University of California, San Diego

Department of Mathematics

Numerical Methods for Convex Optimization and Their Applications

Author: Jiyue Zeng
Advisors: Mareike Dressler, Martin Licht
June 2021

Preface

Convex optimization studies the optimization problem of minimizing or maximizing convex functions over convex sets. When talking about optimization, the simplest and most well-known example is the linear programming problem. If a problem can be formulated as linear functions with linear inequality and equality constraints, then there exist various tools to solve it efficiently. However, the scope of linear programming is limited and restricted because most real life problems do not follow a linear function and also have non-linear boundaries. Therefore, we would like to expand our toolkit and study a special kind of function, convex function. Many real life problems can be formulated as convex optimization problems. Convex optimization is widely applied in various fields, for example, machine learning, signal processing, computer vision, automatic control system, etc. Since convex functions have nice properties, many reliable and useful numerical methods have been developed to quickly find the minimizer of the function. This thesis is an introduction to some fundamentally important numerical methods for solving convex optimization problems.

We will cover gradient descent method, conjugate gradient method, Newton's method, interior-point method, and finite element method. The interior-point method has been successfully used to solve convex optimization problems in the past 40 years. This thesis explains the intuition of these methods and includes some examples and graphs to illustrate the functionality of each method. For some of the methods, we also analyze their convergence analysis. We state the algorithms of many numerical methods in this thesis and illustrate them by examples programmed in MATLAB.

The aim of the thesis is to provide a general idea about various numerical methods that successfully solve convex optimization problems.

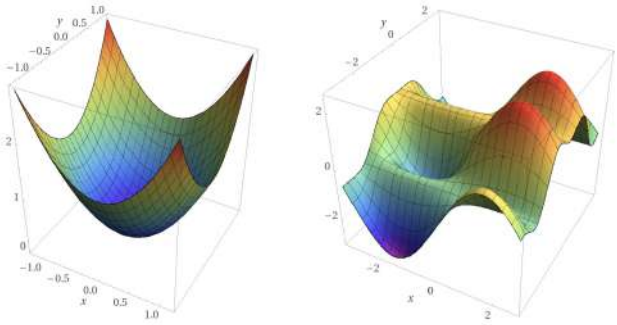
Acknowledgement

My sincere gratitude to my supervisors, Mareike Dressler and Martin Licht. They helped me a lot throughout this project.

Contents

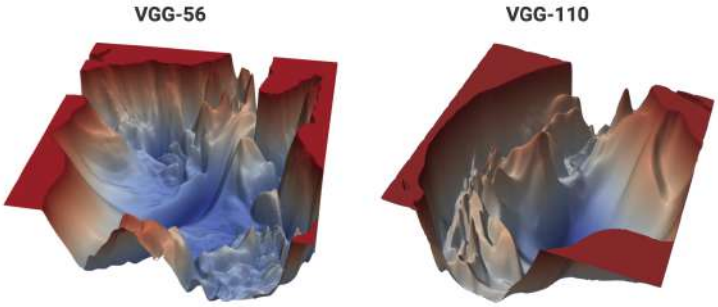
1 Introduction

Why convex functions? Why do we want to study convex optimization? Why is it such an important and interesting topic to explore? Let us first take a look at an example of a convex function and a non-convex function: [1]

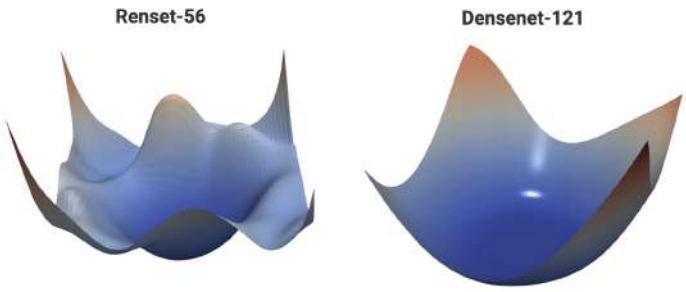


The picture on the left-hand side is a typical convex function which has a bell shape and obviously one unique global minimum. The picture on the right-hand side is a more complicated function. It has some peaks and valleys, which makes it non-convex. It has several local minimum and local maximum, which does not look as nice and simple as the first one. Because convex functions have these nice properties, it is always helpful to use convex functions.

Why studying convex optimization? Numerical methods for convex optimization are useful and powerful tools that help people solve problems in various fields. For example, the following pictures show us how the convex optimization methods can be used to design the network architecture. Pictures on the second row depict two skip connection architectures which cause dramatic "convexification" of loss landscape. [2]



[1] Pictures come from https://www.oreilly.com/radar/the-hard-thing-about-deep-learning/?twitter=@bigdata
[2] Pictures come from https://www.cs.umd.edu/ tomg/projects/landscapes/



It is widely acknowledged that convex optimization are useful in many other fields such as computer engineering, mathematical modeling, statistics etc. Nowadays, there is much active research on this topic.

Brief Summary
We provide a brief summary of the main content of the thesis.

- In the second chapter, we provide a list of background knowledge about vector calculus and convex functions, including essential properties of convex functions, gradient, Hessian matrix, and affine functions.
- In the third chapter, we explain what a convex optimization problem is. We also introduce gradient descent method and conjugate gradient method for symmetric positive definite systems. Several examples and images are presented to describe the main idea.
- The fourth chapter covers Newton's method, including Newton's descent direction, Newton decrement, two phase convergence analysis and some related examples. We also discuss the application of Newton's method to a specific kind of functions, namely the self-concordant functions. The second, third, and fourth chapters finish our discussion about numerical methods for unconstrained convex optimization problems.
- In the fifth chapter, we start from the discussion of equality constrained Newton's method. Then we move on to show the idea of one typical interior-point method, the barrier method, which solves inequality constrained convex optimization problems. Equality constrained Newton's method will play an important role in the development of the barrier method.
- The very last chapter is an outlook. We will briefly talk about the finite element method. This method solves partial differential equations with boundary constraints. We will present one example, the Poisson problem.

For this thesis, we use the book [1] and the paper [3] as major references.

2 Preliminaries

In this section, we introduce all mathematical notations that will be frequently use in this thesis and review background knowledge about vector calculus and convex functions.

2.1 Notation

Here is a list of notations and their meanings:

- $\text{dom}(f)$: the domain of the function f.
- $\|x\|$: the norm of a vector x. The default is 2-norm.
- $\text{span}(B)$: the linear span of a set B of vectors. Usually, B is a set of basis vectors.
- $\mathcal{C}^1$: the set of continuously differentiable functions.
- $\mathcal{C}^2$: the set of twice continuously differentiable functions.
- $\nabla f(x)$: the gradient vector of f evaluated at x.
- $\nabla^2 f(x)$: the Hessian matrix of f evaluated at x.
- Δx: the notation for a descent direction.
- x^*: the optimal solution such that $f(x^*)$ is the minimum value of $f(x)$ over all x in the domain.
- x_i: the point in the domain at the i-th iteration step.
- $X \succeq Y$: X,Y are symmetric matrices of the same dimension and $X - Y$ is a positive semi-definite matrix.
- $\mathbb{1}_S(x)$: the indicator function of the set S where the function gives 1 if $x \in S$, 0 otherwise.
- $\mathcal{N}(A)$: the null space of the matrix A.
- $\mathbb{I}$: the identity matrix.
- $u \succeq 0$: $u_i \geq 0$ entry-wise.

2.2 Affine Sets

Definition 2.2.1. *We say that a set $C \subseteq \mathbb{R}^n$ is **affine** if for every $x, y \in C$ and $\lambda \in \mathbb{R}$, $\lambda x + (1-\lambda)\, y \in C$, i.e., for all $x, y \in C, \alpha, \beta \in \mathbb{R}$, $\alpha x + \beta y \in C$ if $\alpha + \beta = 1$.*

Lemma 2.2.2. *More elements can be added as long as the sum of all coefficients is one. Then we can generalize this definition as follows: if C is an affine set, then for all $x_1, x_2, \ldots, x_n \in C$, $\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n \in C$ if $\lambda_1 + \lambda_2 + \cdots + \lambda_n = 1$.*

Definition 2.2.3. *The linear combination $\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n$ where $\lambda_1 + \lambda_2 + \cdots + \lambda_n = 1$ is called an **affine combination**.*

With this definition, we know that an affine set contains all affine combinations of its elements.

Definition 2.2.4. *A set $S \subseteq \mathbb{R}^n$ is defined to be an **affine subspace** of $\mathbb{R}^n$ if there exist a point $p \in \mathbb{R}^n$ and a subspace $U \subseteq \mathbb{R}^n$ such that*

$$S = p + U = \{p + u \,|\, u \in U\}.$$

Note that since a vector space is closed under addition and scalar multiplication, it is also an affine set. Intuitively speaking, an affine set is developed by shifting the vector space along the direction of a vector away from the origin.

Example 2.2.5. *If $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$, then $C = \{x \in \mathbb{R}^n | Ax = b\}$ is an affine subset of $\mathbb{R}^n$. To see this, choose $x_1, x_2 \in C$. Then we know that $Ax_1 = b$ and $Ax_2 = b$. For an arbitrary $\alpha \in \mathbb{R}$,*

$$A(\alpha x_1 + (1-\alpha)x_2) = \alpha A x_1 + (1-\alpha) A x_2 = b.$$

Hence, for all $\alpha \in \mathbb{R}$, $\alpha x_1 + (1-\alpha)x_2 \in C$, which means that C is an affine set.

Theorem 2.2.6. *Every proper affine subspace V of $\mathbb{R}^n$ has the form $\{x \in \mathbb{R}^n | Ax = b\}$ for some $A \in \mathbb{R}^{m\times n}$ with linearly independent rows and $b \in \mathbb{R}^m$.*

Proof. Since V is an affine subspace, we know that there exists $p \in \mathbb{R}$ and a subspace U of $\mathbb{R}^n$ such that $V = p + U$. Suppose U has a basis $\{u_1, u_2, \ldots, u_k\}$, $k < n$. We want to find an A such that $Ap = b$ and $AU = 0$. Now, we define a matrix $W = [u_1\, u_2 \cdots u_k]$. Then we find a basis $\{a_1, a_2, \ldots, a_m\}$ with $m = n - k$ of the set $\{a_i \in \mathbb{R}^n | W^T a_i = 0\}$. This is verified by the Rank-nullity theorem. We define $A = [a_1\, a_2 \cdots a_m]^T$ and $b = Ap$. Finally, we get $AV = A(p + U) = b$, which gives the result. □

Definition 2.2.7. *For an arbitrary set $S \subseteq \mathbb{R}^n$, the **affine hull**, denoted $\mathrm{aff}(S)$, is a set containing all affine combinations of elements in S:*

$$\mathrm{aff}(S) = \left\{ \sum_{i=1}^{k} \lambda_i x_i \,\middle|\, x_1, x_2, \ldots, x_k \in S, \lambda_1, \lambda_2, \ldots, \lambda_k \in \mathbb{R}, \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

The affine hull $\mathrm{aff}(S)$ is the smallest affine set that contains S.

2.3 Convex Set

Definition 2.3.1. *A set S is a **convex set** if for every two points in S the line connecting these two points is contained in S. Mathematically, S is a convex set if for every $x_1, x_2 \in S, \lambda x_1 + (1-\lambda)x_2 \in S$ for every $0 \le \lambda \le 1$. To generalize this definition, if S is convex, then for every $x_1, x_2, \ldots, x_n \in S$, $\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n \in S$ with $\lambda_1 + \lambda_2 + \cdots + \lambda_n = 1$ and $\lambda_1, \lambda_2, \ldots, \lambda_n \ge 0$.*

From the definition, an affine set is automatically a convex set.

Definition 2.3.2. *The linear combination $\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n$ with $\lambda_1 + \lambda_2 + \cdots + \lambda_n = 1$ and $\lambda_i \ge 0$ for $1 \le i \le n$ is called a **convex combination**.*

Definition 2.3.3. *The **convex hull** of a set S, denoted* $\text{conv}(S)$*, is the set of all convex combinations of elements in S. In other words,*

$$\text{conv}(S) = \left\{ \sum_{i=1}^{n} \lambda_i x_i \mid \sum_{i=1}^{n} \lambda_i = 1, \lambda_1, \ldots, \lambda_n \geq 0, x_1, \ldots, x_n \in S \right\}.$$

The convex hull $\text{conv}(S)$ is the smallest convex set that contains S, i.e., if $S \subseteq U$ and U is a convex set, then we must have $\text{conv}(S) \subseteq U$.

2.4 Cones

Definition 2.4.1. *A set S is a **cone** if for every $x \in S$ and every $\lambda \geq 0, \lambda x \in S$. We say a set S is a **convex cone** if it is convex and is a cone, i.e., for every $x_1, x_2 \in S, \lambda_1, \lambda_2 \geq 0$, we have $\lambda_1 x_1 + \lambda_2 x_2 \in S$.*

Proposition 2.4.2. *The intersection of two convex cones in the same vector space is a convex cone, but the union may not be.*

Proof. Suppose we have two convex cones, S and U. Assume $x, y \in S \cap U$ and $\alpha, \beta \geq 0$. Since S and U are two convex cones, $\alpha x + \beta y \in S$ as well as $\in U$, which means that $\alpha x + \beta y \in S \cap U$. Hence, $S \cap U$ is a convex cone. However, the union may not be. For example, the union of two different lines passing through the origin in $\mathbb{R}^2$. □

2.5 The Gradient and Hessian Matrix

Definition 2.5.1. *The **gradient** of a scalar-valued $\mathcal{C}^1$ function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$,*

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}.$$

To write the gradient as a linear combination of standard basis e_i, we get

$$\nabla f(x) = \frac{\partial f}{\partial x_1}(x)e_1 + \frac{\partial f}{\partial x_2}(x)e_2 + \cdots + \frac{\partial f}{\partial x_n}(x)e_n.$$

It measures how fast the function changes with respect to each standard basis vector e_i.

Theorem 2.5.2. *A differentiable function f increases the fastest along the direction of its gradient.*

Proof. Let u be a unit vector. The dot product $\nabla f(x) \cdot u$ is the directional derivative of f at x along u, which measures the rate of change of f along u. Since $\nabla f(x) \cdot u = ||\nabla f(x)||||u||cos(\theta)$, we choose u to be the unit vector along the direction of the gradient, that is $u = \frac{\nabla f(x)}{||\nabla f(x)||}$. Then $\nabla f(x) \cdot \frac{\nabla f(x)}{||\nabla f(x)||}$ reaches its maximum $||\nabla f(x)||$ since $cos(\theta) = 1$. Hence, $\nabla f(x)$ is the direction that makes the function increase the most. □

Definition 2.5.3. *Suppose we have a $\mathcal{C}^2$ function $f : \mathbb{R}^n \to \mathbb{R}$. The **Hessian matrix** $\mathbf{H_f}$ of f is defined to be*

$$\mathbf{H_f} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}.$$

The Hessian matrix is symmetric because all second order partial derivatives are continuous. $\mathbf{H_f}$ can also be denoted as $\nabla^2 f$.

Definition 2.5.4. *Suppose we have a function $f : \mathbb{R}^n \to \mathbb{R}^m$ whose first-order partial derivatives all exist. The **Jacobian matrix** of f is a $m \times n$ matrix defined as*

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

The gradient is the Jacobian matrix of a scalar-valued function. The Hessian matrix is the Jacobian matrix of the gradient of f, i.e., $\mathbf{H_f} = \mathbf{J}(\nabla f)$.

2.6 Convex function

Definition 2.6.1. *Suppose we have a convex set $S \subseteq \mathbb{R}^n$ and a function $f : S \to \mathbb{R}$. The function f is **convex** if for all $\lambda \in [0, 1]$ and for all $x, y \in S$, we have $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$.*

Note that a function is concave if it satisfies the opposite, i.e., if $f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$, then f is concave.

Definition 2.6.2. *Suppose we have a convex set $S \subseteq \mathbb{R}^n$ and a function $f : S \to \mathbb{R}$. The function f is **strictly convex** if for all $\lambda \in (0, 1)$ and for all $x, y \in S, x \neq y$, we have $f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y)$.*

Theorem 2.6.3. *A local minimum of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is also a global minimum.*

Proof. Suppose we have a convex function $f : \mathbb{R}^n \to \mathbb{R}$ and its local minimum x^*. Then there exists an open ball $B_r(x^*)$ centered at x^* with radius r and for all $x \in B_r(x^*)$, $f(x^*) \leq f(x)$. We choose an arbitrary point $y \neq x$ from the domain of f. Then we choose a constant $\alpha \in (0, 1)$ such that $\alpha x^* + (1 - \alpha)y \in B_r(x^*)$, which means the following:

$$\begin{aligned} &\|\alpha x^* + (1 - \alpha)y - x^*\| \leq r \\ \implies &(1 - \alpha) \|y - x^*\| \leq r \\ \implies &\alpha \geq 1 - \frac{r}{\|y - x^*\|}. \end{aligned}$$

Hence, $\alpha \in (0,1) \bigcap \left[1 - \frac{r}{\|y - x^*\|}, +\infty\right]$. It is easy to see that there exists a possible α. Since f is convex, we know that

$$\begin{aligned} & f(x^*) \leq f(\alpha x^* + (1-\alpha)y) \\ \implies & f(x^*) \leq \alpha f(x^*) + (1-\alpha) f(y) \\ \implies & (1-\alpha) f(x^*) \leq (1-\alpha) f(y) \\ \implies & f(x^*) \leq f(y). \end{aligned}$$

Hence, for any $y \in \operatorname{dom}(f), f(x^*) \leq f(y)$, which tells us that $f(x^*)$ is a global minimum. □

If the function is strictly convex, then it has a unique global minimum. Now we give another way to determine the convexity of a function.

Theorem 2.6.4. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if f is convex along every line, i.e., $g : \mathbb{R} \to \mathbb{R}$, defined by $g(t) = f(x + tv)$, is convex for all $x \in \operatorname{dom}(f), v \in \mathbb{R}^n$.*

Proof. ($\Rightarrow$) Since f is convex over $\mathbb{R}^n$, f is also convex on a line $x+tv$. Hence, g is a convex function. ($\Leftarrow$) For any two points $x, y \in \mathbb{R}^n$, we can find a vector $v \in \mathbb{R}^n$ and $t \in \mathbb{R}$ such that $y = x + tv$. Since g is convex, f is also convex. □

Theorem 2.6.5. *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable. The function f is convex if and only if for all $x, y \in \mathbb{R}^n, f(x) \geq f(y) + \nabla f(y)(x - y)$.*

Proof. ($\Rightarrow$) We choose arbitrary $x, y \in \mathbb{R}^n$. Let $z = \lambda x + (1-\lambda)y$ for some $\lambda \in [0,1]$. Since f is convex, we know that

$$f(z) = f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y).$$

Subtracting $f(y)$ on both sides and then simplifying the equation yield

$$\begin{aligned} & f(y + \lambda(x-y)) - f(y) \leq \lambda f(x) + (1-\lambda) f(y) - f(y), \\ & f(y + \lambda(x-y)) - f(y) \leq \lambda (f(x) - f(y)), \\ & \frac{f(y + \lambda(x-y)) - f(y)}{\lambda} \leq f(x) - f(y), \text{ for } \lambda \in (0,1]. \end{aligned}$$

Taking the limit as λ goes to 0, we get

$$\lim_{\lambda \to 0} \frac{f(y + \lambda(x-y)) - f(y)}{\lambda} = \nabla f(y)(x - y).$$

This computes the directional derivative of f at y in the direction of vector $x - y$. Since the inequality holds for all $\lambda \in (0,1]$, we get for all $x, y \in \mathbb{R}^n$, $f(x) \geq f(y) + \nabla f(y)(x-y)$.
($\Leftarrow$) We choose arbitrary $x, y \in \mathbb{R}^n$ and $\lambda \in [0,1]$. Let $z = \lambda x + (1-\lambda)y$. We know that $f(x) \geq f(z) + \nabla f(z)(x - z)$ and $f(y) \geq f(z) + \nabla f(z)(y - z)$. We multiply λ to the first inequality and $1 - \lambda$ to the second inequality,

$$\begin{aligned} \lambda f(x) &\geq \lambda f(z) + \lambda(1-\lambda)(x-y), \\ (1-\lambda) f(y) &\geq (1-\lambda) f(z) + \lambda(1-\lambda)(y-x). \end{aligned}$$

Adding up these two inequalities, we get

$$\lambda f(x) + (1-\lambda)f(y) \geq f(\lambda x + (1-\lambda)y).$$

Hence, f is convex.

