2005

The eigenstep method: A new iterative method for unconstrained quadratic optimization.

John P. Battaglia
University of Windsor

THE EIGENSTEP METHOD: A NEW ITERATIVE METHOD FOR UNCONSTRAINED QUADRATIC OPTIMIZATION

by

John P. Battaglia

A Thesis
Submitted to the Faculty of Graduate Studies and Research
through the Department of Mathematics and Statistics
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

Your file Votre référence
ISBN: 0-494-04964-2
Our file Notre référence
ISBN: 0-494-04964-2

Abstract

This thesis presents a new method for the unconstrained minimization of convex quadratic programming problems. The method is an iterative method that is a modification of the classical steepest descent method. The methods are the same in the choice of the negative gradient as the search direction, but differ in the choice of step size. The steepest descent method uses the optimal step size, and the proposed method uses the reciprocal of the eigenvalues of the Hessian matrix as step sizes. Thus, the proposed method is referred to as the eigenstep method.

It will be shown that the eigenstep method has finite termination with the number of iterations required being equal to the dimension of the problem, that is, the number of variables. Numerical examples will be provided to illustrate the algorithm, and a comparison is made to other standard optimization methods, including the steepest descent method.

Acknowledgements

I would like to thank my faculty advisor, Dr. R. J. Caron, Dean of the Faculty of Science at the University of Windsor, for his guidance, patience, financial support and encouragement for the past several years. I also would like to thank Dr. A. Alfakih, and Dr. G. Zhang for their time and effort as my thesis committee members. I also would like to thank the University of Windsor for the financial support I have received for my study period at the university. Finally, I would like to thank Cheng Wang, a fellow student, for his computer programming assistance.

Contents

1. Introduction

This thesis presents a new method for the unconstrained minimization of the convex quadratic programming problem (QP) given by

$$\min f(x) = \frac{1}{2}x^{\mathsf{T}}Qx - b^{\mathsf{T}}x,$$

where $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$ and Q is a $n \times n$ symmetric, positive definite matrix. Since Q is real, symmetric and positive definite it has n real positive eigenvalues λ_k, $k = 0, 1, \ldots n-1$. The function f is referred to as the objective function. The vector of first derivatives of $f(x)$ is $\nabla f(x) = Qx - b$. The matrix of second derivatives of $f(x)$ is the Hessian matrix Q.

To solve the QP many algorithms, including the ones discussed in this thesis use the iteration

$$x_{k+1} = x_k + \alpha_k s_k,$$

where α_k is a positive real scalar referred to as the step-size, and $s_k \in \mathbb{R}^n$ is referred to as the search direction. For convenience, we use the notation g_k to refer to the gradient of $f(x)$ at x_k, i.e., $g_k = \nabla f(x_k)$.

The steepest descent method, invented in the nineteenth century by Cauchy [6, p. 339] uses $s_k = -g_k$ as the search direction and uses the optimal step size as described in lemma 2.8. For instances of the QP in which the eigenvalues of the Hessian have different orders of magnitude, it is known that the steepest descent method can exhibit poor performance.

In 1988 Barzilai and Borwein [2] presented alternatives to the optimal step size for the steepest descent method applied to the QP as well as more general problems. They introduced the two point step size or a secant type step

$$\alpha_k = \frac{s_k^{\mathsf{T}} s_k}{s_k^{\mathsf{T}}(g_k - g_{k-1})},$$

where $s_k = x_k - x_{k-1}$.

In 1999 Friedlander et al. [4] and in 2002 Dai and Liao [3] both give commentary on the strategy of steplength choice by Barzilai and Borwein, describing how it improves the steepest descent method.

Inspired by the work of Barzilai and Borwein, Raydan and Svaiter [1] introduced a modification of the steepest descent method applied to the QP by multiplying the optimal step size by a scalar θ, $0 < \theta < 2$, thereby giving what might be called an under relaxation ($0 < \theta < 1$) or an over relaxation ($1 < \theta < 2$). They show, with extensive numerical testing, that with this approach one can improve on the performance of the algorithm.

The new eigenstep method presented in this thesis was motivated by these papers and the results of lemmas 3.1 and 3.2. We present a modifcation of the steepest descent method by taking as step sizes the reciprocals of the eigenvalues of Q.

The above papers stress that altering the step size of the steepest descent method reduces the number of iterations required to solve our QP. However this step size altering cannot ensure termination in n or fewer steps. To the best of our knowledge our new eigenstep method is the *only* method that, for our QP alters the step size of the classical steepest descent method and terminates in n or less steps. Numerical examples are provided to illustrate the algorithm, and a comparison is made to other standard optimization methods, including the steepest descent method.

In chapter 2 we review and summarize some essential unconstrained nonlinear optimization theory as well as provide a brief description of four popular solution methods.

(1) The Steepest Descent Method.
(2) The Conjugate Gradient Method.
(3) The Newton Method.
(4) The Broyden, Fletcher, Goldfarb, and Shanno (BFGS) method.

In chapter 3 we provide two lemmas that led us to the development of our new algorithm. We also prove that our new algorithm terminates in at most n iterations. We then discuss the transformation process which sets the stage for the general proof of finite termination.

