RICE UNIVERSITY

Convergence Properties of the Barzilai and Borwein Gradient Method

by

Marcos Raydan M.

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Richard A. Tapia

Richard A. Tapia, Co-Chairman
Professor of Mathematical Sciences

J E Dennis Jr

John E. Dennis, Jr., Co-Chairman
Noah Harding Professor of Mathematical Sciences

John C. Polking
Professor of Mathematics

W W Symes

William W. Symes
Professor of Mathematical Sciences

Houston, Texas

April, 1991

Convergence Properties of the Barzilai and Borwein Gradient Method

Marcos Raydan M.

Abstract

In a recent paper, Barzilai and Borwein presented a new choice of steplength for the gradient method. Their choice does not guarantee descent in the objective function and greatly speeds up the convergence of the method. We derive an interesting relationship between any gradient method and the shifted power method. This relationship allows us to establish the convergence of the Barzilai and Borwein method when applied to the problem of minimizing any strictly convex quadratic function (Barzilai and Borwein considered only 2-dimensional problems). Our point of view also allows us to explain the remarkable improvement obtained by using this new choice of steplength.

For the two eigenvalues case we present some very interesting convergence rate results. We show that our Q and R-rate of convergence analysis is sharp and we compare it with the Barzilai and Borwein analysis.

We derive the preconditioned Barzilai and Borwein method and present preliminary numerical results indicating that it is an effective method, as compared to the preconditioned Conjugate Gradient method, for the numerical solution of some special symmetric positive definite linear systems that arise in the numerical solution of Partial Differential Equations.

Acknowledgments

I would like to thank the members of my committee: Professors John Dennis, Richard Tapia, John Polking and William Symes, for their help and interest during the course of this work. I am especially grateful to Professor Richard Tapia for his valuable comments and suggestions and also for extending his warmth and friendship.

I wish to thank Professor Richard Byrd at University of Colorado for his careful reading of several drafts of this thesis which led to considerable improvements. I also wish to thank Professors Mario Martinez at Universidade Estadual de Campinas and Yin Zhang at University of Maryland, for providing references and helpful discussions.

I would like to thank Debora Cores, Amr El-Bakry, Cristina Maciel, Susan Minkoff, Michael Pearlman, Cathy Samuelsen and Virginia Torczon who each in their own way provided help during the course of this research. I am also grateful to Universidad Central de Venezuela for financial support throughout my graduate career.

Finally, I wish to dedicate this work to my parents Arlette and Marcos, my wife Debora, and my entire family, as some small acknowledgement of their unfailing love and affection.

Contents

Abstract ii

Acknowledgments iii

List of Tables vi

List of Illustrations vii

1 Introduction 1

2 The Barzilai and Borwein Method 4

3 Relationship to the Shifted Power Method 9

4 Convergence Analysis for the Quadratic Case 14

5 Convergence Rate for the 2 Eigenvalues Case 19

5.1 The Q-rate of Convergence 20

5.2 The R-rate of Convergence 27

6 Numerical Investigation of the R-rate of Convergence in the General Case 31

6.1 Description of Experiments and Numerical Results 31

7 Preconditioned Barzilai and Borwein and Numerical Results for a PDE Application 38

7.1 Preconditioned Barzilai and Borwein 38

7.2 Preconditioned Conjugate Gradient 40

7.3 The SSOR Preconditioning Matrix 41

7.4 Model Problem 43

7.5 Description of Experiments and Numerical Results 44

8 Concluding Remarks **50**

Bibliography **52**

Tables

3.1 Barzilai and Borwein method for Example 3.1 13

5.1 $||e_k||_2$ for 4 randomly generated problems 27

5.2 R-rate of convergence behavior of Algorithm 2.2 with two different sets of initial data 30

Illustrations

2.1 Number of iterations required for the classical gradient method and the Barzilai and Borwein method to achieve $||e_k||_2 \leq 10^{-14}$ when they are applied to the function f in Example 2.1 8

6.1 Effect of reducing $\kappa(A)$ on the R-rate of convergence. 34

6.2 Effect of reducing $\kappa(A)$ on the number of iterations required to achieve $||e_k||_2 \leq 10^{-14}$. 35

6.3 Effect of clustering the eigenvalues of A on the R-rate of convergence. 36

6.4 Effect of clustering the eigenvalues of A on the number of iterations required to achieve $||e_k||_2 \leq 10^{-14}$. 37

7.1 Number of iterations required for the preconditioned Barzilai and Borwein and the preconditioned Conjugate Gradient to achieve $||g_k||_2 \leq 10^{-8}$ when $\alpha = 0$. 47

7.2 Number of iterations required for the preconditioned Barzilai and Borwein and the preconditioned CG to achieve $||g_k||_2 \leq 10^{-8}$ for different values of α and $n = 10^6$. 48

7.3 Computational work (Number of multiplications) required for the preconditioned Barzilai and Borwein and the preconditioned CG for different values of α and $n = 10^6$. 49

Chapter 1

Introduction

In this dissertation, we study the convergence properties of the Barzilai and Borwein gradient method for the smooth unconstrained minimization problem:

$$\min_{x \epsilon \mathbb{R}^n} f(x)$$
$$\text{where } f : \mathbb{R}^n \rightarrow \mathbb{R} .$$

It is well-known that the classical gradient method, also referred to as the steepest descent method, performs poorly when applied to the unconstrained minimization problem, i.e., it converges slowly and is seriously affected by ill-conditioning. In 1988, Barzilai and Borwein [3] presented a new choice of steplength for the gradient method. Their choice of steplength requires less computational work and greatly speeds up the convergence of the gradient method. More interesting, from a theoretical point of view, is that the new method does not guarantee descent in the objective function. Thus, convergence analysis cannot be based on the classical contraction mapping theory. Barzilai and Borwein [3] establish R-superlinear convergence only for the 2-dimensional quadratic case. It is unlikely that their analysis can be extended to higher dimensional problems.

In the present work, we establish the global convergence of the Barzilai and Borwein gradient method for any strictly convex quadratic function. We also present interesting rate of convergence results for some special problems.

The minimization of strictly convex quadratic functions is equivalent to the solution of Symmetric Positive Definite (SPD) linear systems. From this point of view, we study the applicability of the Barzilai and Borwein gradient method on the iterative solution of the large sparse linear systems that arise from the numerical solution of Partial Differential Equations (PDE).

This document is organized as follows :

In Chapter 2, we present the Barzilai and Borwein gradient method. We motivate the new choice of steplength and compare it with the choice of the classical gradient method.

