CONJUGATE GRADIENT LINE SEARCH FOR SOLVING UNCONSTRAINED OPTIMIZATION PROBLEM

ADDIS ABABA UNIVERSITY
COLLEGE OF COMPUTATIONAL AND NATURAL SCIENCES
DEPARTMENT OF MATHEMATICS
"In partial fulfillment of the requirements of the degree of
master of science in mathematics"

By:Bayisa Gosa
Stream: Optimization
Advisor: Birhanu Guta(PhD)

June 14, 2018

ADDIS ABABA UNIVERSITY
DEPARTMENT OF MATHEMATICS

The undersigned hereby certify that they have read and recommend to the department of mathematics for acceptance of this project entitled "**Conjugate gradient line search for solving unconstrained optimization problem** " by **Bayisa Gosa** in partial fulfillment of the requirements for the degree of Master of Science in mathematics.

Advisor: Dr. Birhanu Guta

Signature:_____

Date_____

ADDIS ABABA UNIVERSITY

Author:Bayisa Gosa

Title:Conjugate Gradient method for solving unconstrained optimization Problem

Department: Mathematics

Degree: M.Sc.

Convocation: **June,2018**

Bayisa Gosa:
Signature:_____

Date_____

Examiner 1: Dr. Hunduma L.

Signature:_____

Date_____

Examiner 2: Dr. Nega A.

Signature:_____

Date_____

Contents

Abstract i

Acknowledgment ii

Notations iii

Introduction iv

1 Preliminary 1
- 1.1 Optimality Conditions for Unconstrained Optimization . . . 1
- 1.2 Line search . . . 2
- 1.3 Line Searches and Quadratic Functions . . . 3
  - 1.3.1 Conjugate Direction . . . 4
  - 1.3.2 Method of conjugate Directions: quadratic case . . . 5
- 1.4 Descent properties of Conjugate Direction Method . . . 6

2 Conjugate Gradient Method for solving unconstrained non-linear optimization problem 8
- 2.1 Introduction . . . 8
- 2.2 (Minimizing quadratic function) . . . 9
- 2.3 (Minimizing Non-quadratic Function) . . . 10
- 2.4 Formulation and Criteria . . . 11
- 2.5 Update method of β_k . . . 11
  - 2.5.1 F-R Update . . . 11
  - 2.5.2 Hager and Zhang β_k update . . . 15
- 2.6 Convergence Analysis and Rate of Convergence . . . 15
  - 2.6.1 Global Convergence of F-R, PRP and Hager and Zhang . . . 15
- 2.7 Rate of Convergence . . . 24
- 2.8 Numerical Examples . . . 24

3 Summary 27

Bibliography 28

APPENDIX 30

Abstract

Conjugate gradient methods are a class of important methods for solving linear equations and for solving nonlinear optimization. In this project, a review on conjugate gradient methods for unconstrained optimization is given. They are divided into early conjugate gradient methods, descent conjugate gradient methods and sufficient descent conjugate gradient methods. The general convergence theorems are provided for the conjugate gradient method assuming the descent property of each search direction.

Keywords: Unconstrained optimization, Conjugate Gradient Method, Line search .

Acknowledgment

I express my sincere gratitude to my advisor and instructor **Dr.Birhanu Guta** for helping me on every difficult step of this senior project. I would also like to thank my classmates for giving me suggestions regarding this project. On top of this, my thank goes to Ambo University for its financial support in attending this degree of masters of Science.

Notations

- $\mathbb{R}$ the set of real numbers
- x_k the value of x at k^{th} iteration
- $\nabla f(x_k)$ the gradient of f at x_k
- $g_k := \nabla f(x_k)$
- A positive definite matrix
- β_k update parameter at k^{th} iteration
- β_k^{FR} Fletcher and Reeves update
- β_k^{CD} Fletcher restart update
- β_k^{PRP} Polak and Ribiere and Polyak update
- β_k^{N} Hager and Zhang update
- $s_k := x_{k+1} - x_k$
- $y_k := \nabla f(x_k + 1) - \nabla f(x_k)$
- x^* the minimizer of f
- d_k the search direction vector at k^{th} iteration
- α_k the step size or step length at k^{th} iteration
- $\|.\|$ norm of a vector or norm of matrix.

Introduction

Conjugate gradient methods are a class of important methods for solving unconstrained optimization problem like;

$$minf(x); x \in \mathbb{R}^n \tag{1}$$

especially if the dimension n is large. They are of the form

$$x_{k+1} = x_k + \alpha_k d_k \tag{2}$$

where α_k is a step size obtained by a line search, and d_k is the search direction defined by

$$d_k = \begin{cases} -g_k, & \text{for } k = 1. \\ -g_k + \beta_k d_{k-1}, & k \geq 2. \end{cases} \tag{3}$$

where $\beta_k$is a parameter. It is known from (2) and (3) that only the step size α_k and the parameter β_k remain to be determined in the definition of conjugate gradient methods. In the case that$f$ is a convex quadratic, the choice of β_k should be such that the method (2)-(3) reduces to the linear conjugate gradient method if the line search is exact, namely,

$$\alpha_k = minf(x_k + \alpha d_k); \alpha \geq 0. \tag{4}$$

For nonlinear functions, however, different formulae for the parameter β_k result in different conjugate gradient methods and their properties can be significantly different. To differentiate the linear conjugate gradient method, sometimes we call the conjugate gradient method for unconstrained optimization by nonlinear conjugate gradient method. Meanwhile, the parameter β_k is called conjugate gradient parameter. The linear conjugate gradient method can be dated back to a seminal paper by Hestenes and Stiefel [30] in 1952 for solving a symmetric positive definite linear system $AX = b$, where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. An easy and geometrical interpretation of the linear conjugate gradient method can be founded in Shewchuk. The equivalence of the linear system to the minimization problem of $f(x) = \frac{1}{2}X^T AX + bX$ motivated Fletcher and Reeves[30] to extend the linear conjugate gradient method for nonlinear optimization. This work of Fletcher and Reeves in 1964 not only opened the door of nonlinear conjugate gradient but greatly stimulated the study of nonlinear optimization. In general, the nonlinear conjugate gradient method without restarts is only linearly convergent, while n-step quadratic convergence rate can be established if the method is restarted along the negative gradient every n-step and Some recent reviews on nonlinear conjugate gradient methods can be found in Hager and Zhang [20],Nocedal [19], etc. This project aims to provide a view on the β_k^{FR}, β_k^{CD}, β_k^{PRP}, β_k^{N} update methods.

Chapter 1

Preliminary

In this chapter before study Conjugate Gradient method, we want to present some preliminary concepts.

1.1 Optimality Conditions for Unconstrained Optimization

Consider unconstrained optimization problem

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

Definition 1.1. *A point x^* is called local minimum of f if there exists $\delta > 0$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ satisfying $\|x - x^*\| < \delta$.*

Definition 1.2. *Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Then A is said to be positive definite if $v^T A v > 0$ for every non-zero $v \in \mathbb{R}^n$*

Let x^* be a local minimum of a twice differentiable function f. Then there are two conditions that hold at local minimum of f see e.g. [11, p.58–63].

(1) $\nabla f(x^*) = 0$, this condition is called first order condition

(2) f has non-negative curvature at x^*, i.e. the Hessian matrix of f at x^* should be positive semi-definite ($x^T \nabla^2 f(x^*) x \geq 0$) for all x. We call this condition as second order condition.

These two conditions are called necessary conditions. The necessary condition does not ensure the given point is local minimum, so we need additional conditions. The following conditions are sufficient for x^* to be local minimum of f[11,p.58–63].

(a) $\nabla f(x^*) = 0$.

(b) $\nabla^2 f(x^*)$ is positive definite. i.e. $x^T \nabla^2 f(x^*) x > 0$.

Definition 1.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable at $x \in \mathbb{R}^n$. If there exists a vector $d \in \mathbb{R}^n$ such that $\langle \nabla f(x), d \rangle < 0$ then d is called a descent direction.*

1.2 Line search

To compute the new step update in conjugate gradient methods we use the line search strategy. In multi-variable optimization algorithms, for given x_k, the iterative scheme is

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

The key is to find the direction vector p_k and suitable step size α_k. Let

$$\phi(\alpha) = f(x_k + \alpha p_k).$$

So, the problem that departs from x_k and finds a step size in the direction p_k such that

$$\phi(\alpha_k) < \phi(0)$$

is just line search about α . If we find α_k such that the objective function in the direction p_k is minimized, i.e.,

$$f(x_k + \alpha_k p_k) = \min f(x_k + \alpha p_k), \alpha > 0$$

then such line search is called exact(or optimal) line search [11, p.71]. If we choose α_k such that the objective functions has acceptable descent amount, i.e., such that the descent $f(x_k) - f(x_k + \alpha_k p_k) > 0$ is acceptable by users, such a line is called inexact or approximate line search [11, p.71].

The most commonly used line search method is to find the step length that satisfies the (strong) Wolfe conditions. A popular inexact line search condition stipulates that α_k should first of all give sufficient decrease in the objective function f, as measured by the following inequality

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k$$

for some constant $c_1 \in (0, 1)$. The sufficient decrease is not enough by itself to ensure that the algorithm makes reasonable progress, since it is satisfied for all sufficiently small value of α[11,p.38]. To rule out unacceptable short steps we introduce a second requirement called curvature condition, which requires α_k to satisfy

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k$$

for some constant $c_2 \in (c_1, 1)$. Note that the left-hand side is simply the derivative $\phi'(\alpha_k)$, so that the curvature condition ensures that the slope of $\phi(\alpha_k)$ is greater than c_2 times the gradient $\phi'(0)$. This makes sense because if the slope $\phi'(\alpha)$ is strongly negative, we have an indication that we can reduce f significantly by moving further along chosen direction. The sufficient decrease and curvature conditions are known collectively as the Wolfe conditions.

A step length may satisfy the Wolfe conditions without being particularly close to a minimizer of ϕ. However, we can modify the curvature condition to force α_k to lie in at least a broad neighborhood of local minimizer or stationary point of ϕ. The reason is that we no longer allow the derivative $\phi'(\alpha_k)$ to be too positive. Hence, we exclude points that are far from stationary points of ϕ. Thus the strong Wolfe conditions requires α_k to satisfy

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k \tag{1.1}$$

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |\nabla f_k^T p_k| \tag{1.2}$$

with $0 < c_1 < c_2 < 1$.

Sufficient decrease and Backtracking
We presented that the sufficient decrease condition alone is not sufficient to ensure that the algorithm makes reasonable progress along the given search direction. However, if the line search algorithm chooses its candidate step lengths appropriately, by using backtracking approach, we can dispense with the second requirement called curvature condition and use just the sufficient decrease condition to terminate the line search procedure. Backtracking proceeds as follows.