□

Theorem 2.6.6. *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is $\mathcal{C}^2$. The function f is convex if and only if the Hessian matrix of f is positive semi-definite.*

Proof. ($\Rightarrow$) Since f is convex, from Theorem 2.6.5 we know that

$$\forall x, y \in \mathbb{R}^n, f(x) \geq f(y) + \nabla f(y)(x-y).$$

Therefore,

$$\begin{aligned} f(y) + \nabla f(y)(x-y) \leq f(x) &\leq f(y) + \nabla f(x)(x-y), \\ \nabla f(y)(x-y) \leq f(x) - f(y) &\leq \nabla f(x)(x-y), \\ (\nabla f(x) - \nabla f(y))(x-y) &\geq 0. \end{aligned}$$

Since this inequality holds for all $x, y \in \mathbb{R}^n$, we choose $x, x+h \in \mathbb{R}^n$ for an arbitrary $h \in \mathbb{R}^n$ and we have,

$$(\nabla f(x+h) - \nabla f(x))h \geq 0.$$

Since f is twice continuously differentiable, we have for all $t > 0$,

$$\nabla f(x+th) - \nabla f(x) = \nabla^2 f(x)th + r(th), \text{ and } \lim_{t\to 0} \frac{|r(th)|}{|th|} = 0.$$

Then,

$$h^T(\nabla f(x+th) - \nabla f(x)) = h^T \nabla^2 f(x)th + h^T r(th) \geq 0,$$

for all $h \in \mathbb{R}^n$ and $t > 0$. Taking the limit as t goes to 0, we have

$$\begin{aligned} \lim_{t\to 0} \frac{h^T(\nabla f(x+th) - \nabla f(x))}{t} &= \lim_{t\to 0} h^T \nabla^2 f(x)h + \frac{h^T r(th)}{t} \\ &= h^T \nabla^2 f(x)h + \lim_{t\to 0} \frac{h^T r(th)}{t} \\ &= h^T \nabla^2 f(x)h \geq 0. \end{aligned}$$

This tells us that $\nabla^2 f(x) \succeq 0$, which is the notation for a matrix to be positive semi-definite.
($\Leftarrow$) According to the second order Taylor polynomial for f, we have that $\forall x, y \in \mathbb{R}^n$,

$$f(x) = f(y) + \nabla f(x)(y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(z)(y-x),$$

for some z between x and y. Since $\nabla^2 f$ is positive semi-definite, we know that

$$\frac{1}{2}(y-x)^T \nabla^2 f(z)(y-x) \geq 0,$$

which implies that

$$f(x) \geq f(y) + \nabla f(x)(y-x).$$

Theorem 2.6.5 tells us that f is a convex function. □

2.7 Affine function

Definition 2.7.1. *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is **affine** if there exist a linear function $h : \mathbb{R}^n \to \mathbb{R}^m$ and a constant vector $b \in \mathbb{R}^m$ such that $f(x) = h(x) + b$.*

From the definition, an affine function is a linear function plus a constant vector. An affine function is both concave and convex:

$$\begin{aligned} f(\lambda x + (1-\lambda)y) &= h(\lambda x + (1-\lambda)y) + b \\ &= \lambda h(x) + (1-\lambda)h(y) + \lambda b + (1-\lambda)b \\ &= \lambda f(x) + (1-\lambda)f(y). \end{aligned}$$

3 Convex optimization

In this chapter, we explain the formal statement of a convex optimization problem and present two methods, gradient descent method and conjugate gradient method.

3.1 Introduction

Definition 3.1.1. *Suppose we have a convex function $f_0 : S_0 \to \mathbb{R}$, several other convex functions $f_i : S_i \to \mathbb{R}$ for $1 \leq i \leq m$, and an affine function $h : S_n \to \mathbb{R}^m$, where $S_0, S_1, \ldots, S_n$ are convex subsets of a real vector space.*

*A **convex optimization problem** has the following form,*

$$\begin{aligned} \textit{min} \quad & f_0(x) \\ \textit{subject to} \quad & f_i(x) \leq 0, \ \textit{for } 1 \leq i \leq m, \\ & h(x) = 0. \end{aligned}$$

*We define the **feasible set** of this convex optimization problem, denoted by X, to be*

$$X = \left\{ x \in \left(\bigcap_{i=0}^{p} \operatorname{dom}(f_i) \right) \bigcap \operatorname{dom}(h) \,\middle|\, f_i(x) \leq 0, \ \textit{for } 1 \leq i \leq p, \textit{ and } h(x) = 0 \right\},$$

*where $\operatorname{dom}(f)$ is the notation for the domain of the function f. The **optimal point** of this standard convex optimization problem is*

$$x^* = \inf \{ f_0(x) \,|\, x \in X \}.$$

The problem becomes an unconstrained convex optimization problem if there are no constraints $f_i \leq 0$ and $h = 0$.

Note that $h(x) = 0$ can usually be written in the form of $Ax = b$, or $a_i \cdot x = b_i$, where a_i is the i-th row of A and b_i denotes the i-th entry of b.

Since h is affine and f_0, f_i are convex, we know that the whole system is convex, which immediately tells us that local minimum is the optimal solution to this convex problem.

On top of that, we know that the feasible set X must also be convex since it is an intersection of convex domains.

Theorem 3.1.2. *Suppose f_0 is differentiable and convex. The point x is optimal if and only if for all y in the feasible set X, we have $\nabla f_0(x)(y - x) \geq 0$.*

Proof. ($\Leftarrow$) Since $\nabla f_0(x)(y - x) \geq 0$ and f_0 is convex, we know that for all $y \in X$, $f_0(y) \geq f_0(x) + \nabla f_0(x)(y - x) \geq f_0(x)$. Hence, x is optimal.

($\Rightarrow$) We prove by contradiction. Suppose x is optimal and there exists $y \in X$ such that $\nabla f_0(x)(y - x) < 0$. Let $z = (1 - t)x + ty$. Then

$$\lim_{t \to 0} \frac{f_0(z) - f_0(x)}{t} = \lim_{t \to 0} \frac{f_0(x + t(y - x)) - f_0(x)}{t} = \nabla f_0(x)(y - x) < 0.$$

Hence, for a very small t, we can find z very close to x such that $f_0(z) < f_0(x)$, which contradicts with the fact that x is optimal. □

Corollary 3.1.3. *If there is no inequality or equality constraint, i.e., the problem is an unconstrained convex optimization problem, then x is optimal if $\nabla f_0(x) = 0$.*

Proof. If x is optimal, from Theorem 3.1.2, we know that for all $y \in X$, which denotes the feasible set, $\nabla f_0(x)(y - x) \geq 0$. Choose $\lambda > 0$ small enough such that $y = x - \lambda \nabla f_0(x) \in X$. Then we know that

$$\nabla f_0(x)(x - \lambda \nabla f_0(x) - x) = -\lambda ||\nabla f_0(x)||^2 \geq 0.$$

Hence, $\nabla f_0(x) = 0$. □

This corollary gives a stopping criterion for finding the optimal point, i.e., $\|\nabla f(x)\| \leq \epsilon$. Next we consider the unconstrained convex optimization problem.

3.2 Gradient Descent Method

Suppose we have a compact domain S and a convex function $f : S \to \mathbb{R}$ that is $\mathcal{C}^2$. We know that f must attain its minimum in the domain. Then how do we find this optimal point x^*?

We start from an arbitrary point on the graph and want to find a path that leads us to $\min f$. We can denote this searching path by a sequence of x_i in the domain. We define the path by the following equation,

$$x_{i+1} = x_i + t_i \Delta x_i \text{ for } i \geq 1,$$

where x_i is the current step, x_{i+1} is the next step, and $t_i \Delta x_i$ is the step size. The step size consists of a vector Δx_i that indicates the descent direction and a constant t_i that decides the step size. The following graph is an example of such a searching path.

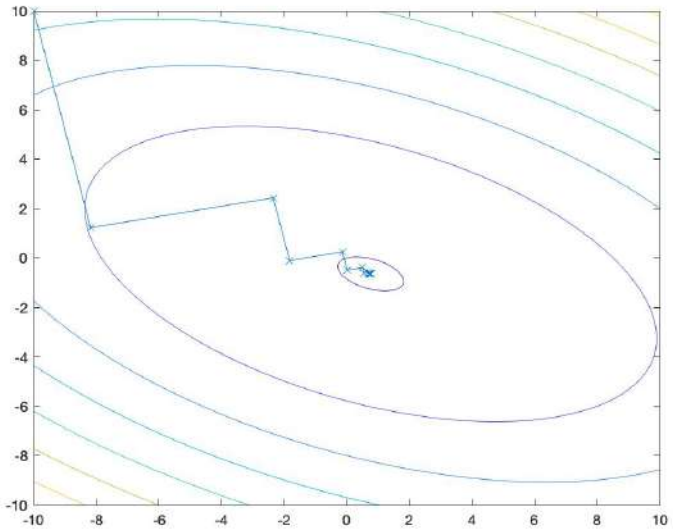


Figure 1: Gradient Descent Method with Exact Line Search

The Gradient Descent Method uses the negative gradient, $-\nabla f(x_i)$, as the descent direction, i.e., $\Delta x_i = -\nabla f(x_i)$. This choice makes sense because Theorem 2.5.2 says that f increases the

fastest along ∇f, which implies that f decreases the fastest along $-\nabla f$. The fact that f is convex tells us that

$$\nabla f(x_i)(x_{i+1} - x_i) \geq 0 \Rightarrow f(x_{i+1}) \geq f(x_i).$$

Therefore, the precondition for obtaining a decreasing sequence is that

$$\nabla f(x_i)(x_{i+1} - x_i) \leq 0. \tag{1}$$

After replacing $(x_{i+1} - x_i)$ by the step size, we have

$$\nabla f(x_i)(-t\nabla f(x_i)) \leq 0,$$

$$-t \left\|\nabla f(x_i)\right\|^2 \leq 0,$$

which is always true if $t \geq 0$. Hence, the precondition for obtaining a decreasing sequence is guaranteed when we use the negative gradient. Notice that $-\nabla f(x_i)$ changes at each step depending on which x_i we use.

Since the direction is already defined, we only need to find the step size t_i. There are two ways to find the constant t_i.

First Way 3.2.1. *Exact Line Search*

Define $t = \{t \geq 0 \,|\, t$ minimizes $f(x_i - t\nabla f(x_i))\}$. This guarantees that $f(x_i - t\nabla f(x_i)) \leq f(x_i)$ because t minimizes the function value. To find t, we take the derivative of $f(x_i - t\nabla f(x_i))$ with respect to t,

$$\begin{aligned} \frac{d}{dt} f(x_i - t\nabla f(x_i)) &= -\nabla f(x_{i+1})^T \nabla f(x_i) \\ &= 0. \end{aligned} \tag{2}$$

Since $\nabla f(x_i)$ is the i-th direction and $\nabla f(x_{i+1})$ is the i+1-th direction, the fact that their dot product is 0 implies that these two directions are orthogonal. Therefore, by using exact line search, we get a zigzag searching path.

Exact line search works well when the function has a nice formula for its gradient so that we can solve for t explicitly. When the cost of computation is expensive, we should switch to the backtracking line search. Here is the algorithm for the gradient descent method with exact line search.

Algorithm 1: Gradient Descent Method with Exact Line Search

GradientDescentExact (f, x_0, ϵ, *max_iter*);
Input : x_0, ϵ, *max_iter*
Output: x^*
$x_i = x_0$;
$t = 0$;
count $= 0$;
while $\|\nabla f(x_i)\| > \epsilon$ & *count* $\leq$ *max_iter* **do**
    count $=$ *count* $+ 1$;
    update t *by solving* $\frac{d}{dt} f(x_i - t\nabla f(x_i)) = 0$;
    $x_i = x_i - t\nabla f(x_i)$;
return x_i;

Second Way 3.2.2. *Backtracking Line Search*

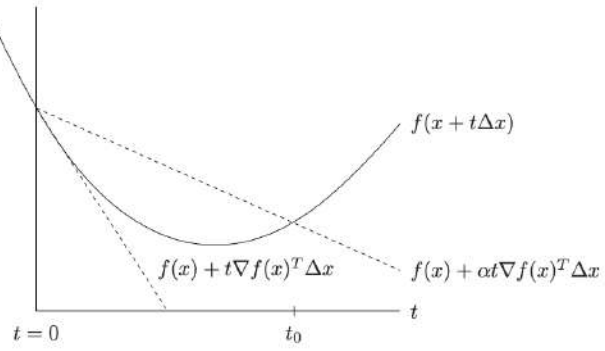


Figure 2: Visualization of Backtracking Line Search

From the graph, we see two lines, $g_1(t) = f(x) + t\nabla f^T\Delta x$ *and* $g_2(t) = f(x) + \alpha t\nabla f^T\Delta x$.[3] g_1 *is the linear approximation of the function* $f(x + t\Delta x)$ *at* $t = 0$. g_2 *has an extra coefficient* α. *The idea in this graph is that we try to start from a large* t *and decrease it until* $t \leq t_0$. *Even though we cannot reach the lowest point, we will get a good amount of decrease in* f. *We need a variable* γ *that decreases* t *at each step and* $\alpha t \|\nabla f(x_i)\|^2$ *is the desired amount of decrease in the function value.* t_0 *is the initial value for* t. *Here is the algorithm for only the backtracking line search.*

Algorithm 2: Backtracking Line Search

initialization: x_i, $x_{i+1} = x_i - t\nabla f(x_i)$, $t = t_0 > 0$, $\gamma \in (0, 1)$, $\alpha \in (0, \frac{1}{2})$;
while $f(x_i) - f(x_{i+1}) < \alpha t \|\nabla f(x_i)\|^2$, **do**
  $t = \gamma t$, $x_{i+1} = x_i - t\nabla f(x_i)$;
end

The while loop will quit after finite iterations because of the following.
By the second order Taylor expansion of the function, we have

$$f(x_{i+1}) = f(x_i) - t\|\nabla f(x_i)\|^2 + \frac{1}{2}t^2\nabla f(x_i)^T\nabla^2 f(y)\nabla f(x_i),$$

for some y *between* x_i *and* x_{i+1}. *When* t *is very small, we know that the last term is of* $\mathcal{O}(t^2)$ *and can be ignored. Then we know for* $\alpha < \frac{1}{2}$,

$$f(x_i) - f(x_{i+1}) = t\|\nabla f(x_i)\|^2 > \alpha t\|\nabla f(x_i)\|^2.$$

We know that t *can be arbitrarily small within finite iterations because* $\gamma < 1$.

To precisely compute the possible range of t, *we assume one more condition. Suppose the largest eigenvalue of* $\nabla^2 f$ *is bounded by* M, *i.e.,* $\nabla^2 f \preceq M\mathbb{I}$. *The stopping criterion for the while loop can be rewritten as*

$$t\|\nabla f(x_i)\|^2 - \frac{1}{2}t^2\nabla f(x_i)^T\nabla^2 f(y)\nabla f(x_i) \geq \alpha t\|\nabla f(x_i)\|^2.$$

After simplifying the inequality, we get

$$(1-\alpha)t\|\nabla f(x_i)\|^2 \geq \frac{1}{2}t^2\nabla f(x_i)^T\nabla^2 f(y)\nabla f(x_i). \tag{3}$$

[3]The picture comes from [1].

With the upper bound on $\nabla^2 f$, we know that

$$\frac{1}{2}t^2\nabla f(x_i)^T\nabla^2 f(y)\nabla f(x_i) \leq \frac{1}{2}t^2 \left\|\nabla f(x_i)\right\|^2 M.$$

If there exists t such that $(1-\alpha)t \left\|\nabla f(x_i)\right\|^2 \geq \frac{1}{2}t^2 \left\|\nabla f(x_i)\right\|^2 M$, then (3) is satisfied. So

$$t \leq \frac{2(1-\alpha)}{M}. \tag{4}$$

Therefore, the while loop must end when $t \in (0, \min\{\frac{2(1-\alpha)}{M}, t_0\})$, which implies that $t \geq \frac{2(1-\alpha)\gamma}{M}$. The algorithm guarantees that there is a decent amount of decrease in f and the step size is not too big or too small Below is the full algorithm for the gradient descent method with backtracking line search. Here we give exact values to α and γ.