In Chapter 3, we derive a relationship between any gradient method for minimizing a quadratic function and the shifted power method for approximating eigenvectors and eigenvalues. We believe that this connection is the key to understanding the convergence properties of the Barzilai and Borwein method.

In Chapter 4, we establish the convergence of the Barzilai and Borwein method applied to a quadratic function with a SPD Hessian. In particular, we explain why the objective function f increases at some iterations.

In Chapter 5, we show that the convergence is unusual and at least (3,2)-step Q-quadratic when the Hessian has exactly two distinct and positive eigenvalues. This means that in 3 steps of the algorithm we obtain a quadratic decrease in the error and in the next 2 steps we also obtain a quadratic decrease in the error. This convergence property implies an R-convergence rate of at least $\sqrt[5]{4}$. We compare this results with the results obtained by Barzilai and Borwein [3] for the 2-dimensional quadratic case, and we show that our Q-rate and R-rate of convergence analysis is sharp. For a further discussion of Q-rate and R-rate of convergence, see Ortega and Rheinboldt [11] or Potra [12].

In Chapter 6, we present a numerical investigation of the R-rate of convergence when the Hessian has more than two distinct eigenvalues. We conclude, based on our numerical results, that the Barzilai and Borwein method has a much better performance when the condition number of the Hessian matrix is not large or when the eigenvalues are clustered.

In Chapter 7, we introduce the preconditioned Barzilai and Borwein method and compare it with the preconditioned Conjugate Gradient method when they are both applied to the iterative solution of large sparse linear systems of equations that arise in the numerical solution of elliptic problems.

Finally, in Chapter 8 we summarize our results and discuss issues for further research.

Chapter 2

The Barzilai and Borwein Method

In order to solve the unconstrained minimization problem, we consider the nonlinear equations problem :

$$\text{find } x_\star \in I\!R^n \text{ such that } \nabla f(x_\star) = 0, \tag{2.1}$$

where $f : I\!R^n \longrightarrow I\!R$. The numerical solution of (2.1) is usually iterative, moving at each iteration from an estimate x_c of $x_\star$ to a better estimate x_+. In many algorithms, each iteration involves the calculation of a Quasi-Newton step, $s_{QN} = -A_c^{-1}\nabla f(x_c)$, where $A_c \in I\!R^{nXn}$ is an approximation of the Hessian of f at x_c. After each iteration the current A_c is updated to A_+, an approximation of the Hessian of f at x_+. The approximation usually is chosen to satisfy the secant equation,

$$A_+ s_c = y_c, \tag{2.2}$$

where $s_c = x_+ - x_c$ and $y_c = \nabla f(x_+) - \nabla f(x_c)$. For a further discussion of Quasi-Newton methods, see Dennis and Schnabel [5].

In the one dimensional case the secant equation completely determines A_+; however if $n > 1$, then many matrices will satisfy the secant equation. So, in addition to obeying (2.2), the update A_+ must be further restricted to a set of matrices that have desirable properties. Barzilai and Borwein in [3] considered a related but somewhat different approach. They observed that the scalar $\alpha_+ \epsilon I\!R$ that uniquely solves the overdetermined linear system $y_c = \alpha_+ s_c$ in the least squares sense is given by

$$\alpha_+ = \frac{s_c^t y_c}{s_c^t s_c} \tag{2.3}$$

if $s_c \neq 0$. Hence by restricting the update matrix in the quasi-Newton method to the class of scalar multiples of the identity and then asking that the secant equation be satisfied in the least squares sense they devised the following algorithm

Algorithm 2.1 (Barzilai and Borwein Method)

```
Given $x_0 \epsilon \mathbb{R}^n$,$\alpha_0 \epsilon \mathbb{R}$

For k=0,1,...,(until convergence) do

  1. Set $s_k = -\frac{1}{\alpha_k}\nabla f(x_k)$
  2. Set $x_{k+1} = x_k + s_k$
  3. Set $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
  4. Set $\alpha_{k+1} = \frac{s_k^t y_k}{s_k^t s_k}$

End do
```

Notice that every iteration of the Barzilai and Borwein method requires two inner products, one scalar-vector multiplication, two vector additions and a gradient evaluation.

If we consider problem (2.1) when $f(x) = \frac{1}{2}x^tAx - b^tx + c$ is a quadratic function and A is a symmetric positive definite (SPD) matrix, then α_+ in (2.3) becomes

$$\alpha_+ = \frac{s_c^t A s_c}{s_c^t s_c} \tag{2.4}$$

and Algorithm 2.1 becomes

Algorithm 2.2 (Barzilai and Borwein Method for Quadratics)

```
Given $x_0 \epsilon \mathbb{R}^n$,$\alpha_0 \epsilon \mathbb{R}$

For k=0,1,...,(until convergence) do

  1. Set $s_k = -\frac{1}{\alpha_k}\nabla f(x_k)$
  2. Set $x_{k+1} = x_k + s_k$
  3. Set $\alpha_{k+1} = \frac{s_k^t A s_k}{s_k^t s_k}$

End do
```

In the quadratic case, α_{k+1} turns out to be the Rayleigh quotient of A at the vector s_k. Since A is SPD,

$$0 < \lambda_{min} \leq \alpha_k \leq \lambda_{max} \quad for \;\; all \;\; k, \tag{2.5}$$

where λ_{min} and λ_{max} are respectively the smallest and largest eigenvalues of A. And so, in step 1 there is no danger of dividing by zero.

Barzilai and Borwein [3] also observed, by symmetry, that the scalar $\hat{\alpha}_+$ that uniquely solves the overdetermined linear system $\hat{\alpha}_+ y_c = s_c$ in the least squares sense is given by

$$\hat{\alpha}_+ = \frac{y_c^t y_c}{s_c^t y_c}.$$

In the quadratic case, $\hat{\alpha}_+$ becomes

$$\hat{\alpha}_+ = \frac{s_c^t A^2 s_c}{s_c^t A s_c},$$

which is the Rayleigh quotient of A at the vector $\sqrt{A} s_c$. Hence, $\hat{\alpha}_+$ also satisfies (2.5).

In the rest of this work, we will only consider the Barzilai and Borwein method with the choice of α_+ defined by (2.3) in the general case and by (2.4) in the quadratic case. The reason for this is that all results established for Algorithm 2.2 with the choice α_+ also hold with the choice $\hat{\alpha}_+$.