In chapter 4 two examples are provided as well as numerical experiments comparing our new eigenstep method to the conjugate gradient method and the steepest descent method. The steepest descent method is hopelessly inefficient at solving our QP, due to its choice of step size. This is the main issue of this thesis, we alter the step size of the steepest descent method and attain optimality in n or fewer iterations.

2. Background

In this chapter we present some definitions and results pertinent to our QP and to iterative methods for its solution. We then describe the four methods listed in the first chapter.

Definition 2.1. *The directional derivative for a function f is defined as the instantaneous rate of change of f along a direction s and is given by*

$$\lim_{\alpha \to 0} \frac{f(x+\alpha s) - f(x)}{\alpha} = \nabla f(x)^{\mathsf{T}} s.$$

Definition 2.2. *A function f is strictly convex if and only if*

$$f(\theta x_1 + (1-\theta)x_2) < \theta f(x_1) + (1-\theta) f(x_2)$$

for all $x_1, x_2 \in \mathbb{R}^n$, $0 < \theta < 1$.

Definition 2.3. *A symmetric matrix Q is positive definite if $x^{\mathsf{T}} Q x > 0$ for all $x \in \mathbb{R}^n$ $x \neq 0$.*

The next result follows from Taylor's theorem.

Lemma 2.4. *If $f(x)$ is quadratic, then*

$$\begin{aligned}
(a) \quad f(x_2) &= f(x_1) + \nabla f(x_1)^{\mathsf{T}}(x_2 - x_1) + \frac{1}{2}(x_2 - x_1)^{\mathsf{T}} Q (x_2 - x_1), \\
(b) \quad f(x_{k+1}) &= f(x_k) + \alpha_k g_k^{\mathsf{T}} s_k + \frac{1}{2}\alpha^2 s_k^{\mathsf{T}} Q s_k, \\
(c) \quad g_{k+1} &= g_k + \alpha_k Q s_k.
\end{aligned}$$

Theorem 2.5. *The differentiable function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, is strictly convex if and only if for all $x_1, x_2 \in \mathbb{R}^n$, $x_1 \neq x_2$*

$$f(x_2) > f(x_1) + \nabla f(x_1)^{\mathsf{T}}(x_2 - x_1)$$

Proof. Assume f is strictly convex. By definition 2.2 for all $0 < \theta < 1$,

$$f(\theta x_1 + (1-\theta)x_2) < \theta f(x_1) + (1-\theta)f(x_2)$$

$$\Leftrightarrow f(x_2 + \theta(x_1 - x_2)) - f(x_2) < \theta(f(x_1) - f(x_2)).$$

Divide both sides of the above inequality by θ and let $\theta \longrightarrow 0$, to obtain by definition 2.1

$$\nabla f(x_2)^\top (x_1 - x_2) < f(x_1) - f(x_2)$$

$$\Leftrightarrow f(x_2) > f(x_1) + \nabla f(x_1)^\top (x_2 - x_1).$$

Conversely, let $x_1, x_2 \in \mathbb{R}^n$, $\overline{x} = \theta x_1 + (1-\theta)x_2$ for all $0 < \theta < 1$, be arbitrary but fixed. By hypothesis

$$f(x_1) > f(\overline{x}) + \nabla f(\overline{x})^\top (x_1 - \overline{x}) \quad (1)$$

$$f(x_2) > f(\overline{x}) + \nabla f(\overline{x})^\top (x_2 - \overline{x}) \quad (2)$$

substituting $x_1 - \overline{x} = (1-\theta)(x_1 - x_2)$ and $x_2 - \overline{x} = \theta(x_2 - x_1)$ in (1) and (2) above we obtain

$$f(x_1) > f(\overline{x}) + \nabla f(\overline{x})^\top (x_1 - x_2)(1-\theta) \quad (3)$$

$$f(x_2) > f(\overline{x}) + \nabla f(\overline{x})^\top (x_2 - x_1)(\theta) \quad (4)$$

now multiplying (3) by θ and (4) by $(1-\theta)$ then adding (3) and (4)we obtain

$$f(\theta x_1 + (1-\theta)x_2) < \theta f(x_1) + (1-\theta)f(x_2).$$

Therefore f is convex. □

Lemma 2.6. *Q is positive definite if and only if $f(x)$ is strictly convex.*

Proof. Assume Q is positive definite. Let $x_1, x_2 \in \mathbb{R}^n$ be arbitrary but fixed, $x_1 \neq x_2$. From lemma 2.4(a)

$$f(x_2) = f(x_1) + \nabla f(x_1)^\top (x_2 - x_1) + \frac{1}{2}(x_2 - x_1)^\top Q(x_2 - x_1)$$

since Q is positive definite $\frac{1}{2}(x_2 - x_1)^\top Q(x_2 - x_1) > 0$ therefore

$$f(x_2) > f(x_1) + \nabla f(x_1)^\top (x_2 - x_1)$$

therefore by theorem 2.5 it follows that f is strictly convex. Conversley, assume f is strictly convex. From lemma 2.4(a)

$$f(x_2) = f(x_1) + \nabla f(x_1)^\top (x_2 - x_1) + \frac{1}{2}(x_2 - x_1)^\top Q(x_2 - x_1).$$