Procedure. (Backtracking line search)
Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable at $x \in \mathbb{R}^n$ and $d \in \mathbb{R}^n$ is a direction of strict descent atx_c , i.e., $\langle \nabla f(x_c), d\rangle < 0$
Initialization :Choose $\gamma \in (0, 1)$ and $c \in (0, 1)$
having x_c obtain x_+ as follows.
Step 1 : Compute backtracking step size until $f(x_c + \gamma^v d) \leq f(x_c) + c\gamma^v f'(x_c, d)$
Step 2: $x_+ = x_c + t^* d$

Now we need to show that the backtracking line search is well defined and finitely terminating.
Since $f'(x_c, d) \leq 0$ and $0 \leq c \leq 1$, we know that $f'(x_c, d) \leq cf'(x_c, d) \leq 0$.
Hence

$$f'(x_c, d) = \lim_{t\to 0} \frac{f(x_c + td) - f(x_c)}{t} < cf'(x_c, d)$$

Therefore, there is a $\bar{t} > 0$ such that

$$\frac{f(x_c + td) - f(x_c)}{t} < cf'(x_c, d), \forall t \in (0, \bar{t})$$

that is

$$f(x_c + td) < f(x_c) + ctf'(x_c, d), \forall t \in (0, \bar{t})$$

so

$$f(x_c + td) < f(x_c) + ctf'(x_c, d), \forall t \in (0, \bar{t})$$

since $0 < \gamma < 1$, $\gamma^v \to 0$ as $v \to \infty$, there is a v_0 such that $\gamma^v < \bar{t}\ \forall v \geq v_0$.
Consequently, $f(x_c + \gamma^v d) \leq f(x_c) + c\gamma^v f'(x_c, d)$
Now we generalize the above procedure by the following algorithm.

Algorithm 1. (Backtracking - Armijo line search)
Choose $\bar{\alpha} > 0$, ρ, $c \in (0, 1)$; set $\alpha \leftarrow \bar{\alpha}$;
repeat untill $f(x_k + \alpha p_k) < f(x_k) + c\alpha \nabla f_k^T p_k$
$\alpha \leftarrow \rho\alpha$
end repeat
Terminate with $\alpha_k = \alpha$.

1.3 Line Searches and Quadratic Functions

Most of the algorithm we have developed for finding minima involve a series of line searches. Consider performing a line search on the function $f(x) = \frac{1}{2}x^T Ax + x^T b + c$ from some base point a in the direction v, i.e minimizing f along the line $L(t) = a + tv$. This amounts to minimizing the function

$$g(t) = f(a+tv) = \tfrac{1}{2}v^T Avt^2 + (v^T Aa + v^T b)t + \tfrac{1}{2}a^T Aa + a^T b + c \qquad (2)$$

Since $\frac{1}{2}v^T Av \geq 0$ (A is positive definite), the quadratic function $g(t)$ always has a unique global minimum in t, for g is quadratic in t with a positive coefficient in front of t^2. The minimum occurs when $g^{'}(t) = (v^T Av)t + (v^T Aa + v^T b) = 0$, i.e., at $t = t^*$ where

$$t^* = -\frac{v^T(Aa+b)}{v^T Av} = -\frac{v^T \nabla f(a)}{v^T Av} \qquad (3)$$

Note that if v is a descent direction ($v^T \nabla f(a) \leq 0$) then $t^* \geq 0$. Another thing to note is this: Let $p = a + t^* v$, so p is the minimum of f on the line $a + tv$. Since $g^{'}(t) = \nabla f(a+tv).v$ and $g^{'}(t) = 0$, we see that the minimizer p is the unique point on the line $a + tv$ for which

$$\nabla f(p).v = 0. \qquad (4)$$

Thus if we are at some point $b \in R^n$ and want to test whether we are at a minimum for a quadratic function f with respect to some direction v, we merely test whether $\nabla f(b).v = 0$.

1.3.1 Conjugate Direction

Definition 1.4. *Let H be a symmetric positive definite matrix of order n.*
We say that two vectors $d_1, d_2 \in R^n$ are conjugate with respect to H if
$d_1^T H d_2 = 0$.
A set of vectors $d_1, ..., d_k$ are conjugate if $d_i^T H d_j = 0$ whenever $i \neq j$.

Lemma 1.1. *: Let H be symmetric positive definite matrix,and $d_0, d_1, ..., d_k$ be non-zero conjugate vectors. Then the coefficients α_i in a linear combination*

$$d = \sum_{i=0}^{k} \alpha_i d_i \qquad (1.3)$$

can be restored by explicit formulae

$$\alpha_i = \frac{d^T H d_i}{d_i^T H d_i}, i = 0, ..., k \qquad (1.4)$$

The denominators are nonzero due to positive definiteness of H recall that d_i are nonzero by assumption.

Proof. taking the usual inner product of both sides on (1.3) with Hd_j, we get

$$d^T H d_j = \alpha_i d_j^T$$

due to the orthogonality of d_i with $i \neq j$ to d_j, he right hand side terms associated with $i \neq j$ vanish, and we come to (1.4). In fact relations (1.4) are nothing but the standard formulae for Fourier coefficients of a vector in orthogonal (in the Euclidean structure $\langle u, v \rangle = u^T Hv$ basis $d_0, ..., d_{n-1}$

$$x = \sum_{i=0}^{n-1} \frac{\langle x, d_i \rangle}{\langle d_i, d_i \rangle} d_i.$$

□

Theorem 1.1. *:If the set $S = \{d_0, d_1, ..., d_{n-1}\}$ is conjugate and all d_i are non- zero then S forms a basis for R^n.*

Proof. :Since S has n vectors ,all we need to show is that S is linearly independent, since it then automatically spans R^n. Start with

$$c_1 d_1 + c_2 d_2 + ... + c_n d_n = 0.$$

Multiply on the right by A distribute over the sum and multiply the whole mess by d_i^T to obtain

$$c_i d_i^T A d_i = 0.$$

we immediately conclude that $c_i = 0$, so the set is linearly independent and hence a basis for R^n. □

Proposition 1.1. *Let H be a positive definite n x n matrix,and let $d_0, ..., d_{n-1}$ be a system of n non-zero H orthogonal vectors. Then the solution x^* to the system*

$$Hx = b$$

is given by the formula

$$x^* = \sum_{i=0}^{n-1} \frac{b^T d_i}{d_i^T H d_i} d_i$$

1.3.2 Method of conjugate Directions: quadratic case

It is convinient for us to formulate the statement of proposition as an assertion about certain iterative algorithm:

Theorem 1.2 (Conjugate Direction Theorem)**.** *Let H be a positive definite symmetric matrix of order n, b be a vector and*

$$f(x) = \frac{1}{2} x^T H x - b^T x \tag{1.5}$$

be the quadratic form associated with H and b. Let further, $d_0, ..., d_{n-1}$ be a system of non-zero H-orthogonal vectors, and let x_0 be an arbitrary starting point. The iterative process

$$x_{t+1} = x_t + \beta_{t+1} d_t, \quad \beta_{t+1} = -\frac{d_t^T g_t}{d_t^T H d_t}, t = 0, ..., n-1 \tag{1.6}$$

where g_t is the gradient of f at x_t:

$$g_t = \nabla f(x_t) = Hx - b \tag{1.7}$$

converges to the unique minimizer x^* of f ($\equiv$ the unique minimizer to the linear system $Hx = b$) in n steps: $x_n = x^*$

Proof. From (1.6) above we have

$$x_k - x_0 = \sum_{t=0}^{k-1} \beta_{t+1} d_t,$$

whence

$$d_k^T H(x_k - x_0) = \sum_{t=0}^{k-1} \beta_{t+1} d_k^T H d_t = 0 \tag{1.8}$$

On the other hand by (1.3)

$$x^* - x_0 = \sum_{t=0}^{n-1} \frac{d_t^T H(x^* - x_0)}{d_t^T H d_t} d_t = \sum_{t=0}^{n-1} \frac{d_t^T H(x^* - x_t)}{d_t^T H d_t} \quad \text{by} \quad (1.8)$$

$[SinceH(x^* - x_t) = b - Hx_t = -\nabla f(x_t) = -g_t]$

$= \sum_{t=0}^{n-1} [\frac{d_t^T g_t}{d_t^T H d_t}] = \sum_{t=0}^{n-1} \beta_{t+1} d_t = x_n - x_0$ (by definition of $\beta_t + 1$) so that $x^* = x_n$

□

1.4 Descent properties of Conjugate Direction Method

Let as above

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

be a strongly convex quadratic form on $\mathbb{R}^n$ (so that H is a positive definite symmetric matrix of order n),and let $d_0, ..., d_n - 1$ be a system of n non-zero H- orthogonal vectors. Our goal is to establish certain property of the trajectory$x_0, ..., x_n$ of the Conjugate Direction method associated with $d_0, ..., d_n - 1$. We already know that x_n is the minimizer of f on the entire space $\mathbb{R}^n$; and the indicated property is that every $x_t, 0 \leq t \leq n$, is the minimizer of f on the affine subspace

$$M_t = x_0 + \lambda_t - 1, \quad \lambda_t - 1 = Lin\{d_0, ..., d_t - 1\}$$

is the linear span of the vectors $d_0, ..., d_n - 1$ (here, for the homogeneity, the linear span of an empty set of vectors is $0 : \lambda_0 = 0$.) The affine sets $M_0, ..., M_n$ form an increasing sequence:

$$x_0 = M_0 \subset M_1 \subset ... \subset M_n - 1 \subset M_n = \mathbb{R}^n$$

which links x_0 and the entire space.This property is given by the following.