Notice that, in the Barzilai and Borwein gradient method, the search direction is always the negative gradient of f at x_c as in the gradient method, but the choice of steplength is not the classical choice. In fact, Algorithm 2.2 would be the classical gradient method for quadratics if we changed (2.4) to

$$\alpha_{k+1} = \frac{g_{k+1}^t A g_{k+1}}{g_{k+1}^t g_{k+1}}, \tag{2.6}$$

where $g_{k+1} = \nabla f(x_{k+1})$.

Despite the similarities between these two methods, Algorithm 2.2 is significantly faster than the classical gradient method at the same cost per iteration. We now present an example to illustrate this difference.

Example 2.1

Let $f(x) = \frac{1}{2}x^t Ax$ where $A = diag(1, 2, ..., n)$. Clearly, f has a unique minimizer at $x_* = (0, 0, ..., 0)^t$. Figure 2.1 shows the number of iterations required by both algorithms for different values of n, to achieve $\|x_k - x_*\| \leq 10^{-14}$. In both cases the starting point was $x_0 = (.5, .5, ..., .5)^t$ and for Algorithm 2.2 $\alpha_0 = 1.5$. One may observe the remarkable difference between these two methods.

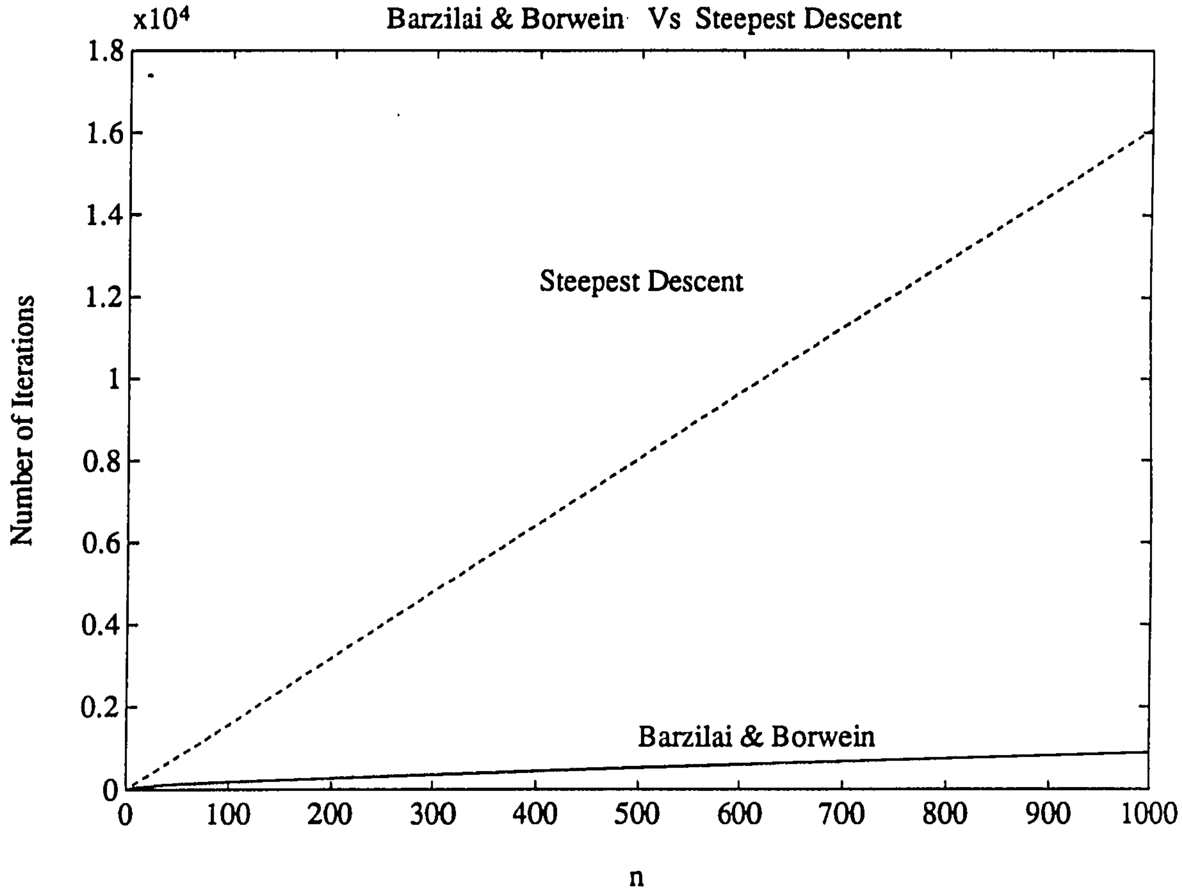

Figure 2.1 Number of iterations required for the classical gradient method and the Barzilai and Borwein method to achieve $\|e_k\|_2 \leq 10^{-14}$ when they are applied to the function f in Example 2.1

Chapter 3

Relationship to the Shifted Power Method

In this chapter we present an interesting relationship between any gradient method and the shifted power method to approximate eigenvectors and eigenvalues.

Let us consider the gradient method for problem (2.1) when f is a differentiable function. For the purpose of comparison we will write the steplength choice in a slightly different way.

Algorithm 3.1 (Gradient Method)

```
Given $x_0 \in I\!R^n$
For k=0,1,...,(until convergence) do
    1. Choose steplength $\frac{1}{\alpha_k}$
    2. Set $s_k = -\frac{1}{\alpha_k}\nabla f(x_k)$
    3. Set $x_{k+1} = x_k + s_k$
End do
```

Both the classical gradient method and the Barzilai and Borwein method for quadratics are special cases of Algorithm 3.1 . They differ only in the way the scalars α_k are chosen. Lemma 3.1 demonstrates a connection between Algorithm 3.1 and the shifted power method. This relationship will be extensively used to establish our global and local convergence results.

Lemma 3.1 Let $f(x) = \frac{1}{2}x^tAx - b^tx + c$ where A is a SPD matrix. Further let x_* be the unique minimizer of f, $\{x_k\}$ the sequence generated by Algorithm 3.1 and $e_k = x_* - x_k$ for all k. Then

1. $Ae_k = \alpha_k s_k$
2. $e_{k+1} = \frac{1}{\alpha_k}(\alpha_k I - A)e_k$
3. $s_{k+1} = \frac{1}{\alpha_{k+1}}(\alpha_k I - A)s_k$

Proof: Using the fact that $\nabla f(x_k) = Ax_k - b$ and the definition of the step s_k in Algorithm 3.1, the three claims in Lemma 3.1 follow directly . □

Since A is SPD, the scalars α_k satisfy (2.5) when they are generated by either (2.6) or by (2.4). And so, claim 1 in Lemma 3.1 allows us to conclude that $\|e_k\|$ tends to zero if and only if $\|s_k\|$ tends to zero. Thus, for the minimization of a quadratic function with a SPD Hessian it suffices to study the behavior of $\{s_k\}$.