Algorithm 3: Gradient Descent Method with Backtracking Line Search

GradientDescentBacktracking (f, x_0, ϵ, *max_iter*);
Input : x_0, ϵ, *max_iter*
Output: x^*
$x_i = x_0$;
$x_{i+1} = x_i - t\nabla f(x_i)$;
$t = 1$;
count $= 0$;
$\alpha = 0.25$;
$\gamma = 0.5$;
while $\|\nabla f(x_i)\| > \epsilon$ & *count* $\leq$ *max_iter* **do**
  count $=$ *count* $+ 1$;
  while $f(x_i) - f(x_{i+1}) < \alpha t \left\|\nabla f(x_i)\right\|^2$, **do**
    $t = \gamma t$;
    $x_{i+1} = x_i - t\nabla f(x_i)$;
  $x_i = x_i - t\nabla f(x_i)$;
return x_i;

3.3 Convergence Analysis of Gradient Descent Method

Assume the function $f : \mathbb{R}^n \to \mathbb{R}$ is strongly convex, which means that we can bound its Hessian matrix, $m\mathbb{I} \preceq \nabla^2 f \preceq M\mathbb{I}$. As before, we let $f(x^*)$ denote the minimum value of f. By the second order Taylor's expansion, we have for all $x, y \in \mathrm{dom}(f)$,

$$\begin{aligned} f(y) &= f(x) + \nabla f(x)(y-x) + \frac{1}{2}(y-x)^T\nabla^2 f(z)(y-x) \\ &\geq f(x) + \nabla f(x)(y-x) + \frac{1}{2}m\left\|y-x\right\|^2. \end{aligned}$$

Consider the right-hand side as a quadratic function with variable $y - x$. It reaches the minimum when $y - x = -\frac{1}{m}\nabla f(x)$. After taking the minimum on both sides, we get

$$\begin{aligned} f(x^*) &\geq f(x) + \nabla f(x)(-\frac{1}{m}\nabla f(x)) + \frac{m}{2}\left\|\frac{1}{m}\nabla f(x)\right\|^2 \\ &= f(x) - \frac{1}{2m}\|\nabla f(x)\|^2. \end{aligned}$$

Therefore,

$$\|\nabla f(x)\|^2 \geq 2m(f(x) - f(x^*)). \tag{5}$$

From the second order Taylor expansion, we get

$$f(x_i - t\nabla f(x_i)) = f(x_i) - t\|\nabla f(x_i)\|^2 + \frac{1}{2}t^2\nabla f(x_i)^T\nabla^2 f(x_i)\nabla f(x_i). \tag{6}$$

Applying the upper bound $M\mathbb{I}$ on (6),

$$f(x_i - t\nabla f(x_i)) \leq f(x_i) - t\|\nabla f(x_i)\|^2 + \frac{1}{2}t^2 M\|\nabla f(x_i)\|^2. \tag{7}$$

Consider both sides as functions of t. Then the right-hand side is simply a real-valued single variable quadratic function whose minimum is obtained at $t = \frac{1}{M}$. The minimum of the left-hand side via exact line search is just $f(x_{i+1})$. Hence, after minimizing both sizes over the variable t and applying (5), we have

$$\begin{aligned} f(x_i + 1) &\leq f(x_i) - \frac{1}{2M}\|\nabla f(x_i)\|^2 \\ &\leq f(x_i) - \frac{m}{M}(f(x_i) - f(x^*)). \end{aligned}$$

Then subtract $f(x^*)$ on both sides,

$$f(x_i + 1) - f(x^*) \leq \left(1 - \frac{m}{M}\right)(f(x_i) - f(x^*)). \tag{8}$$

Since $0 < m/M < 1$, we know that the sequence $\{f(x_i) - f(x^*)\}_{i=1}^{\infty}$ converges to 0. By recursively applying (8), we get

$$f(x_i) - f(x^*) \leq \left(1 - \frac{m}{M}\right)^i (f(x_0) - f(x^*)).$$

Things are sightly different when the backtracking line search is used. Based on the while condition from the algorithm of backtracking line search, we know that

$$f(x_i) - f(x_{i+1}) \geq \alpha t\|\nabla f(x_i)\|^2.$$

We can replace t by $\frac{2(1-\alpha)\gamma}{M}$ from (4) and use (5) again,

$$\begin{aligned} f(x_i) - f(x_{i+1}) &\geq \frac{2\alpha(1-\alpha)\gamma}{M}\|\nabla f(x_i)\|^2 \\ &\geq \frac{4m\alpha(1-\alpha)\gamma}{M}(f(x_i) - f(x^*)). \end{aligned}$$

Subtracting $f(x^*)$ on both sides,

$$f(x_i) - f(x^*) \geq f(x_{i+1}) - f(x^*) + \frac{4m\alpha(1-\alpha)\gamma}{M}(f(x_i) - f(x^*)),$$

$$\left(1 - \frac{4m\alpha(1-\alpha)\gamma}{M}\right)(f(x_i) - f(x^*)) \geq f(x_{i+1}) - f(x^*). \tag{9}$$

We know that $0 < \frac{4m\alpha(1-\alpha)\gamma}{M} < 1$ because $\alpha(1-\alpha) \leq \frac{1}{4}$ and $\frac{m}{M}, \gamma < 1$. Hence, after applying (9) recursively, we get

$$f(x_i) - f(x^*) \leq \left(1 - \frac{4m\alpha(1-\alpha)\gamma}{M}\right)^i (f(x_0) - f(x^*)),$$

which finishes our convergence analysis. Next, we apply the gradient descent method to a quadratic function with exact line search and backtracking line search respectively.

Example 3.3.1. *Suppose we have a convex quadratic function* $f(x) = x^T Ax + bx + c$ *where* $A = \begin{bmatrix} 2 & 1 \\ 5 & 7 \end{bmatrix}, b = \begin{bmatrix} 4 \\ 6 \end{bmatrix}, c = 5$. *The optimal point* x^* *that gives the minimum value of* f *is* $x^* = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$. *We fix a starting point* $x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. *Figure 3 and 4 are graphs of gradient descent method with exact line search and backtracking line search respectively. The blue lines on the graph represent the path of* x_i *starting from* $[1, 1]$, *ending at* $[0, -1]$. *Figure 5 and 6 are the projections of Figure 3 and 4 onto the* x, y *plane.*

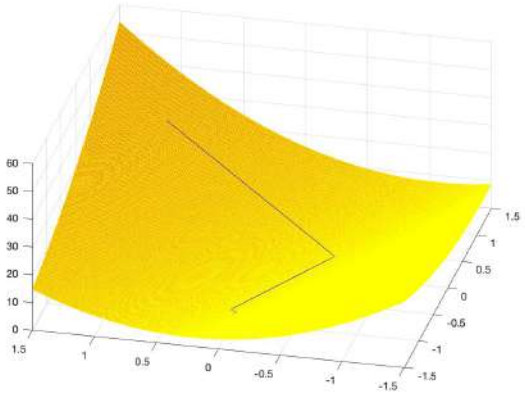


Figure 3: Gradient Descent with Exact Line Search

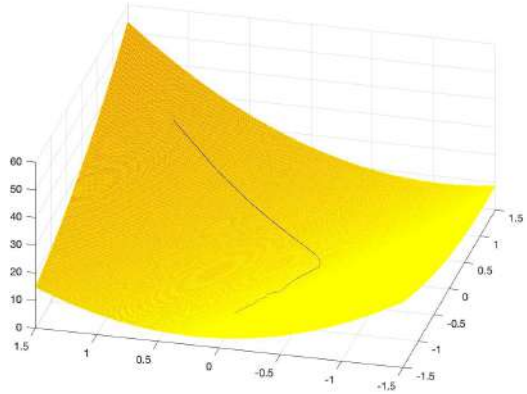


Figure 4: Gradient Descent with Backtracking Line Search

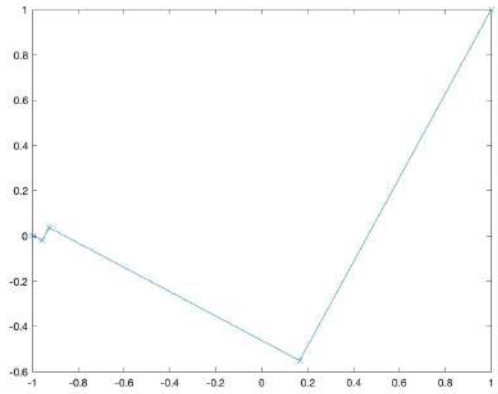


Figure 5: The Searching Path with Exact Line Search

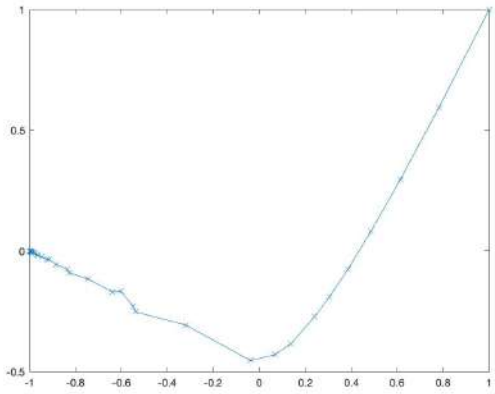


Figure 6: The Searching Path with Backtracking Line Search

From Figure 3 and 5, we see that the searching path of the exact line search has a zigzag shape. Comparing Figure 4 and 6, we also see that exact line search takes much fewer steps than backtracking line search. This is because on each step, exact line search finds the best x_{i+1} that minimize the function value while backtracking line search takes a small range of t that provides some decrease in f.

3.4 Conjugate Gradient Method

We narrow down the problem from minimizing a convex function $f \in \mathcal{C}^2$ to minimizing a quadratic function, $f(x) = \frac{1}{2}x^TAx + bx + c$. Before presenting the conjugate gradient method, we introduce the concept of A-conjugacy.

Definition 3.4.1. *Suppose we have a positive definite square matrix $A \in \mathbb{R}^{n\times n}$. Two nonzero vectors $u, v \in \mathbb{R}^n$ are* ***A-conjugate*** *if $u^TAv = 0$.*

Figure 8 shows two A-conjugate vectors geometrically.

Theorem 3.4.2. *Suppose A is a positive definite $n \times n$ square matrix. If u, v are nonzero vectors that are mutually A-conjugate, then u, v are linearly independent.*

Proof. It suffices to show that $c_1u + c_2v = 0$ implies $c_1 = 0$, $c_2 = 0$. Multiplying Av on both sides, we get $c_1u^TAv + c_2v^TAv = 0$. Since A is positive definite, v is not a zero vector, we get $v^TAv > 0$. Since u, v are A-conjugate, $u^TAv = 0$. Hence, $c_2 = 0$ and then $c_1 = 0$ since u is nonzero. □

If A is the identity matrix, then A-conjugacy is the same as orthogonality. If the matrix A is instead a symmetric positive definite n × n square matrix and if we have a set of n mutually A-conjugate vectors $\{u_i\}_{i=1}^n \in \mathbb{R}^n$ with respect to A, then $\{u_i\}_{i=1}^n$ form a basis of $\mathbb{R}^n$. Any vector $x \in \mathbb{R}^n$ can be written as $x = \sum_{i=1}^n \alpha_i u_i$, $\alpha_i \in \mathbb{R}$.

Theorem 3.4.3. ***Gram-Schmidt Algorithm***

Suppose A is a symmetric positive definite $n \times n$ square matrix. If we start with a sequence of linearly independent vectors $\{v_i\}_{i=1}^n \in \mathbb{R}^n$, we can generate a sequence of mutually A-conjugate vectors $\{u_k\}_{k=1}^n \in \mathbb{R}^n$ from that sequence.

Proof. Set $u_1 = v_1$. Let $u_2 = v_2 + \gamma_2 u_1$, for some $\gamma \in \mathbb{R}$. Since we want u_2, u_1 to be mutually A-conjugate, we multiply Au_1 on both sides:

$$u_2^T A u_1 = {v_2}^T A u_1 + \gamma_2 {u_1}^T A u_1,$$

$$0 = v_2^T A u_1 + \gamma_2 u_1^T A u_1,$$

$$\gamma = -\frac{v_2^T A u_1}{{u_1}^T A u_1}.$$

To finish the induction process, let $u_k = v_k + \sum_{j=1}^{k-1} \gamma_j u_j$. We multiply Au_m, for $m \in \{1, 2, \ldots, k-1\}$, on both sides of the equation:

$$u_k^T A u_m = v_k^T A u_m + \sum_{j=1}^{k-1} \gamma_j u_j^T A u_m.$$

We want u_k to be A-conjugate to all previous u_i for $i \in \{1, 2, \ldots, k-1\}$. Notice that u_i, u_j are already mutually A-conjugate for $i, j \in \{1, 2, \ldots, k-1\}, i \neq j$. Therefore, we obtain

$$0 = v_k^T A u_m + \gamma_m u_m^T A u_m.$$

$$\gamma_m = -\frac{v_k^T A u_m}{u_m^T A u_m}. \tag{10}$$

We can get all coefficients γ_m for $m \in \{1, 2, \cdots, k-1\}$ by doing the same computation. Hence, we have created a sequence of mutually A-conjugate vectors $\{u_k\}_{k=1}^n$. □

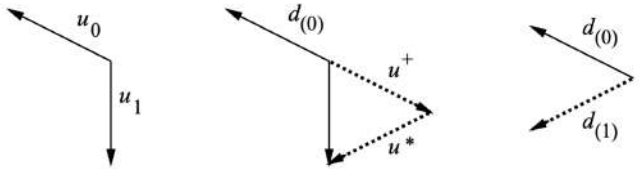


Figure 7: Visualization of Gram-Schmidt Algorithm

Here is a visualization of the Gram-Schmidt Algorithm.[4] We want to remove the component of u_1 that is not A-conjugate to u_0. From this graph, u^+ is removed and u^* is left so that u^* and u_0 are A-conjugate.

Next, we explain the main idea of the conjugate gradient descent method. Suppose we want to minimize a quadratic function $f(x) = \frac{1}{2}x^T Ax + bx + c$, where A is a symmetric positive definite $n \times n$ square matrix. Here we use the same notations x^*, x_i, x_0 as in the last section. We generate

[4]This picture comes from [3].

a set of n mutually A-conjugate vectors $\{u_k\}_{k=1}^n \in \mathbb{R}^n$ via Gram-Schmidt Algorithm. We define the **error** $e_i = x_i - x_0$. We can write e_i as a linear combination of $\{u_k\}_{k=1}^n$, i.e., $e_i = \sum_{k=1}^n \alpha_k u_k$.

The conjugate gradient method says that we just need to remove the error in one direction u_k at each step, that is

$$\begin{aligned} x_0 &= x^* + e_0 = x^* + \sum_{k=1}^{n} \alpha_k u_k, \\ x_1 &= x^* + e_1 = x^* + \sum_{k=2}^{n} \alpha_k u_k, \\ &\vdots \\ x_{n-1} &= x^* + e_{n-1} = x^* + \alpha_n u_n, \\ x_n &= x^*. \end{aligned}$$

After n steps, we get all errors removed since we have traversed through all directions u_k. One question left to be discussed is: why is orthogonal basis not a good choice? The reason is that we need A-conjugacy to compute coefficients α_k.

Suppose instead we have a set of orthogonal basis $\{b_1, b_2, \ldots, b_n\}$, such that $b_i \cdot b_j = 0$, for any $i \neq j$. Now, e_0 can be written as a linear combination of this set, i.e., $e_0 = \sum_{i=1}^n \beta_i b_i$. To compute one coefficient β_i, we multiply b_i^T on both sides:

$$b_i^T e_0 = \sum_{i=1}^{n} \beta_i b_i^T b_i,$$

$$\beta_i = -\frac{b_i^T e_0}{b_i^T b_i}.$$

Here, the problem is that we have no idea what e_0 is. If we know it, $x_0 - e_0$ already gives the optimal solution x^*. However, using A-conjugacy solves this problem.

Now, a new problem arises. From which set of linear independent vectors should we generate $\{u_k\}_{k=1}^n$? The conjugate gradient method suggests using gradients:

$$B_{n-1} = \{\nabla f(x_0), \nabla f(x_1), \cdots, \nabla f(x_{n-1})\}.$$

The gradient can be written in several forms: for some $1 \leq i \leq n-1$,

$$\nabla f(x_i) = Ax_i + b = Ax_i - Ax^* = A(x_i - x^*) = Ae_i. \tag{11}$$

We will explain how to generate the A-conjugate basis from B_{n-1} via (15). We first explain the mechanism of this method.

At each step, we choose one direction and minimize f along that direction. The iteration formula for x_i is $x_{i+1} = x_i + \beta_{i+1}u_{i+1}$. To minimize, we set

$$\begin{aligned}\frac{d}{d\beta_{i+1}}f(x_{i+1}) &= \nabla f(x_{i+1})^T u_{i+1} \\ &= {u_{i+1}}^T \nabla f(x_i + \beta_{i+1}u_{i+1}) \\ &= {u_{i+1}}^T (A(x_i + \beta_{i+1}u_{i+1} - x^*)) \\ &= {u_{i+1}}^T \nabla f(x_i) + \beta_{i+1}{u_{i+1}}^T A u_{i+1} \\ &= 0.\end{aligned}$$

$$\beta_{i+1} = -\frac{{u_{i+1}}^T \nabla f(x_i)}{{u_{i+1}}^T A u_{i+1}}. \tag{12}$$

Since $e_i = \sum_{k=1}^{n} \alpha_k u_k$, we multiply ${u_{i+1}}^T A$ on both side,

$$\begin{aligned}{u_{i+1}}^T A e_i &= \sum_{k=1}^{n} \alpha_k {u_{i+1}}^T A u_k \\ \implies {u_{i+1}}^T A e_i &= \alpha_{i+1}{u_{i+1}}^T A u_{i+1} \\ \implies {u_{i+1}}^T A e_i &= \alpha_{i+1}{u_{i+1}}^T A u_{i+1} \\ \implies \alpha_{i+1} &= \frac{{u_{i+1}}^T A e_i}{{u_{i+1}}^T A u_{i+1}} = \frac{{u_{i+1}}^T \nabla f(x_i)}{{u_{i+1}}^T A u_{i+1}}.\end{aligned} \tag{13}$$

From (12) and (13), we see that $\alpha_{i+1} = -\beta_{i+1}$. This guarantees that once we minimize the function along one direction, we remove the error in that direction completely. Hence, if we set the step size to be $\frac{{u_i}^T \nabla f(x_{i-1})}{{u_i}^T A u_i}$, we can finish the minimization process in at most n steps.

3.4.1 Some Properties

Claim 3.4.4. *$\nabla f(x_i)$ is orthogonal to u_j for all $j \le i$, $i \in \{0, 1, \ldots, n-1\}$, $j \in \{1, \ldots, i\}$.*

Proof. We know that $e_i = \sum_{k=i+1}^{n} \alpha_k u_k$, for $i \in \{0, 1, \ldots, n-1\}$. We multiply ${u_j}^T A$, for some $j \le i$, on both sides:

$${u_j}^T \nabla f(x_i) = {u_j}^T A e_i = \sum_{k=i+1}^{n} \alpha_k {u_j}^T A u_k = 0.$$

□

Claim 3.4.5. *Let $B_{i-1} = \{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_{i-1})\}$, $D_i = \{u_1, u_2, \ldots, u_i\}$, for any $i \in \{1, 2, \ldots, n\}$. Then,* $\operatorname{span}\{B_{i-1}\} = \operatorname{span}\{D_i\}$

Proof. First, we prove that $\{u_1, u_2, \ldots, u_i\} \subseteq \operatorname{span}\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_{i-1})\}$.

$$u_1 = \nabla f(x_0) \in \operatorname{span}\{B_0\},$$

$$u_2 = \nabla f(x_1) + \gamma_1 u_1 = \nabla f(x_1) + \gamma_1 \nabla f(x_0) \in \operatorname{span}\{B_1\},$$

$$\vdots$$

$$u_i = \nabla f(x_{i-1}) + \sum_{k=1}^{i-1} \gamma_k u_k \in \text{span}\{B_{i-1}\}.$$

By definition, u_i is a linear combination of $\nabla f(x_i)$ and $u_1, u_2, \ldots, u_{i-1}$. By recursion, each $u_i \in \text{span}\{B_{i-1}\}$. Hence, the forward direction is proved.

Next, we prove that $\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_{i-1})\} \subseteq \text{span}\{u_1, u_2, \ldots, u_i\}$. This direction is obvious if we move some terms and express $\nabla f(x_i)$ as a linear combination of u_i:

$$\nabla f(x_0) = u_1 \in \text{span}\{u_1\},$$

$$\nabla f(x_1) = u_2 - \gamma_1 u_1 \in \text{span}\{u_1, u_2\},$$

$$\vdots$$

$$\nabla f(x_{i-1}) = u_i - \sum_{k=1}^{i-1} \gamma_k u_k \in \text{span}\{u_1, \ldots, u_i\}.$$

□

From claim 3.4.4 and claim 3.4.5, we know that $\nabla f(x_i)$ is orthogonal to B_{i-1}.

Claim 3.4.6. *Let* $E_{i-1} = \{\nabla f(x_0), A\nabla f(x_0), \ldots, A^{i-1}\nabla f(x_0)\}$. *Then* $\text{span}\{E_{i-1}\} = \text{span}\{B_{i-1}\}$, *for any* $i \in \{1, 2, \ldots, n\}$.