Next from theorem 2.5

$$f(x_2) > f(x_1) + \nabla f(x_1)^\top (x_2 - x_1).$$

Therefore $\frac{1}{2}(x_2 - x_1)^\top Q(x_2 - x_1) > 0$ for all $x_1, x_2 \in \mathbb{R}^n$, therefore Q is positive definite. □

Lemma 2.7. *If $f(x)$ is strictly convex then $\nabla f(x^*) = 0$ is a necessary and sufficent condition for x^* to be the unique global minimizer of $f(x)$.*

Proof. Assume that $\nabla f(x^*) = 0$. Since f is convex, it follows from theorem 2.5, that for all $x \in R^n$, $x \neq x^*$,

$$f(x) > f(x^*) + \nabla f(x^*)^\top (x - x^*) \Rightarrow f(x) > f(x^*)$$

which establishes that x^* is the unique global minimizer. Conversely, assume that x^* is the unique global minimizer of $f(x)$. Thus, for all directions s, $\nabla f(x^*)^\top s > 0$. Next we assume $\nabla f(x^*) \neq 0$. Selecting $s = -\nabla f(x^*)$ we obtain

$$\nabla f(x^*)^\top s = -\nabla f(x^*)^\top \nabla f(x^*) < 0.$$

The contradiction is evident. □

Next we review the optimal step size of a line search. Consider the standard iteration, $x_{k+1} = x_k + \alpha_k s_k$. We say that α_k is the optimal step size, which we will denote by α_k^*, if

$$f(x_k + \alpha_k s_k) = \min_{\alpha \in \mathbb{R}} \{f(x_k + \alpha s_k)\}.$$

Lemma 2.8. *For our QP, the optimal step size is*

$$\alpha_k^* = -\frac{g_k^\top s_k}{s_k^\top Q s_k}.$$

Proof. From lemma 2.4(b), we have

$$f(x_k + \alpha s_k) = f(x_k) + \alpha g_k^\top s_k + \frac{\alpha^2}{2} s_k^\top Q s_k.$$

so that

$$\frac{df(x_k + \alpha s_k)}{d\alpha} = g_k^\top s_k + \alpha s_k^\top Q s_k.$$

Since Q is positive definite, we know that $s_k^\top Q s_k > 0$, so that setting the derivative to zero gives the optimal step size

$$\alpha^* = -\frac{g_k^\top s_k}{s_k^\top Q s_k}.$$

□

The next lemma gives the optimal step size property.

Lemma 2.9. *If α_k is the optimal step size, that is, if $\alpha_k = \alpha_k^*$ then*

$$g_{k+1}^\top s_k = 0.$$

Proof. From lemma 2.4(c)

$$\begin{aligned} g_{k+1}^\top s_k &= g_k^\top s_k + \alpha^* (Q s_k)^\top s_k \\ &= g_k^\top s_k - \frac{(g_k^\top s_k)(s_k^\top Q s_k)}{s_k^\top Q s_k} \\ &= g_k^\top s_k - g_k^\top s_k \\ &= 0 \end{aligned}$$

□

2.1. The Steepest Descent Method.

The steepest descent is an iterative method with $s_k = -g_k$ and $\alpha_k = \alpha_k^*$. The following lemma shows that $-g_k$ is a direction of steepest descent.

Lemma 2.10. *The direction of maximum local decrease for $f(x)$ at x_k is $-g_k$.*

Proof. We know $g_k^\top s_k$ describes the instantaneous rate of change of f along the direction s_k. Since

$$g_k^\top s_k = \| g_k \| \| s_k \| cos\theta,$$

where θ is the angle between g_k and s_k, a solution to

$$\min_{s_k} g_k^\top s_k$$

is attained when $cos\theta = -1$, i.e., when $\theta = \pi$. Thus, $s_k = -g_k$ is a direction of steepest descent. □

Since $s_k = -g_k$ we can rewrite the optimal step size (see Lemma 2.8) as

$$\alpha_k = \frac{g_k^\top g_k}{g_k^\top Q g_k}.$$

Next we discuss convergence of the steepest descent method. Recall that a sequence $\{x_k\}$ converges to x^* with rate r if

$$\lim_{k\to\infty} \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|^r} < \infty.$$

For linear convergence we have, $\| x_{k+1} - x^* \| = C \| x_k - x^* \|$ where $C < \infty$. For the steepest descent method we utilize the Q-norm of x denoted $\| x \|_Q^2 = x^\top Q x$. It can be shown [6,p.342] that

$$\frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|} \leq \frac{(\lambda_n - \lambda_1)^2}{(\lambda_n + \lambda_1)^2}$$

where $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ are the eigenvalues of Q. In other words the steepest descent method converges Q-linearly.

