 22 | 39 |
| 24 | 4 | 12 | 4 | 12 | 5 | 19 |
| 25 | 7 | 16 | 7 | 16 | 7 | 17 |
| 26 | 4 | 8 | 4 | 8 | 16 | 47 |
| 27 | 27 | 52 | 27 | 46 | 42 | 93 |
| 28 | 19 | 65 | 19 | 65 | 14 | 45 |
| 29 | 23 | 54 | 23 | 54 | 16 | 26 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 30 | 31 | 52 | 31 | 52 | 22 | 33 |
| 31 | 25 | 54 | 25 | 54 | 23 | 34 |
| 32 | 23 | 46 | 24 | 48 | 25 | 36 |
| 33 | 56 | 124 | 55 | 123 | 35 | 46 |
| 34 | 78 | 167 | 79 | 164 | 62 | 96 |
| 35 | 23 | 59 | 23 | 59 | 18 | 53 |
| 36 | 17 | 46 | 17 | 46 | 14 | 25 |
| 37 | 28 | 44 | 28 | 44 | 28 | 40 |
| 38 | 23 | 62 | 23 | 62 | 30 | 62 |
| 39 | 29 | 57 | 29 | 57 | 23 | 58 |
| 40 | 60 | 138 | 58 | 134 | 27 | 92 |
| 41 | 19 | 55 | 19 | 55 | 16 | 26 |
| 42 | 44 | 98 | 43 | 99 | 36 | 62 |
| 43 | 20 | 44 | 20 | 44 | 8 | 20 |
| 44 | 62 | 142 | 64 | 149 | 34 | 45 |
| 45 | 10 | 35 | 10 | 35 | 18 | 26 |
| 46 | 13 | 43 | 13 | 43 | 11 | 44 |
| 47 | 7 | 27 | 7 | 27 | 11 | 19 |
| 48 | 16 | 72 | 16 | 72 | 10 | 22 |
| 49 | 34 | 76 | 36 | 78 | 22 | 48 |
| 50 | 24 | 65 | 24 | 65 | 22 | 41 |
| 51 | 95 | 219 | 99 | 222 | 74 | 170 |
| 52 | 4 | 12 | 4 | 12 | 4 | 16 |
| 53 | 73 | 146 | 76 | 158 | 75 | 120 |
| 54 | 8 | 16 | 8 | 16 | 8 | 20 |
| 55 | 21 | 60 | 21 | 60 | 15 | 27 |
| Total | 2952 | 6377 | 3001 | 6465 | 1432 | 2887 |

**Table 5.4 Running Percentage Performance of Fast SCG against BK and LDW**

| Tools | LDW | BK | Fast SCG |
|---|---|---|---|
| **NOI** | 100% | 98% | 47% |
| **NOF** | 100% | 98% | 44% |

The numerical results of this table indicates that the **Fast SCG** algorithm saves about 53% NOI and 56% NOF compared with the LDW-algorithm and about 51% NOI and 54% NOF against the spectral BK-algorithm to complete solving the set of selected complicated nonlinear test problems.