Proposition 1.2. *For every $t, 1 \leq t \leq n$, the vector x_t is the minimizer of f on the affine plane*

$$x_0 + \lambda_t - 1$$

where

$$\lambda_t - 1 = Lin\{d_0, ..., d_t - 1\}$$

is the linear span of $d_0, ..., d_t - 1$. In particular x_t minimizes f on the line

$$l_t = x_t - 1 + \beta d_t - 1 | \beta \in \mathbb{R}.$$

Proof. by construction, $l_t \subset x_0 + \lambda_t - 1$ and $x_t \in l_t$; knowing that x_t minimizes f on $x_0 + \lambda_t + 1$, we could immediately conclude that it minimizes f on $l_t \subset x_0 + \lambda_{t-1}$ as well. To prove that x_t minimizes f on $x_0 + \lambda_t - 1$, it is suffices to prove that the gradient g_t of f at x_t is orthogonal to $\lambda_t - 1$, i.e., is orthogonal to $d_0, ..., d_{t-1}$. According to the Conjugate Direction Theorem, we have $x_n = x^* \equiv H^{-1} b$ whence

$$x_t - x^* = x_t - x_n = -\sum_{i=t}^{n-1} \beta_{i+1} d_i$$

whence

$$g_t = Hx_t - b = Hx_t - Hx^* = H(x_t - x^*) = -\sum_{i=t}^{n-1} \beta_{i+1} [Hd_i]$$

Since $d_0, ..., d_n - 1$ are H-orthogonal, each vector $Hd_i, i \geq t$, is orthogonal to all vectors $d_j, 0 \leq j \leq t$, and consequently is orthogonal to $\lambda_t - 1$ as just we have seen g_t is a linear combination of $Hd_i, i = 0, ..., n-1$, so that g_t is also orthogonal to λ_{t-1} □

Theorem 1.3. *(Principal Theorem of Conjugate Direction Method) For a quadratic function with positive definite Hessian G, the conjugate direction method terminates in at most n exact line searches. Each x_{i+1} is the minimizer in the subspace generated by x_0 and the directions $d_0, ..., d_i$, that is $\{x | x = x_0 + \sum_{j=0}^{i} \alpha_j d_j\}$*

Proof. Since G is positive definite and the conjugate directions $d_0, d_1, ...$ are linearly independent, it is enough to prove for all $i \leq n1$ that

$$g_{i+1}^T d_j = 0, \quad j = 0, ..., i \tag{1.9}$$

(Note that if (1.9) holds, we immediately have $g_n^T d_j = 0, \quad j = 0, ..., n1$ and $g_n = 0$, therefore x_n is a minimizer.)
To prove (1.9), we consider two cases $j < i$ and $j = i$. Keep in mind that

$$yk =^{Def} g_{k+1} - g_k = G(x_{k+1} - x_k) = \alpha_k G d_k. \tag{1.10}$$

When $j < i$, by use of exact line search and the conjugacy, we have

$$\begin{aligned} g_{i+1}^T d_j &= g_{j+1}^T d_j + \sum_{k=j+1}^{i} y_k^T d_j \\ &= -g_{j+1}^T d_j + \sum_{k=j+1}^{i} \alpha_k d_k^T G d_j \\ &= 0 \end{aligned} \tag{1.11}$$

When $j = i$, (1.9) is a direct result from the exact line search. Thus (1.9) holds and we complete the proof. □

Algorithm 2. (General conjugate Direction method)

1 *Given $x_0 \in \mathbb{R}^n, \epsilon > 0, k : 0. compute g_0 = g(x_0)$; compute d_0 such that $d_0^T g_0 < 0$*

2 *if $||g_k|| \leq \epsilon$ stop, else*

3 *Compute α_k by line search minimization. and set*

$$x_{k+1} = x_k + \alpha_k d_k$$

4 *Compute d_{k+1} by some conjugate gradient method, such that $d_{k+1}^T A d_j = 0, j = 0, ..., k$*

5 *Set $k := k + 1$, go to step 2*

Chapter 2

Conjugate Gradient Method for solving unconstrained non-linear optimization problem

2.1 Introduction

Conjugate gradient (CG) methods comprise a class of unconstrained optimization algorithms which are characterized by low memory requirements and strong local and global convergence properties. CG history, surveyed by Golub and O'Leary in [31], begins with research of Cornelius Lanczos and Magnus Hestenes and others (Forsythe, Motzkin, Rosser, Stein) at the Institute for Numerical Analysis (National Applied Mathematics Laboratories of the United States National Bureau of Standards in Los Angeles), and with independent research of Eduard Stiefel at Eidg. Technische Hochschule Zurich. In the seminal 1952 paper [30] of Hestenes and Stiefel, the algorithm is presented as an approach to solve symmetric, positive-definite linear systems. In this survey, we focus on conjugate gradient methods applied to the nonlinear unconstrained optimization problem

$$minf(x) : x \in \mathbb{R}^n; \tag{2.1}$$

where $f : \mathbb{R}^n \longmapsto \mathbb{R}$ is a continuously differentiable function, bounded from below. A nonlinear conjugate gradient method generates a sequence $\{x_k\}, k \geq 1$, starting from an initial guess $x_0 \in \mathbb{R}^n$, using the recurrence

$$x_{k+1} = x_k + \alpha_k d_k; \tag{2.2}$$

where the positive step size α_k is obtained by a line search, and the directions d_k are generated by the rule:

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad d_0 = -g_0. \tag{2.3}$$

Here β_k is the CG update parameter and $g_k = \nabla f(x_k)^T$, where the gradient, $\nabla f(x_k)$ of f at x_k is a row vector and g_k is a column vector.
One drawback of conjugate direction method is that we need an entire set of conjugate directions d_i before we start. Here we discuss a way to generate the conjugate directions. First let us see for quadratic function and next we will see the extension to a general nonlinear function.

2.2 (Minimizing quadratic function)

Let $f(x) = \frac{1}{2}x^T Ax + b^T x$ be a quadratic function. now we generate a set of conjugate directions as follows. First, we take $d_0 = -g_0 = -(Ax_0 + b)$. we then perform the line minimization as of conjugate direction method to obtain point $x_1 = x_0 + \alpha_0 d_0$, where α_0 is obtained by

$$\alpha_k = -\frac{d_k^T g_k}{d_k^T A d_k} \tag{2.4}$$

The next direction d_1 is computed as a linear combination $d_1 = -g_1 + \beta_0 d_0$ of the current gradient and the last search direction. We choose β_0 so that $d_1^T A d_0 = 0$, which gives

$$\beta_0 = \frac{g_1^T A d_0}{d_0^T A d_0}$$

Of course d_0 and d_1 form a conjugate set of directions, by construction. We then perform a line search from x_1 in the direction d_1 to obtain x_2.
In general we compute the search direction d_k as a linear combination

$$d_k = -g_k + \beta_{k-1} d_{k-1} \tag{2.5}$$

of the current gradient and the last search direction. We choose β_{k-1} so that $d_k^T A d_{k-1} = 0$,which forces

$$\beta_{k-1} = \frac{g_{x_k}^T A d_{k-1}}{d_{k-1}^T A d_{k-1}} \tag{2.6}$$

We then perform a line search direction x_k in the direction d_k to locate x_{k+1},and then repeat the whole procedure . This is early conjugate gradient algorithm. By construction we have $d_k A d_{k-1} = 0$,that is each search direction is conjugate to the previous direction, but we need $d_i^T A d_j = 0, \forall i \neq j$.

Theorem 2.1. *The set of search directions defined by equations (2.5) and (2.6) for $k = 1$ to $k = n$ (with $d_0 = -g_0$) satisfy $d_i^T A d_j = 0$, and so are conjugate.*

Proof. Well do this by induction. We have already seen that d_0 and d_1 are conjugate. Now suppose that the set $d_0, ..., d_m$ are all conjugate. Well show that $d_{m+1}$is conjugate to each of these directions. First, it will be convenient to note that if the directions d_k are generated according to (2.5) and (2.6) then g_{k+1} is orthogonal to g_j for $0 \leq j \leq k \leq m$. to see this we write the equation (2.5) as

$$g_j = \beta_{j-1} d_{j-1} - d_j$$

multiply both sides by g_{k+1}^T to obtain

$$g_{k+1}^T g_j = \beta_{j-1} g_{k+1}^T d_{j-1} - g_{k+1}^T d_j$$

But from the induction hypothesis (the directions $d_0, ..., d_m$ are already known to be con-jugate) and it follows that the right side above is zero, so

$$g_{k+1}^T g_j = 0 \tag{2.7}$$

for $0 \leq j \leq k$ as asserted. now we already know that d_{m+1} is conjugate to d_m, so we need only show that d_{m+1} is conjugate to d_k for $0 \leq k < m$. To this end recall that

$$d_{m+1} = -g_{m+1} + \beta_m d_m$$

Multiply both sides by d_k^T where $0 \leq k < m$ to obtain

$$d_k^T A d_{m+1} = -d_k^T A g_{m+1} + \beta_m d_k^T A d_m.$$

But $d_k^T A d_m = 0$ by the induction hypothesis, so we have

$$d_k^T A d_{m+1} = -d_k^T A g_{m+1} = -g_{m+1}^T A d_k \tag{2.8}$$

We need to show that the right side of equation (2.8) is zero. First, start with equation (2.2) and multiply by A to obtain $Ax_{k+1} - Ax_k = \alpha_k A d_k$ or $(Ax_{j+1} + b) - (Ax_k + b) = \alpha_k A d_k$. Since $g = Ax + b$ we have

$$g_{k+1} - g_k = \alpha_k A d_k$$

or

$$A d_k = \frac{1}{\alpha_k}(g_{k+1} - g_k) \tag{2.9}$$

Substitute the right side of (2.9) into equation (2.8) for Ad_k to obtain

$$d_k^T A d_{m+1} = \frac{1}{-\alpha_k}(g_{m+1})^T g_{k+1} - g_{m+1}^T g_k$$

for $k < m$. From equation (2.7) the right side above is zero,$d_k^T A d_{m+1} = 0$ for $k < m$. Since $d_m^T A d_{m+1} = 0$ by construction, weve shown that if $d_0, ..., d_m$ satisfies $d_i^T A d_j = 0$ fo $i \neq j$ then so does $d_0, ..., d_m, d_{m+1}$. From induction it follows that $d_0, ..., d_n$ also has this property. □

2.3 (Minimizing Non-quadratic Function)

The above algorithm works only for quadratic function, since it explicitly uses the matrix A. We are going to modify the algorithm so that the specific quadratic nature of the objective function does not appear explicitly in fact, only ∇f will appear, but the algorithm will remain unchanged if f is truly quadratic. However, since only ∇f will appear, the algorithm will immediately generalize to any function f for which we can compute ∇f.

Here are the modifications. The first step in the algorithm of conjugate gradient for quadratic function involves computing $d_0 = -\nabla f(x_0) = -g_0$. There is no mention of A, here we can compute ∇f for any differentiable function. In step 2 of the algorithm of conjugate gradient for quadratic function we do a line search from x_k in direction d_k. For the quadratic case we have the luxury of a simple formula (involving t_k) for the unique minimum. But the computation of t_k involves A. Let us replace this formula with a general (exact) line search, using Golden Section or whatever line search method you like. This will change nothing in the quadratic case, but generalizes things, in that we know how to do line searches for any function. Note this gets rid of any explicit mention of A in step 2.

Step 4 is the only other place we use A, in which we need to compute Ad_k in order to compute β_k. There are several ways to modify this to eliminate explicit mention of A. First (still thinking of f as quadratic) note that since (from step 2) we have $x_{k+1}x_k = cd_k$ for some constant c (where c depends on how far we move to get from x_k to x_{k+1}) we have

$$g_{k+1} - g_k = (Ax_{k+1} + b) - (Ax_k + b) = cAd_k \tag{2.10}$$

Thus $Ad_k = \frac{1}{c}(g_{k+1}g_k)$. Use this fact in the numerator and denominator of the definition for

β_k in step 4 to obtain

$$\beta_k = \frac{g_{k+1}^T(g_{k+1} - g_k)}{d_k^T(g_{k+1} - g_k)} \tag{2.11}$$

Note that the value of c was irrelevant! The search direction dk+1 is as before, $d_{k+1} = g_{k+1} + \beta_k d_k$. If f is truly quadratic then definitions (2.6) and (2.11) are equivalent. But the new algorithm can be run for ANY differentiable function. Equation (2.11) is the Hestenes-Stiefel formula. It yields one version of the conjugate gradient algorithm for non-quadratic problems.