For any s_0, there exist constants $c_1, c_2, ..., c_n$ such that:

$$s_0 = \sum_{i=1}^{n} c_i v_i, \tag{3.1}$$

where $\{v_1, v_2, ..., v_n\}$ are orthonormal eigenvectors of A associated with the eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_n\}$.

From claim 3 in Lemma 3.1 we can see that the generation of $\{s_k\}$ resembles a shifted power method iteration. In fact, for any integer k,

$$s_{k+1} = \frac{1}{\gamma_k} \sum_{i=1}^{n} (\prod_{j=0}^{k} (\alpha_j - \lambda_i)) c_i v_i \tag{3.2}$$

where

$$\gamma_k = \prod_{j=1}^{k+1} \alpha_j.$$

From (3.2) we can see that if we use the exact eigenvalues of A as the scalars α_k in Algorithm 3.1, in any order, then we find the exact solution in p iterations, where p is the number of distinct eigenvalues of A. At each iteration we eliminate at least one coefficient from the eigenvector expansion (3.1). Unfortunately, we do not know the eigenvalues of A in advance. However, we can use the Rayleigh quotient at s_k (the choice of steplength for the Algorithm 2.2) to approximate the eigenvalues associated with the large coefficients in the eigenvector expansion (3.2) of s_k, and obtain a significant reduction in the norm of s_{k+1}. In fact, let us suppose that for some integer q,

$$s_q = \hat{c}_1 v_1 + \hat{c}_2 v_2 + \sum_{i=3}^{n} \epsilon_i v_i,$$

where

$$|\epsilon_i| \ll |\hat{c}_1| < |\hat{c}_2|.$$

Then α_{q+1} in Algorithm 2.2 can be viewed as an estimate of λ_2. If $|\lambda_2 - \alpha_{q+1}| \ll |\lambda_1 - \alpha_{q+1}|$, then the coefficient of v_2 in s_{q+1} will be greatly reduced relative to that of v_1. After a few iterations the scalar α_k might move towards λ_1.

So, by properties of the Rayleigh quotient, the scalars α_k generated by Algorithm 2.2 are approaching the eigenvalues associated with the largest coefficient of the eigenvector expansion of s_k, and this explains the remarkable behavior of the Barzilai and Borwein method when compared to the classical gradient method. Notice that, the choice of steplength for the clasical gradient method is given by (2.6) which is also a Rayleigh quotient but is not a good approximation to any of the eigenvalues of A.

The behavior of the sequence $\{\alpha_k\}$ for the Barzilai and Borwein method, in particular the approximation of the eigenvalues of the Hessian, is most easily appreciated by considering a particular example.

Example 3.1

Let $f(x) = \frac{1}{2}x^t Ax$ where $A = diag(1, 2, 12)$. Clearly, f has a unique minimizer at $x_\star = (0,0,0)^t$. The first 10 iterations generated by Algorithm 2.2 starting at $x_0 = (1,1,1)^t$ and $\alpha_0 = 1$ are shown in Table 3.1. The Table lists the 2-norm of the error, and the 2-norm of the gradient. Also shown are the scalars α_k and the coefficients in the eigenvector expansion (3.2) associated with the three eigenvalues of A.

Since $\alpha_0 = \lambda_1 = 1$, the column with the coefficients of the eigenvector associated with λ_1 in Table 3.1 contains zero after the first iteration, and the scalar α_k approximates only the eigenvalues $\lambda_2 = 2$ and $\lambda_3 = 12$ during the rest of the process. Notice that α_k approximates, at each iteration, the eigenvalue with the larger coefficient in the previous iteration. In fact, the bigger the difference between the two coefficients the closer the scalar will be to the eigenvalue.

Notice also that this is not a descent algorithm. Under special circumstances, that will be studied in the next section, the 2-norm of the error e_k, as well as the objective function, increases at some iterations. This is in sharp contrast to the classical gradient method. In fact, if the classical gradient method is used to minimize the function $f(x)$ with the same initial guess x_0, then 165 iterations are required to achieve an error of $.3\text{x}10^{-29}$. Observe that the Barzilai and Borwein method achieved this accuracy in only 10 iterations.

iteration	$\|e_k\|_2$	$\|\nabla f(x_k)\|_2$	α_k	coefficients of eigenvectors associated with		
				$\lambda_1 = 1$	$\lambda_2 = 2$	$\lambda_3 = 12$
0	0.17d+01	0.12d+02	0.1000d+01	-.10d+01	-.20d+01	-.12d+02
1	0.11d+02	0.13d+03	0.1165d+02	0.00d+00	0.17d+00	0.11d+02
2	0.88d+00	0.42d+01	0.1199d+02	0.00d+00	0.14d+00	-.32d+00
3	0.69d+00	0.13d+01	0.1045d+02	0.00d+00	0.13d+00	0.71d-04
4	0.55d+00	0.11d+01	0.2000d+01	0.00d+00	0.56d+00	-.55d-04
5	0.45d-04	0.54d-03	0.2000d+01	0.00d+00	0.80d-06	0.27d-03
6	0.22d-03	0.27d-02	0.1199d+02	0.00d+00	0.65d-14	-.23d-03
7	0.16d-08	0.19d-07	0.1200d+02	0.00d+00	0.54d-14	0.16d-08
8	0.26d-13	0.53d-13	0.1200d+02	0.00d+00	0.45d-14	0.00d+00
9	0.22d-13	0.44d-13	0.2000d+01	0.00d+00	0.22d-13	0.00d+00
10	0.31d-29	0.63d-29	0.2000d+01	0.00d+00	-.32d-29	0.00d+00

Table 3.1 Barzilai and Borwein method for Example 3.1

Chapter 4

Convergence Analysis for the Quadratic Case

In this chapter we will establish the convergence of the Barzilai and Borwein method when applied to the minimization of a strictly convex quadratic function.

For any initial error e_0, there exist constants $d_1^0, d_2^0, ..., d_n^0$ such that:

$$e_0 = \sum_{i=1}^{n} d_i^0 v_i,$$

where $\{v_1, v_2, ..., v_n\}$ are orthonormal eigenvectors of A associated with the eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_n\}$.