Proof. From (11), we get that

$$\begin{aligned} \nabla f(x_i) &= A(x_i - x^*) \\ &= A(x_{i-1} + \beta_i u_i - x^*) \\ &= \nabla f(x_{i-1}) + \beta_i A u_i. \end{aligned} \tag{14}$$

From Claim (3.4.5) and (14), we know that

$$\nabla f(x_i) = \nabla f(x_{i-1}) + \beta_i A \sum_{k=0}^{i-1} a_k \nabla f(x_k),$$

for some coefficients a_k since u_i can be expressed as a linear combination of $\nabla f(x_k)$. We just need to prove the base cases:

$$\nabla f(x_0) = u_1 \in \text{span}\{E_0\},$$

$$\nabla f(x_1) = \nabla f(x_0) + \beta_1 A u_1 = \nabla f(x_0) + \beta_1 A \nabla f(x_0) \in \text{span}\{E_1\}.$$

Using recursion finishes the proof. □

From claim 3.4.5 and claim 3.4.6, we get the result that $\text{span}\{E_{i-1}\} = \text{span}\{D_i\}$ and $\text{span}\{AD_i\} \subseteq \text{span}\{E_i\} = \text{span}\{B_i\}$. Therefore, $\nabla f(x_{i+1})$ is orthogonal to $\text{span}\{AD_i\}$ for $i \in \{1, 2, \ldots, n-2\}$.

Next, we have proved a key fact that will be very helpful when we generate $\{u_k\}_{k=1}^n$. By using claim 3.4.6, we know that $\nabla f(x_k)^T A u_m = 0$ for all $m \in \{1, 2, \ldots, k-1\}$. Therefore, we can determine all coefficients by Gram-Schmidt Algorithm:

$$\begin{aligned} \gamma_1 = \gamma_2 = \cdots = \gamma_{k-1} = 0, \\ \gamma_k = -\frac{\nabla f(x_k)^T A u_k}{u_k^T A u_k}, \\ u_{k+1} = \nabla f(x_k) + \gamma_k u_k. \end{aligned} \tag{15}$$

Now we present the algorithm of conjugate gradient method. The following algorithm is written with respect to the function $f(x) = \frac{1}{2}x^T A x + bx + c$.

Algorithm 4: Conjugate Gradient Method

ConjugateGradient (A, b, x_0, max_iter);
Input : A, b, x_0, max_iter
Output: x^*
$\beta = 0$;
$\gamma = 0$;
$count = 0$;
$n = size(A)$;
$x = x_0$;
$\nabla f(x) = Ax + b$;
$u = Ax + b$;
while $count \leq n - 1$ *and* $count \leq max\_iter$ **do**
  $count = count + 1$;
  $\beta = -\frac{u^T \nabla f(x)}{u^T A u}$;
  $x = x + \beta u$;
  $\nabla f(x) = Ax + b$;
  $\gamma = -\frac{\nabla f(x)^T A u}{u^T A u}$;
  $u = \nabla f(x) + \gamma u$;
return x;

3.4.2 Simplification of the Algorithm

We can simplify the coefficients β and γ by modifying the following equations:

$$\begin{aligned} x_{i+1} = x_i + \beta_{i+1} u_{i+1}, \\ Ax_{i+1} + b = Ax_i + b + \beta_{i+1} A u_{i+1}, \\ \nabla f(x_{i+1}) = \nabla f(x_i) + \beta_{i+1} A u_{i+1}. \end{aligned}$$

Multiplying $\nabla f(x_{i+1})$ and $\nabla f(x_i)$ to the equation yields

$$\begin{aligned} \|\nabla f(x_{i+1})\|^2 = \beta_{i+1} \nabla f(x_{i+1})^T A u_{i+1}, \\ \|\nabla f(x_i)\|^2 = -\beta_{i+1} \nabla f(x_i)^T A u_{i+1}, \end{aligned} \tag{16}$$

since $\nabla f(x_{i+1}) \cdot \nabla f(x_i) = 0$. From (15), we know that $\nabla f(x_i) = u_{i+1} + \gamma_i u_i$. Since u_{i+1} is A-conjugate to u_i, we combine (15) and (16):

$$\|\nabla f(x_i)\|^2 = -\beta_{i+1} {u_{i+1}}^T A u_{i+1}.$$

The variable γ used in the algorithm can be computed by:

$$\gamma = \frac{\|\nabla f(x_i)\|^2}{\|\nabla f(x_{i-1})\|^2}.$$

From claim 3.4.4 we know that $\nabla f(x_{i-1})$ is orthogonal to u_{i-1} and from (15), we get $u_i = \nabla f(x_{i-1}) - \gamma_i u_{i-1}$. Then the variable β used in the algorithm can be computed in the following way:

$$\beta = -\frac{\nabla f(x_{i-1})^T (\nabla f(x_{i-1}) - \gamma_i u_{i-1})}{u_i^T A u_i} = -\frac{\|\nabla f(x_{i-1})\|^2}{{u_i}^T A u_i}.$$

Notice that we can not change the denominator into $\|\nabla f(x_i)\|^2$ because by the time we compute β, x_i has not been updated yet. i.e., we still do not know the value of $\|\nabla f(x_i)\|^2$.

Hence, we can simplify the above algorithm by replacing β, γ by what we just computed. Since all the setup of function and variables remain the same, we only present the new version of the while loop:

Algorithm 5: Conjugate Gradient Method's While Loop

while *count* $\leq n-1$ *and count* $\leq$ *max_iter* **do**
- $count = count + 1$;
- $\beta = -\frac{\|\nabla f(x)\|^2}{u^T A u}$;
- $x = x + \beta u$;
- $\nabla f(x)_{pre} = \nabla f(x)$;
- $\nabla f(x) = Ax + b$;
- $\gamma = \frac{\|\nabla f(x)\|^2}{\|\nabla f(x)_{pre}\|^2}$;
- $u = \nabla f(x) + \gamma u$;

return x;

Next, we give two applications of the conjugate gradient method.

Example 3.4.7. *In the first example, we use conjugate gradient method to find the optimal point of a quadratic function with a symmetric positive definite matrix. Suppose* $f(x) = \frac{1}{2} x^T A x + bx + c$, *where* $A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}, b = \begin{bmatrix} -2 \\ 5 \end{bmatrix}, c = 3$. $\nabla f(x) = Ax + b$. *The optimal point* x^* *that gives the minimum value of* f *is* $x^* = \begin{bmatrix} \frac{11}{7} \\ \frac{-19}{14} \end{bmatrix}$. *We choose an initial starting point* $x = \begin{bmatrix} -3 \\ 5 \end{bmatrix}$.

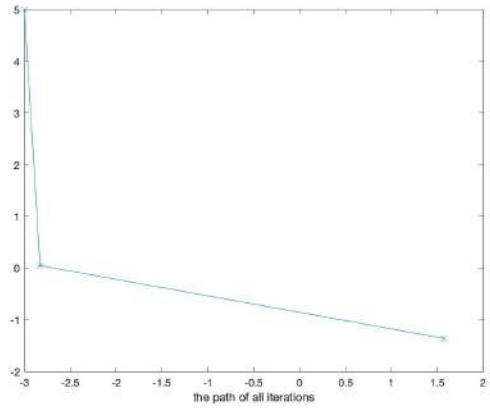


Figure 8: Searching Path

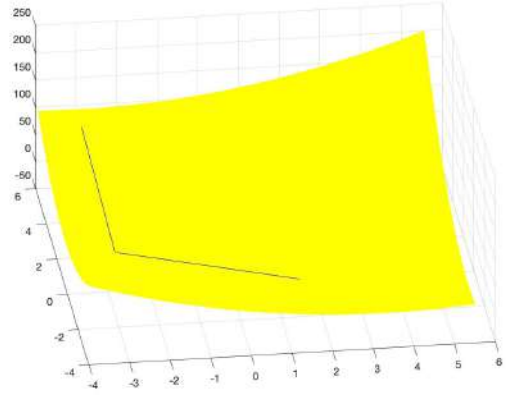


Figure 9: Conjugate Gradient Method

Figure 8 is the projection of Figure 9 onto the xy plane. The graph shows that conjugate gradient method only takes 2 steps because the matrix A has dimension two. Figure 8 shows a searching path with two directions. If we imagine two unit vectors in these two directions, the Figure 8 shows the geometric explanation of A-conjugacy. Notice these two directions would be orthogonal to each other if $A = \mathbb{I}$.

Example 3.4.8. *In the second example, we use a higher dimensional matrix A. The following graph shows an example of using conjugate gradient method to solve the minimum value of the function $f(x) = \frac{1}{2}x^T Ax + bx + c$.*

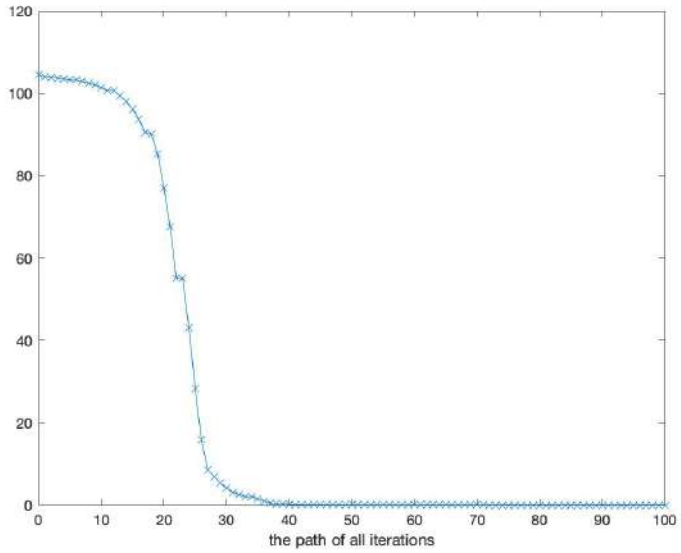


Figure 10: The Conjugate Gradient Method in a Higher Dimension

Suppose we have an arbitrary 100×100 symmetric positive definite matrix A. We can construct a random vector b with 100 entries and an arbitrary starting point x_0 as a vector of 100 entries.

Figure 10 is the graph of the change of error e_i with respect to each iteration step x_i. It shows that if we have a matrix with clustered eigenvalues, then conjugate gradient method takes much fewer steps than the maximum number of iterations, i.e, the number of dimension of the matrix.

Here is a brief explanation of how the above matrix is designed. We first create a random matrix A with size 100×100. Create a new symmetric matrix B by adding A with A^T. Do eigendecomposition on B to get its eigenvalue matrix D and eigenvector matrix V. Apply absolute value to the eigenvalue matrix and add one, which guarantees that all eigenvalues are positive. We denote the new eigenvalue matrix as $D^{'}$. Finally, we can get a random symmetric positive definite matrix by multiplying $A^T D^{'} A$.

Algorithm 6: Create a Random Symmetric Positive Definite Matrix

while $count \leq n - 1$ *and* $count \leq max\_iter$ **do**

$A = rand(100, 100)$;
$B = A + A^T$;
$[V, D] = eig(B)$;
$D^{'} = abs(D) + 1$;
$B^{'} = A^T D^{'} A$;

In summary, we conclude several advantages of the conjugate gradient method:

- The update formula of the conjugate gradient method is simple.
- For quadratic functions, the conjugate gradient method always converges in a finite number of iterations.
- When the matrix has a very high dimension n but its eigenvalues are clustered, then the method converges much faster than n steps.
- The algorithm need no storage of the matrix A, hence, memory efficient.

4 Newton's Method

4.1 The descent direction

Suppose the objective function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is $\mathcal{C}^2$ and its Hessian matrix is positive definite. Notice that the second order differentiability guarantees the symmetry of its Hessian matrix and positive definiteness guarantees the invertibility of the matrix. The second order Taylor approximation of this function at x is

$$f(x+\epsilon) = f(x) + \nabla f(x)^T \epsilon + \frac{1}{2}\epsilon^T \nabla^2 f(x)\epsilon + \mathcal{O}(\|\epsilon\|^3), \epsilon > 0.$$

We assume the approximation is local and ϵ is not too big. Therefore, we can ignore the last term that contains a third power of ϵ. Since $\nabla^2 f(x)$ is positive definite, we can view the approximate function as a convex and quadratic function of the variable ϵ. To achieve its minimum, we choose $\epsilon = -\nabla^2 f(x)^{-1}\nabla f(x)$. The intuitive idea is that we try to minimize the approximate function that satisfies the first and second order derivatives of the objective function and the minimizer is good enough to serve as the descent direction for the objective function. For future notation, we denote this quadratic approximation function as $\hat{f}(a+x) := f(a) + \nabla f(a)^T x + \frac{1}{2}x^T \nabla^2 f(a)x$, for some fixed point a.

Here is a picture illustrating the second order Taylor approximation of the objective function $f(x, y) = x^2 + y^2 + e^{x^2+y^2}$ at the origin. In this case, the minimum of the quadratic approximation function is the same as the minimum of the objective function.

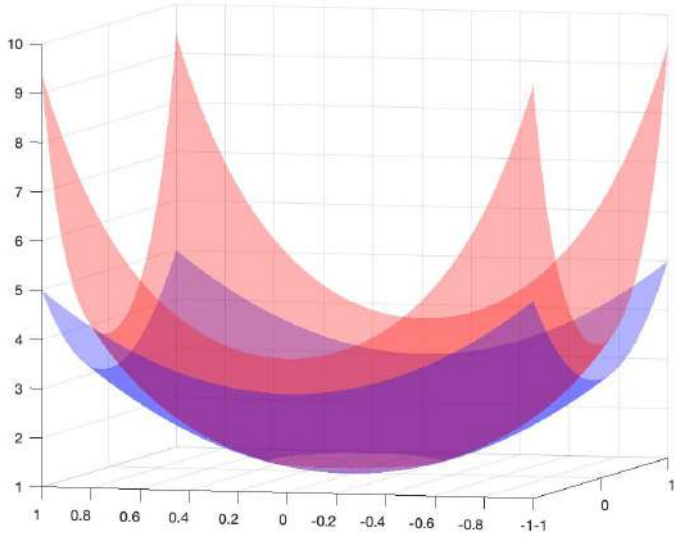


Figure 11: Second Order Taylor Approximation

Another intuition for Newton's method is the linear approximation of the gradient. The function reaches its minimum specifically when $\nabla f(x) = 0$. By linear approximation, $\nabla f(x+\epsilon) \approx \nabla f(x) + \nabla^2 f(x)\epsilon = 0$. This suggests the same idea: $\epsilon = -\nabla^2 f(x)^{-1}\nabla f(x)$.

The above explanations imply the idea of Newton's method. The descent direction for Newton's method is $\Delta x = -\nabla^2 f(x)^{-1}\nabla f(x)$. The iteration step is $x_{i+1} = x_i + t\Delta x_i$, where $\Delta x_i = -\nabla^2 f(x)^{-1}\nabla f(x)$. As before, t is a variable that determines the step size.

Theorem 4.1.1. *Suppose we have a twice continuously differentiable function f whose Hessian matrix is positive definite. Assume we have x_i. By using Newton's descent direction, we can find $x_{i+1} = x_i - t\nabla^2 f(x)^{-1}\nabla f(x)$ with an appropriate t such that $f(x_{i+1}) < f(x_i)$.*

Proof. Denote $y = -\nabla^2 f(x)^{-1}\nabla f(x)$. Define a function $g(t) = f(x_i + ty)$. The derivative of this function is $g'(t) = \nabla f(x_i + ty) \cdot y$. Plug in $t = 0$:

$$g'(0) = \nabla f(x_i) \cdot y = -\nabla f(x_i)^T \nabla^2 f(x)^{-1}\nabla f(x).$$

Since $\nabla^2 f(x)$ is positive definite, we know that its inverse is also positive definite, which means that $g'(0) < 0$. Then in a small neighborhood $(-\epsilon, \epsilon)$, we know the function g is decreasing. There exists $t \in (-\epsilon, \epsilon)$ such that $g(t) < g(0)$, which is the same as saying $f(x_i - t\nabla^2 f(x)^{-1}\nabla f(x)) < f(x_i)$. This guarantees that with a proper choice of t, the Newton's iteration step gives a sequence of points with decreasing function value. □

4.2 The Newton Decrement

The Newton decrement of f at x is defined to be

$$\lambda(x) = (\nabla f(x)^T \nabla^2 f(x)^{-1}\nabla f(x))^{\frac{1}{2}} = (-\nabla f(x)^T \Delta x)^{\frac{1}{2}}.$$

The $\lambda(x_i)$ is obtained by computing the difference between $f(x_i)$ and its quadratic approximation:

$$\begin{aligned} f(x_i) - \inf_{\epsilon} \hat{f}(x_i + \epsilon) &= f(x_i) - \hat{f}(x_i + \Delta x_i) \\ &= f(x_i) - (f(x_i) + \nabla f(x_i)^T \Delta x_i + \frac{1}{2}\Delta x_i^T \nabla^2 f(x_i)\Delta x_i) \\ &= -\nabla f(x_i)^T \Delta x_i - \frac{1}{2}\Delta x_i^T \nabla^2 f(x_i)\Delta x_i \\ &= \lambda(x_i)^2 - \frac{1}{2}\lambda(x_i)^2 \\ &= \frac{1}{2}\lambda(x_i)^2. \end{aligned} \tag{17}$$

Therefore, $\lambda(x)$ measures the difference between the objective function and the minimum of the quadratic approximation at each x. Even though the minimum of the quadratic approximation is not exactly the minimum of f, it is a good approximation. When x is close enough to the optimal point x^*, the difference between $\inf f$ and $\inf \hat{f}$ should be fairly small, which means that $\lambda(x^*)$ is very small. Hence, $\lambda(x)$ is a good stopping criterion for Newton's method. As $\lambda(x)$ gets small enough, we can say that x is very close to x^*.

On top of that, the Newton decrement is also used in the backtracking line search. Recall that the while condition for the backtracking line search is

$$f(x + t\Delta x) \leq f(x) + \alpha t \nabla f(x)^T \Delta x, \tag{18}$$

where $\nabla f(x)^T \Delta x = -\lambda(x)^2$.

However, why is the exact line search not a good approach? Remember that for the exact line search we have to compute t such that $\frac{d}{dt} f(x_i + t\Delta x_i) = 0$. Therefore, we need

$$\begin{aligned}\frac{d}{dt} f(x_i + t\Delta x_i) &= \nabla f(x_i + t\Delta x_i)^T \Delta x_i \\ &= -\nabla f(x_i + t\Delta x_i)^T \nabla^2 f(x)^{-1} \nabla f(x) = 0.\end{aligned}$$

Since both $\nabla^2 f(x)^{-1}$ and $\nabla f(x)$ cannot be zero, we have $\nabla f(x_i + t\Delta x_i) = 0$. Solving this equation exactly is difficult unless there is a nice formula for ∇f.

Next, we talk about the affine invariant property of the Newton's descent direction.

Suppose $A \in \mathbb{R}^{n \times n}$ is a nonsingular matrix. Suppose we have a function $f(x) : \mathbb{R}^n \to \mathbb{R}$. Define a new function $f_1(y) = f(Ay)$, such that $x = Ay$. Then

$$\nabla f_1(y) = A^T \nabla f(x), \nabla^2 f_1(y) = A^T \nabla^2 f(x) A. \tag{19}$$

Then Newton's step for this new function f_1 is

$$\begin{aligned}\Delta y &= -\nabla^2 f_1(y)^{-1} \nabla f_1(y) \\ &= -(A^T \nabla^2 f(x) A)^{-1} A^T \nabla f(x) \\ &= -A^{-1} \nabla^2 f(x)^{-1} \nabla f(x) \\ &= A^{-1} \Delta x.\end{aligned} \tag{20}$$

Hence, $A\Delta y = \Delta x$. The descent directions of f_1 and f follow the same affine transformation. For Newton steps,

$$x + t\Delta x = A(y + t\Delta y).$$

4.3 Convergence Analysis of Newton's Method

To analyze the convergence of Newton's method, we assume the following conditions about the objective function f:

- f is $\mathcal{C}^2$.
- $\nabla^2 f(x) \succeq m\mathbb{I}, m > 0$, meaning that the smallest eigenvalue of $\nabla^2 f(x)$ is at least m.
- $\nabla^2 f(x) \preceq M\mathbb{I}, M > 0$, meaning that the largest eigenvalue of $\nabla^2 f(x)$ is at most M.
- $\nabla^2 f(x)$ is a Lipschitz function with constant L, i.e., $\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L \|x - y\|_2$ for any $x, y \in \mathrm{dom}(f)$.