**Table 5.5 Comparisons between BA, BH & Fast SCG Algorithms**

| No. of Test Functions | BA | | BH | | Fast SCG | |
|---|---|---|---|---|---|---|
| | **NOI** | **NOF** | **NOI** | **NOF** | **NOI** | **NOF** |
| 1 | 229 | 395 | 232 | 413 | 48 | 56 |
| 2 | 23 | 52 | 23 | 52 | 24 | 72 |
| 3 | 45 | 120 | 64 | 158 | 39 | 79 |
| 4 | 53 | 148 | 53 | 148 | 26 | 103 |
| 5 | 76 | 157 | 77 | 150 | 69 | 77 |
| 6 | 17 | 46 | 17 | 46 | 14 | 25 |
| 7 | 25 | 60 | 22 | 48 | 24 | 82 |
| 8 | 29 | 71 | 29 | 71 | 32 | 62 |
| 9 | 20 | 55 | 20 | 55 | 18 | 51 |
| 10 | 22 | 58 | 22 | 58 | 9 | 20 |
| 11 | 22 | 50 | 22 | 49 | 15 | 24 |
| 12 | 161 | 335 | 158 | 326 | 72 | 101 |
| 13 | 30 | 95 | 51 | 135 | 8 | 20 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 14 | 7 | 25 | 7 | 25 | 7 | 22 |
| 15 | 30 | 90 | 30 | 90 | 29 | 77 |
| 16 | 29 | 59 | 29 | 59 | 27 | 83 |
| 17 | 33 | 73 | 31 | 67 | 20 | 52 |
| 18 | 29 | 78 | 28 | 78 | 24 | 83 |
| 19 | 18 | 33 | 18 | 33 | 14 | 38 |
| 20 | 4 | 12 | 4 | 12 | 8 | 16 |
| 21 | 1433 | 2332 | 1357 | 2325 | 85 | 132 |
| 22 | 25 | 69 | 25 | 69 | 27 | 45 |
| 23 | 112 | 235 | 59 | 183 | 22 | 39 |
| 24 | 4 | 12 | 4 | 12 | 5 | 19 |
| 25 | 7 | 16 | 7 | 16 | 7 | 17 |
| 26 | 4 | 8 | 4 | 8 | 16 | 47 |
| 27 | 34 | 59 | 29 | 52 | 42 | 93 |
| 28 | 19 | 65 | 19 | 65 | 14 | 45 |
| 29 | 23 | 54 | 23 | 54 | 16 | 26 |
| 30 | 31 | 52 | 31 | 52 | 22 | 33 |
| 31 | 25 | 54 | 25 | 54 | 23 | 34 |
| 32 | 24 | 48 | 24 | 48 | 25 | 36 |
| 33 | 61 | 135 | 58 | 125 | 35 | 46 |
| 34 | 76 | 157 | 81 | 162 | 62 | 96 |
| 35 | 23 | 59 | 23 | 59 | 18 | 53 |
| 36 | 17 | 46 | 17 | 46 | 14 | 25 |
| 37 | 24 | 36 | 56 | 104 | 28 | 40 |
| 38 | 23 | 62 | 23 | 62 | 30 | 62 |
| 39 | 30 | 60 | 29 | 57 | 23 | 58 |
| 40 | 54 | 135 | 56 | 125 | 27 | 92 |
| 41 | 19 | 55 | 19 | 55 | 16 | 26 |
| 42 | 41 | 95 | 48 | 107 | 36 | 62 |
| 43 | 20 | 44 | 20 | 44 | 8 | 20 |
| 44 | 56 | 132 | 52 | 116 | 34 | 45 |
| 45 | 10 | 35 | 10 | 35 | 18 | 26 |
| 46 | 13 | 43 | 13 | 43 | 11 | 44 |
| 47 | 7 | 27 | 7 | 27 | 11 | 19 |
| 48 | 16 | 72 | 16 | 72 | 10 | 22 |
| 49 | 33 | 73 | 31 | 67 | 22 | 48 |
| 50 | 24 | 64 | 24 | 64 | 22 | 41 |
| 51 | 119 | 252 | 93 | 210 | 74 | 170 |
| 52 | 4 | 12 | 4 | 12 | 4 | 16 |
| 53 | 74 | 149 | 74 | 151 | 75 | 120 |
| 54 | 8 | 16 | 8 | 16 | 8 | 20 |
| 55 | 21 | 60 | 5 | 14 | 15 | 27 |
| Total | 3416 | 6835 | 3311 | 6784 | 1432 | 2887 |

**Table 5.6 Running Percentage Performance of Fast SCG against BA and BH**

| Tools | BA | BH | Fast SCG |
|---|---|---|---|
| **NOI** | 100% | 97% | 42% |
| **NOF** | 100% | 99% | 42% |

Again our numerical results in this table indicates that the **Fast SCG** algorithm saves about 58% for both (NOI and NOF) compared with the BA-algorithm and about 55% NOI and 57% NOF when compared against the BH-algorithm to complete solving the set of 55-selected complicated unconstrained nonlinear test problems.