Heres another way to get rid of A. Again, assume f is quadratic. Multiply out the denominator in (2.11) to find

$$d_k^T(g_{k+1} - g_k) = d_k^T g_{k+1} - d_k^T g_k$$

But from equation (1.9) we have $d_k^T g_{xk+1} = 0$, so really

$$d_k^T(g_{k+1} - g_k) = -d_k^T g_k$$

Now note $d_k = -g_k + \beta_{k-1} d_{k-1}$ so that

$$d_k^T(g_{k+1} - g_k) = -(-g_k^T + \beta_{k-1} d_{k-1}^T)g_k = ||g_k||^2$$

since by equation (1.9) again, $d_{k1}^T g_k = 0$. All in all, the denominator in (2.11) is, in the quadratic case, $||g_k||^2$. We can thus give an alternate definition of

β_k as

$$\beta_k = \frac{g_{k+1}^T(g_{k+1} - g_k)}{||g_k||^2} \qquad (PRP\ Formula) \tag{2.12}$$

Different CG methods correspond to different choices for the scalar β_k.

2.4 Formulation and Criteria

The formulation of conjugate gradient method is to be presented,as well as some important criteria. A conjugate gradient method generates a sequence $\{x_k\}, k \geq 1$,starting from initial guess $x_0 \in R^n$ using the recurrence

$$x_{k+1} = x_k + \alpha_k d_k$$

where the positive step size α_k is obtained by line search. The search direction d_k is obtained by

$$d_{k+1} = -g_{k+1} + \beta_k d_k, d_0 = -g_0$$

should be conjugate to each other. Here β_k is the CG update parameter and $g_k = \nabla f(x_k)^T$. The aim is to generate a descent direction. i.e.,$g_k^T d_k < 0$.

2.5 Update method of β_k

2.5.1 F-R Update

Now we derive the conjugate gradient method for the quadratic case. Let

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c, \tag{2.13}$$

where A is an n x n symmetric positive definite matrix, $b \in \mathbb{R}^n$ and c is a real number. Obviously, the gradient of $f(x$ is

$$g_{(x)} = Ax + b. \tag{2.14}$$

Set

$$d_0 = -g_0 \tag{2.15}$$

then we have

$$x_1 = x_0 + \alpha_0 d_0 \tag{2.16}$$

whereα_0 is generated by an exact line search. Then we have

$$g_1^T d_0 = 0 \tag{2.17}$$

Set

$$d_1 = -g_1 + \beta_0 d_0 \tag{2.18}$$

and choose β_0 such that

$$d_1^T A d_0 = 0 \tag{2.19}$$

It follows from multiplying (2.18) by $d_0^T A$ that

$$\beta_0 = \frac{g_1^T A d_0}{d_0^T A d_0} = \frac{g_1^T (g_1 - g_0)}{d_0^T (g_1 - g_0)} = \frac{g_1^T g_1}{g_0 g_0} \tag{2.20}$$

In general, in the k-th iteration, set

$$d_k = -g_k + \sum_{i=0}^{k-1} \beta_i d_i \tag{2.21}$$

Choosing β_i such that ${d_k}^T A d_i = 0, i = 0, ..., k-1$ and noticing from Theorem 1.3 that

$$g_k T d_i = 0, g_k^T g_i = 0 i = 0, ..., k-1, \tag{2.22}$$

it follows from multiplying (2.21) by $d_j^T A, \quad j = 0, ..., k-1$ that

$$\beta_j = \frac{g_k^T A d_j}{d_j^T A d_j} = \frac{g_k^T (g_{j+1} - g_j)}{d_j (g_{j+1} - g_j)}, \quad j = 0, ..., k-1 \tag{2.23}$$

Then

$$\beta_j = 0, j = 0, 1, , k2 \tag{2.24}$$

$$\beta_{k-1} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{2.25}$$

The above derivation establishes the iterative scheme of the conjugate gradient method:

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.26}$$

$$d_k = -g_k + \beta_{k-1} + d_{k-1} \tag{2.27}$$

Where

$$\beta_{k-1} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \qquad (F - RFormula) \tag{2.28}$$

and α_k is an exact step size, in particular, for the quadratic case,

$$\alpha_k = \frac{-g_k d_k}{d_k A d_k} \tag{2.29}$$

Theorem 2.2. *(Property theorem of conjugate gradient method) For positive definite quadratic function (2.13), the conjugate gradient method (2.25)- (2.27) with exact line searches terminates after $m \leq n$ steps, and the following properties hold for all $i, (0 \leq i \leq m)$,*

$$d_i^T A d_j = 0, j = 0, ..., i-1 \tag{2.30}$$

$$g_i^T g_j = 0, j = 0, ..., i-1 \tag{2.31}$$

$$d_i^T g_i = -g_i^T g_i \tag{2.32}$$

$$[g_0, g_1, ..., g_i] = [g_0, Ag_0, ..., A^i g_0] \tag{2.33}$$

$$[d_0, d_1, ...d_i] = [g_0, Ag_0, ..., A^i g_0] \tag{2.34}$$

where m is the number of distinct eigenvalues of A

Proof. We prove (2.29)(2.31) by induction. For$i = 1$, it is trivial. Suppose (2.29)(2.31) hold for some$i < m$. We show that they also hold for$i + 1$. For quadratic function (2.13), we have obviously

$$g_{i+1} = g_i + A(x_{i+1} - x_i) = g_i + \alpha_i A d_i \tag{2.35}$$

From (2.28) α_i can be written as

$$\alpha_i = \frac{g_i^T g_i}{d_i^T A d_i} \neq 0 \tag{2.36}$$

using (2.35) and (2.26) gives

$$\begin{aligned} g_{i+1}^T g_i &= g_i^T g_j + \alpha_i d_i^T A g_j \\ &= g_i^T g_j - \alpha_i d_i^T A(d_j - \beta_{j-1} d_{j-1}) \end{aligned} \tag{2.37}$$

when$j = 1$ (2.35) becomes

$$g_{i+1}^T g_i = g_i^T g_i - \frac{g_i^T g_i}{d_i^T A d_i} d_i^T A d_i = 0$$

when $j \leq i$ (2.37) is zero directly by induction hypothesis.So(2.30) follows
Now from (2.26) and (2.35) it follows that

$$\begin{aligned} d_{i+1}^T A d_j &= -g_{i+1}^T A d_j + \beta_i d_i^T A d_j \\ &= \frac{-g_{i+1}^T (g_j - g_{j+1})}{\alpha_j + \beta_i d_i^T A d_j} \end{aligned} \tag{2.38}$$

When$j = i$, it follows from (2.38), (2.30), (2.35) and (2.27) that

$$d_{i+1}^T A d_j = -\frac{g_{i+1}^T g_{i+1}}{g_i^T g_i} d_i^T A d_i + \frac{g_{i+1}^T g_{i+1}}{g_i^T g_i} d_i^T A d_i = 0$$

When$j < i$, (2.38) is also zero from induction hypothesis. Then (2.29) follows. Also, from (2.26) and the exact line search, we have

$$\begin{aligned} d_{i+1}^T g_{i+1} &= g_{i+1}^T g_{i+1} + \beta_i d_i^T g_{i+1} \\ &= g_{i+1}^T g_{i+1} \end{aligned} \tag{2.39}$$

which shows (2.31) holds for $i+1$. Finally, we show (2.32) and (2.33) by induction. It is trivial for $i = 0$. Now suppose they hold for some i, and we prove that they hold also for $i+1$. From the induction hypothesis, both g_i and Ad_i belong to

$$[g_0, Ag_0, ..., A^i g_0, A^{i+1} g_0]$$

Then it follows from (2.35) $g_{i+1} \in [g_0, Ag_0, ..., A^i g_0, A^{i+1} g_0]$. Furthermore, we need to show

$$g_{i+1} \notin [g_0, Ag_0, ..., A^i g_0] = [d_0, d_1, ...d_i]$$

In fact, since vectors$d_0, ..., d_i$ are conjugate, it follows from Theorem 4.1.3 that$g_{i+1}[d_0, ..., d_i]$. If $g_{i+1} \in [g_0, Ag_0, ..., A^i g_0] = [d_0, ..., d_i]$, then it results in $g_{i+1} = 0$. This is a contradiction. Therefore (2.32) follows. Similarly, by (2.26) and induction hypothesis, we can get (2.33). □

If exact line search was used in the previous iteration, then $g_k^T d_{k-1} = 0$ and hence $g_k^T d_k = -g_k^T g_k \leq 0$ which guarantees that d_k is a descent direction. However, if inexact line search was used in the previous iteration, the quantity$\beta_{k-1} g_k^T d_{k-1}$ may be positive and larger than $-g_k^T g_k$ consequently $-g_k^T g_k + \beta_{k-1} g_k^T g_{k-1}$ is possibly larger than zero. In this case d_k will not be a descent direction. A typical remedy for such an eventuality is to restart the algorithm with d_k as the steepest descent direction g_k. However, frequently setting d_k to the steepest descent direction will lessen the efficiency of the algorithm, and make the behavior of the algorithm incline to a steepest descent method. This situation requires care. The following control measure can be used to overcome this difficulty. Let $\bar{g}_{k+1}$, $\bar{d}_{k+1}$, and $\bar{\beta}k$ denote the computed values of g_{k+1}, d_{k+1}, and βk at $x_k + \alpha^j d_k$ respectively, where $\{\alpha^j\}$ is a test step size sequence generated from a step size algorithm. If

$$-\bar{g}_{k+1} + \bar{d}_{k+1} \geq \sigma ||\bar{g}_{k+1}|| ||\bar{d}_{k+1}||, \tag{2.40}$$

where σ is a small positive number, then α^j is accepted as α_k. If (2.40) is not satisfied at any trial points, we will use exact line search to produce α_k. The following algorithm is a restart conjugate gradient method with exact line search.