The Lipschitz condition provides a bound on the third derivative of f. When the function is quadratic, we have $\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 = 0$ and L can be chosen to be 0. So when L is small or close to zero, the function can be well approximated by a quadratic function. As we will show in the next section, if f is a quadratic function, Newton's method takes only one step to reach the optimal point x^*. Therefore, Newton's method works well for functions that have small Lipschitz constants.

Choose a number $\eta \in (0, 3(1-2\alpha)\frac{m^2}{L})$ where α is the coefficient used in the backtracking line search, and m, L are defined above. Since α is an arbitrary number less than $\frac{1}{2}$, we assume α is not too small, $\alpha \in (\frac{1}{3}, \frac{1}{2})$.We know that $3(1-2\alpha)\frac{m^2}{L}$ is greater than 0 because α is chosen to be less than $\frac{1}{2}$ and m, L are both positive.

There are two phases for the convergence of Newton's method. The first phase is called the **Damped Newton Phase** where $\|\nabla f(x)\|_2 \geq \eta$. The second phase is called the **Quadratically Convergent Phase** where $\|\nabla f(x)\|_2 < \eta$.

4.3.1 Damped Newton Phase

In this phase, Newton's method uses the backtracking line search to determine the step size t. We are going to show that there exists a number γ such that each iteration step results in a decrease of at least γ in the objective function. We assume that $\|\nabla^2 f(x)\| \geq \eta$. By the Taylor's expansion theorem,

$$f(x + t\Delta x) = f(x) + t\nabla f(x)^T \Delta x + \frac{1}{2}t^2 \Delta x^T \nabla^2 f(y) \Delta x,$$

for some y between x and $x + t\Delta x$. Since $\nabla^2 f(x) \preceq M\mathbb{I}$, we get

$$f(x + t\Delta x) \leq f(x) + t\nabla f(x)^T \Delta x + \frac{1}{2}t^2 M \|\Delta x\|_2^2. \tag{21}$$

By definition, $\nabla f(x)^T \Delta x = -\lambda(x)^2$ and

$$\lambda(x)^2 = \Delta x^T \nabla^2 f(x) \Delta x \geq m\|\Delta x\|_2^2. \tag{22}$$

Therefore, (21) can be simplified into:

$$f(x + t\Delta x) \leq f(x) - t\lambda(x)^2 + \frac{t^2 M}{2}\frac{\lambda(x)^2}{m}. \tag{23}$$

Since (23) holds for all t, we choose $\tilde{t} = \frac{m}{M}$ which is the minimizer of the right hand side.

$$\begin{aligned} f(x + \frac{m}{M}\Delta x) &\leq f(x) - \frac{m}{M}\lambda(x)^2 + \frac{m}{2M}\lambda(x)^2 \\ &= f(x) - \frac{m}{2M}\lambda(x)^2 \\ &= f(x) - \alpha\frac{m}{M}\lambda(x). \end{aligned} \tag{24}$$

Thus, $\tilde{t} = \frac{m}{M}$ satisfies the exit condition of the while loop in the backtracking line search. We know that it returns a step size $t \geq \gamma\frac{m}{M}$. Recall that γ is a variable used to shrink t at each iteration. Notice that by definition,

$$\lambda(x)^2 = \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x) \geq \frac{1}{M}\|\nabla f(x)\|_2^2.$$

Hence, plugging in $t \geq \gamma\frac{m}{M}$ and reorganizing (24) yield:

$$\begin{aligned} f(x_i) - f(x_{i+1}) &\geq \alpha\gamma\frac{m}{M}\lambda(x)^2 \\ &\geq \alpha\gamma\frac{m}{M^2}\|\nabla f(x)\|_2^2 \\ &\geq \alpha\gamma\frac{m}{M^2}\eta^2. \end{aligned} \tag{25}$$

Therefore, at each iteration step, the amount of decrease is at least $\alpha\gamma\frac{m}{M^2}\eta^2$. The total number of iterations will be bounded by

$$\frac{f(x_0) - f(x^*)}{\alpha\gamma\frac{m}{M^2}\eta^2},$$

where x_0 is the starting point and x^* is the optimal point.

4.3.2 Quadratically Convergent Phase

In the Quadratically Convergent Phase, Newton's method needs no backtracking line search because we are going to prove that $t = 1$ satisfies the quit condition for the while loop. More importantly, in this phase, f decreases quadratically at each step.

Remember we have two assumptions: $\eta \leq 3(1-2\alpha)\frac{m^2}{L}$ and $\|\nabla f(x)\|_2 < \eta$. We have not used the Lipschitz condition in the Damped Newton Phase and we will use it here.

By the Lipschitz condition,

$$\left\|\nabla^2 f(x + t\Delta x) - \nabla^2 f(x)\right\|_2 \leq L \|t\Delta x\|_2 . \tag{26}$$

Define a new function $\tilde{f}(t) = f(x + t\Delta x)$. Then $\tilde{f}''(t) = \Delta x^T \nabla^2 f(x + t\Delta x)\Delta x$. This suggests that we can multiply Δx on both sides of (26) and get an inequality involving the second derivative of $\tilde{f}$:

$$\begin{aligned} |\Delta x^T[\nabla^2 f(x + t\Delta x) - \nabla^2 f(x)]\Delta x| &\leq tL \|\Delta x\|_2^3 , \\ |\Delta x^T \nabla^2 f(x + t\Delta x)\Delta x - \Delta x^T \nabla^2 f(x)\Delta x| &\leq tL \|\Delta x\|_2^3 , \\ |\tilde{f}''(t) - \tilde{f}''(0)| &\leq tL \|\Delta x\|_2^3 , \\ |\tilde{f}''(t) - \tilde{f}''(0)| &\leq tL\frac{\lambda(x)^3}{m^{\frac{3}{2}}}, \end{aligned} \tag{27}$$

where the last inequality follows by (22).

We can get an upper bound on $\tilde{f}$ by computing the integral twice. We know that

$$\tilde{f}''(0) = \Delta x^T \nabla^2 f(x)\Delta x = \lambda(x)^2,$$

and

$$\tilde{f}'(0) = \nabla f(x)^T \Delta x = -\lambda(x)^2, \tilde{f}(0) = f(x),$$

which are conditions for determining constants in the indefinite integral. Then with the help of (22) and above conditions, we integrate (27) one time and get:

$$\begin{aligned} \tilde{f}''(t) &\leq \tilde{f}''(0) + tL \|\Delta x\|_2^3 , \\ \tilde{f}'(t) &\leq t\tilde{f}''(0) + \frac{t^2 L}{2m^{\frac{3}{2}}}\lambda(x)^3 - \lambda(x)^2, \end{aligned} \tag{28}$$

where $\tilde{f}'(0) = -\lambda(x)^2$ and $m^{\frac{3}{2}}$ comes from (22). Integrating again gives

$$\begin{aligned} \tilde{f}(t) &\leq \frac{t^2}{2}\lambda(x)^2 + \frac{t^3 L}{6m^{\frac{3}{2}}}\lambda(x)^3 - \lambda(x)^2 t + f(x) \\ &= -\frac{t^2}{2}\lambda(x)^2 + \frac{t^3 L}{6m^{\frac{3}{2}}}\lambda(x)^3 + f(x) \\ &= -(\frac{t^2}{2} - \frac{t^3 L\lambda(x)}{6m^{\frac{3}{2}}})\lambda(x)^2 + f(x). \end{aligned} \tag{29}$$

Notice the above inequality looks similar to the while condition in the backtracking line search except that the coefficient is more complicated than α. Now, we need to use our two assumptions:

$$\lambda(x) \leq \frac{1}{\sqrt{m}} \|\nabla f(x)\|_2 \leq 3(1-2\alpha)\frac{m^{\frac{3}{2}}}{L},$$
$$\alpha \leq \frac{1}{2} - \frac{L\lambda(x)}{6m^{\frac{3}{2}}}.$$

Therefore, when $t = 1$,

$$\begin{aligned} f(x+\Delta x) &\leq -(\frac{1}{2} - \frac{L\lambda(x)}{6m^{\frac{3}{2}}})\lambda(x)^2 + f(x) \\ &\leq -\alpha\lambda(x)^2 + f(x). \end{aligned}$$

This means that the step size $t = 1$ satisfies the backtracking line search exit condition. Next, we prove that the rate of convergence is quadratic. Assume $t = 1$.

Theorem 4.3.1. *If g is $\mathcal{C}^1$ and ∇g is a Lipschitz function with constant L, i.e., $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x-y\|_2$, then*

$$\left\|g(x+y) - g(x) - \nabla g(x)^T y\right\|_2 \leq \frac{L}{2}\|y\|_2^2.$$

Proof. Change the left hand side into an integral,

$$\begin{aligned} g(x+y) - g(x) - \nabla g(x)^T y &= \int_0^1 (\nabla g(x+ty) - \nabla g(x))y \; dt \\ &\leq \int_0^1 L\|y\|_2^2 t \; dt = \frac{L}{2}\|y\|_2^2. \end{aligned}$$

□

By using the above theorem with $g = \nabla f$, $y = \Delta x$, we get

$$\begin{aligned} \|\nabla f(x+\Delta x)\|_2 &= \left\|\nabla f(x+\Delta x) - \nabla f(x) - \nabla^2 f(x)^T \Delta x\right\|_2 \\ &\leq \frac{L}{2}\|\Delta x\|_2^2 \\ &\leq \frac{L}{2m^2}\|\nabla f(x)\|_2^2. \end{aligned} \tag{30}$$

Therefore, we get the desired result, $\|\nabla f(x_{i+1})\|_2 \leq \frac{L}{2m^2}\|\nabla f(x_i)\|_2^2$.

When $\|\nabla f(x_i)\|_2 < \eta$,

$$\|\nabla f(x_{i+1})\|_2 \leq \frac{L}{2m^2}\eta^2 \leq 3(1-2\alpha)\frac{m^2}{L}\frac{L}{2m^2}\eta \leq \frac{1}{2}\eta,$$

where the last inequality comes from the fact that $\alpha \in (\frac{1}{3}, \frac{1}{2})$ implies $3(1-2\alpha) < 1$.

Therefore, once the condition $\|\nabla f(x_i)\|_2 < \eta$ holds for some x_i, it is going to be true for all steps afterwards. This means that after x_i the method moves into Quadratically convergent phase and $t = 1$ is always satisfied.

Recall from the Gradient Descent Method, we get an inequality (5):

$$\|\nabla f(x)\|^2 \geq 2m(f(x) - f(x^*)).$$

Here, we can reuse this inequality because the objective function f satisfies all the conditions that we used to derive this inequality. Readers can refer to the previous section to see the proof.

Suppose that starting from x_i the method gets into the Quadratically convergent phase. Then by applying (30) recursively, we get

$$\begin{aligned} f(x_k) - f(x^*) &\leq \frac{1}{2m} \|\nabla f(x_k)\|_2^2 \\ \|\nabla f(x_k)\|_2 &\leq \frac{L}{2m^2} \|\nabla f(x_{k-1})\|_2^2 \leq (\frac{L}{2m^2} \|\nabla f(x_i)\|)_2^{2^{k-i}} \end{aligned} \tag{31}$$

We can simplify this even further:

$$\frac{L}{2m^2} \|\nabla f(x_i)\| \leq \frac{L}{2m^2} 3(1-2\alpha)\frac{m^2}{L} \leq \frac{1}{2},$$

$$\|\nabla f(x_k)\|_2^2 \leq \left(\frac{1}{2}\right)^{2^{k-i}}.$$

Therefore, we finally get

$$f(x_k) - f(x^*) \leq \frac{1}{2m} \left(\frac{1}{2}\right)^{2^{k-i}}.$$

Suppose we want to obtain x_k such that $f(x_k) - f(x^*) \leq \epsilon$. We just need to make the upper bound less than ϵ :

$$\begin{aligned} &\frac{1}{2m} \left(\frac{1}{2}\right)^{2^{k-i}} \leq \epsilon \\ &2^{k-i} \geq \log_2 (2m\epsilon)^{-1} \\ &k \geq i + \log_2 \log_2 (2m\epsilon)^{-1}. \end{aligned}$$

In practice, we only need 5 or 6 steps because $\left(\frac{1}{2}\right)^{2^6}$ is already $5.42 * 10^{-20}$. Therefore, combining two phases, the total number of iterations are bounded by

$$\frac{f(x_0) - f(x^*)}{\alpha\gamma\frac{m}{M^2}\eta^2} + 6.$$

4.4 Algorithm

The following algorithm uses the backtracking line search to determine each iteration step size.

Algorithm 7: Newton's Method

```
Newton'sMethod (f, x_0, ϵ, α, γ);
Compute the initial λ^2 and Δx for x_0;
while λ^2/2 ≥ ϵ do
    x_{i+1} = x_i + Δx;
    while f(x_i) − f(x_{i+1}) < −αt∇f(x_i)^T Δx, do
        t = γt;
        x_{i+1} = x_i + tΔx;
    x_i = x_{i+1};
    Δx = −∇^2 f(x_i)^{-1} ∇f(x_i);
    λ^2 = ∇f(x_i)^T ∇^2 f(x_i)^{-1} ∇f(x_i);
return x_i;
```

4.5 Examples

In what follows, we illustrate the method with two examples.

4.5.1 Quadratic function

Consider a quadratic function $f(x) = \frac{1}{2}x^TAx + bx + c$ where A is symmetric and invertible. We know that $\Delta x = -A^{-1}(Ax + b) = -x - A^{-1}b$ and $\nabla f(x) = Ax + b$. Thus,

$$\begin{aligned} x_1 &= x_0 + \Delta x_0 \\ &= x_0 - x_0 - A^{-1}b \\ &= -A^{-1}b, \end{aligned}$$

which gives the minimum value for f. Hence, for any quadratic function, Newton's method converges in one step.

4.5.2 Exponential Function in $\mathbb{R}^2$

Consider a convex function

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}.$$

We can compute its gradient and Hessian matrix directly:

$$\nabla f(x_1, x_2) = e^{x_1+3x_2-0.1}\begin{bmatrix}1\\3\end{bmatrix} + e^{x_1-3x_2-0.1}\begin{bmatrix}1\\-3\end{bmatrix} + e^{-x_1-0.1}\begin{bmatrix}-1\\0\end{bmatrix}$$

and

$$\nabla^2 f(x_1, x_2) = e^{x_1+3x_2-0.1}\begin{bmatrix}1 & 3\\3 & 9\end{bmatrix} + e^{x_1-3x_2-0.1}\begin{bmatrix}1 & -3\\-3 & 9\end{bmatrix} + e^{-x_1-0.1}\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}.$$

Therefore, we know Δx and λ. By using Newton's method with a starting point $x_0 = \begin{bmatrix}-5\\-5\end{bmatrix}$, we get the following two graphs:

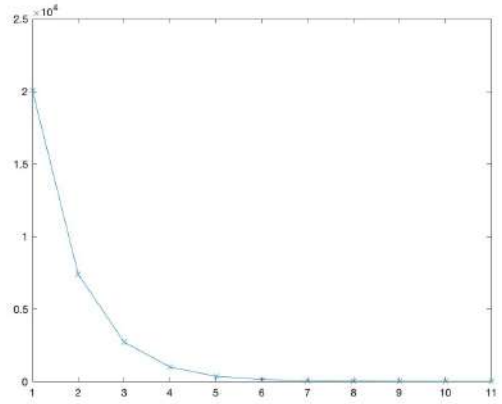


Figure 12: Error $f(x_i) - f(x^*)$

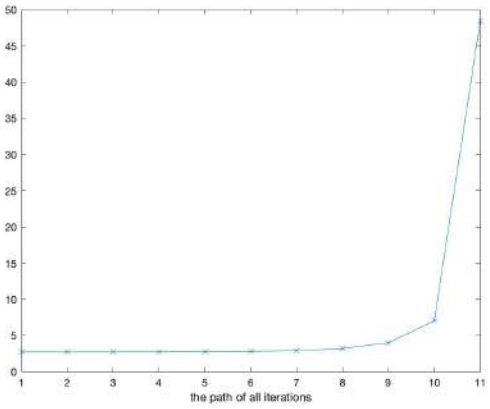


Figure 13: Damped Phase and Quadratically Convergent Phase: Plot of ζ_i

Figure 12 shows how the error $f(x_i) - f(x^*)$ decreases at each step. Here we see that Newton's method gives a convergent sequence of points which leads to x^*. In Figure 13, the vertical axis represents $\zeta_i = \frac{f(x_i)-f(x^*)}{f(x_{i+1})-f(x^*)}$, the old error divide by the new error. Once the algorithm gets into the quadratically convergent phase(starting from index 9), ζ_i decreases quadratically, from approximately 3 to 7 to 49. This means that $(f(x_i)-f(x^*))^2 \approx f(x_{i+1})-f(x^*)$ and $\zeta(i)^2 \approx \zeta(i+1)$. From the graph, we see that at step $i = 10$ and $i = 11$, $\zeta(10)^2 \approx \zeta(11)$. This matches our convergence analysis.

In summary, once the iteration step moves into the Quadratically Convergent Phase, Newton's method converges rapidly within 5 or 6 steps with high precision. However, storing the Hessian matrix for each iteration is memory inefficient and the cost of matrix and vector multiplication is expensive if the function has large dimensions.

In practice, we usually do not have precise estimates for constants, m, M, L, which are used to set up a range for η and separate two convergence phases. This is a theoretically correct analysis. Next, we are going to introduce a new type of function, called self-concordant function, whose convergence analysis does not depend on these constants.

4.6 Newton's Method for Self-concordant Functions

Definition 4.6.1. *A function $f(x) : \mathbb{R} \to \mathbb{R}$ that has a third derivative and satisfies for all $x \in \mathrm{dom}(f)$,*

$$|f'''(x)| \leq 2f''(x)^{\frac{3}{2}}$$

is called a self-concordant function.

From the definition, a self-concordant function f is convex because $f''(x) \geq 0$ and is also $\mathcal{C}^2$.