2.2. The Conjugate Gradient Method.

It can be shown [6,p.389] that this method solves our QP in n steps using the standard iteration, $x_{k+1} = x_k + \alpha_k s_k$, hence it is much more efficient than the the steepest descent method. This method gets its name from the fact that it generates a set of vectors $s_0, s_1, ..., s_{n-1}$ that are conjugate with

respect to Q that is, $s_i^T Q s_j = 0$ for $i \neq j$. In the case where $Q = I$ the conjugate vectors are just orthogonal vectors. As with the steepest descent method this algorithm uses the optimal step size and the negative gradient as its initial search direction i.e., $s_0 = -g_0$. The conjugate vectors are generated as follows:

$$s_{k+1} = -\nabla f(x_{k+1}) + \frac{\| \nabla f(x_{k+1}) \|^2}{\| \nabla f(x_k) \|^2} s_k.$$

2.3. Newton's Method.

Newton's method is a second order method and not directly related to this thesis. This method was used in our experiments only for illustrative purposes and to compare CPU times. Newton's method for optimization uses a unit step size and is given by the iterative step

$$x_{k+1} = x_k - [\nabla^2 f(x)]^{-1} \nabla f(x_k)$$

Note that for our QP Newton's method terminates with minimizer x^* in one step since

$$\begin{aligned} x_1 &= x_0 - Q^{-1}(Qx_0 - b) \\ &= Q^{-1}b \\ &= x^* \end{aligned}$$

2.4. The Broyden Fletcher Goldfarb Shanno Method.

The BFGS method is a Quasi-Newton method which is based on approximating the Hessian by another matrix that is available for certain functions at a lower computational cost. The BFGS method uses a unit step size and is given by the iterative step

$$x_{k+1} = x_k - (D_k)^{-1} \nabla f(x_k)$$

where D_k is the secant approxiamtion of the Hessian as follows:
Let $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ thus we have

$$D_{k+1} = D_k + \frac{y_k^\top y_k}{y_k^\top s_k} - \frac{D_k s_k s_k^\top D_k}{s_k^\top D_k s_k}$$

The algorithm normally begins with D_0 as the identity however any matrix may be used.

3. The Eigenstep Method

The eigenstep method is defined using $s_k = -g_k$ and for k=0,1,...,n-1, $\alpha_k = 1/\lambda_k$ where each λ_k represents an eigenvalue of Q. This algorithm is motivated by the following lemmas.

Lemma 3.1. *If g_k is an eigenvector of Q with eigenvalue λ, and if $s_k = -g_k$ then $\alpha_k^* = 1/\lambda$.*

Proof. The proof follows from the definition of α_k^*, from $s_k = -g_k$ and from $Qg_k = \lambda g_k$. □

Lemma 3.2. *Suppose that $(x^* - x_k)$ is an eigenvector of Q with eigenvalue λ. If $s_k = -g_k$ and $\alpha_k = 1/\lambda$, then $x_{k+1} = x^*$, the global minimizer of $f(x)$.*

Proof. Since $\nabla f(x^*) = 0$ and $\nabla f(x) = Qx - b$, we have

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k s_k \\ &= x_k - \frac{1}{\lambda} g_k \\ &= x_k + \frac{1}{\lambda}(\nabla f(x^*) - g_k) \\ &= x_k + \frac{1}{\lambda}(Qx^* - b - (Qx_k - b)) \\ &= x_k + \frac{1}{\lambda} Q(x^* - x_k) \\ &= x_k + \frac{1}{\lambda}\lambda(x^* - x_k) \\ &= x^* \end{aligned}$$

□

A drawback of this method, a modification of the steepest descent method, is that we need to calculate the eigenvalues of Q but an advantage is that we guarantee termination in n steps or less. We reinforce what previous authors have shown, i.e., that the steepest descent algorithm can be improved by changes to the step size procedure.

In the following sections of this chapter, we prove finite termination for $n = 2$, we transform the problem to standard form, we prove finite termination for the standard form in n variables, and we then prove finite termination for our QP in n variables.

3.1. The Eigenstep Method for $n = 2$..

In this section we prove that for $n = 2$ the eigenstep method will terminate in n or fewer iterations. We include this specific proof as it differs from the general proof presented in the next section, and we feel it adds understanding.

Theorem 3.3. *If* $n = 2$, $s_k = -g_k$, *and* $\alpha_k = 1/\lambda_k$, *Then* $x_2 = x^*$.

Proof. Let x_0 be arbitrary but fixed. Since the eigenvectors of Q form a basis for $\mathbb{R}^2$, we can write

$$x_0 = \beta_0 v_0 + \beta_1 v_1$$

for some real scalars β_0 and β_1, and eigenvectors v_0 and v_1 corresponding to λ_0 and λ_1, respectively. For convenience, let $w_0 = \beta_0 v_0$ and $w_1 = \beta_1 v_1$, and recall that w_0 and w_1 are also eigenvectors of Q. For the first iteration, the eigenstep method gives

$$\begin{aligned} x_1 &= x_0 - \frac{1}{\lambda_0} g_0 \\ &= w_0 + w_1 - \frac{1}{\lambda_0}(Qw_0 + Qw_1 - b) \\ &= w_0 + w_1 - \frac{1}{\lambda_0}(\lambda_0 w_0 + \lambda_1 w_1 - b) \\ &= w_1 - \frac{\lambda_1}{\lambda_0} w_1 + \frac{1}{\lambda_0} b \end{aligned}$$