**Table 5.7 Comparisons of CPU-Times (in Seconds) Between All SCG-Algorithms**

| N. T. Fs. | LDW | BK | BA | BH | Fast SCG |
|---|---|---|---|---|---|
| 1 | 0,09 | 0,03 | 0,04 | 0,08 | 0,01 |
| 2 | 0,1 | 0,02 | 0,01 | 0,06 | 0,04 |
| 3 | 0,04 | 0,01 | 0,01 | 0,03 | 0 |
| 4 | 0,03 | 0,01 | 0 | 0,02 | 0 |
| 5 | 0,07 | 0,02 | 0,02 | 0,05 | 0,02 |
| 6 | 0,01 | 0,01 | 0 | 0,01 | 0 |
| 7 | 0,02 | 0,01 | 0,01 | 0,02 | 0,01 |

| | | | | | |
|---|---|---|---|---|---|
| 8 | 0,03 | 0,01 | 0 | 0,03 | 0,01 |
| 9 | 0 | 0 | 0 | 0,01 | 0 |
| 10 | 0 | 0 | 0 | 0,01 | 0,01 |
| 11 | 0,04 | 0,01 | 0,01 | 0,02 | 0 |
| 12 | 0,06 | 0,01 | 0,02 | 0,09 | 0,01 |
| 13 | 0,01 | 0,01 | 0 | 0,01 | 0 |
| 14 | 0,02 | 0 | 0,01 | 0,04 | 0 |
| 15 | 0,01 | 0 | 0 | 0,01 | 0,01 |
| 16 | 0,04 | 0,01 | 0,01 | 0,04 | 0,01 |
| 17 | 0,05 | 0,01 | 0 | 0,04 | 0,01 |
| 18 | 0,02 | 0,01 | 0,01 | 0,02 | 0,01 |
| 19 | 0,02 | 0,01 | 0 | 0,02 | 0 |
| 20 | 0,01 | 0 | 0 | 0 | 0 |
| 21 | 0,28 | 0,06 | 0,09 | 0,3 | 0,01 |
| 22 | 0 | 0 | 0 | 0 | 0,01 |
| 23 | 0,12 | 0,03 | 0,03 | 0,12 | 0,01 |
| 24 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 0,01 |
| 27 | 0,01 | 0 | 0,01 | 0 | 0,01 |
| 28 | 0,02 | 0 | 0 | 0,02 | 0,01 |
| 29 | 0,02 | 0 | 0,01 | 0,02 | 0 |
| 30 | 0,02 | 0 | 0 | 0,01 | 0,01 |
| 31 | 0,02 | 0,01 | 0,01 | 0,02 | 0 |
| 32 | 0,03 | 0 | 0 | 0,02 | 0,01 |
| 33 | 0,58 | 0,15 | 0,16 | 0,6 | 0,08 |
| 34 | 0,03 | 0 | 0 | 0,03 | 0 |
| 35 | 0,01 | 0,01 | 0 | 0 | 0 |
| 36 | 0,01 | 0 | 0,01 | 0,01 | 0,01 |
| 37 | 0,01 | 0 | 0 | 0,01 | 0 |
| 38 | 0,01 | 0 | 0 | 0,01 | 0 |
| 39 | 0,02 | 0 | 0,01 | 0,02 | 0,01 |
| 40 | 0,06 | 0,02 | 0,01 | 0,05 | 0,01 |
| 41 | 0,01 | 0 | 0 | 0,01 | 0 |
| 42 | 0,02 | 0 | 0 | 0,02 | 0 |
| 43 | 0 | 0 | 0 | 0 | 0,01 |
| 44 | 0,05 | 0,01 | 0,03 | 0,04 | 0 |
| 45 | 0 | 0 | 0 | 0,01 | 0 |
| 46 | 0,01 | 0 | 0,02 | 0,01 | 0,01 |
| 47 | 0,01 | 0 | 0 | 0,02 | 0 |
| 48 | 0,02 | 0,01 | 0,01 | 0,03 | 0 |
| 49 | 0,03 | 0,01 | 0,01 | 0,03 | 0,01 |