From Lemma 3.1 we can see that the generation of $\{e_k\}$ resembles a shifted power method iteration. In fact, for any integer k,

$$e_{k+1} = \sum_{i=1}^{n} d_i^{k+1} v_i, \tag{4.1}$$

where

$$d_i^{k+1} = \prod_{j=0}^{k} (\frac{\alpha_j - \lambda_i}{\alpha_j}) d_i^0.$$

We observe that the convergence properties of the sequence $\{e_k\}$ will depend on the behavior of each one of the sequences $\{d_i^k\}$, $1 \le i \le n$. Later in this chapter, we will prove that each of these sequences converges to zero. First let us establish the Q-linear convergence of Algorithm 2.2 applied to a quadratic function with a SPD Hessian that satisfies the admittedly restrictive condition

$$\lambda_{max} < 2 * \lambda_{min}. \tag{4.2}$$

Lemma 4.1 Let $f(x) = \frac{1}{2}x^t A x - b^t x + c$ where A is SPD and satisfies (4.2). Let $\{x_k\}$ be the sequence generated by Algorithm 2.2 and x_* the unique minimizer of f. Then the sequence $\{x_k\}$ converges Q-linearly to x_* in the Euclidean norm with convergence factor $\hat{c} = (\lambda_{max} - \lambda_{min})/\lambda_{min}$.

Proof: Using Lemma 3.1 and (4.1) we obtain for any k,

$$e_{k+1} = \frac{1}{\alpha_k}(\alpha_k I - A)\sum_{i=1}^{n} d_i^k v_i = \sum_{i=1}^{n}(\frac{\alpha_k - \lambda_i}{\alpha_k})d_i^k v_i.$$

By the orthonormality of the eigenvectors we have

$$\|e_{k+1}\|_2^2 = \sum_{i=1}^{n}(d_i^k)^2(\frac{\alpha_k - \lambda_i}{\alpha_k})^2 \leq \max_i(\frac{\alpha_k - \lambda_i}{\alpha_k})^2\|e_k\|_2^2. \tag{4.3}$$

From (4.2), recalling that α_k obeys (2.5),

$$\max_i |\frac{\alpha_k - \lambda_i}{\alpha_k}| \leq \frac{\lambda_{max} - \lambda_{min}}{\lambda_{min}} < 1. \tag{4.4}$$

Combining (4.3) and (4.4) gives

$$\|e_{k+1}\|_2 \leq \hat{c}\|e_k\|_2 \quad where \quad \hat{c} = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{min}} < 1 \qquad \square$$

Now we can explain why the norm of the error might increase at some iterations when the spectrum of A does not satisfy (4.2). Let us first divide the spectrum of A into two subintervals

$$Left = [\lambda_{min}, \frac{\lambda_{max}}{2}] \quad and \quad Right = (\frac{\lambda_{max}}{2}, \lambda_{max}].$$

Clearly, if the spectrum of A obeys (4.2) then the Left interval is empty and Lemma 4.1 says that the error decreases at each iteration. If we force the scalars α_k to be in the Right interval, by a similar argument, $\{x_k\}$ converges Q-linearly to x_* for any $x_0 \in I\!R^n$. But Algorithm 2.2 moves the scalars α_k dynamically within the spectrum of A. If at the j^{th} iteration $\alpha_j \in Left$, then the coefficient associated with the eigenvalues λ_i to the right of α_j will be amplified by the factor $|\frac{\alpha_j - \lambda_i}{\alpha_j}| > 1$ (i.e., $|d_i^{j+1}| > |d_i^j|$), and this might lead to an increase of $\|e_{j+1}\|$ with respect to $\|e_j\|$. In general, the sequences $\{d_i^k\}$ defined in (4.1) will increase at some iterations. However, the sequence $\{d_1^k\}$ associated with the eigenvalue λ_{min} will decrease at every iteration.

Lemma 4.2 The sequence $\{d_1^k\}$ defined by (4.1) converges to zero Q-linearly with convergence factor $\hat{c} = 1 - (\lambda_{min}/\lambda_{max})$.

Proof: For any positive integer k,

$$d_1^{k+1} = (\frac{\alpha_k - \lambda_{min}}{\alpha_k})d_1^k.$$

Since α_k satisfies (2.5), we have

$$0 < \frac{\lambda_{min}}{\lambda_{max}} \leq \frac{\lambda_{min}}{\alpha_k} \leq 1.$$

And so,

$$|d_1^{k+1}| = (1 - \frac{\lambda_{min}}{\alpha_k})|d_1^k| \leq \hat{c}|d_1^k|,$$

where

$$\hat{c} = 1 - \frac{\lambda_{min}}{\lambda_{max}} < 1 \quad \square$$

In the proof of our convergence theorem, we will use the following result.

Lemma 4.3 Assume that $\{d_1^k\}, \{d_2^k\}, ..., \{d_l^k\}$ all converge to zero for a fixed integer l, $1 \leq l < n$. Also assume that $\{d_{l+1}^k\}$ does not converge to zero. Then, for any $\delta > 0$ there exists $\hat{k}$ sufficiently large such that the sequence $\{\alpha_k\}$ generated by Algorithm 2.2 satisfies:

$$(\lambda_{l+1} - \delta) \leq \alpha_k \leq \lambda_{max}, \quad for \ all \ k \geq \hat{k}.$$

Proof: The proof is built upon two properties of the Rayleigh quotient.

If at the j-th iteration the vector e_j could be written as a linear combination of the eigenvectors $\{v_{l+1}, ..., v_n\}$ (i.e., the coefficients $d_1^j, d_2^j, ..., d_l^j$ in (4.1) are all zero) then, by Lemma 3.1, the vector s_j could also be written as a linear combination of the same eigenvectors. And so, by a well-known property of the Rayleigh quotient (see Noble and Daniel [10]), we would have

$$\lambda_{l+1} \leq \alpha_{j+1} \leq \lambda_{max}.$$

On the other hand, by (4.1), Lemma 3.1 and the orthonormality of the eigenvectors $\{v_1, v_2, ..., v_n\}$, we can see that the Rayleigh quotient α_{k+1} can be written as

$$\alpha_{k+1} = \frac{\sum_{i=1}^n (d_i^k)^2 \lambda_i^3}{\sum_{i=1}^n (d_i^k)^2 \lambda_i^2} \, . \tag{4.5}$$

Since $\{d_{l+1}^k\}$ does not converge to zero, $d_{l+1}^k \neq 0$ for all k, for if $d_{l+1}^{k_1} = 0$ for some integer k_1 then,

$$d_{l+1}^k = \prod_{j=k_1+1}^{k} (\frac{\alpha_j - \lambda_{l+1}}{\alpha_j}) d_{l+1}^{k_1} = 0 \quad for \quad all \quad k > k_1 \ .$$

Hence, at least one of the coefficients d_i^k, $l+1 \leq i \leq n$, is different from zero. We conclude that there is no danger of dividing by zero in (4.5), and so, α_{k+1} is a well defined and a continuous function of the variables $(d_1^k, d_2^k, ..., d_n^k)$.