4.6.1 Affine Invariant Property

Theorem 4.6.2. *Suppose we have a self-concordant function $f(x) : \mathbb{R} \to \mathbb{R}$. Define a new function $\tilde{f}(x) = f(ax + b)$ for some constant $a \neq 0$ and b. Then $\tilde{f}(x)$ is also self-concordant.*

Proof. Since $\tilde{f}$ is an affine transformation of f, $\tilde{f}$ is also convex and has its third derivative. $\tilde{f}''' = a^3 f'''(ax+b)$ and $\tilde{f}'' = a^2 f''(ax+b)$. Since f is self-concordant and $ax+b \in \text{dom}(f)$, we get

$$|a^3 f'''(ax+b)| \leq 2(a^2 f''(x))^{\frac{3}{2}},$$
$$|\tilde{f}'''(x)| \leq 2\tilde{f}''(x)^{\frac{3}{2}}.$$

Therefore, $\tilde{f}$ is also self-concordant. □

Theorem 4.6.3. *Self-concordance is preserved under scalar multiplication if the constant $c \geq 1$ and addition.*

Proof. The first part is straightforward. If $c \geq 1$, then $c^{\frac{3}{2}} \geq c$. So $c|f'''(x)| \leq 2(cf''(x))^{\frac{3}{2}}$. Next, suppose we have two self-concordant functions $f_1,\ f_2 : \mathbb{R} \to \mathbb{R}$. By definition and triangle inequality, we have

$$\begin{aligned} |f_1'''(x) + f_2'''(x)| &\leq |f_1'''(x)| + |f_2'''(x)| \\ &\leq 2({f_1}''(x)^{\frac{3}{2}} + {f_2}''(x)^{\frac{3}{2}}) \\ &\leq 2({f_1}''(x) + {f_2}''(x))^{\frac{3}{2}}. \end{aligned}$$

The last inequality comes from the fact that $u^{\frac{3}{2}} + v^{\frac{3}{2}} \leq (u+v)^{\frac{3}{2}}$. □

Definition 4.6.4. *A function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is self-concordant if it is self-concordant along every line in the domain, i.e., $\hat{f}(t) = f(x+tv)$ is a self-concordant function of t for all directions v and for all $x \in \text{dom}(f)$. In other words, $f(x) : \mathbb{R}^n \to \mathbb{R}$ satisfies*

$$\frac{d}{dt}\nabla^2 f(x+tv)\bigg|_{t=0} \preceq 2\sqrt{v^T\nabla^2 f(x)v}\nabla^2 f(x).$$

In higher dimensional cases, the affine invariant property becomes the following:

Suppose we have a self-concordant function $f(x) : \mathbb{R}^n \to \mathbb{R}$. Let $A \in \mathbb{R}^{n\times m}, b \in \mathbb{R}^n$. Then $f(Ax+b)$ is also self-concordant.

Example 4.6.5. *Consider the negative logarithm.*

Let $f(x) = -\log(x), \text{dom}(f) = \{x > 0\}$. Second derivative $f''(x) = \frac{1}{x^2}$. Third derivative $f'''(x) = -\frac{2}{x^3}$. Obviously, $|f'''(x)| = 2f''(x)^{\frac{3}{2}}$. From here, we know that functions $f(x) = \sum_{k=1}^n -\log(b_k - a_k x)$ are all self-concordant because self-concordance is preserved under affine transformation and addition. We can then generalize to functions of higher dimensions $f(x) = \sum_{k=1}^n -\log(b_k - {a_k}^T x)$.

Example 4.6.6. *Consider another function $f(x) = x\log(x) - \log(x)$.*

The domain of the function is $\text{dom}(f) = \{x > 0\}$. Second derivative $f''(x) = \frac{x+1}{x^2} > 0$. Third derivative $f'''(x) = -\frac{x+2}{x^3}$. Then

$$\left|\frac{f'''(x)}{2f''(x)^{\frac{3}{2}}}\right| = \frac{x+2}{x^3}\cdot\frac{x^3}{2(x+1)^{\frac{3}{2}}} = \frac{x+2}{2(x+1)^{\frac{3}{2}}} = \frac{1}{2}\left(\frac{1}{(x+1)^{\frac{1}{2}}} + \frac{1}{(x+1)^{\frac{3}{2}}}\right),$$

which reaches it maximum 1 at $x = 0$. Therefore,

$$\left|\frac{f'''(x)}{2f''(x)^{\frac{3}{2}}}\right| = 1$$

and f is self-concordant.

We know that if the Hessian matrix of a function f is positive definite, then the function is strictly convex. However, the converse is not true. For example, $f(x) = x^4$ is strictly convex. The second derivative at 0 is $f''(0) = 0$, which is not positive definite. It can be proved that the Hessian matrix of a strictly convex self-concordant function is positive definite everywhere. Readers who are interested can refer to [1].

Recall from the previous section that the Newton decrement is

$$\lambda(x) = (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))^{\frac{1}{2}}.$$

We are going to prove another way to express the Newton decrement.

Theorem 4.6.7. *Let v be a descent direction. From (1), we know that v satisfies $v^T \nabla f \leq 0$. The Newton decrement can be written as*

$$\lambda(x) = \sup_{v \neq 0} \frac{-v^T \nabla f(x)}{(v^T \nabla^2 f(x) v)^{\frac{1}{2}}}.$$

Proof. We know that $\nabla^2 f(x)$ is positive definite. Define $w = (\nabla^2 f(x))^{\frac{1}{2}} v$, then $v = (\nabla^2 f(x))^{-\frac{1}{2}} w$ and $(\|w\|_2)^2 = v^T \nabla^2 f(x) v$. Then

$$\begin{aligned}
\sup_{v^T \nabla^2 f(x) v = 1} -v^T \nabla f(x) &= \sup_{\|w\|_2 = 1} -w^T (\nabla^2 f(x))^{-\frac{1}{2}} \nabla f(x) \\
&= \left\| (\nabla^2 f(x))^{-\frac{1}{2}} \nabla f(x) \right\|_2 \\
&= (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))^{\frac{1}{2}} \\
&= \lambda(x).
\end{aligned}$$

The second line comes from

$$w = \frac{-(\nabla^2 f(x))^{-\frac{1}{2}} \nabla f(x)}{\left\| (\nabla^2 f(x))^{-\frac{1}{2}} \nabla f(x) \right\|_2}.$$

Since w can be any unit vector, to get the supremum of the right hand side, we let w point in the direction of $\nabla^2 f(x))^{-\frac{1}{2}} \nabla f(x)$. Therefore,

$$\lambda(x) = \sup_{v^T \nabla^2 f(x) v = 1} -v^T \nabla f(x) = \sup_{v \neq 0} \frac{-v^T \nabla f(x)}{(v^T \nabla^2 f(x) v)^{\frac{1}{2}}}.$$

The last equality is obvious because $v^T \nabla^2 f(x) v = 1$. □

From the previous theorem, we immediately get an inequality

$$\lambda(x) \geq \frac{-v^T \nabla f(x)}{(v^T \nabla^2 f(x) v)^{\frac{1}{2}}}, \tag{32}$$

since $\lambda(x)$ is the supremum. The equality is obtained when $v = -\Delta x = \nabla^2 f(x)^{-1} \nabla f(x)$.

Theorem 4.6.8. *Suppose f is a strictly convex self-concordant function. Then the self-concordance inequality can be rewritten as*

$$\left| \frac{d}{dt} \left(f''(t) \right)^{-\frac{1}{2}} \right| \leq 1,$$

for all $t \in \mathrm{dom}(f)$.

Proof. The proof is very easy. The main point of this proof is to develop the upper and lower bounds on $f''(t)$.

$$\left| \frac{d}{dt} \left(f''(t) \right)^{-\frac{1}{2}} \right| = \left| -\frac{1}{2} \left(f''(t) \right)^{-\frac{3}{2}} f'''(t) \right| \leq 1,$$

which is just another way of saying $|f'''(x)| \leq 2f''(x)^{\frac{3}{2}}$. □

Assume $t \geq 0$ and the interval $[0, t]$ is contained in $\mathrm{dom}(f)$. Then we can integrate the derivative between 0 and t:

$$\int_0^t \frac{d}{dx} \left(f''(x) \right)^{-\frac{1}{2}} dx = f''(t)^{-\frac{1}{2}} - f''(0)^{-\frac{1}{2}} \in [-t, t].$$

Hence, we get

$$-t \leq f''(t)^{-\frac{1}{2}} - f''(0)^{-\frac{1}{2}} \leq t.$$

We can isolate $f''(t)$:

$$\begin{gathered} f''(0)^{-\frac{1}{2}} - t \leq f''(t)^{-\frac{1}{2}} \leq t + f''(0)^{-\frac{1}{2}}, \\ \frac{f''(0)}{(1 + t\sqrt{f''(0)})^2} \leq f''(t) \leq \frac{f''(0)}{(1 - t\sqrt{f''(0)})^2}. \end{gathered} \tag{33}$$

The right hand side inequality is valid when $f''(0)^{-\frac{1}{2}} - t \geq 0$, that is $0 \leq t \leq f''(0)^{-\frac{1}{2}}$.

4.6.2 Bound on $f(x) - f(x^*)$

Assume the function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is a strictly convex self-concordant function and v is a descent direction but does not need to be the Newton direction. Let $\tilde{f}(t) : \mathbb{R} \to \mathbb{R}$ be $\tilde{f}(t) = f(x + tv)$, which is also strictly convex and self-concordant.

By using the lower bound of (33), we can integrate to get that

$$\begin{aligned} \tilde{f}'(t) - \tilde{f}'(0) &= \int_0^t \tilde{f}''(x) dx \geq \int_0^t \frac{\tilde{f}''(0)}{(1 + x\sqrt{\tilde{f}''(0)})^2} dx \\ &= -\frac{\sqrt{\tilde{f}''(0)}}{1 + x\sqrt{\tilde{f}''(0)}} \Bigg|_0^t = \sqrt{\tilde{f}''(0)} - \frac{\sqrt{\tilde{f}''(0)}}{1 + t\sqrt{\tilde{f}''(0)}}. \end{aligned}$$

$$\tilde{f}'(t) \geq \tilde{f}'(0) + \sqrt{\tilde{f}''(0)} - \frac{\sqrt{\tilde{f}''(0)}}{1 + t\sqrt{\tilde{f}''(0)}}. \tag{34}$$

We integrate (34) again and get

$$\tilde{f}(t) \geq \tilde{f}(0) + t\tilde{f}'(0) + t\sqrt{\tilde{f}''(0)} - \log\left(1 + t\sqrt{\tilde{f}''(0)}\right). \tag{35}$$

Notice that the right hand side is a convex function of t. Then we can find t^* that reaches its minimum, i.e., to solve for t when the derivative of the right-hand side is 0. We get

$$t^* = \frac{-\tilde{f}'(0)}{\tilde{f}''(0) + \sqrt{\tilde{f}''(0)}\tilde{f}'(0)}.$$

Since (34) is always true for $t \geq 0$, we can plug in t^* and get:

$$\begin{aligned}
\inf_{t\geq 0} \tilde{f}(t) &\geq \tilde{f}(0) + t^*\tilde{f}'(0) + t^*\sqrt{\tilde{f}''(0)} - \log\left(1 + t^*\sqrt{\tilde{f}''(0)}\right) \\
&= \tilde{f}(0) + \left(\tilde{f}'(0) + \sqrt{\tilde{f}''(0)}\right)\frac{-\tilde{f}'(0)}{\tilde{f}''(0) + \sqrt{\tilde{f}''(0)}\tilde{f}'(0)} \\
&\quad - \log\left(1 - \frac{\tilde{f}'(0)}{\tilde{f}''(0) + \sqrt{\tilde{f}''(0)}\tilde{f}'(0)}\sqrt{\tilde{f}''(0)}\right) \\
&= \tilde{f}(0) + \frac{-\tilde{f}'(0)}{\sqrt{\tilde{f}''(0)}} - \log\left(\frac{\tilde{f}''(0)}{\tilde{f}''(0) + \sqrt{\tilde{f}''(0)}\tilde{f}'(0)}\right) \\
&= \tilde{f}(0) - \frac{\tilde{f}'(0)}{\sqrt{\tilde{f}''(0)}} + \log\left(1 + \frac{\tilde{f}'(0)}{\sqrt{\tilde{f}''(0)}}\right).
\end{aligned} \tag{36}$$

We did all these calculations because now we can treat $\frac{\tilde{f}'(0)}{\sqrt{\tilde{f}''(0)}}$ as a variable. Notice that $\tilde{f}'(0) = v^T\nabla f(x)$ and $\tilde{f}''(0) = v^T\nabla^2 f(x)v$. Recall the inequality we got for the Newton decrement. Here, (32) can be rewritten as

$$\lambda(x) \geq \frac{\tilde{f}'(0)}{\sqrt{\tilde{f}''(0)}}.$$

Consider the function $g(x) = x + \log(1-x)$, (log is of base e). We know that $g(0) = 0$ and $g'(x) = \frac{-x}{1-x}$, which is negative on $(0,1)$. So the function g is decreasing on $(0,1)$. Then the following inequality is true for any descent direction v provided that $\lambda(x) < 1$,

$$-\frac{\tilde{f}'(0)}{\sqrt{\tilde{f}''(0)}} + \log\left(1 + \frac{-\tilde{f}'(0)}{\sqrt{\tilde{f}''(0)}}\right) \geq \lambda(x) + \log(1-\lambda(x)). \tag{37}$$

Combining (36),

$$f(x^*) = \inf_{t\geq 0} \tilde{f}(t) \geq \tilde{f}(0) + \lambda(x) + \log(1-\lambda(x)). \tag{38}$$

On the last line, we have $f(x^*) = \underset{t\geq 0}{inf}\, \tilde{f}(t)$ because we can choose v to be any descent direction. Notice that $\lambda(x) + \log(1-\lambda(x)) \geq -\lambda(x)^2$ on the interval $\lambda(x) \in (0, 0.68)$. Then (38) can be simplified into

$$f(x^*) \geq f(x) - \lambda(x)^2,\ \lambda(x)^2 \geq f(x) - f(x^*), \tag{39}$$

provided that $\lambda(x) < 0.68$. Here we get the desired result.

Remember the termination condition for the general Newton's method is $\frac{\lambda(x)^2}{2} \leq \epsilon$. If the objective function is self-concordant, then double the value still gives us a valid upper bound. In conclusion, the termination condition of Newton's method for self-concordant functions become $\lambda(x)^2 < \epsilon$ where $\epsilon < 0.68^2$.

4.6.3 Convergence Analysis of Newton's Method for Self-concordant Functions

Assume the objective function is strictly convex and self-concordant. Now we do not need upper, lower bound on $\nabla^2 f(x)$ or the Lipschitz condition. Instead, we only use the assumption: self-concordance and the Newton decrement will replace $\|\nabla f(x)\|_2$. Similar to the classic convergence analysis of the Newton method, there are two phases: **Damped Newton Phase** where $\lambda(x_k) > \eta$ and **Quadratically Convergent Phase** where $\lambda(x_k) \leq \eta$, $\eta \in (0, 1/4)$.

4.6.4 Damped Newton Phase

Define $\tilde{f}(t) = f(x + t\Delta x)$. So far, we have not used the upper bound (33) for the second derivative of self-concordant functions. We are going to use it now. Similar idea as before, we integrating the upper bound of (33):

$$\begin{aligned}
\tilde{f}'(t) - \tilde{f}'(0) &= \int_0^t \tilde{f}''(x)dx \leq \int_0^t \frac{\tilde{f}''(0)}{(1 - x\sqrt{\tilde{f}''(0)})^2}dx \\
&= \left.\frac{\sqrt{\tilde{f}''(0)}}{1 - x\sqrt{\tilde{f}''(0)}}\right|_0^t = \frac{\sqrt{\tilde{f}''(0)}}{1 - t\sqrt{\tilde{f}''(0)}} - \sqrt{\tilde{f}''(0)}.
\end{aligned}$$

$$\tilde{f}'(t) \leq \tilde{f}'(0) + \frac{\sqrt{\tilde{f}''(0)}}{1 - t\sqrt{\tilde{f}''(0)}} - \sqrt{\tilde{f}''(0)}. \tag{40}$$

We integrate (40) again:

$$\tilde{f}(t) \leq \tilde{f}(0) + t\tilde{f}'(0) - t\sqrt{\tilde{f}''(0)} - \log\left(1 - t\sqrt{\tilde{f}''(0)}\right).$$

Plugging in $\tilde{f}'(0) = -\lambda(x)^2$ and $\tilde{f}''(0) = \lambda(x)^2$, we get

$$\tilde{f}(t) \leq \tilde{f}(0) - t\lambda(x)^2 - t\lambda(x) - \log\left(1 - t\lambda(x)\right). \tag{41}$$

Remember the above inequality is valid when $0 \leq t \leq f''(0)^{-\frac{1}{2}} = \frac{1}{\lambda(x)}$.

Claim 4.6.9. *The backtracking line search always ends up with a step size*

$$t \geq \frac{\gamma}{1+\lambda(x)}.$$

Proof. First of all, $\frac{\gamma}{1+\lambda(x)} < \frac{1}{\lambda(x)}$ because $\gamma < 1$, which is within the possible range for t. Let $\tilde{t} = \frac{1}{1+\lambda(x)}$. Plugging in $\tilde{t}$ to (41), we get

$$\begin{aligned}\tilde{f}(\tilde{t}) &\leq \tilde{f}(0) - \tilde{t}\lambda(x)^2 - \tilde{t}\lambda(x) - \log\left(1 - \tilde{t}\lambda(x)\right) \\ &= \tilde{f}(0) - \lambda(x) + \log(1+\lambda(x)).\end{aligned}$$

Now consider the function $h(x) = -x + \log(1+x) + \frac{x^2}{2(x+1)}$. We know that for $x > -1$,

$$h(0) = 0,\ h'(x) = \frac{-x^2}{2(1+x)^2} \leq 0.$$

Since the derivative is always less than 0, we know h is a monotonically decreasing function. Since $h(0) = 0$, we know that

$$h(x) = -x + \log(1+x) + \frac{x^2}{2(x+1)} \leq 0.$$

Replacing x by $\lambda(x)$, we get that

$$\begin{aligned}h(\lambda(x)) &= -\lambda(x) + \log(1+\lambda(x)) + \frac{\lambda(x)^2}{2(\lambda(x)+1)} \leq 0, \\ -\lambda(x) + \log(1+\lambda(x)) &\leq \frac{-\lambda(x)^2}{2(\lambda(x)+1)} \leq \frac{-\alpha\lambda(x)^2}{(\lambda(x)+1)} = \frac{-\alpha\lambda(x)^2}{\tilde{t}}.\end{aligned} \tag{42}$$

Therefore, $f(x) - f(x_+) = \tilde{f}(0) - \tilde{f}(\tilde{t}) \geq \frac{-\alpha\lambda(x)^2}{\tilde{t}}$, where $f(x_+)$ denotes the next step. □

In conclusion, at the end of the backtracking line search, we have $t \geq \frac{\gamma}{1+\lambda(x)}$. At each iteration step of the Damped Newton Phase, the function value decreases at least

$$\alpha\gamma\frac{\lambda(x)^2}{1+\lambda(x)} \geq \alpha\gamma\frac{\eta^2}{1+\eta},$$

since $\frac{x^2}{1+x}$ is an increasing function for $x \geq 0$ and $\lambda(x) > \eta$.