| | | | | | |
|---|---|---|---|---|---|
| 50 | 0,01 | 0 | 0 | 0 | 0,01 |
| 51 | 0,02 | 0,01 | 0,01 | 0,02 | 0 |
| 52 | 0,01 | 0 | 0 | 0,01 | 0 |
| 53 | 0,02 | 0 | 0 | 0,02 | 0 |
| 54 | 0,01 | 0,01 | 0 | 0 | 0 |
| 55 | 0 | 0 | 0 | 0,01 | 0 |
| **Total** | **2,14** | **0,52** | **0,57** | **2,08** | **0,38** |

**Table 5.8 Running Percentages Performance of CPU-Times (for Fast SCG) Against All other SCG-Algorithms**

| **ALGORITHM / TOOL** | **LDW** | **BK** | **BA** | **BH** | **Fast SCG** |
|---|---|---|---|---|---|
| **CPU (Sec)** | 100% | 24% | 26% | 97% | 18% |

Now, this table indicates a very nice comparison between the CPU time required by each SCG-methods, regarding in that too small numbers can be represented by zero as the computer program was designed for that purpose. The above results indicates that the **Fast SCG**-algorithm saves about 82% of CPU time compared with the SCG-algorithm LDW-method; saves about 79% of CPU time, when compared against spectral BH-method; saves about 6% of CPU time, when compared against the spectral BK-method and saves about 8% of CPU time, when compared against the spectral BA-method for solving the set of 55-selected complicated unconstrained nonlinear test problems.

## 5.3. DISCUSSIONS OF THE NUMERICAL RESULTS

There exists a large variety of CG-algorithms. Here, we have presented a Fast SCG-algorithm in which the parameter $g_k$ is computed as $\theta_k g_k$. For uniformly convex functions, if the step-size $d_k$ approaches zero, the gradient is bounded and the line search satisfies the strong Wolfe conditions, then our Fast SCG-algorithm is globally convergent. Also, for general nonlinear functions, if the parameter $\theta_k$ is bounded, then our Fast SCG-algorithm is globally convergent. The performance percentage of our fast proposed fast SCG-algorithm is very effective compared with other established CG-algorithms for a selected set of test problems found in the CUTE library. However, in general among the eight CG algorithms mentioned in this thesis, we have found that the CD algorithm is the worse and the proposed Fast SCG algorithm is the best. The argument of these algorithms as shown here is given:

**Table 5.9 Running Percentages Performance of (Fast SCG) Algorithm Against All other Standard and SCG-Algorithms with Respect to NOI**

| METHOD / TOOL | CD | PR | BA | FR | BH | LDW | BK | Fast SCG |
|---|---|---|---|---|---|---|---|---|
| **NOI** | 100% | 98% | 93% | 92% | 90% | 82% | 80% | 39% |

**NOI: CD** (the worst) **---PR----BA-----FR-----BH------LDW-----BK---- Fast SCG** (the best and fastest SCG-Method)

**Table 5.10 Running Percentages Performance of (Fast SCG) Algorithm Against All other Standard and SCG-Algorithms with Respect to NOF**

| METHOD / TOOL | CD | BA | FR | BH | PR | LDW | BK | Fast SCG |
|---|---|---|---|---|---|---|---|---|
| **NOF** | 100% | 97% | 96% | 96% | 95% | 92% | 90% | 41% |

**NOF: CD** (the worst) **---BA----FR-----BH------PR----LDW-----BK---- Fast SCG** (the best and fastest SCG-Method)

# CHAPTER 6

# CONCLUSION

## 6.1 CONTRIBUTIONS

Our study is restricted to the non-linear CG- optimization algorithms, widely used in optimization, especially for large scale complicated non-linear unconstrained optimization problems in extended or generalized form with four different dimensions:

1) The new SCG method is proposed by adding a new spectral parameter $\Theta$ and by modifying the Wolfe line search algorithm.