The conclusion now follows by combining these two properties of the Rayleigh quotient with the fact that the sequences $\{d_1^k\}, \{d_2^k\}, ..., \{d_l^k\}$ are converging to zero. □

Theorem 4.1 establishes the convergence of the Barzilai and Borwein method when applied to a quadratic function with a SPD Hessian.

> **Theorem 4.1** Let $f(x)$ be a strictly convex quadratic function. Let $\{x_k\}$ be the sequence generated by Algorithm 2.2 and x_* the unique minimizer of f. Then, either $x_j = x_*$ for some finite integer j, or the sequence $\{x_k\}$ converges to x_*.

Proof: We need only consider the case in which there is no finite integer j such that $x_j = x_*$. Hence, it suffices to prove that the sequence $\{e_k\}$ converges to zero.

From (4.1) and the orthonormality of the eigenvectors we have

$$\|e_k\|_2^2 = \sum_{i=1}^{n} (d_i^k)^2.$$

And so, the sequence of errors $\{e_k\}$ converges to zero if and only if each one of the sequences $\{d_i^k\}$ for $i = 1, 2, ..., n$ converges to zero.

Suppose, by way of contradiction, that some of the sequences $\{d_i^k\}$ are not converging to zero. In particular, let us suppose that p is the smallest integer between 1 and n for which the sequence $\{d_p^k\}$ does not converge to zero. By Lemma 4.2, we can see that $p \geq 2$.

By Lemma 4.3, taking $\delta = (\lambda_p - \lambda_1)/4$, it follows that there exists $\hat{k}$ sufficiently large, such that

$$\lambda_p - \frac{(\lambda_p - \lambda_1)}{4} \leq \alpha_k \leq \lambda_{max}, \quad \textit{for all } k \geq \hat{k}.$$

And so, by using the fact that $d_p^{k+1} = ((\alpha_k - \lambda_p)/\alpha_k)d_p^k$, we obtain for all $k \geq \hat{k}$,

$$|d_p^{k+1}| = |1 - \frac{\lambda_p}{\alpha_k}||d_p^k| \leq \hat{c}|d_p^k|,$$

where

$$\hat{c} = \max(\frac{1}{3}, 1 - \frac{\lambda_p}{\lambda_{max}}) < 1 ,$$

which is a contradiction. Therefore, x_k converges to x_* □

Notice that with the choice of $\hat{\alpha}_{k+1} = \frac{s_k^t A^2 s_k}{s_k^t A s_k}$ instead of $\alpha_{k+1} = \frac{s_k^t A s_k}{s_k^t s_k}$, equality (4.5) can be written as

$$\hat{\alpha}_{k+1} = \frac{\sum_{i=1}^n (d_i^k)^2 \lambda_i^4}{\sum_{i=1}^n (d_i^k)^2 \lambda_i^3}.$$

Then, by a similar argument, we conclude that $\hat{\alpha}_{k+1}$ is a well defined and a continuous function of the variables $(d_1^k, ..., d_n^k)$. Thus, the convergence result established in Theorem 4.1 for Algorithm 2.2 with the choice of α_{k+1} also holds with the choice of $\hat{\alpha}_{k+1}$.

Chapter 5

Convergence Rate for the 2 Eigenvalues Case

In this chapter we study the rate of convergence of the Barzilai and Borwein gradient method applied to a quadratic function with a SPD Hessian A that has only two distinct eigenvalues $0 < \lambda_1 < \lambda_2$. In this special case the behavior of the scalars α_k is very predictable and we prove that under mild conditions the sequence $\{x_k\}$ generated by the method converges (3,2)-step Q-quadratically to x_*, i.e., for any norm, there exists $\hat{k}$ sufficiently large and positive constants $\hat{d_1}$ and $\hat{d_2}$ such that

$$\frac{\|e_{5k+3}\|}{\|e_{5k}\|^2} \leq \hat{d_1} \quad \textit{and} \quad \frac{\|e_{5k+5}\|}{\|e_{5k+3}\|^2} \leq \hat{d_2} \quad \textit{for all} \quad k \geq \hat{k}\,.$$

This unusual convergence property implies an R-convergence rate of at least $\sqrt[5]{4}$. Barzilai and Borwein [3] outline a convergence analysis of the method only for the 2-dimensional quadratic case. They observe, for this particular case, that the gradient converges to zero with an R-rate of $\sqrt{2}$ when the process begins with $x_0 = (\lambda_2, \frac{\lambda_1}{\lambda_2})^t$ and $\alpha_0 = \frac{(\lambda_2^2+\lambda_2)}{(\lambda_2^2+\lambda_1)}$. However, at the end of this chapter, we present an example where the generated sequence has an R-convergence rate of $\sqrt[5]{4}$. This means that the R-convergence rate of the algorithm for this special case is indeed $\sqrt[5]{4}$. Hence, the Barzilai and Borwein $\sqrt{2}$ R-convergence result occurred because of the special starting value.

First, we present an example where the (3,2)-step Q-quadratic local convergence is not attained because the initial data x_0 and α_0 are chosen so that the coefficients in the eigenvector expansion of the vector s_k have the same absolute value and the scalar α_k remains constant for all k, i.e., it does not do a good job of approximating either one of the eigenvalues. As a consequence, the convergence is just Q-linear, and its R-convergence is also only linear.

Example 5.1
Let $f(x) = \frac{1}{2}x^tAx$ where $A = diag(1,2)$ and let the starting data be $x_0 = (2\ ,\ 1)^t$ and let $\alpha_0 = 1.5$. Then for all k in Algorithm 2.2, we have

$$\alpha_k = 1.5 \quad and \quad \|e_{k+1}\|_2 = \frac{1}{3}\|e_k\|_2\ .$$

However, if we induce a slight perturbation in either α_0 or x_0 in Example 5.1, then the scalars α_k will oscillate within the spectrum of A and the convergence will be much faster. In fact, as we will discuss in the next section, if $\alpha_0 \epsilon I\!R$ and $x_0 \epsilon I\!R^n$ are chosen at random then the probability of the scalar α_k being fixed during the process is zero, and the sequence $\{\alpha_k\}$ will attempt to approach λ_1 and λ_2 alternatively.