4.6.5 Quadratically Convergent Phase

In order to show that the unit step size is always valid for the backtracking line search, we need to restrict η to a smaller range. Take $\eta = \frac{1-2\alpha}{4} \leq \frac{1}{4}$. Then $\lambda(x) \leq \frac{1-2\alpha}{4}$. Plugging in $t = 1$ to (41), we get

$$\begin{aligned}\tilde{f}(1) &\leq \tilde{f}(0) - \lambda(x)^2 - \lambda(x) - \log\left(1 - \lambda(x)\right) \\ &\leq \tilde{f}(0) - \frac{1}{2}\lambda(x)^2 + \lambda(x)^3.\end{aligned} \tag{43}$$

The above inequality comes from the fact that $-x - log(1-x) \leq \frac{1}{2}x^2 + x^3$ for $x \in [0, 0.816]$ and $\lambda(x) \leq \frac{1-2\alpha}{4} < 0.816$. We can simplify the inequality even further:

$$\begin{aligned} \lambda(x) \leq \frac{1-2\alpha}{4}, \quad \frac{1}{2} - \alpha \geq \lambda(x), \\ \lambda(x)^2(\frac{1}{2} - \alpha - \lambda(x) \geq 0) \geq 0, \\ \frac{1}{2}\lambda(x)^2 - \lambda(x)^3 \geq \alpha\lambda(x)^2, \\ -\frac{1}{2}\lambda(x)^2 + \lambda(x)^3 \leq -\alpha\lambda(x)^2. \end{aligned} \tag{44}$$

Therefore, (43) can be simplified into

$$\tilde{f}(1) \leq \tilde{f}(0) - \alpha\lambda(x)^2,$$

or

$$f(x) - f(x_+) = \tilde{f}(0) - \tilde{f}(1) \geq \alpha\lambda(x)^2.$$

From this, we know that the unit step size $t = 1$ satisfies the exit condition of the backtracking line search. To prove that the convergent rate is quadratic, we need the following inequality:

$$\lambda(x_+) \leq \left(\frac{\lambda(x)}{1-\lambda(x)}\right)^2,$$

which is true for $\lambda(x) < 1$. The proof will not be presented here but can be found in ***. Since we have an even smaller upper bound $\lambda(x) \leq 1/4$, we get that

$$\begin{aligned} \frac{1}{(1-\lambda(x))^2} \leq 2, \\ \lambda(x_+) \leq 2\lambda(x)^2. \end{aligned} \tag{45}$$

Hence, by (39), $f(x_+) - f(x^*) \leq \lambda(x_+)^2 \leq (2\lambda(x))^2$. Applying (45) recursively, we get that

$$\begin{aligned} f(x_k) - f(x^*) \leq \lambda(x_k)^2 &\leq \left(\frac{1}{2}\right)^{1-2^k} (\lambda(x_0))^{2^k} \\ &\leq \left(\frac{1}{2}\right)^{1-2^k} \left(\frac{1}{4}\right)^{2^k} \\ &= \left(\frac{1}{2}\right)^{2^k+1}. \end{aligned} \tag{46}$$

Lastly, similar to the classic Newton's method, we can find an upper bound for the total number of iterations:

$$(f(x_0) - f(x^*))\frac{1+\eta}{\alpha\gamma\eta^2} + 6.$$

From this convergence analysis, we see that self-concordance can replace strong convexity and Lipschitz condition. The result is simpler and involves fewer variables.

5 Interior-point Method

In this section, we introduce the interior-point method, an algorithm that solves inequality constrained convex optimization problems. Recall the general form of such a problem from Definition 3.1.1:

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \text{ for } 1 \leq i \leq m, \\ & h_j(x) = 0, \text{ for } 1 \leq j \leq n, \end{aligned} \tag{47}$$

where $f_0, f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex and $\mathcal{C}^2$ functions and $h_1, \ldots, h_n : \mathbb{R}^n \to \mathbb{R}$ are affine functions. Let $\mathcal{D}$ be the domain of this optimization problem and let $\mathcal{X} \subseteq \mathcal{D}$ be the set of feasible points, i.e., for all $x \in \mathcal{X}$, $f_i(x) \leq 0$ and $h_i(x) = 0$. Let f^* be the optimal value and x be the optimal point that gives $f_0(x) = f^*$. Before we go into the details, we first introduce some basic definitions about primal and dual problems.

5.1 Primal and Dual Problem

We refer to the above constrained convex optimization problem as the **primal** problem. Define the **Lagrangian** equation $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ as

$$\mathcal{L}(x, u, v) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x) + \sum_{j=1}^{n} v_i h_i(x),$$

where $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$ are called **Lagrange multipliers**. We restrict the domain of u to be $\mathbb{R}^m_+$, i.e., $u_i \geq 0$ for all $1 \leq i \leq m$, such that inequality constraints f_i make sense.

For every $x \in \mathcal{X}, u \succeq 0$, we have $\mathcal{L}(x, u, v) \leq f_0(x)$. If we pick $u_i = 0$ whenever $f_i(x) < 0$, then

$$\inf_x \sup_{u \succeq 0,\, v} \mathcal{L}(x, u, v) = \inf_x f_0(x).$$

There $u \succeq 0$ means $u_i \geq 0$ entry-wise.

Now we define the **Lagrange Dual Function** to be

$$\begin{aligned} g(u, v) &= \inf_x \mathcal{L}(x, u, v) \\ &= \inf_x \left(f_0(x) + \sum_{i=1}^{m} u_i f_i(x) + \sum_{j=1}^{n} v_i h_i(x) \right). \end{aligned} \tag{48}$$

Given (47) and (48), we define the **Lagrange Dual Problem** as:

$$\begin{aligned} \underset{u \succeq 0, v}{\text{maximize}} \quad & g(u, v) \\ \text{subject to} \quad & u_i \geq 0, \text{ for } 1 \leq i \leq m. \end{aligned} \tag{49}$$

Let g^* denote the optimal value of the Lagrange Dual Problem. The pair (u^*, v^*) with which g^* is obtained is called **dual optimal**. For any fixed $x \in \mathcal{X}$, $\mathcal{L}(x, u, v)$ is an affine function of u and

v. Then g can be viewed as the point-wise infimum of the affine function of u and v, and thus is concave. The constraint $u \succeq 0$ is an affine constraint. So system (49) is a concave maximization problem, which is also a convex optimization problem.

In general, we do not have $f^* = g^*$, but f^* gives an upper bound for g^*, which leads to the following result.

Theorem 5.1.1. *The optimal value for the Primal problem is always greater than or equal to the optimal value for the Dual problem* (49)*, i.e., $f^* \geq g^*$. This inequality is called the* ***Weak Duality****.*

Proof. We know that

$$f^* = \inf_x \sup_{u \succeq 0,\, v} \mathcal{L}(x, u, v),$$

$$g^* = \sup_{u \succeq 0,\, v} \inf_x \mathcal{L}(x, u, v).$$

Notice that the following inequality is necessary:

$$\inf_x \mathcal{L}(x, u, v) \leq \sup_{u \succeq 0,\, v} \mathcal{L}(x, u, v).$$

Then we first take the infimum over x on both sides to get

$$\inf_x \mathcal{L}(x, u, v) \leq \inf_x \sup_{u \succeq 0,\, v} \mathcal{L}(x, u, v),$$

and then take the supremum over $u \succeq 0$, v on both sides,

$$\sup_{u \succeq 0,\, v} \inf_x \mathcal{L}(x, u, v) \leq \inf_x \sup_{u \succeq 0,\, v} \mathcal{L}(x, u, v) \Rightarrow f^* \geq g^*.$$

□

The difference $f^* - g^*$ is called the **Duality Gap** and **Strong Duality** states that there are x^*, u^*, v^* such that $f^* = g^*$. The following result presents a sufficient condition under which strong duality holds.

Theorem 5.1.2. ***Slater's Condition***

Suppose we have a convex optimization problem. If there exists at least one strictly feasible x in the domain, then the strong duality holds.

Geometrically speaking, Slater's Condition holds if the feasible region has an interior point.

Finally, we discuss the Karush-Kuhn-Tucker conditions or KKT conditions. These give us additional necessary conditions. Given (47), the KKT conditions are

- $\nabla f(x) + \sum_{i=1}^{m} u_i \nabla f_i(x) + \sum_{j=1}^{n} v_i \nabla h_i(x) = 0$ (Stationarity)
- $u_i f_i(x) = 0$ for $1 \leq i \leq m$ (Complementary slackness)
- $f_i(x) \leq 0, h_j(x) = 0$ for $1 \leq i \leq m$ (Primal feasibility)
- $u_i \geq 0$ for $1 \leq i \leq m$ (Dual feasibility)

Theorem 5.1.3. *(x^*, u^*, v^*) are primal and dual solutions such that the strong duality holds if and only if (x^*, u^*, v^*) satisfies the KKT conditions.*

Proof. ($\Rightarrow$) Since the strong duality holds with (x^*, u^*, v^*), we know that

$$\begin{aligned} f(x^*) &= g(u^*, v^*) \\ &= \inf_x \left(f_0(x) + \sum_{i=1}^m u_i^* f_i(x) + \sum_{j=1}^n v_i^* h_i(x) \right) \\ &\leq f_0(x^*) + \sum_{i=1}^m u_i^* f_i(x^*) + \sum_{j=1}^n v_i^* h_i(x^*) \\ &\leq f(x^*). \end{aligned}$$

We have the first inequality because plugging in any value for x would give a value greater than the infimum over all x. The second inequality comes from the fact that $f_i(x) \leq 0$, $u_i^* \geq 0$, and $h_i(x) = 0$. Since we cannot have $f(x^*) < f(x^*)$, all inequalities should be equality. Therefore, we get x^* gives the infimum of $\mathcal{L}(x, u, v) = f_0(x) + \sum_{i=1}^m u_i^* f_i(x) + \sum_{j=1}^n v_i^* h_i(x)$, which shows the stationarity condition. On top of that, the last equality tells us that $\sum_{i=1}^m u_i^* f_i(x^*) = 0$, which is the complementary slackness condition. Primal feasibility and dual feasibility are obviously true. ($\Leftarrow$) By integrating the stationarity condition, we get the following:

$$\begin{aligned} g(u^*, v^*) &= f_0(x^*) + \sum_{i=1}^m u_i^* f_i(x^*) + \sum_{j=1}^n v_i^* h_i(x^*) \\ &= f(x^*). \end{aligned}$$

We get the last equality from complementary slackness and primal feasibility conditions. Hence, we get that (x, u^*, v^*) satisfy that $f(x^*) = g(u^*, v^*)$, which gives the strong duality. □

5.2 Newton's Method with Equality constraints

In this section, we talk about the extension of Newton's method to optimization problems with equality constraints. This method will play an important role for our later discussion about the interior-point Method. Consider the following convex quadratic function with equality constraints:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T A x + b^T x + c \\ \text{subject to} \quad & Qx = q, \end{aligned} \tag{50}$$

where A is a $\mathbb{R}^{n \times n}$ symmetric positive semi-definite matrix and $Q \in \mathbb{R}^{p \times n}$, $\text{rank}(Q) = p < n$. The assumption on the dimension of Q says that there are fewer equality constraints than variables.

Since there is no inequality constraints, the KKT conditions can be simplified into the following:

- $Ax + b + Q^T v^* = 0$,
- $Qx = q$.

From the previous section, we know that x is the primal solution and v^* is the dual solution if and only if they satisfy the KKT conditions. The KKT conditions can be rewritten in the matrix form:

$$\begin{bmatrix} A & Q^T \\ Q & 0 \end{bmatrix} \begin{bmatrix} x \\ v^* \end{bmatrix} = \begin{bmatrix} -b \\ q \end{bmatrix}, \tag{51}$$

which is a set of $(n+p)$ linear equations. The coefficient matrix is called the KKT matrix. We can solve for x, v^* if the KKT matrix is non-singular. To answer this question, we introduce the following theorem:

Theorem 5.2.1. *Suppose A is a $\mathbb{R}^{n\times n}$ symmetric positive semi-definite matrix and $Q \in \mathbb{R}^{p\times n}$ such that rank$(Q) = p < n$. Then the following are equivalent to KKT matrix being non-singular:*

- $\mathcal{N}(A) \cap \mathcal{N}(Q) = \{0\}$, *i.e., the only vector that satisfies $Ax = Qx = 0$ is the zero vector.*
- $Qx = 0, x \neq 0$ *implies* $x^T Ax > 0$.
- $F^T AF \succ 0$, *where $F \in \mathbb{R}^{n\times(n-p)}$ is a matrix such that $\mathcal{R}(F) = \mathcal{N}(Q)$.*

Proof. $(1 \Rightarrow 2)$ Choose $x \neq 0 \in \mathcal{N}(Q)$. Since $\mathcal{N}(A) \cap \mathcal{N}(Q) = \{0\}$, we know that $x \notin \mathcal{N}(A)$, which means that $x^T Ax > 0$. Proved 2.

$(2 \Rightarrow 3)$ Choose any $x \in \mathbb{R}^{n-p}$. Then $Fx = z$ for some $z \in \mathcal{N}(Q)$. $x^T F^T AFx = z^T Az$. By the Rank-Nullity Theorem, $rank(A)$+dim$(\mathcal{N}(A)) = n$, i.e., dim$(\mathcal{R}(F)) = n - p$. So dim$(\mathcal{N}(F)) = 0$, which means that $\mathcal{N}(F) = \{0\}$. If $z \neq 0$, then we have $x \neq 0$ and $x^T F^T AFx = z^T Az > 0$. Therefore, for all $x \neq 0$, $x^T F^T AFx > 0$, which means that $F^T AF \succ 0$.

$(3 \Rightarrow 1)$ Choose any $x \neq 0$. dim$(\mathcal{N}(F)) = 0$ implies that $z = Fx \neq 0$. $\mathcal{R}(F) = \mathcal{N}(Q)$ implies that $z \in \mathcal{N}(Q)$. Since $F^T AF \succ 0$, we get that $z^T Az > 0$ for all $z \neq 0 \in \mathcal{N}(Q)$. This means that $\mathcal{N}(A) \cap \mathcal{N}(Q) = \{0\}$. □

In conclusion, if A is instead a symmetric positive definite matrix, then the KKT matrix is always non-singular.

Next, we discuss how Newton's method can be applied. The previous discussion about convex quadratic function is useful because we need to use the second order Taylor approximation. Suppose we are interested in finding the minimum of a $\mathcal{C}^2$ function $f : \mathbb{R}^n \to \mathbb{R}$ whose Hessian matrix is positive definite. Let Q be the same matrix that denotes the equality constraints. Assume that we start at a feasible point x_0 such that $Qx_0 = q$. We approximate $f(x_0)$ by its second order Taylor expansion $\hat{f}(x_0 + v)$ and get the following minimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} v^T \nabla^2 f(x_0) v + \nabla f(x_0)^T v + f(x_0) \\ \text{subject to} \quad & Q(x_0 + v) = q, \text{equivalently } Qv = 0. \end{aligned} \tag{52}$$

Here, the variable is v which is a descent direction that decreases the function value. Define the Newton step Δx to be the previous optimal solution x and v^* as before to be the dual solution. Similar to (51), the KKT conditions for this minimization problem become:

- $\nabla^2 f(x_0)\Delta x + \nabla f(x_0) + Q^T v^* = 0$,
- $Q\Delta x = 0$.

To write this in the matrix form, we get

$$\begin{bmatrix} \nabla^2 f(x_0) & Q^T \\ Q & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ v^* \end{bmatrix} = \begin{bmatrix} -\nabla f(x_0) \\ 0 \end{bmatrix}. \tag{53}$$

Since $\nabla^2 f(x_0)$ is positive definite, we know that the KKT matrix must be non-singular. So we can always invert the matrix and solve for Δx and v^*. Hence, we have computed the Newton step Δx which is the optimal solution that satisfies the equality constraints and minimize the second order approximation function. However, we still get some confusions that have not been verified yet. How do we know that Δx is a descent direction for $f(x)$, i.e., $f(x + t\Delta x) < f(x)$? Similar to the classic Newton's method, how is Newton decrement $\lambda(x)$ defined? Is $\lambda(x)$ still a good estimate of the distance between $f(x)$ and $\inf_v \hat{f}(x + v)$?

5.2.1 Newton Decrement

We define the Newton decrement for the equality constrained optimization problem to be:

$$\lambda(x) = (\Delta x^T \nabla^2 f(x) \Delta x)^{\frac{1}{2}}.$$

This is actually the same definition as in chapter 4. Previously, we defined $\Delta x = -\nabla^2 f(x)^{-1} \nabla f(x)$, and $\lambda(x) = (\Delta x^T \nabla^2 f(x) \Delta x)^{\frac{1}{2}}$, which is our definition for the classic Newton's method. Now, we discover the relation between $\lambda(x)$ and $f(x) - \inf_v f(x + v)$

Theorem 5.2.2. *The difference between f and its second order Taylor approximation satisfies $f(x) - \inf\{\hat{f}(x + v) \,|\, Qv = 0\} = \frac{\lambda(x)^2}{2}$.*

Proof. Note that $\hat{f}$ denotes the second order Taylor approximation of f. $\hat{f}(x + \Delta x) = \inf\{\hat{f}(x + v) \,|\, Qv = 0\}$. The Newton step is defined by the KKT conditions:

$$\begin{bmatrix} \nabla^2 f(x) & Q^T \\ Q & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ v^* \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix}. \tag{54}$$

From the first row of the KKT matrix, we get that

$$\nabla^2 f(x) \Delta x + Q^T v^* = -\nabla f(x).$$

We multiply Δx^T to both sides of the equation:

$$\Delta x^T \nabla^2 f(x) \Delta x = -\Delta x^T \nabla f(x), \tag{55}$$

because $(Q\Delta x)^T v^* = 0$. Then

$$\begin{aligned} \hat{f}(x + \Delta x) &= \frac{1}{2} \Delta x^T \nabla^2 f(x) \Delta x + \nabla f(x)^T \Delta x + f(x) \\ &= \frac{1}{2} \Delta x^T \nabla^2 f(x) \Delta x - \Delta x^T \nabla^2 f(x) \Delta x + f(x) \\ &= -\frac{1}{2} \lambda(x)^2 + f(x). \end{aligned} \tag{56}$$

So $f(x) - \hat{f}(x + \Delta x) = \frac{1}{2} \lambda(x)^2$ finishes the proof. □

This theorem means that similar to the classic Newton's method, $\frac{1}{2}\lambda(x)^2$ is a good indicator of the precision and serves as the stopping criterion. Now, we explain why Δx is a descent direction for $f(x)$. We just need to check that the directional derivative of f in the direction Δx is negative:

$$\frac{d}{dt} f(x + t\Delta x)\bigg|_{t=0} = \nabla f(x)^T \Delta x = -\lambda(x)^2 < 0.$$

So the Newton step is a descent direction for $f(x)$.

5.2.2 Algorithm for Equality Constrained Newton's Method

The following algorithm uses the backtracking line search to determine each iteration step size.

Algorithm 8: Newton's Method with Equality Constraints

NewtonMethod (f, Q, x_0, ϵ, α, γ);
Compute the initial λ^2 and Δx for x_0;
while $\frac{\lambda^2}{2} \geq \epsilon$ **do**
  $x_{i+1} = x_i + \Delta x$;
  while $f(x_i) - f(x_{i+1}) < -\alpha t \nabla f(x_i)^T \Delta x$, **do**
    $t = \gamma t$;
    $x_{i+1} = x_i + t\Delta x$;
  $x_i = x_{i+1}$;
  $\begin{bmatrix} \Delta x \\ v^* \end{bmatrix} = \begin{bmatrix} \nabla^2 f(x) & Q^T \\ Q & 0 \end{bmatrix}^{-1} \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix}$;
  $\lambda^2 = \Delta x^T \nabla^2 f(x) \Delta x$;
return x_i;

5.3 Barrier Method and Logarithmic Barrier Function

Our goal now is to use equality constrained Newton's method to solve the minimization problem with both inequality and equality constraints. Recall the statement of the problem:

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \text{ for } 1 \leq i \leq m, \\ & Ax = b, \end{aligned} \tag{57}$$

where $f_0, f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex and twice continuously differentiable functions and $A \in \mathbb{R}^{p \times n}$, $\text{rank}(A) = p < n$. Let $\mathcal{D}$ be the domain of this optimization problem. How can we transform the problem to be a minimization problem with only equality constraints? The clever idea is that we use a differentiable function to approximate the inequality constraint.