2) The new algorithm is compared with standard and spectral CG-methods; namely (FR; PR and CD) and (BK; BH; BA and LDW) using 55- well-known non-linear test functions by obtaining very promising results for both groups of the above algorithms.

3) The new algorithm is generic and easy to implement in all gradient-based optimization processes.

4) The new algorithm does not need the storage of any matrix, and so it is easily implemented both theoretically and experimentally.

## 6.2 LIMITATIONS

There are several limitations in this study that can be mentioned here:

1) There is no standard CG algorithm that is suitable for solving all complicated non-linear test functions and that is used to compare all studies.

2) Some authors do not publish their instance problems, thus making comparisons impossible.

3) The computers hardware specifications of different papers with different published years is another issue in comparing the numerical results.

## 6.3. FURTHER WORKS

We hope to follow the following steps for our future implementations:

1- This work may be extended for the field of non-quadratic models, especially for the conic section models.

2- This work may also be extended for the constrained optimization algorithms, especially for large-scale optimization problems.

3- The modified Wolfe line searches are used in this study, so this may be improved further by implementing modified Goldstein Line search techniques.

4- Implementing all the new algorithms developed in this thesis with other type of inexact line search techniques such as backtracking or Armijo line searches

5- The domain of these algorithms may be extended to training artificial neural networks which is used to solve different types of optimization problems.

6- The parallel algorithms entered many algorithms by customizing the CG-algorithm to fit parallel programming requirements.

7- Finally, for a practical purposes, one may form, a mathematical model for any complicated project; the model must be first analyzed further to find the appropriate CG-type algorithms used to solve it; after that a computer code will be prepared to solve such nonlinear-complicated problem.

## APPENDIX

The details of all selected test problems can be found in CUTE (Bongartz et al, 1995):

1) Extended Freudenstein & Roth
2) Extended Trigonometric Function
3) Extended Beale Function
4) Extended Penalty Function
5) Raydan 1 Function
6) Raydan 2 Function
7) Diagonal2 Function
8) Hager Function
9) Generalized Tridiagonal-1 Function
10) Extended Tridiagonal-1 Function
11) Extended Three Exponential Terms
12) Generalized Tridiagonal-2
13) Diagonal4 Function
14) Diagonal5 Function
15) Extended Himmelblau Function
16) Generalized PSC1 Function
17) Extended PSC1 Function
18) Extended Block Diagonal BD1 Function

19) Extended Cliff

20) Quadratic Diagonal Perturbed Function

21) Extended Wood Function

22) Extended Quadratic Penalty QP1 Function

23) Extended Quadratic Penalty QP2 Function

24) Extended EP1 Function

25) Extended Tridiagonal-2 Function

26) ARWHEAD

27) NONDQUAR

28) EG2

29) DIXMAANA

30) DIXMAANB

31) DIXMAANC

32) DIXMAANE

33) Partial Perturbed Quadratic

34) Broyden Tridiagonal

35) EDENSCH Function

36) DIAGONAL 6

37) DIXON3DQ

38) ENGVAL1

39) DENSCHNA

40) DENSCHNC

41) DENSCHNB

42) DENSCHNF

43) BIGGSB1

44) Extended Block-Diagonal BD2

45) Generalized Quartic GQ1 function

46) Diagonal 7

47) Diagonal 8

48) Full Hessian

49) SINCOS

50) Generalized Quartic GQ2 function

51) EXTROSNB

52) ARGLINB

53) FLETCHCR

54) HIMMELBG

55) HIMMELBH

# REFERENCES


Al-Bayati, A. and Abdullah, R. (2008), A New Family of Spectral CG-Algorithm, Rafidain Journal of Computer Science & Mathematics, Iraq, 5(1), 69-80.

Al-Bayati, A. and Hassan, B. (2011), A Spectral Conjugate Gradient Method With An Inexact Line Search, Iraqi Journal of Statistical Science, Iraq, 11(20), 155-163.