5.1 The Q-rate of Convergence

Equality (3.1) can be written as

$$s_0 = \sum_{i=1}^{m} c_i v_i + \sum_{i=m+1}^{n} c_i v_i \qquad (5.1)$$

where $\{v_1, ..., v_m\}$ are orthonormal eigenvectors associated with λ_1 and $\{v_{m+1}, ..., v_n\}$ are orthonormal eigenvectors associated with λ_2. Using Lemma 3.1 and (5.1) we obtain

$$\begin{aligned} s_1 &= \frac{1}{\alpha_1}(\alpha_0 I - A)s_0 \\ &= (\frac{\alpha_0 - \lambda_1}{\alpha_1})\sum_{i=1}^{m} c_i v_i + (\frac{\alpha_0 - \lambda_2}{\alpha_1})\sum_{i=m+1}^{n} c_i v_i \\ &= (\frac{\alpha_0 - \lambda_1}{\alpha_1})y_1 + (\frac{\alpha_0 - \lambda_2}{\alpha_1})y_2 \qquad (5.2) \end{aligned}$$

where $y_1 = \sum_{i=1}^{m} c_i v_i$ and $y_2 = \sum_{i=m+1}^{n} c_i v_i$.

Let us now assume that the initial data α_0 and x_0 are given such that the components of y_1 and y_2 in (5.2) have distinct absolute values. Without loss of generality, we can say that

$$s_1 = y_1 + \epsilon\, y_2 \quad where \quad |\epsilon| < 1. \qquad (5.3)$$

Next, the scalar α_2 is computed as the Rayleigh quotient evaluated at s_1. It is a very well-known fact that (see Noble and Daniel [10],page 432)

$$\begin{aligned}\alpha_2 &= \lambda_1 + (\lambda_2 - \lambda_1)\frac{y_2^t y_2}{(y_1^t y_1 + \epsilon^2 y_2^t y_2)}\epsilon^2 \\ &= \lambda_1 + O(\epsilon^2). \qquad (5.4)\end{aligned}$$

Using Lemma 3.1 and (5.3) we obtain

$$s_2 = \frac{1}{\alpha_2}(\alpha_1 I - A)(y_1 + \epsilon\, y_2) = \frac{\alpha_1 - \lambda_1}{\alpha_2} y_1 + O(\epsilon) y_2. \qquad (5.5)$$

Since α_1 is not neccesarily close to either of the two eigenvalues, we have

$$\alpha_3 = \lambda_1 + O(\epsilon^2)\ .$$

Now, using Lemma 3.1, (5.4) and (5.5) we obtain

$$\begin{aligned}s_3 &= \frac{1}{\alpha_3}((\lambda_1 + O(\epsilon^2))I - A)(\frac{\alpha_1 - \lambda_1}{\alpha_2} y_1 + O(\epsilon) y_2) \\ &= O(\epsilon^2) y_1 + O(\epsilon) y_2\end{aligned}$$

and the scalar α_4 is now moving towards λ_2,

$$\alpha_4 = \lambda_2 + O(\epsilon^2)\ .$$

By the same arguments, we obtain

$$\begin{aligned}s_4 &= O(\epsilon^4) y_1 + O(\epsilon) y_2 \\ \alpha_5 &= \lambda_2 + O(\epsilon^6) \\ s_5 &= O(\epsilon^4) y_1 + O(\epsilon^3) y_2 \\ \alpha_6 &= \lambda_2 + O(\epsilon^2) \\ s_6 &= O(\epsilon^4) y_1 + O(\epsilon^9) y_2\ ,\end{aligned}$$

and now the scalar α_7 moves back towards λ_1

$$\begin{aligned}\alpha_7 &= \lambda_1 + O(\epsilon^{10}) \\ s_7 &= O(\epsilon^4) y_1 + O(\epsilon^{11}) y_2\ .\end{aligned}$$

The vector s_7 and the scalar α_7 share some important features with the vector s_2 and the scalar α_2. Both α_7 and α_2 are approximating λ_1, and the coefficient associated with y_2 is much smaller in absolute value than the coefficient associated with y_1 in the expansion of the vector s_2 and also of the vector s_7. So, by an induction argument we see that the sequence $\{\alpha_k\}$ is demonstrating a cyclic behavior. The scalars α_k are alternatively approaching λ_1 and λ_2 and the approximation is closer to both eigenvalues at every cycle. Therefore, it is reasonable to assume for our next result that eventually, the scalar α_k will be very close to either λ_1 or λ_2. Theorem 5.1 establishes the local convergence of Algorithm 2.2 applied to a quadratic function with a Hessian that has only 2 distinct eigenvalues.

Theorem 5.1 Let $f(x) = \frac{1}{2}x^tAx - b^tx + c$ where A is a SPD matrix that has only two distinct eigenvalues $\lambda_1 < \lambda_2$. Assume that for some integer k, α_k is sufficiently close to λ_1 or λ_2. Then the sequence $\{x_j\}$ from Algorithm 2.2 converges at least (3,2)-step Q-quadratically to x_*. Furthermore, if α_k is equal to either λ_1 or λ_2 then $x_{k+3} = x_*$.

Proof: Since p-step Q-quadratic convergence is independent of norm, we can work with the 2-norm. From Lemma 3.1 we know that $\|e_k\|_2$ tends to zero if and only if $\|s_k\|_2$ tends to zero. Let us assume without loss of generality that the coefficients associated with y_1 and y_2 in the expansion of the vector s_k are sufficiently close to zero.

Assume

$$\alpha_k = \lambda_1 + \epsilon \ , \qquad 0 < \epsilon \ll 1 \ , \tag{5.6}$$

and

$$s_k = \beta_1 \epsilon \, y_1 + \beta_2 \epsilon \, y_2 \tag{5.7}$$

where $\beta_1, \beta_2 \in I\!R$, $|\beta_1| > |\beta_2| > 0$ and $\epsilon \ll \min(\lambda_1, |\beta_2|, \frac{1}{|\beta_1|}, \frac{1}{\lambda_2}, \|y_1\|_2, \|y_2\|_2)$. We need to prove, as a preliminary result, that the sequence $\{s_j\}$ from Algorithm 2.2 converges at least (3,2)-step Q-quadratically to the vector 0.