In the second iteration, we have

$$\begin{aligned} x_2 &= x_1 - \frac{1}{\lambda_1} g_1 \\ &= x_1 - \frac{1}{\lambda_1}(Qx_1 - b) \\ &= (w_1 - \frac{\lambda_1}{\lambda_0} w_1 + \frac{1}{\lambda_0} b) - \frac{1}{\lambda_1}(Q(w_1 - \frac{\lambda_1}{\lambda_0} w_1 + \frac{1}{\lambda_0} b) - b) \\ &= w_1 - \frac{\lambda_1}{\lambda_0} w_1 + \frac{1}{\lambda_0} b - \frac{1}{\lambda_1} Qw_1 + \frac{1}{\lambda_0} Qw_1 - \frac{1}{\lambda_0 \lambda_1} Qb + \frac{1}{\lambda_1} b \\ &= w_1 - \frac{\lambda_1}{\lambda_0} w_1 + \frac{1}{\lambda_0} b - \frac{\lambda_1}{\lambda_1} w_1 + \frac{\lambda_1}{\lambda_0} w_1 - \frac{1}{\lambda_0 \lambda_1} Qb + \frac{1}{\lambda_1} b \\ &= \frac{1}{\lambda_0} b - \frac{1}{\lambda_0 \lambda_1} Qb + \frac{1}{\lambda_1} b \end{aligned}$$

Thus,

$$\begin{aligned} g_2 &= Q(\frac{1}{\lambda_0} b - \frac{1}{\lambda_0 \lambda_1} Qb + \frac{1}{\lambda_1} b) - b \\ &= \frac{1}{\lambda_0} Qb - \frac{1}{\lambda_0 \lambda_1} QQb + \frac{1}{\lambda_1} Qb - b. \end{aligned}$$

As we did for x_0, we write $b = u_0 + u_1$, where $Qu_0 = \lambda_0 u_0$ and $Qu_1 = \lambda_1 u_1$. Thus, $Qb = \lambda_0 u_0 + \lambda_1 u_1$. From the previous equation it follows

$$\begin{aligned} g_2 &= \frac{1}{\lambda_0}(\lambda_0 u_0 + \lambda_1 u_1) - \frac{1}{\lambda_0 \lambda_1} Q(\lambda_0 u_0 + \lambda_1 u_1) + \frac{1}{\lambda_1}(\lambda_0 u_0 + \lambda_1 u_1) - b \\ &= u_0 + \frac{\lambda_1}{\lambda_0} u_1 - \frac{1}{\lambda_0 \lambda_1}(\lambda_0^2 u_0 + \lambda_1^2 u_1) + \frac{\lambda_0}{\lambda_1} u_0 + u_1 - b \\ &= u_0 + \frac{\lambda_1}{\lambda_0} u_1 - \frac{\lambda_0}{\lambda_1} u_0 - \frac{\lambda_1}{\lambda_0} u_1 + \frac{\lambda_0}{\lambda_1} u_0 + u_1 - b \\ &= 0. \end{aligned}$$

□

If it happens that $\lambda_1 = \lambda_0$, then

$$\begin{aligned} g_1 &= Q(w_1 - \frac{\lambda_1}{\lambda_0} w_1 + \frac{1}{\lambda_0} b) - b \\ &= Q(w_1 - w_1 + \frac{1}{\lambda_0} b) - b \\ &= \frac{Qb}{\lambda_0} - b \\ &= \frac{\lambda_0 u_0 + \lambda_0 u_1}{\lambda_0} - b \\ &= 0. \end{aligned}$$

3.2. **Transformation to Standard Form.**

Since Q positive definite and symmetric, there exists an orthogonal matrix P such that

$$D = P^\top Q P = diag\{\lambda_0, \ldots, \lambda_{n-1}\}$$

If we let $x = Py$ we can rewrite

$$f(x) = f(Py) = \frac{1}{2} y^\top P^\top Q P y - b^\top P y.$$

Letting $\hat{f}(y) = f(Py)$ and $d = P^\top b$, we have transformed the objective function to

$$\hat{f}(y) = \frac{1}{2} y^\top D y - d^\top y.$$

We now set $y = z + D^{-1}d$ since $\nabla \hat{f}(D^{-1}d) = 0$ to obtain

$$\hat{f}(z+D^{-1}d) = \frac{1}{2}(z+D^{-1}d)^{\top}D(z+D^{-1}d) - d^{\top}(z+D^{-1}d) = \frac{1}{2}z^{\top}Dz - \frac{1}{2}d^{\top}D^{-1}d.$$

Letting $h(z) = \hat{f}(z + D^{-1}d) = \hat{f}(z + D^{-1}P^{\top}b)$ and ignoring the constant $\frac{1}{2}d^{\top}D^{-1}d$ (it doesn't affect the value of the minimizer z^*) we have transformed the objective function to standard form

$$h(z) = \frac{1}{2}z^{\top}Dz.$$

The next lemma shows a one to one correspondence between the minimizers of our QP and our QP in standard form.