Al-Bayati, A. and Al-Khayat, H. (2013), A Global Convergent Spectral Conjugate Gradient Method, Australian Journal of Basic and Applied Sciences, 7(7), 302-309.

Andreessen, N; Evgrafov, A. and Patriksson, M. (2005), An Introduction to Continuous Optimization: Foundations and Fundamental Algorithms, Germany.

Andrei, N. (2007a), Numerical Comparison of Conjugate Gradient Algorithms for Unconstrained Optimization, Studies in Informatics and Control, 16, 333–352.

Andrei, N. (2007b), Scaled Conjugate Gradient Algorithms for Unconstrained Optimization, Computational Optimization and Applications, 38, 401- 416.

Andrei, N. (2007c), Scaled Memoryless BFGS Preconditioned CG Algorithms for Unconstrained Optimization Methods and Software, 22, 561-571.

Andrei, N. (2008), A Scaled Nonlinear Conjugate Gradient Algorithm for Unconstrained Optimization, A Journal of Mathematical Programming and Operations Research, 57, 549-570

Andrei, N. (2009a), Another Nonlinear Conjugate Gradient Algorithm for Unconstrained Optimization, Optimization Methods and Software, 24(1), 89-104.

Andrei, N. (2009b), Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization, Journal Optimization Theory Application, 141, 249–264.

Andrei, N. (2009c), Acceleration of Conjugate Gradient Algorithms for Unconstrained Optimization, Applied Mathematical Computational, 213, 361–369.

Baldick, R. (2009), Applied Optimization: Formulation and Algorithms for Engineering Systems Slides, www.cambridge.org/resources/ 0521855640/ 3529.

Barzilai, J. and Borwein, J. (1988), Two-Point Step Size Gradient Methods, IMA Journal of Numerical Analysis, 8(1), 141-148.

Biggs, M. (2005), Nonlinear Optimization with Financial Applications, Kluwer Academic Publishers, England.

Birgin, E., Mart´ınez, J. and Raydan, M. (2000), Non-Monotone Spectral Projected Gradient Methods on Convex Sets, SIAM Journal on Optimization, 10,1196–1211.

Birgin, E. and Mart´ınez, J. (2001), A Spectral Conjugate Gradient Method for Unconstrained Optimization, Applied Mathematical Optimization, 43,117–128.

Bongartz, I. Conn, A. Gould, N. and Toint, P. (1995), CUTE Constrained and Unconstrained Testing Environment, ACM Transactions on Mathematical Software, 21(1), 123–160.

Dai, Y. and Yuan, Y. (2001), A Three Parameter Family of Hybrid Conjugate Gradient Method, Mathematics of Computation, 70, 1155-1167.

Dai, Y. and Yuan, Y. (1999), A Nonlinear Conjugate Gradient Method with a Strong Global Convergence Property, SIAM J. Optimization, 10, 177-182.

Dai, Y. and Liao, L. (2001), New Conjugacy Conditions and Related Non-Linear CG Methods, Applied Mathematical Optimization, Spring-Verlag, New York, 43, 87-101.

Dai, Y. Yuan, Y. (2001), A three-parameter family of hybrid conjugate gradient method, Mathematics of Computation, 70, 1155-1167.

Fletcher, R. (1987), Practical Methods of Optimization: Unconstrained Optimization, John Wiley & Sons, New York, USA.

Fletcher, R. and Reeves, G. (1964), Function Minimization by Gradients, Computer Journal, 7, 149-154.

Frandsen, P. Jonasson, K. Nielsen, H. and Tingleff, O. (2004), Unconstrained Optimization, Informatics and Mathematical Modeling Technical University of Denmark (IMM-DTU), (3rded), Denmark.

Ghanbari, M.; Ahmad, T.; Alias, N. and Askaripour, M. (2013), Global Convergence of Two Spectral Conjugate Gradient Methods, Science Asia, 39, 306–311.