Algorithm 3. (Conjugate Gradient algorithm using F-R update)
Initial step: Given $x_0 \in \mathbb{R}^n$ an initial point, $\epsilon > 0$ a termination scalar

1 Set $k := 0$ compute $g_0 = g(x_0)$.

2 If $||g_0|| \leq \epsilon$, stop.otherwise set $d_0 = -g_0$.

3 Compute the step size α_k using line search.

4 Set $x_{k+1} = x_k + \alpha_k d_k, k := k + 1$.

5 Compute $g_k = g(x_k)$. If $||g_k|| \leq \epsilon$, stop. otherwise go to step 6

6 If$k = n$ set $x_0 = x_k$, and go to step 1, otherwise go to step 7.

7 Compute $\beta_k = \dfrac{g_k^T g_k}{g_{k-1}^T g_{k-1}}, d_k = -g_k + \beta_{k-1} d_{k-1}$.

8 If $d_k^T g_k > 0$ set $x_0 = x_k$, and go to step 1 otherwise go to step 3.

2.5.2 Hager and Zhang β_k update

Another β_k update formula is given by

$$\beta_k^N = \left(y_k - 2d_k \frac{||y_k||^2}{d_k^T y_k} \right) \frac{g_{k+1}}{d_k^T y_k} \tag{2.41}$$

Where the step size α_k in (2.2) is obtained by a line search , and the direction d_k are generated by the rule;

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad d_0 = -g_0 \tag{2.42}$$

The above β_k update formula is proposed by Hager and Zhang [73]. To make the iteration to converge we make the lower bound to β_k as follows.

$$d_{k+1} = -g_{k+1} \bar{\beta_k}^N d_k, \quad do = -g_0 \tag{2.43}$$

$$\bar{\beta_k}^N d_k = max\{\beta_k^N d_k, \eta_k\}, \quad \eta_k = \frac{-1}{||d_k|| min\{\eta, ||g_k||\}} \tag{2.44}$$

where $\eta > 0$ is a constant; we take 0.1 in the experiments. An attractive feature of the new conjugate gradient scheme is that the search directions always yield descent when $d_k^T y_k \neq 0$, a condition which is satisfied when f is strongly convex, or the line search satisfies the Wolfe conditions. convergence analysis of this method is given below.

2.6 Convergence Analysis and Rate of Convergence

2.6.1 Global Convergence of F-R, PRP and Hager and Zhang

This subsection is divided into two parts. The first part discusses the global convergence of conjugate gradient methods with exact line search, and consists of three theorems which state respectively global convergence of Fletcher-Reeves (F-R) conjugate gradient method, and Polak-Ribi‘ere-Polyak (PRP) conjugate gradient method. The second part discusses the global

convergence of F-R conjugate gradient method with inexact line search. Now, we start the discussion by proving the global convergence result of F-R method in the case of exact line search.

Theorem 2.3. *(Global convergence of F-R conjugate gradient method) Suppose that$f : R^n \to$ Ris continuously differentiable on a bounded level set $L = x \in R^n | f(x) \leq f(x0)$, and that F-R conjugate gradient method is implemented with exact line search. Then the produced sequence xk has at least one accumulation point which is a stationary point, i.e.,*

(1) when $\{x_k\}$ is a finite sequence, then the final point x^* is a stationary point of f;

(2) when $\{x_k\}$is an infinite sequence, it has limit point, and any limit point is a stationary point.

Proof. (1) When $\{x_k\}$ is finite, from the termination condition, it follows that the final point x^* satisfies $\nabla f(x^(*)) = 0$, and hence x^* is a stationary point of f.
(2) When $\{x_k\}$ is infinite, we have $\nabla f(x_k \neq 0), \forall k$. Noting that $d_k = -g_k + \beta_{k-1} d_{k-1}$ and $g_k^T d_{k-1} = 0$ by exact line search, we have

$$g_k^T d_k = -||g_k||^2 + \beta_{k-1} g_k^T d_{k-1} = -||g_k||^2 \leq 0 \tag{2.45}$$

which means that d_k is a descent direction, $f(x_k)$ is a monotone descent sequence, and thus $x_k \subset L$. Therefore x_k is a bounded sequence and must have a limit point. Let x^* be a limit point of x_k. Then there is a subsequence $\{x_k\}k_1$ converging to x^* , where k_1 is an index set of a subsequence of $\{x_k\}$.. Since $\{x_k\}k_1 \subset \{x_k\}$, $f(\{x_k\}k_1) \subset f(\{x_k\})$. It follows from the continuity of f that for $k \in k_1$,

$$f(x^*) = f(\lim_{k\to\infty} x_k) = \lim_{k\to\infty} f(x_k) = f^* \tag{2.46}$$

Similarly, $\{x_{k+1}\}$ is also a bounded sequence. Hence there exists a subsequence $\{x_{k+1}\}k_2$ converging to $\bar{x^*}$, where k_2 is an index set of a subsequence of x_{k+1}. In this case,

$$f(\bar{x}^*) = f(\lim_{k\to\infty} x_{k+1}) = \lim_{k\to\infty} f(x_{k+1}) = f^* \tag{2.47}$$

Then

$$f(\bar{x}^*) = f(x^*) = f^* \tag{2.48}$$

Now we prove $\nabla f(x^*) = 0$ by contradiction. Suppose that $\nabla f(x^*) = 0$, then, for α sufficiently small, we have

$$f(x^* + \alpha d^*) < f(x^*) \tag{2.49}$$

since

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq f(\alpha d_k), \forall \alpha > 0,$$

then for$k \in K_2$, passing to limit $k \to \infty$ and using (2.34), we get

$$f(\bar{x}^*) \leq f(x^* + \alpha d^*) < f(x^*) \tag{2.50}$$

which contradicts (2.48). This proves $\nabla f(x^*) = 0$ i.e., x^* is a stationary point of f. □

Theorem 2.4. *Let $f(x)$ be twice continuously differentiable and the level set $L = \{x \in R^n | f(x) < f(x)\}$ be bounded. Suppose that there is a constant $m > 0$ such that for $x \in L$,*

$$m||y||^2 \leq y^T \nabla^2 f(x) y, \forall y \in R^n \tag{2.51}$$

Then the sequence $\{x_k\}$ generated by PRP method with exact line search converges to the unique minimizer x^* of f.

Proof. From Theorem 2.2.4,[20, p-79] we know that it is enough to prove that, there exists a constant $\rho > 0$ such that

$$-g_k^T d_k \geq \rho ||g_k|| ||d_k|| \tag{2.52}$$

which means

$$\cos\theta_k \geq \rho > 0$$

Then, from Theorem 2.2.4,[20, p-79] we have $g_k \longrightarrow 0$ and $g(x^*) = 0$. From (2.51) it follows that $\{x_k \longrightarrow x^*\}$ which is a unique minimizer. By using $g_k^T d_{k1} = 0$ and (2.26), we have

$$g_k^T d_k = -||g_k||^2$$

Then 2.37 is equivalent to

$$\frac{||g_k||}{||d_k||} \geq \rho \tag{2.53}$$

From (2.28) and (2.26), it follows that

$$\alpha_{k-1} = -\frac{g_{k-1}^T d_{k-1}}{d_{k-1}^T G_{k-1} d_{k-1}} = \frac{||g_{k-1}||^2}{d_{k-1}^T G_{k-1} d_{k-1}} \tag{2.54}$$

where

$$G_{k-1} = \int_0^1 G(x_{k-1} + t\alpha_{k-1}) dt \tag{2.55}$$

By (2.55), the integral form of the mean-value theorem is

$$g_k - g_{k-1} = g(x_{k-1} + \alpha_{k-1} d_{k-1}) - g(x_{k-1}) = \alpha_{k-1} G_{k-1} d_{k-1} \tag{2.56}$$

Then, by (2.55) and (2.54), (2.12) becomes

$$\begin{aligned} \beta_{k-1} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T g_{k-1}} &= \alpha_{k-1} \frac{g_k^T G_{k-1} d_{k-1}}{||g_{k-1}||^2} \\ &= \frac{g_k^T G_{k-1} d_{k-1}}{d_{k-1}^T G_{k-1} d_{k-1}} \end{aligned} \tag{2.57}$$

Since the level set L is bounded, there is a constant $M > 0$, such that

$$y^T DG(x) y \leq M||y||^2, x \in L, \forall y \in R^n \tag{2.58}$$

Then, by (2.57), (2.58) and (2.51), we have

$$|\beta_{k-1}| \leq \frac{||g_k|| ||G_{k-1} d_{k-1}||}{m||d_{k-1}||^2} \leq \frac{M}{m} \frac{||g_k||}{||d_{k-1}||} \tag{2.59}$$

Therefore

$$\begin{aligned} ||d_k|| &\leq ||g_k|| + |\beta_{k-1}|||d_{k-1}|| \\ &\leq ||g_k|| + \frac{M}{m}||g_k|| \\ &= (1 + \frac{M}{m})||g_k|| \end{aligned} \tag{2.60}$$

which gives

$$\frac{||g_k||}{||d_k||} \geq (1 + \frac{M}{m})^{-1} \tag{2.61}$$

The above inequality shows that (2.53) holds □

Theorem 2.5. *if* $d_k^T y_k \neq 0$ *and*

$$d_{k+1} = -g_{k+1} + \tau d_k, d_0 = -g_0 \tag{2.62}$$

for any $\tau \in [\beta_k^N, max\{\beta_k^N, 0\}]$ *, then*

$$g_k^T d_k \leq -\frac{7}{8}||g_k||^2 \tag{2.63}$$

Proof. Since $d_0 = -g_0$ we have $g_0^T = -||g_0||^2$ which satisfies (2.63). Suppose $\tau = \beta_k^N$. Multiplying (2.62) by g_{k+1}^T we have

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -||g_{k+1}||^2 + \beta_k^N g_{k+1}^T d_k \\ &= -||g_{k+1}||^2 + g_{k+1}^T d_k \left(\frac{y_k^T g_{k+1}}{d_k^T y_k} - 2\frac{||y_k||^2 g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \\ &= \frac{y_k^T g_{k+1}(d_k^T y_k)(g_{k+1}^T d_k) - ||g_{k+1}||^2 (d_k^T y_k)^2 - 2||y_k||^2 (g_{k+1}^T d_k)^2}{(d_k^T y_k)^2} \end{aligned} \tag{2.64}$$

We apply the inequality

$$U^T V \leq \frac{1}{2}(||U||^2 + ||V||^2)$$

to the first term in (2.64) with

$$U = \frac{1}{2}(d_k^T y_k) g_{k+1} \quad and \quad V = 2(g_{k+1}^T d_k) y_k$$

to obtain (2.63). On the other hand, if $\tau \neq \beta_k^N$, then $\beta_k^N \leq \tau \leq 0$. After multiplying (2.62) by g_{k+1}^T, we have

$$g_{k+1}^T d_{k+1} = -||g_{k+1}||^2 + \tau g_{k+1}^T d_k$$

If $g_{k+1}^T d_k \geq 0$, then (2.63) follows immediately since $\tau \leq 0$. If $g_{k+1}^T d_k < 0$, then

$$g_{k+1}^T d_k = -||g_{k+1}||^2 + \tau g_{k+1}^T d_k \leq -||g_{k+1}||^2 \beta_k^N g_{k+1}^T d_k$$

since $\beta_k^N \leq \tau \leq 0$ Hence, (2.63) follows by our previous analysis.
By taking $\tau = \beta_k^N$, we see that the directions generated by (2.2) and (2.42) are descent directions. Since η_k in (2.44) is negative, it follows that

$$\bar{\beta_k}^N = max\{\beta_k^N, \eta_k\} \in [\beta_k^N, max\{\beta_k^N, 0\}]$$

Hence, the direction given by (2.43) and (2.45) is a descent direction. Dai and Yuan [20, 21] present conjugate gradient schemes with the property that $d_k^T g_k < 0$ when $d_k^T y_k > 0$. If f is strongly convex or the line search satisfies the Wolfe conditions, then $d_k^T y_k > 0$ and the Dai/Yuan schemes yield descent. Note that in (2.7) we bound $d_k^T g_k$ by $-\frac{7}{8}||g_k||^2$, while for the schemes [20, 21], the negativity of $d_k^T g_k$ is established. □

Lemma 2.1. *Suppose that d_k is a descent direction and ∇f satisfies the Lipschitz condition*

$$||\nabla f(x) - \nabla f(x_k)|| \leq L||x - x_k||$$

for all x on the line segment connecting x_k and x_{k+1}, where L is a constant. If the line search satisfies the Goldstein conditions, then

$$\alpha_k \geq \frac{1-\delta_1}{L}\frac{|g_k^T d_k|}{||d_k||^2} \tag{2.65}$$

If the line search satisfies the Wolfe conditions, then

$$\alpha_k \geq \frac{1-\sigma}{L}\frac{|g_k^T d_k|}{||d_k||^2} \tag{2.66}$$

Proof. See [12, 13]. □

We now prove convergence of the unrestricted scheme (2.2) and (2.42) with $\beta_k = \beta_k^N$ when f is strongly convex.