Lemma 3.4. *z^* is the unique global minimizer for $h(z)$, if and only if $x^* = Py^* = P(z^* + D^{-1}d) = P(z^* + D^{-1}P^{\top}b)$ is the unique global minimizer for $f(x)$.*

Proof. We know $D = P^{\top}QP = diag\{\lambda_0, \ldots, \lambda_{n-1}\}$ is nonsingular therefore $z^* = 0$. Since, $x^* = Py^* = PD^{-1}d$, we have

$$\begin{aligned}
\nabla f(x^*) &= Qx^* - b \\
&= Q(Py^*) - b \\
&= QPD^{-1}d - b \\
&= QPD^{-1}P^{T}b - b \\
&= QQ^{-1}b - b \\
&= 0.
\end{aligned}$$

Conversely, assume x^* is the unique global minimizer for $f(x)$, thus $x^* = Q^{-1}b$ and $z^* = P^{-1}(x^* - Q^{-1}b) = 0$. □

Theorem 3.5. *The eigenstep method will solve a QP in standard form in at most n iterations.*

Proof. The QP to be solved is $\min h(z) = \frac{1}{2}z^{\mathsf{T}}Dz$. Let z_0 be arbitrary but fixed. The eigenstep method gives

$$\begin{aligned} z_1 &= z_0 - \frac{1}{\lambda_0}Dz_0 = (I - \frac{1}{\lambda_0}D)z_0 \text{ and} \\ z_2 &= z_1 - \frac{1}{\lambda_1}Dz_1 = (I - \frac{1}{\lambda_1}D)(I - \frac{1}{\lambda_0}D)z_0. \end{aligned}$$

Continuing, we see that

$$z_n = \prod_{i=0}^{n-1}(I - \frac{1}{\lambda_i}D)z_0$$

The $i-th$ row of $(I-\frac{1}{\lambda_i}D)$ is a zero row since the diagonal element is $1-(\frac{1}{\lambda_i})\lambda_i$. Thus, the product is the zero matrix and it follows that $z_n = 0$, the global minimizer. □

3.3. **Proof of Termination.**

Theorem 3.6. *The eigenstep method will solve QP in at most n iterations.*

Proof. We will show a one to one correspondence between the iterates of the eigenstep method applied to our QP and the iterates of the eigenstep method applied to the QP in standard form. The result will then follow from Theorem 3.5.

We proceed by induction. Let x_0 be arbitrary but fixed, and set $z_0 = P^{\mathsf{T}}x_0 - D^{-1}P^{\mathsf{T}}b$. Assume that for $i = 1, \ldots, k$, $x_i = P(z_i + D^{-1}P^{\mathsf{T}}b) = Pz_i + Q^{-1}b$.

We have

$$\begin{aligned} x_{k+1} &= x_k - \frac{1}{\lambda_k}(Qx_k - b) \\ &= Pz_k + Q^{-1}b - \frac{1}{\lambda_k}(Q(Pz_k + Q^{-1}b) - b) \\ &= Pz_k + Q^{-1}b - \frac{1}{\lambda_k}(QPz_k + b - b) \\ &= Pz_k + Q^{-1}b - \frac{1}{\lambda_k}(QPz_k) \\ &= Pz_k - \frac{1}{\lambda_k}PDP^{\mathsf{T}}Pz_k + Q^{-1}b \\ &= Pz_k - \frac{1}{\lambda_k}PDz_k + Q^{-1}b \\ &= P(z_k - \frac{1}{\lambda_k}Dz_k) + Q^{-1}b \\ &= Pz_{k+1} + Q^{-1}b \end{aligned}$$

□

4. Examples and Numerical Experiments

We begin the chapter with two examples to demonstrate the eigenstep method, and we follow up with the results of limited numerical testing.

Example 4.1. Consider the QP

$$f(x) = \frac{1}{2}x^{\mathsf{T}} \begin{bmatrix} 21 & 28 & 28 & 17 \\ 28 & 38 & 38 & 21 \\ 28 & 38 & 40 & 22 \\ 17 & 21 & 22 & 19 \end{bmatrix} x - \begin{bmatrix} 10 & -212 & 303 & -417 \end{bmatrix} x.$$

The solution to the QP is

$$x^* = Q^{-1}b = \begin{bmatrix} 9484.125 & -7136.625 & 1347.6875 & -2180.375 \end{bmatrix}^{\mathsf{T}}$$

giving $f(x^*) = -1462685.7187$. The eigenvalues of Q are

$$\lambda_0 = 110.7270,\ \lambda_1 = 5.9953,\ \lambda_2 = 1.2585,\ \text{and}\ \lambda_3 = 0.0191.$$

We begin the algorithm with

$$x_0 = \begin{bmatrix} -50 & -15 & 78 & 23 \end{bmatrix}^T.$$

Iteration 0

$$\begin{aligned} s_0 &= -\begin{bmatrix} 1095 & 1689 & 1353 & 1405 \end{bmatrix}^T, \\ \alpha_0 &= 1/\lambda_0 = 1/110.7270 \\ x_1 &= \begin{bmatrix} -59.9 & -30.3 & 65.8 & 10.3 \end{bmatrix}^T, \\ f(x_1) &= -16169.9, \\ g_1 &= \begin{bmatrix} -97.6 & 101.7 & -271.4 & 406.64 \end{bmatrix}^T, \\ \| g_1 \| &= 508.8 \end{aligned}$$