One immediate answer would be to use an indicator function to denote the inequality constraint, that is

$$\mathbb{1}_{\{f_i(x) > 0\}} = \begin{cases} 0 & \text{if } f(x) \neq 0 \\ 1 & \text{if } f(x) > 0 \end{cases}.$$

Then the original minimization problem can be changed into the following:

$$\begin{aligned} \text{minimize} \quad & f_0(x) + \sum_{i=1}^{m} \mathbb{1}_{\{f_i(x)>0\}} \cdot \infty \\ \text{subject to} \quad & Ax = b. \end{aligned}$$

We get rid of the inequality constraints but the objective function is no longer differentiable. So using the indicator function is not a good choice. Our next step is to approximate this indicator function.

We use the following log function to approximate the indicator function:

$$I_t(x) = -\frac{1}{t} log(-x), \text{dom}(I_t) = -\mathbb{R}_+,$$

where $t > 0$ is a variable that determines the accuracy of the approximation. The following Figure 14 shows the approximation. The red curve has a much bigger value of t compared to the blue curve. As a result, the red curve is a better approximation to the indicator function than the blue curve. From the picture, we know that as t gets larger, $I_t(x)$ approximates the indicator function better. Unlike the indicator function, $I_t(x)$ is convex, differentiable and it increases to positive infinity as x goes to 0.

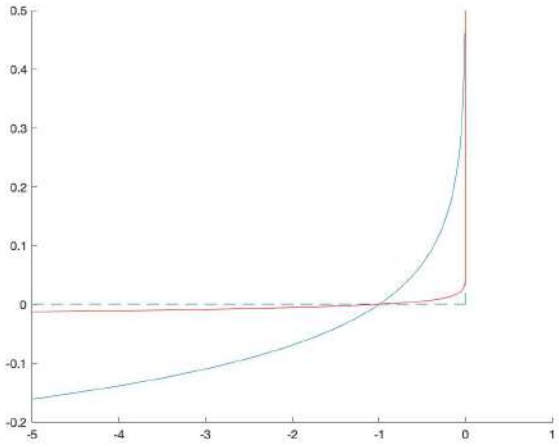


Figure 14: The Log Function Approximates the Indicator Function

We can update our minimization problem by replacing the indicator function with this log function:

$$\begin{aligned} \text{minimize} \quad & f_0(x) + \sum_{i=1}^{m} I_t(f_i(x)) = f_0(x) + \sum_{i=1}^{m} -\frac{1}{t} log(-f_i(x)) \\ \text{subject to} \quad & Ax = b. \end{aligned}$$

We define $\phi(x) = -\sum_{i=1}^{m} log(-f_i(x))$ with $\text{dom}(\phi(x)) = \{x \in \mathcal{D} | f_i(x) < 0, i = 1, 2, \ldots, m\}$ to be the **logarithmic barrier function**. The above new objective function is convex and twice

continuously differentiable. To simplify the problem, we multiply t to the objective function and get the following equivalent problem:

$$\begin{aligned} \text{minimize} \quad & tf_0(x) + \phi(x) \\ \text{subject to} \quad & Ax = b. \end{aligned} \tag{58}$$

This new problem has a different minimum function value but the minimizer x^* is the same as in the previous problem. For now, we assume that the above minimization problem can be solved by the equality constrained Newton's method. Here we compute the gradient and Hessian matrix of the objective function that will be useful when using Newton's method:

$$\begin{aligned} \nabla(tf_0(x) + \phi(x)) &= t\nabla f_0(x) + \nabla\phi(x) = t\nabla f_0(x) - \sum_{i=1}^{m} \frac{\nabla f_i(x)}{f_i(x)}, \\ \nabla^2(tf_0(x) + \phi(x)) &= t\nabla^2 f_0(x) + \nabla^2\phi(x) \\ &= t\nabla^2 f_0(x) + \sum_{i=1}^{m} \frac{1}{f_i(x)^2}\nabla f_i(x)^T\nabla f_i(x) + \sum_{i=1}^{m} \frac{1}{-f_i(x)}\nabla^2 f_i(x). \end{aligned}$$

For each $t > 0$, we have a corresponding minimization problem (58), from which we get a unique solution $x^*(t)$. We define the **central path** to be the set of points $x^*(t), t > 0$. Notice that x^*, u^*, v^* all depend on t. Different t gives different value for these three variables. For future reference, we use x^*, u^*, v^* to mean $x^*(t), u^*(t), v^*(t)$. By the KKT conditions, we know that points on the central path x^* that is strictly feasible satisfies

- $t\nabla f_0(x^*) + \nabla\phi(x^*) + A^T\hat{v} = 0$, for some $\hat{v} \in \mathbb{R}^p$,
- $f_i(x^*) < 0,$ for $i = 1, 2, \ldots, m, Ax^* = b.$

From the above conditions, we can derive a dual solution (u^*, v^*) that satisfy

$$\nabla f(x^*) + \sum_{i=1}^{m} {u_i}^*\nabla f_i(x^*) + A^T v^* = 0. \tag{59}$$

Define u^* and v^* by the following formula:

$${u_i}^* = \frac{-1}{tf_i(x^*)} \text{ for } i = 1, 2, \ldots, m, \text{ and } v^* = \frac{\hat{v}}{t}. \tag{60}$$

We see that ${u_i}^* > 0$ because $f_i(x^*) < 0$ and $t > 0$. In addition,

$$\begin{aligned} 0 &= t\nabla f_0(x^*) + \nabla\phi(x^*) + A^T\hat{v} \\ &= t\left(\nabla f_0(x^*) - \sum_{i=1}^{m} \frac{\nabla f_i(x)}{tf_i(x)} + A^T v^*\right) \\ &= t\left(\nabla f_0(x^*) - \sum_{i=1}^{m} {u_i}^*\nabla f_i(x) + A^T v^*\right). \end{aligned}$$

Therefore, we find the dual solution such that (59) is true. This means that x^* minimizes the Lagrangian equation $\mathcal{L}(x, u^*, v^*) = f_0(x) + \sum_{i=1}^{m} {u_i}^* f_i(x) + (Ax - b)^T v^*$. Since the dual function $g(u^*, v^*)$ is defined to be the infimum of $\mathcal{L}(x, u^*, v^*)$ over all x, we get that

$$\begin{aligned} g(u^*, v^*) &= f_0(x^*) + \sum_{i=1}^{m} {u_i}^* f_i(x^*) + (Ax^* - b)^T v^* \\ &= f_0(x^*) + \sum_{i=1}^{m} \frac{-1}{t f_i(x^*)} f_i(x^*) + (Ax^* - b)^T v^* \\ &= f_0(x^*) - \frac{m}{t}. \end{aligned} \tag{61}$$

The last term $(Ax^* - b)^T v^*$ disappears because $Ax^* = b$. By Weak Duality, we know that

$$\frac{m}{t} = f_0(x^*) - g(u^*, v^*) \geq f_0(x^*) - {f_0}^*,$$

where we proved the important fact that $f_0(x^*(t))$ approaches to the infimum of f_0 as $t \to \infty$. As said before, x^* is a variable that depends on t. In order to see this clearly, we write x^* for $x^*(t)$.

5.3.1 Algorithm for Barrier Method

Here, we present the algorithm. The following algorithm uses nested while loop. The outer while loop generates a new objective function $tf_0 + \phi$ and updates t and x^*. The inner while loop uses Newton's method to compute the minimizer x^* given the objective function.

Algorithm 9: Newton's Method With Equality Constraints

BarrierMethod (f_0, f_i, A, x_0, ϵ, α, γ, $t_0 > 0, \beta > 1$);
Compute the $\phi(x)$, KKT matrix and Δx for x_0;
$t = t_0$;
while $m/t \geq \epsilon$ **do**
1. Use equality constrained Newton's method to minimize $tf_0 + \phi$ with x_0 as the starting point and solve for x^*;
2. $x_0 = x^*$;
3. $t = \beta t$;

return x_0;

The iteration step for Newton's method is the following:

Algorithm 10: Newton's Method Iteration Step

while $\frac{\lambda^2}{2} \geq \epsilon$ **do**
$x_{i+1} = x_i + \Delta x$;
while $f(x_i) - f(x_{i+1}) < -\alpha t \nabla f(x_i)^T \Delta x$, **do**
$t = \gamma t$;
$x_{i+1} = x_i + t\Delta x$;
$x_i = x_{i+1}$;
$$\begin{bmatrix} t\nabla^2 f_0(x) + \nabla^2 \phi(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ v^* \end{bmatrix} = \begin{bmatrix} -t\nabla f(x) - \nabla \phi(x) \\ 0 \end{bmatrix};$$
$\lambda^2 = \Delta x^T \nabla^2 f(x) \Delta x$;

6 Finite Element Method

In this section, we use one simple example to explain the essential idea of the Finite Element Method without diving to much into the technical details. The example we will use is the Poisson Problem. We also discuss the correction to convex optimization. Consider the following problem

$$-\Delta u = f \text{ in } \Omega,$$

$$u = 0 \text{ on } \partial\Omega,$$

where $\Omega \in \mathbb{R}^d$ is a bounded domain and $\partial\Omega$ is the boundary of the domain Ω. Here, $\Delta u = \sum_{i=1}^{d} \partial_i^2 u$ is the Laplacian, which involves second order derivatives. To simplify the problem, we want to lower the degree of the derivative by using integration by parts.

We let $\mathcal{H}(\Omega)$ be the Hilbert space of functions v such that $v \in L^2(\Omega)$, derivatives $\partial_1 v, \partial_2 v, \ldots, \partial_n v$ are in $L^2(\Omega)$, and $v = 0$ on $\partial\Omega$.

The **Weak Formulation** of the Poisson problem is derived from the following. We multiply the equation $\Delta u = f$ by a function v that vanishes along $\partial\Omega$, that is, $v = 0$ on $\partial\Omega$. Then we integrate:

$$\int_\Omega -(\Delta u)\, v \; dx = \int_\Omega f v \; dx.$$

By using integration by parts, we get

$$\begin{aligned} \int_\Omega -(\Delta u)\, v \; dx &= \int_\Omega -\mathrm{div}(\nabla u \cdot v) \; dx + \int_\Omega \nabla u \cdot \nabla v \; dx \\ &= \int_\Omega \nabla u \cdot \nabla v \; dx \end{aligned} \tag{62}$$

Here, we have used Gauss's Theorem in vector analysis. Therefore, we have got

$$\int_\Omega \nabla u \cdot \nabla v \; dx = \int_\Omega f v \; dx.$$

We redefine the integrals in the following way:

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \; dx, \; b(v) = \int_\Omega f v \; dx.$$

So we have got $a(u, v) = b(v)$ for all $v \in \mathcal{H}(\Omega)$, $u \in \mathcal{H}(\Omega)$. This is the weak formulation of the original Poisson problem.

Instead of computing the exact solution for u, we want to obtain a good approximation. What we will do is to try to solve the same problem in a finite-dimensional subspace of $\mathcal{H}(\Omega)$.

Let $S \subseteq \mathcal{H}(\Omega)$ be a finite dimensional subspace. We know that $a : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is a continuous and bilinear function because

$$a(\alpha u + w, v) = \int_\Omega \alpha \nabla u \cdot \nabla v \; dx + \int_\Omega \nabla w \cdot \nabla v \; dx = \alpha a(u, v) + a(w, v).$$

Since a are symmetric, it follows that $a(u, v)$ is bilinear. Similarly, b is also a continuous and linear function because integration is a continuous linear function. The discrete version of the Poisson problem becomes

$$\text{finding } U \in S \text{ such that } a(U, V) = b(V) \text{ for all } V \in S,$$

and U is called the **discrete solution**.

Now, we have got a modified version of the Poisson problem that is defined on a finite-dimensional space. Then the next thing we want to do is to construct a basis for the subspace S and express functions $a(U, V)$ and $b(V)$ in terms of that basis.

Suppose we have a basis $(\eta_1, \eta_2, \ldots, \eta_N)$ of S. Consider the matrix $A \in \mathbb{R}^{N\times N}$ formed by the function $a(U, V)$ evaluated at these basis functions. So we get

$$A(i, j) = a(\eta_i, \eta_j), 1 \leq i, j \leq N.$$

The matrix A is called the **stiffness matrix** of the discrete Poisson problem. Notice that A is a symmetric matrix. Evaluating the function b by using the same basis functions, we get a vector in $\mathbb{R}^N$, that is

$$b_j = b(\eta_j), 1 \leq j \leq N.$$

This vector b is called the **load vector** of the discrete problem. With what we defined just now, we present the following theorem.

Theorem 6.0.1. *Suppose we are still using the same set of basis functions. The discrete solution can be written as a linear combination of the basis, i.e.,*

$$U = \sum_{j=1}^{N} x_j \eta_j.$$

Then the coefficient vector $x = (x_1, x_2, \ldots, x_n)$ is the unique solution of the linear system of equations

$$Ax = b$$

where A is the stiffness matrix and b is the load vector.

Proof. Choose any $V \in S$ such that $V = \sum_{i=1}^{N} y_i \eta_i$. By the definition of A and b, we know that $a(U, V) = b(V)$. Replacing U and V by their linear combinations and using the bilinearity of $a(U, V)$ yield

$$\begin{aligned} a(U, V) &= a(U, \sum_{i=1}^{N} y_i \eta_i) = \sum_{i=1}^{N} y_i a(U, \eta_i) = \sum_{i=1}^{N} y_i a(\sum_{j=1}^{N} x_j \eta_j, \eta_i) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} y_i x_j a(\eta_j, \eta_i) = \sum_{i=1}^{N}\sum_{j=1}^{N} y_i x_j A_{ji} = y^T A^T x \\ &= y^T A x. \end{aligned} \tag{63}$$

A similar computation for $b(V)$ yields

$$b(V) = b(\sum_{i=1}^{N} y_i \eta_i) = \sum_{i=1}^{N} y_i b(\eta_i) = \sum_{i=1}^{N} y_i b_i = y^T b. \tag{64}$$

So $y^T Ax = y^T b$ for all $y \in \mathbb{R}^N$. This means that $Ax = b$. In addition, $a(V, V) = y^T Ay > 0$ whenever $y \neq 0$. This means that A is positive definite. □

Example 6.0.2. *We consider a specific one-dimensional Poisson problem:*

$$-u'' = 1 \text{ in } (0,1) = \Omega \text{ and } u(0) = u(1) = 0.$$

The Weak Formulation of this problem is the following:

$$\text{for all } v \in \mathcal{H}((0,1)),\ v(0) = v(1) = 0,\ a(u,v) = \int_{(0,1)} u'v'\,dx, \text{ and } b(v) = \int_{(0,1)} v\,dx.$$

Next, we try to generate a discrete version of the problem. We start from constructing the basis functions. For simplicity, we choose the following set of trigonometric functions as the basis:

$$B = \{sin(k\pi x) \,|\, k = 1, 2, \ldots, N\}.$$

Then the subspace spanned by this set of basis is

$$S = \left\{ \sum_{k=1}^{N} c_k\, sin(k\pi x) \ :\ c_k \in \mathbb{R} \right\}.$$

The entry (i, j) of the stiffness matrix becomes

$$\begin{aligned} A(i,j) = a(\eta_i, \eta_j) &= \int_{(0,1)} sin'(i\pi x) sin'(j\pi x)\,dx \\ &= i\pi j\pi \int_{(0,1)} cos(i\pi x) cos(j\pi x)\,dx \\ &= \frac{ij\pi^2}{2} \int_{(0,1)} cos((i+j)\pi x) + cos((i-j)\pi x)\,dx \\ &= \begin{cases} \frac{(i\pi)^2}{2} \int_{(0,1)} cos(2i\pi x) + 1\,dx & \text{if } i = j \\ \frac{ij\pi^2}{2} \int_{(0,1)} cos((i+j)\pi x) + cos((i-j)\pi x)\,dx & \text{if } i \neq j \end{cases} \\ &= \begin{cases} \frac{(i\pi)^2}{2} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}. \end{aligned} \tag{65}$$

Notice that the stiffness matrix A has positive value on the diagonal and zero everywhere else.

The ith entry of the load vector b becomes

$$\begin{aligned} b_i = b(\eta_i) &= \int_{(0,1)} sin(i\pi x)\,dx \\ &= \frac{-1}{i\pi} cos(i\pi x)\Big|_0^1 = \frac{-1}{i\pi} cos(i\pi) + \frac{1}{i\pi} \\ &= \begin{cases} \frac{2}{i\pi} & \text{if } i \text{ is odd} \\ 0 & \text{if } i \text{ is even} \end{cases}. \end{aligned} \tag{66}$$

To find the discrete solution $U = \sum_{j=i}^{N} x_j \eta_j$, we just need to find the solution for $Ax = b$.

In this example, we have used a subspace spanned by trigonometric functions. The finite element method uses a space S that is spanned by continuous functions that are piecewise linear with respect to a partition of the interval into non-overlapping sub-intervals. The basis B are the so-called hat functions.

Here is a picture of the discrete solution U and the real solution $u = \frac{1}{2}(x - x^2)$.[5] The blue crosses show the approximation U and the read smooth curve denotes the real solution u. We see that the approximation is very precise.

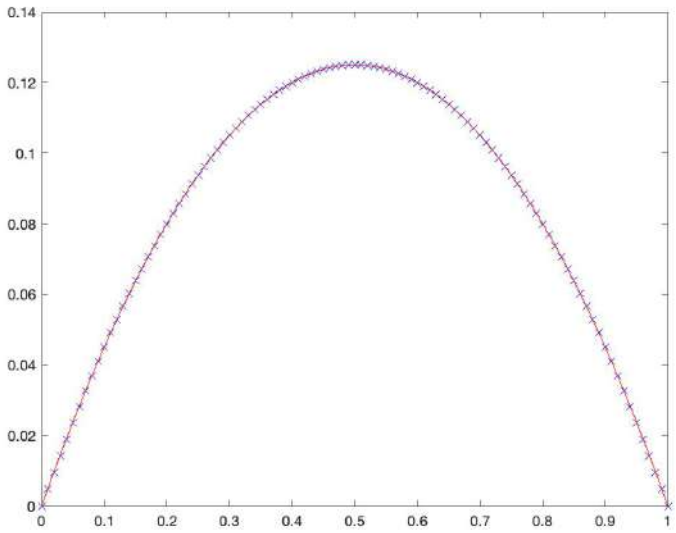


Figure 15: Finite Element Method of the Poisson Problem

We now outline the relation to convex optimization. For the finite element method, we need to solve the system $Ax = b$. This is equivalent to finding the minimum of the convex function

$$f_0(x) = \frac{1}{2}x^T Ax - b^T x.$$

We have seen such quadratic functions before. This shows how convex optimization is related to the finite element method.

We can also use equality and inequality constraints. We can give the equality constraint $x_i = 0$ or the inequality constraint $x_i \geq 0$. For example, that way we can fix the values of the function u or require that u is non-negative.

[5]The picture is obtained from MATLAB. The code is provided in [2].

References

[1] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[2] Andreas Byfut. Lecture notes in computational partial differential equations I, May 4 2007.

[3] Jonathan Richard Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. NSERC, 1994.

[4] Wikipedia. Finite element method — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Finite\%20element\%20method&oldid=1014865140`, 2021. [Online; accessed 18-May-2021].

[5] Wikipedia. Interior-point method — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Interior-point\%20method&oldid=1008695689`, 2021. [Online; accessed 06-May-2021].