Theorem 2.6. *Suppose that f is strongly convex and Lipschitz continuous on the level set*

$$L = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\} \tag{2.67}$$

That is, there exists constants L and $\mu > 0$ such that

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y|| \tag{2.68}$$

$$\mu||x - y||^2 \leq (\nabla f(x) - \nabla f(y))(x - y)$$

for all x and $y \in L$. If the conjugate gradient method (2.2)(2.3) is implemented using a line search that satisfies either the Wolfe or the Goldstein conditions in each step, then either $g_k = 0$ for k or

$$\lim_{k\to\infty} g_k = 0 \tag{2.69}$$

Proof. Suppose that $g_k \neq$ for all k. By the strong convexity assumption

$$y_k^T d_k = (g_{k+1} - g_k)^T d_k \geq \mu\alpha_k||d_k||^2 \tag{2.70}$$

Theorem 2.5 and the assumption $g_k \neq 0$ imply that $d_k \neq 0$. Since $\alpha_k > 0$, it follows from (2.17) that$y_k^T d_k > 0$. Since f is strongly convex over L, f is bounded from below. After summing over k the upper bound in (1.1) or (1.2), we conclude that

$$\sum_{k=0}^{\infty} \alpha_k g_k^T d_k > -\infty$$

Combining this with the lower bound for α_k given in Lemma 2.1 and the descent property (2.63) gives

$$\sum_{k=0}^{\infty} \frac{||g_k||^4}{||d_k||^2} \leq \infty \tag{2.71}$$

By Lipschitz continuity (2.68),

$$||y_k|| = ||g_{k+1} - g_k|| = ||\nabla f_(x_k\alpha_k + d_k) - \nabla f(x_k)|| \leq L\alpha_k||d_k|| \tag{2.72}$$

Utilizing (2.70) and (2.42), we have

$$\begin{aligned} |\beta_k^N| &= \left| \frac{y_k^T g_{k+1}}{d_k^T y_k} - 2\frac{||y_k||^2 d_k g_{k+1}}{(d_k^T y_k)^2} \right| \\ &\leq \frac{||y_k||||g_{k+1}||}{\mu\alpha_k||d_k||^2} + \frac{||y_k||^2||d_k||||g_{k+1}||}{\mu^2\alpha_k^2||d_k||^4} \\ &\leq \frac{L\alpha_k||d_k||||g_{k+1}||}{\mu\alpha_k||d_k||^2} + 2\frac{L^2\alpha_k^2||d_k||^3||g_{k+1}||}{\mu^2\alpha_k^2||d_k||^4} \\ &\leq \left(\frac{L}{\mu} + \frac{2L^2}{\mu^2}\right)\frac{||g_{k+1}||}{||d_k||} \end{aligned} \tag{2.73}$$

Hence, we have

$$||d_{k+1}|| \leq ||g_{k+1}|| + |\beta_k^N|||d_k|| \leq \left(1 + \frac{L}{\mu} + \frac{2L^2}{\mu^2}\right)||g_{k+1}||$$

Inserting this upper bound for d_k in (2.71) yields

$$\sum_{k=1}^{\infty} ||g_k||^2 \leq \infty$$

which completes the proof. □

Lemma 2.2. *If the level set (2.67) is bounded and the Lipschitz condition (2.68) holds, then for the scheme (2.43)(2.44) and a line search that satises the Wolfe conditions (1.1)(1.2), we have*
$d_k \neq 0$ *for each* k *and* $\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 < \infty$.
whenever $inf\ \{\|g_k\| : k \geq 0\} > 0$.

Proof. Define $\gamma = inf\ \{\|g_k\| : k \geq 0\}$. Since $\gamma > 0$ by assumption, it follow from the descent Theorem 4 that $d_k \neq 0$ for each k. Since L is bounded,f is bounded from below,and by(1.1)and(2.66),
$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty$.
Again,the descent property yields

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} \leq \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \leq \frac{64}{49} \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \tag{2.74}$$

Define the quantities:
$\beta_k^+ = max\{\bar{\beta}_k^N, 0\},\ r_k = \frac{-g_k + \beta_{k-1}^- d_{k-1}}{\|d_k\|},\ \delta_k = \beta_{k-1}^+ \frac{\|d_{k-1}\|}{\|d_k\|}$.

By(2.43)(2.44),we have

$$u_k = \frac{d_k}{\|d_k\|} = \frac{-g_k + (\beta_{k-1}^+ + \beta_{k-1}^-)d_{k-1}}{\|d_k\|} = r_k + \delta_k u_{k-1}.$$

Since the u_k are unit vectors,

$$\|r_k\| = \|u_k - \delta_k u_{k-1}\| = \|\delta_k u_k - u_{k-1}\|.$$

Since $\delta_k > 0$,it follows that

$$\begin{aligned} \|u_k - u_{k-1}\| &\leq \|(1+\delta_k)(u_k - u_{k-1})\| \\ &\leq \|u_k - \delta_k u_{k-1}\| + \|\delta_k u_k - u_{k-1}\| \\ &= 2\|r_k\|. \end{aligned} \tag{2.75}$$

By the definition of β_k^- and the fact that $\eta_k < 0$ and $\bar{\beta}_k^N \geq \eta_k$ in(2.44),we have the following bound for the numerator of r:

$$\begin{aligned} \|-g_k + \beta_{k-1}^- d_{k-1}\| &\leq \|-g_k\| - min\{\bar{\beta}_{k-1}^N, 0\}\|d_{k-1}\| \\ &\leq \|-g_k\| - \eta_{k-1}\|d_{k-1}\| \\ &\leq \|-g_k\| + \frac{1}{\|d_{k-1}\| min\{\eta,\gamma\}}\|d_{k-1}\| \\ &\leq \Gamma + \frac{1}{min\{\eta,\gamma\}} \end{aligned} \tag{2.76}$$

Where

$$\Gamma = \max_{x \in L} \|\nabla f(x)\|. \tag{2.77}$$

Let c denote the expression $\Gamma + \frac{1}{min\{\eta,\gamma\}}$ in(2.76).This bound for the numerator of r_k coupled with(2.75)gives

$$\|u_k - u_{k-1}\| \leq 2\|r_k\| \leq \frac{2c}{\|d_k\|}. \tag{2.78}$$

Finally,squaring(2.78),summing over k,and utilizing(2.74),the proof is com- plete. □

Theorem 2.7. *If the level set(2.67)is bounded and the Lipschitz condition(2.68) holds,then for the scheme(2.43) and (2.44) and a line search that satisfies the Wolfe conditions(1.1)-(1.2),either $g_k = 0$ for some k or*

$$\lim_{k \to \infty} inf\|g_k\| = 0. \tag{2.79}$$

Proof. We suppose that$g_k \neq 0$ for all k,and $\lim_{k\to\infty} inf\|g_k\| > 0$,and we obtain a contradiction. Defining $\gamma = inf\{\|g_k\|\} : k \geq 0$,we have $\gamma > 0$ due to(2.79) and the fact that $g_k \neq 0$ for all k.The proof is divided into 3 steps.

I . A bounded for $\bar{\beta}_k^N$:

By the Wolfe condition $g_{k+1}^T d_k \geq \sigma g_k^T d_k$,we have

$$y_k^T d_k = (g_{k+1} - g_k)^T d_k \geq (\sigma - 1) g_k^T d_k = -(1-\sigma) g_k^T d_k \tag{2.80}$$

By Theorem 2.5,
$-g_k^T d_k \geq \frac{7}{8}\|g_k\|^2 \geq \frac{7}{8}\gamma^2$.

Combining this with(2.80)gives

$$y_k^T d_k \geq (1-\sigma)\frac{7}{8}\gamma^2. \tag{2.81}$$

Also,observe that

$$g_{k+1}^T d_k = y_k^T d_k + g_k^T d_k \leq y_k^T d_k. \tag{2.82}$$

Again,the Wolfe condition gives

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k = -\sigma y_k^T d_k + \sigma g_{k+1}^T d_k. \tag{2.83}$$

Since $\sigma < 1$,we can rearrange(2.83)to obtain
$g_{k+1}^T d_k \geq \frac{-\sigma}{(1-\sigma)y_k^T d_k}$.
Combining this lower bound for $g_{k+1}^T d_k$ with the upper bound(2.82)yields

$$\left| \frac{g_{k+1}^T d_k}{y_k^T d_k} \right| \leq \max\left\{ \frac{\sigma}{1-\sigma}, 1 \right\} \tag{2.84}$$

By the definition of $\bar{\beta}_k^N$ in(2.44),we have
$\bar{\beta}_k^N = \beta_k^N$, if $\beta_k^N \geq 0$ and $0 \geq \bar{\beta}_k^N \geq \beta_k^N \; if \; \beta_k^N < 0$.
Hence,$|\bar{\beta}_k^N| \leq \beta_k^N$ for each k.We now insert the upper bound(2.84)for $|g_{k+1}^T d_k|/|y_k^T d_k|$,the lower bound(2.81)for $y_k^T d_k$,and the Lipschitz estimate(2.72)for y_k into the expression(2.42)to obtain:

$$\begin{aligned} |\bar{\beta}_k^N| &\leq \beta_k^N \qquad (2.85)\\ &\leq \frac{1}{|d_k^T y_k|}\left(|y_k^T| + 2\|y_k\|^(2)\frac{|g_{k+1}^T d_k|}{|y_k^T d_k|} \right) \\ &\leq \frac{8}{7}\frac{1}{(1-\sigma)\gamma^2}\left(L\Gamma\|s_k\| + 2L^2\|s_k\|^2 \max\left\{ \frac{\sigma}{1-\sigma}, 1 \right\} \right), \\ &\leq C\|s_k\|, \end{aligned}$$

Where Γ is defined in (2.77)

$$\begin{aligned} C &= \frac{8}{7}\frac{1}{(1-\sigma)\gamma^2}\left(L\Gamma + 2L^2 \max\left\{ \frac{\sigma}{1-\sigma}, 1 \right\} \right) \qquad (2.86)\\ D &= \max\{\|y-z\| \; y, z \in L\} \qquad (2.87) \end{aligned}$$

Here D is the diameter of L.