Iteration 1:

$$\begin{aligned} s_1 &= -\begin{bmatrix} -97.6 & 101.7 & -271.4 & 406.64 \end{bmatrix}^T, \\ \alpha_1 &= 1/\lambda_1 = 1/5.9953 \\ x_2 &= \begin{bmatrix} -43.6 & -47.2 & 111.1 & -57.5 \end{bmatrix}^T, \\ f(x_2) &= -44851.9, \\ g_2 &= \begin{bmatrix} -115.7 & 209.5 & -140.9 & 34.8 \end{bmatrix}^T, \\ \| g_2 \| &= 279.9 \end{aligned}$$

Iteration 2:

$$\begin{aligned} s_2 &= -\left[\, -115.7 \quad 209.5 \quad -140.9 \quad 34.8 \,\right]^{\mathsf{T}}, \\ \alpha_2 &= 1/\lambda_2 = 1/1.2585 \\ x_3 &= \left[\, 48.3 \quad -213.7 \quad 223.0 \quad -85.2 \,\right]^{\mathsf{T}}, \\ f(x_3) &= -97075.59 \\ g_3 &= \left[\, -180.7 \quad 132.6 \quad -21.5 \quad 40.1 \,\right]^{\mathsf{T}}, \\ \| g_3 \| &= 228.7 \end{aligned}$$

Iteration 3:

$$\begin{aligned} s_3 &= -\left[\, -180.7 \quad 132.6 \quad -21.5 \quad 40.1 \,\right]^{\mathsf{T}}, \\ \alpha_3 &= 1/\lambda_3 = 1/0.0191 \\ x_4 &= \left[\, -9484.1 \quad -7136.6 \quad 1347.7 \quad -2180.4 \,\right]^{\mathsf{T}}, \\ f(x_4) &= -1462685.7187, \\ g_4 &= \left[\, 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \,\right]^{\mathsf{T}}, \\ \| g_4 \| &= 0.0000 \end{aligned}$$

Also for this example the steepest descent method terminated after 14695 iterations with a CPU time of 84.98 seconds compared to a CPU time of 0.22 seconds using the eigenstep method.

Example 4.2. In this example we compare the performance of the steepest descent (S), conjugate gradient (C) and eigenstep(E) methods in the solution of

$$\min f(x) = \frac{1}{2} x^{\mathsf{T}} \begin{bmatrix} 2 & 1 \\ 1 & 10 \end{bmatrix} x - \left[\, -4 \quad -6 \,\right] x$$

for which

$$x^* = Q^{-1} b = \left[\, -1.78947 \quad -0.42105 \,\right]^{\mathsf{T}}.$$

We begin all of the alogrithms with

$$x_0 = \begin{bmatrix} -30 & 2 \end{bmatrix}^\top.$$

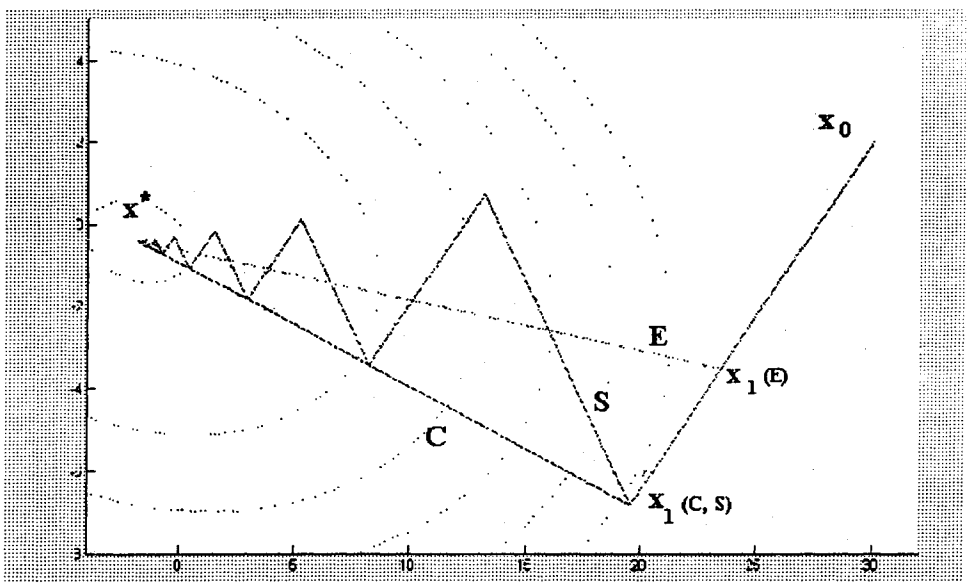
The two iterates for the eigenstep method are

$$x_1 = \begin{bmatrix} 23.48026 & -3.53189 \end{bmatrix}^\top \text{ and } x_2 = \begin{bmatrix} -1.78947 & -0.42105 \end{bmatrix}^\top.$$

The two iterates for the conjugate gradient method are

$$x_1 = \begin{bmatrix} 19.58217 & -6.83937 \end{bmatrix}^\top \text{ and } x_2 = \begin{bmatrix} -1.78947 & -0.42105 \end{bmatrix}^\top.$$

The steepest descent method terminated after 31 iterations with $\| g_{30} \| < 10^{-6}$.