Griva I.; Nash, S. and Sofer, A. (2009), Linear and Non-Linear Optimization, (2nded), Society for Industrial and Applied Mathematics (SIAM), Philadelphia.

Hager, W. and Zhang, H. (2005), CG-Descent, a Conjugate Gradient Method with Guaranteed Descent, SIAM Journal on Optimization, 16, 170-192.

Hager, W. and Zhang, H. (2006), Algorithms 851: CG-descent, A Conjugate Gradient Method with Guaranteed Descent, ACM. Transactions on Mathematical Software, 32(1), 113–137.

Hestenes, M and Stiefel, E. (1952), Methods of CG for Solving Linear Systems, J. Research Nat. Bar . standards Sec. B, 49, 409-436.

Liu, J. and Jiang, Y. (2012), Global Convergence of a Spectral Conjugate Gradient Method for Unconstrained Optimization, Hindawi Publishing Corporation, Abstract and Applied Analysis.

Liu, J., Du, X. and Wang, K. (2012), A Mixed Spectral CD-DY Conjugate Gradient Method, Hindawi Publishing Corporation, Journal of Applied Mathematics.

Liu, D. and Story, C. (1991), Efficient Generalized CG Algorithms, Part l: Theory, Journal of Optimization Theory and Applications, 69, 129–137.

Livieris I. and Pintelas P. (2012), A New Class of Spectral Conjugate Gradient Methods Based on a Modified Secant Equation for Unconstrained Optimization, Journal of Computational and Applied Mathematics, 239, 396–405.

Liu, D. Zhang, L. and Xu, G. (2014), Spectral Method and Its Application to the Conjugate Gradient Method, Applied Mathematics and Computation, 240, 339–347.

Navon I. and Legler M. (1987), Conjugate Gradient Methods for Large-Scale Minimization in Meteorology, Monthly Weather Review American Meteorological Society, 115(4),1479-1502.

Nazareth, J. (1999), Note Sur la Convergence de Methods de Directions Conjugates, 3(16), 35-43.

Polak, E. and Ribière. R. (1969), Not sur la Convergence de Directions Conjugates. Rev. Franaisse Informants. Research operational, 3e Anne., 16, 35-43.

Powell, M. (1977), Restart Procedure for the Conjugate Gradient Methods, Mathematics Programming, 12, 241-254.

Raydan, M. (1997), The Brazilian and Borwein Gradient Method for the Large Unconstrained Minimization Problem, SIAM Journal on Optimization, 7, 26–33.

Shanno, D. (1978), Conjugate Gradient Methods with Inexact Searches, Mathematics of Operations Research 3, 244–256.

Sun, M. and Liu, J. (2013), A New Spectral Conjugate Gradient Method and Its Global Convergence, Journal of Information and Computing Science, 8(1),75-79.

Touti-Ahmed, D. and Story, C. (1990), Efficient Hybrid Conjugate Gradient Techniques for Unconstrained Optimization, Journal of Optimization Theory and Applications, 64, 379-397.

Yu, G.; Guan, L. and Chen, W. (2008), Spectral Conjugate Gradient Methods with Sufficient Descent Property for Large-Scale Unconstrained Optimization, Optimization Methods and Software, 23(2), 275–293.

Zhang, J., Xiao, Y. and Wei, Z. ( 2009 ), Non-Linear Conjugate Gradient Methods with Sufficient Descent Condition for Large-Scale Unconstrained Optimization, Hindawi publishing corporation. Mathematical Problems in Engineering Volume.

Zoutendijk, G. (1970), Nonlinear Programming, Computational Methods, In Integer and Nonlinear Programming, J. Abadie (Ed), Holland, 37-86.

Wenyu, S. and Yuan Y. (2006), Optimization Theory and Methods; Nonlinear Programming, Springer Science and Business Media, LLC, USA.

Wolfe, P. (1969), Convergence Conditions for Ascent Methods, SIAM Review, 11, 226–235.

Zeng, W. and Liu, H. (2015), The Global Convergence of a New Spectral Conjugate Gradient Method, International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2015).