II .A bound on the steps s_k:
This is a modied version of [60, Thm. 4.3]. Observe that for any $l \geq k$,

$$X_l - x_k = \sum_{j=k}^{l-1} x_{j+1} - x_j = \sum_{j=k}^{l-1} \|s_j\| u_j = \sum_{j=k}^{l-1} \|s_j\| u_k + \sum_{j=k}^{l-1} \|s_j\|(u_j - u_k).$$

By the triangle inequality:

$$\sum_{j=k}^{l-1}\|s_j\| \le \|X_l - x_k\| + \sum_{j=k}^{l-1}\|s_j\|\|u_j - u_k\| \le D + \sum_{j=k}^{l-1}\|s_j\|\|u_j - u_k\|. \tag{2.88}$$

Let Δ be a positive integer, chosen large enough that

$$\Delta \ge 4CD \tag{2.89}$$

where C and D appear in (2.86) and (2.87). Choose k_0 large enough that

$$\sum_{i\ge k_0}\|u_{i+1} - u_i\|^2 \le \frac{1}{4\Delta} \tag{2.90}$$

By Lemma 2.2,k_0 can be chosen in this way. If $j > k \ge k_0$ and $j - k \le \Delta$, then by (2.90) and the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
\|u_{i+1} - u_i\| &\le \sum_{i=k}^{j-1}\|u_{i+1} - u_i\| \\
&\le \sqrt{j-k}\left(\sum_{i=k}^{j-1}\|u_{i+1} - u_i\|^2\right)^{\frac{1}{2}} \\
&\le \sqrt{\Delta}(\frac{1}{4\Delta})^{\frac{1}{2}} = \frac{1}{2}
\end{aligned}$$

Combining this with (2.89) yields

$$\sum_{j=k}^{l-1}\|s_j\| \le 2D, \tag{2.91}$$

When $l > k \ge k_0$ and $l - k \le \Delta$.

II .A bound on the directions d_l:

By (2.43) and the bound on $\bar{\beta}_k^N$ given in Step I, we have

$$\|d_l\|^2 \le (\|g_l\| + \|\bar{\beta}_{l-1}^N\|\|d_{l-1}\|)^2 \le 2\Gamma^2 + 2C^2\|s_{l-1}\|^2\|d_{l-1}\|^2,$$

where Γ is the bound on the gradient given in (2.77). Defining $S_i = 2C^2\|s_i\|^2$, we conclude that for $l > k_0$,

$$\|d_l\|^2 \le 2\Gamma^2\left(\sum_{i=k_0+1}^{l}\prod_{j=i}^{l-1}S_j\right) + \|d_{k_0}\|^2\prod_{j=k_0}^{l-1}S_j \tag{2.92}$$

Above, the product is defined to be one whenever the index range is vacuous. Let us consider a product of Δ consecutive S_j where $k \ge k_0$:

$$\begin{aligned}
\prod_{j=k}^{k+\Delta-1}S_j &= \prod_{j=k}^{k+\Delta-1}2C^2\|S_j\|^2 = \left(\prod_{j=k}^{k+\Delta-1}\sqrt{2}C\|S_j\|\right)^2 \\
&\le \left(\frac{\sum_{j=k}^{k+\Delta-1}\sqrt{2}C\|S_j\|}{\Delta}\right)^{2\Delta} \le \left(\frac{2\sqrt{2}CD}{\Delta}\right)^{2\Delta} \le \frac{1}{2^{\Delta}}
\end{aligned}$$

The rst inequality above is the arithmetic-geometric mean inequality, the second is due to (2.91), and the third comes from (2.89). Since the product of Δ consecutive S_j is bounded by $1/2^{\Delta}$ it follows that the sum in (2.92) is bounded, independent of l. This bound for $\|d_l\|$, independent of $l > k$, contradicts (2.74). Hence,

$$\gamma = \lim_{k\to\infty} inf\|g\| = 0.$$

□

2.7 Rate of Convergence

We have already seen that the conjugate gradient method has quadratic termination, that is, for a convex quadratic function, the conjugate gradient method with exact line search terminates after n iterations.
Note that the conjugate gradient method with exact line search can find the minimizer of a convex quadratic function in at most n iterations, which corresponds to one step of Newton method. Hence we can say that if n iterations of the conjugate gradient method are regarded as a big iteration, the conjugate gradient method should have a similar convergence rate as Newton method. Burmeister [29], and McCormick and Ritter [30] studied the n-step quadratic convergence rate. We now state this result without proof in the following theorem.
Assume that

(A1) $f : \mathbb{R}^n \to \mathbb{R}$ is three times continuously differentiable;

(A2) there exist constants $M > m > 0$ such that

$$m||y||^2 \leq y^T \nabla^2 f(x) y \leq M||y||^2 \forall x \in L$$

where L is a bounded level set.

Theorem 2.8. *Assume that the conditions* $(A1)$ *and* $(A2)$ *are satisfied, then the sequence* $\{x_k\}$ *generated by PRP-CG and F-R-CG restart methods have* n *step quadratic convergence rate, that is, there exists a constant* $c > 0$, *such that*

$$\limsup_{k\to\infty} \frac{||x_{kr+n} - x^*||}{||x_{kr} - x^*||^2} \leq c < \infty$$

where r *means that the methods restarts per* r *iterations.*

Further, Ritter [9] shows that the convergence rate is n step super-quadratic, that is

$$||x_{k+n} - x^*|| = O(||x_k - x^*||^2)$$

The other results on convergence rate of conjugate gradient methods can consult Stoer [6].

2.8 Numerical Examples

In these numerical examples, we need to solve two quadratic and non-quadratic problems and discuss over the result. We want to minimize using our matlab code which is written on appendix D.

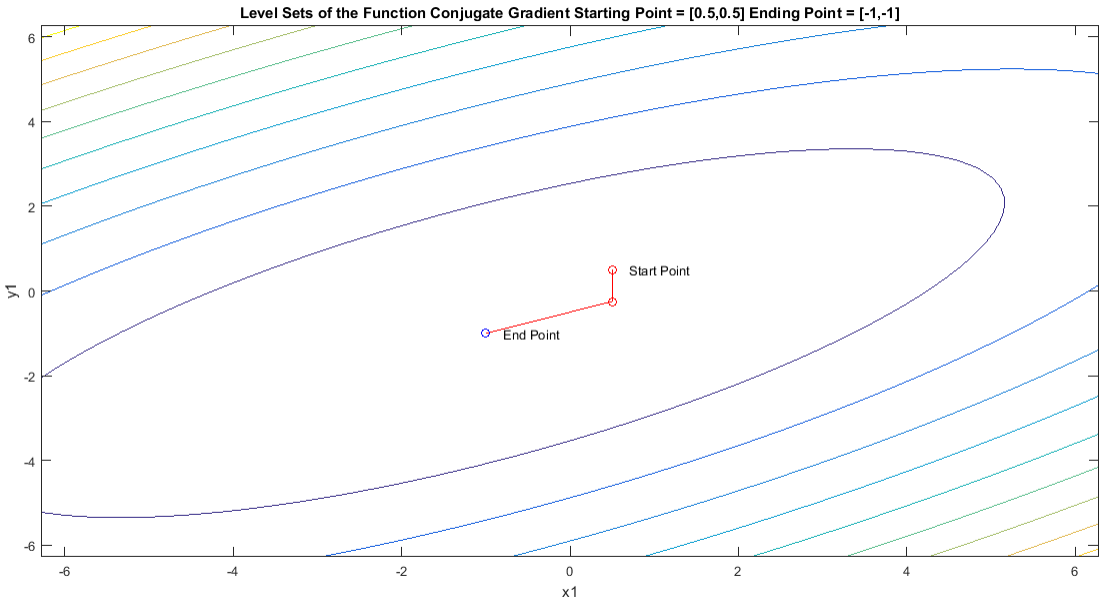
Example 1. *min* $f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 + 2x_2 + 2$,

Solution. We want to minimize using matlab code which is written on appendix D and using secant line search and numerical line search. Let see the result obtained using secant line search and F-R algorithm by taking starting point, $x^0 = [0.5, 0.5]$.

Algorithm: Conjugate Gradient
Linesearch: Secant Linesearch

Iteration	x_current	alpha	||g(xc)||
0.0000	[0.5000 0.5000]	0.2500	1.5000e+00
1.0000	[0.5000 −0.2500]	1.0000	0.0000e+00
2.0000	[−1.0000 −1.0000]	NaN	0.0000e+00

minimum point is $x^* = [-1, -1]$ ◀



Example 2. *min* $f(x) = 3x_1^2 + x_2^2 + 3x_3^2 + 4x_1x_3 + 2x_2x_3 + x_1 - x_3$ *with starting point* $x^0 = [1, 1, 1]$

Solution. The following result is obtained when we solve by using matlab code written in appendix D

Algorithm: Conjugate Gradient
Linesearch: Secant Linesearch

Iteration	x_current	alpha	||g(xc)||
0.0000	[1.0000 1.0000 1.0000]	0.0982	1.2225e+00
1.0000	[−0.0799 0.6073 −0.0799]	1.4604	1.3017e+00
2.0000	[−0.4664 −0.9669 0.6806]	0.4359	5.2925e−17
3.0000	[−1.0000 −1.2500 1.2500]	0.7811	5.2925e−17

◀

Example 3. $minf(x) = (x_1 - 1)^4 + (x_1 - x_2)^2$

Solution. If we solve the above non- quadratic problem by using matlab, with starting point $x^0 = [0, 0]$ the following result is obtained.

Algorithm: Conjugate Gradient
Linesearch: Secant Linesearch

Iteration	x_current	alpha	||g(xc)||
0.0000	[0.0000 0.0000]	0.1026	8.2049e−01
1.0000	[0.4102 0.0000]	0.7112	5.3353e−01
2.0000	[0.5299 0.5835]	0.3373	1.4200e−01
3.0000	[0.7302 0.6644]	0.2843	7.1036e−02
4.0000	[0.7151 0.7018]	3.6196	3.6304e−02
5.0000	[0.9054 0.9174]	0.2619	1.9980e−03
6.0000	[0.9162 0.9152]	0.8202	2.9880e−03
7.0000	[0.9165 0.9168]	7.2845	6.8757e−04

◀

Here the approximate solution is $x^* = [0.9165, 0.9168]$

Chapter 3

Summary

Conjugate gradient methods are a class of important methods for solving unconstrained optimization problem

$$minf(x); x \in \mathbb{R}^n \tag{3.1}$$

especially if the dimension n is large. They are of the form

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.2}$$

where α_k is a step size obtained by a line search, and d_k is the search direction defined by

$$d_k = \begin{cases} -g_k, & \text{for } k = 1. \\ -g_k + \beta_k d_{k-1}, & k \geq 2. \end{cases} \tag{3.3}$$

where $\beta_k$is a parameter that is obtained by one of β_k update method. If we apply a CG method to a quadratic function with exact line search, it converges at n- iterations.