In the graph above note that all 3 methods have the same search direction after the first iteration. The steepest descent method and conjugate gradient method also have the same step size, α^* and arrive at the same point after the first iteration, however the eigenstep method, initially has a shorter step. Next note the different directions generated by all methods. The eigenstep

method and the conjugate gradient method terminate with x^* after two iterations , however the steepest desecent method produces a zig zag route to x^* terminating after 31 steps.

Numerical experiments were conducted comparing CPU time required to solve the QP for the eigenstep method (E), the steepest descent method, and the conjugate gradient method (C). We also included CPU time required to solve the system of equations, $Qx = b$. Matlab was used to conduct all experiments. During all experiments Matlab was the only program running on our pentium 4 computer. We used Matlab's random symmetric positive definite matrix generator and the random vector generator to generate all required dense matrices and all vectors for each of our experiments. CPU time was recorded by Matlab, and for the eigenstep method CPU time included the time to calculate the required eigenvalues. The results are given below in Tables 1 and 2.

Table 1: CPU Time (seconds)			
n	E	C	Qx = b
50	0.492	0.451	0.000
100	1.682	1.192	0.010
250	8.098	7.361	0.040
500	34.724	31.896	0.221
750	107.250	91.361	0.611
1000	261.956	193.108	1.241

Table 2: The Steepest Descent Method		
n	CPU Time (seconds)	Iterations
20	115.642	11010
50	255.387	32527
100	5051.674	542641

The number of variables in the first columns of both tables is represented by n. In Table 1 for each value of n five experiments were conducted and the

average CPU time in seconds is given to compare the eigenstep method and the conjugate gradient method. Also, we included CPU time to solve $Qx = b$ in column three of Table 1 since this is equivalent to solving the QP. In Table 2 one experiment for each value of n was conducted. Table 1 shows that the conjugate gradient method out performs the eigenstep method however both methods have CPU times that are approximately of the same order of magnitude. This result is what one might expect since the eigenstep method requires the calculation of eigenvalues, an expensive step. Table 2 results reinforce the fact that the steepest descent method is hopelessly inefficient.

5. Conclusion

In this thesis we presented theorem 3.6 which validates a new iterative method for unconstrained convex quadratic optimization which we named the eigenstep method. We compared the eigenstep method with two other well known first order methods, the conjugate gradient method and the steepest descent method. The conjugate gradient method and the eigenstep method both terminate in at most n iterations, however our numerical experiments show that the conjugate gradient method has a better performance than the eigenstep method, and furthermore we observe that both of these methods are superior in performance to the steepest descent method. Future work will include modifying the eigenstep method to deal with more general functions. For example, for a function where the Hessian is not constant one might consider using the reciprocal of the largest eigenvalue as a step size to move efficiently to the next point. In fact, modifications of the conjugate gradient method along with step size formulas by *Fletcher – Reeves* and *Polak – Ribiére* [6,p.399] are used today to deal with more general problems since these methods have low storage requirements. We feel with the evidence presented in this thesis our new method warrants further investigation.

Bibliography

(1) Raydan, M., and Svaiter, B. F.,"Relaxed steepest descent and Cauchy-Barzilai-Borwein method", International Journal of Computational Optimization and Applications, Vol.21, No.2, pp155-167, February 2002.

(2) Barzilai,J., and Borwein,J.M.,"Two point step size gradient method", IMA Journal of Numerical Analysis, Vol.8, No.1, pp.141-148, 1988.

(3) Dai,Y.H.,and Liao,L.Z.,"R-Linear convergence of the Barzilai and Borwein gradient method", IMA Journal of Numerical Analysis, Vol.22 , No.1,pp.1-10, January 2002.

(4) Friedlander,A., Martinez,J.M., Molina,B.,Raydan,M., "Gradient Method With Retards and Generalizations", SIAM Journal on Numerical Analysis, Volume 36, number 1, pp.275-289,1999.

(5) Raydan,M., "On the Barzilai and Borwein choice of steplength for the gradient method", IMA Journal of Numerical Analysis, Vol.13, No.1,pp. 321-326, 1993.

(6) Nash,S., Sofer,A.," Linear and Nonlinear Programming", McGraw-Hill publishers, 1996.

(7) Johnson,L., Riess,R., Arnold,J.," Introduction to Linear Algebra", Addison-Wesley publishers, third edition, 1993.

Vita Auctoris

NAME:	John Peter Battaglia.
DATE OF BIRTH:	April 29, 1953.
PLACE OF BIRTH:	Windsor Ontario, Canada.
EDUCATION AND WORK HISTORY:	Secondary School Graduate (grade 13) from F.J. Brennan High School, 1972. Bachelor of Arts, Mathematics University of Windsor, 1974. Bachelor of Education University of Windsor, 1975. High school mathematics teacher in Windsor from 1975 to 1992. Appointed as head of the mathematics department at Catholic Central High School in Windsor from 1992-2003. Bachelor of Arts, Honours Mathematics, University of Windsor, 1991. Master of Science, Mathematics University of Windsor, 2004.