There are several strategies for updating b_k. However we focused only on F-R, PRP and Hager and Zhang update. Various CG method obeys, the search direction obtained is a descent direction and hence we can minimize our function in that direction. The line search strategy is used to find step length and enforces these methods as to converge toward their local minimizer.

Bibliography

[1] M. Al-Baali, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, IMA J. Numer. Anal., 5 (1985),.

[2] A. Buckley, Conjugate gradient methods, in Nonlinear Optimization 1981, M. J. D. Powell, ed., Academic Press, London, 1982,

[3] A. Cohen, Rate of convergence of several conjugate gradient algorithms, SIAM J. Numer. Anal., 9 (1972),

[4] Wilhelm Forst -Dieter Hoffmann: Optimization theory and practice, springer, New York, 2010.

[5] M.S.Bazaraa, H.D .Sherall, C.M.Shetty: Non-linear programming-Theory and algorithms, second edition, 1993.

[6] J. Stoer, On the relation between quadratic termination and convergence properties of minimization algorithms, Part I: Theory, Numerische Mathematik 28 (1977) 343-366.

[7] J.E. Dennis and R.B. Schnabel, Numerical methods for unconstrained optimization, Prentice-Hall, Englewood Cliffs, NJ, 1983

[8] G.R.WALSH, Methods of optimazation, John Wiley & Sons, New york, 1975

[9] S.S. Oren, Self-scaling variable metric algorithm for unconstrained minimization, Ph.D. Dissertation, Computer Science Department, Stanford University, USA, (1972).

[10] Y. H. Dai and Y. Yuan, Convergence properties of the conjugate descent method, Adv. Math. (China), 26 (1996),.

[11] Wenyu Sun, YA-Xing Yuan: Optimization theory and methods, springer, New York, 2006.

[12] W. W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim., 16:170192, 2005.

[13] H. Zhang and W. W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim., 14:10431056, 2004.

[14] Y. H. Dai and Y. Yuan, Further studies on the Polak-Ribiere-Polyak method, Research report ICM-95-040, Institute of Computational Mathematics and Scienti

c/Engineering Computing, Chinese Academy of Sciences, 1995.

[15] Fletcher, R., Powell, M. J. D.A rapidly convergent descent method for minimization, 1963/1964

[16] R. Fletcher, Practical Methods of Optimization vol. 1: Unconstrained Optimization, John Wiley and Sons, New York, 1987.

[17] R. Fletcher and C. Reeves, Function minimization by conjugate gradients, Comput. J., 7 (1964),

[18] R. W. Freund, G. H. Golub, and N. M. Nachtigal, Iterative solution of linear systems, Acta Numerica, (1991), pp. 57-100.

[19] J. C. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, SIAM J. Optim., 2 (1992).

[20] William W. Hager and Hongchao Zhang, A survey of non linear Conjugate gradient methods, 2006.

[21] Wenyu Sun Nanjing, Normal University, Nanjing, China,2006.

[22] Hongchao Zhang,Gradient method for large scale optimization, University of Florida 2006

[23] J. Nocedal, S.J. Wright: Numerical optimization, springer, Berlin, Heidelberg, New york, 2006

[24] Y. H. Dai and Y. Yuan, Nonlinear Conjugate Gradient Methods, Shanghai Science and Technology Publisher, Shanghai,2000.

[25] Y. H. Dai and Y. Yuan, Further studies on the Polak-Ribiere-Polyak method, Research report ICM-95-040, Institute of Computational Mathematics and Scienti

c/Engineering Computing, Chinese Academy of Sciences, 1995.

[26] Won Young Yang, W. Cao, T.S. Chung and J.Morris: Applied Numerical method using matlab, 2005

[27] G. McCormick and K. Ritter, Alternative proofs of the convergence properties of the conjugate gradient method, J. Optimization Theory and Applications 13 (1974) 497-518.

[28] A. Nemirovski,Nonlinear continuous optimization,1999.

[29] W. Burmeister, Die konvergenzordnung des Fletcher-Powell algorithmus, Z. Angew. Math. Mech. 53 (1973) 693-699.

[30] M. R. Hestenes and E. L. Stiefel, Methods of conjugate gradients for solving linear systems, J. Research Nat. Bur. Standards, 49 (1952),

[31] G. H. Golub and D. P. O'Leary, Some history of the conjugate gradient methods and Lanczos algorithms: 1948 - 1976,

APPENDIX

Appendix A: Numerical line search

```
function alpha=numsrch(g,xc,dc)
% NUMSRCH Numerical Search, Performs a Numerical Search algorithm for finding
% a minimum of a function f(x+ad), with initial position xc, direction dc,
% and gradient function g, returning the value of a that minimizes the
% function.
% EXAMPLE:
% (given a function Df, vector xcurr, vector d)
% >>alpha=numsrch(Df,xcurr,?d);

syms x y z
X=symvar(g);
gc=dc.'*subs(g,X,xc);
inc=1.5*10^-2; % Increment Step Size
alpha=inc;
i=0;
while gc<0, % Keep Incrementing alpha while descending
    alpha=alpha+inc;
    gc=dc.'*subs(g,X,xc+alpha*dc');
    if i>1000,
        break;
    end % Max Iteration
end
alpha=alpha-inc;
```

Appendix B: Secant line search

```
function [alpha]=secsrch(f,x,d)
% SECSRCH Secant Search, Performs the Secant Search algorithm for finding
% a minimum of function f(x+ad), with initial position x, direction d,
% and gradient function f, returning the value of a that minimizes the
% function.
% EXAMPLE:
% (given a function Df, vector xcurr, vector d)
% >>alpha=secsrch(Df,xcurr,?d);

syms aa x1 x2 x3
A=symvar(f).';
% Get the function to be minimized
```

```
G=d'*subs(f,A,x'+aa*d);
% Initial conditions 0, .001
a(1)=0;
g(1)=subs(G,a(1));
a(2)=.001;
g(2)=subs(G,a(2));
% Compute first Iteration
a(3)=(a(2)-(a(2)-a(1))/(g(2)-g(1))*g(2));
E(1)=0;
E(2)=0;
E(3)=norm(g(2));
i=3;
% Iterate
while E(i)>(10^-4),
    g(i)=subs(G,a(i));
    i=i+1;
    a(i)=(a(i-1)-(a(i-1)-a(i-2))/(g(i-1)-g(i-2))*g(i-1));
    E(i)=norm(g(i-1));
    if i+3>20,
        break
    end
end
alpha=a(i);
```

Appendix C: Plot

```
function out=pltpts(xnew,xcurr)
% Plots Two points and connects them via red line
plot([xcurr(1),xnew(1)],[xcurr(2),xnew(2)],'r-',xnew(1),xnew(2),...
    'bo',xcurr(1),xcurr(2),'ro');
drawnow update; % Draws current graph now
out = [];
```

Appendix D: Conjugate Gradient

```
function [xn, i]=conjgrad14(f,xc,line)
% CONJGRAD Conjugate Gradient Method. Perform the conjugate gradient method
% for minimizing input function f, starting at location xc. Directions
% are cumputed after each iteration using a method guaranteeing Q
% conjugacy, where f=(1/2)x'Qx?x'b+c. The function outputs the final
% position of the algorithm xn, and the number of iterations i. Line=0
% selects a secant line search, line=1 selects an accurate, but
% computationally heavy numeric line search.
syms x1 x2
% Compute Gradient Function
g=jacobian(f).';
A=symvar(f);
i=0;
xs=xc;
```

```
alpha=0;
% Compute Gradient at xc
gc=subs(g,A,xc);
dc=-gc;
% Plot of Level Sets
figure(1)
ezcontour(f);
%ezsurf(f);
% Output Table Header and Initial Values
if line==0, linstr='Secant Line search';
elseif line==1, linstr='Numeric Line search';
else linstr='Line search Unspecified';
end
fprintf('Algorithm: Conjugate Gradient \nLinesearch: %s \n',linstr);
fprintf('Iteration \t x_current \t\t\t alpha \t\t ||g(xc)|| \n');
text(xc(1)+.2,xc(2),'Start Point')
hold on
% Iterate until stopping condition is met
while i>=0,
    % Alpha computed using a secant line?search
    if line==0
        alpha=secsrch(g,xc,dc);
    elseif line==1
        alpha=numsrch(g,xc,dc);
    else display('Linesearch Unspecified'); break;
    end
    xn=xc+alpha*dc';
    gn=subs(g,A,xn);
    D=norm(gn);
    %E=vpa(D);
    % Output Table

    d1=num2str(i,'%10.4f'); d2=num2str(double(xc(1)),'%10.4f');
    d3=num2str(double(xc(2)),'%10.4f'); d4=num2str(alpha,'%10.4f');
    d6=num2str(double(D),'%10.4e');
    fprintf(' %s \t [%s %s] \t %s \t %s\n',d1,d2,d3,d4,d6);
    % Stopping Condition

    if D<10^(-4) && D ~= 0,
        break
    end

    % Beta Calculation
    beta=(gn.'*gn)/(gc.'*gc);
    i=i+1;
    % Get new direction with reset
    if mod(i,length(A)+1)>0,
        dc=-gn+beta*dc;
```

```
    else
        dc=-gn;
    end
    % Plot the points and connect them with a line.
    pltpts(xn,xc);
    xc=xn;
    gc=gn;
end
% Label Plots
text(double(xn(1)+.2),double(xn(2)),'End Point')
str1='Level Sets of the Function'; str2=' Conjugate Gradient';
str3=' Starting Point = ['; str4=num2str(xs(1));
str5=num2str(xs(2)); str6=']'; str7=','; str8=' Ending Point = [';
str9=num2str(double(xn(1))); str10=num2str(double(xn(2))); str11=']'; str12=',';
str13=strcat(str1,str2,str3,str4,str7,str5,str6,str8,str9,str12,str10,str11);
title(str13); xlabel('x1'); ylabel('y1');
```

Appendix E: Backtracking line search

```
function alpha=blinesearch1(f,xc)

alpham=1; r1=0.5;%choose r1 number between 0 and 1 , alpham>0.
alpha=alpham ;
c=0.0001;
g=jacobian(f);
A=symvar(f);
gc=subs(g,A,xc);
dc=-gc;
fk=subs(f,A,xc);
gp=gc*dc';
xcc=xc+alpha*dc;
cfk=subs(f,A,xcc);
%ind=2;
while (cfk>fk+c*alpha*gp)
    alpha=r1*alpha;
    xcc=xc+alpha*dc;
    cfk=subs(f,A,xcc);
    %ind=ind+1;
end
```