From (5.7) and the orthonormality of the eigenvectors we obtain

$$\begin{aligned} \|s_k\|_2^2 &= (\beta_1\epsilon)^2\, y_1^t y_1 + (\beta_2\epsilon)^2\, y_2^t y_2 \\ &= c_1\epsilon^2\,, \end{aligned} \tag{5.8}$$

where

$$c_1 = \beta_1 y_1^t y_1 + \beta_2 y_2^t y_2\,.$$

Using Lemma 3.1, (5.6) and (5.7) we have

$$\begin{aligned} s_{k+1} &= \frac{1}{\alpha_{k+1}}((\lambda_1+\epsilon)I - A)(\beta_1\epsilon\, y_1 + \beta_2\epsilon\, y_2) \\ &= c_2\epsilon^2 y_1 + c_3\epsilon y_2\,, \end{aligned} \tag{5.9}$$

where $c_2 = \frac{\beta_1}{\alpha_{k+1}}$ and $c_3 = \frac{(\lambda_1-\lambda_2)\beta_2}{\alpha_{k+1}}$. Since $\lambda_1 \le \alpha_{k+1} \le \lambda_2$, $|c_2|$ and $|c_3|$ are away from ϵ and away from infinity.

By (2.4) and a well-known property of the Rayleigh quotient (see Noble and Daniel [10], page 432), we obtain

$$\alpha_{k+2} = \lambda_2 + c_4\epsilon^2\,, \tag{5.10}$$

where

$$c_4 = \frac{c_5(\lambda_2-\lambda_1)y_1^t y_1}{y_2^t y_2 + c_5\epsilon^2 y_1^t y_1} \quad \mathit{and} \quad c_5 = \frac{\beta_1^2}{\beta_2^2(\lambda_1-\lambda_2)^2}.$$

Now, using Lemma 3.1, (5.9) and the fact that $\alpha_{k+1} = \frac{\lambda_1+c_6\lambda_2}{1+c_6}$ where

$$c_6 = \frac{(\beta_2^2 y_2^t y_2)}{(\beta_1^2 y_1^t y_1)}\,,$$

we have

$$\begin{aligned} s_{k+2} &= \frac{1}{\alpha_{k+2}}(\alpha_{k+1}I - A)(c_2\epsilon^2 y_1 + c_3\epsilon y_2) \\ &= c_7\epsilon^2 y_1 + c_8\epsilon y_2\,, \end{aligned} \tag{5.11}$$

where

$$c_7 = \frac{c_6(\lambda_2-\lambda_1)}{\alpha_{k+2}(1+c_6)} \quad \mathit{and} \quad c_8 = \frac{(\lambda_1-\lambda_2)}{\alpha_{k+2}(1+c_6)}\,.$$

Since $\lambda_1 \leq \alpha_{k+2} \leq \lambda_2$, $|c_7|$ and $|c_8|$ are away from ϵ and away from infinity. Finally, by means of Lemma 3.1, (5.10) and (5.11) it follows that

$$\begin{aligned} s_{k+3} &= \frac{1}{\alpha_{k+3}}((\lambda_2 + c_4\epsilon^2)I - A)(c_7\epsilon^2 y_1 + c_8\epsilon y_2) \\ &= c_9\epsilon^2 y_1 + c_{10}\epsilon^3 y_2 \ , \end{aligned} \tag{5.12}$$

where

$$c_9 = \frac{c_7(\lambda_2 + c_4\epsilon^2\lambda_1)}{\alpha_{k+3}} \quad \textit{and} \quad c_{10} = \frac{c_4 c_8}{\alpha_{k+3}}.$$

Both $|c_9|$ and $|c_{10}|$ are away from ϵ and away from infinity. Hence,

$$\|s_{k+3}\|_2 = c_{11}\epsilon^2 \ , \tag{5.13}$$

where $c_{11} = (c_9^2 y_1^t y_1 + c_{10}^2\epsilon^2 y_2^t y_2)^{\frac{1}{2}}$.

Combining (5.8) and (5.13) gives

$$\|s_{k+3}\|_2 \leq \ c_{12}\|s_k\|_2^2 \ , \tag{5.14}$$

where

$$c_{12} = \frac{c_{11}}{c_1} = \frac{c_{11}}{(\beta_1 y_1^t y_1 + \beta_2 y_2^t y_2)} \ < \infty \, .$$

By the same arguments, we obtain

$$\begin{aligned} \alpha_{k+3} &= \lambda_2 + c_{13}\epsilon^2 \\ \alpha_{k+4} &= \lambda_1 + c_{14}\epsilon^2 \\ s_{k+4} &= c_{15}\epsilon^2 y_1 + c_{16}\epsilon^5 y_2 \\ \alpha_{k+5} &= \lambda_1 + c_{17}\epsilon^6 \\ s_{k+5} &= c_{18}\epsilon^4 y_1 + c_{19}\epsilon^5 y_2 \ , \end{aligned}$$

where

$$c_{13} = \frac{c_6^2(\lambda_1 - \lambda_2)y_1^t y_1}{y_2^t y_2 + c_6^2\epsilon^2 y_1^t y_1} \quad , \quad c_{14} = \frac{(\lambda_2 - \lambda_1)y_2^t y_2}{y_1^t y_1 + (\frac{c_{10}}{c_9})^2\epsilon^2 y_2^t y_2} \ ,$$

$$c_{15} = \frac{c_9}{\alpha_{k+4}}(\lambda_2 + c_{13}\epsilon^2 - \lambda_1) \quad , \quad c_{16} = \frac{c_{13}c_{10}}{\alpha_{k+4}} \ ,$$

$$c_{17} = \frac{(\lambda_2 - \lambda_1) y_2^t y_2}{y_1^t y_1 + (\frac{c_{16}}{c_{15}})^2 \epsilon^6 y_2^t y_2} ,$$

$$c_{18} = \frac{c_{14} c_{15}}{\alpha_{k+5}} \quad \textit{and} \quad c_{19} = \frac{c_{16}(\lambda_1 + c_{14}\epsilon^2 - \lambda_2)}{\alpha_{k+5}} .$$

Hence,

$$\|s_{k+5}\|_2 = c_{20}\epsilon^4 , \tag{5.15}$$

where $c_{20} = (c_{18}^2 y_1^t y_1 + c_{19}^2 \epsilon^2 y_2^t y_2)^{\frac{1}{2}}$.

Combining (5.13) and (5.15) gives

$$\|s_{k+5}\|_2 \leq c_{21} \|s_{k+3}\|_2^2 , \tag{5.16}$$

where

$$c_{21} = \frac{c_{20}}{c_{11}^2} < \infty .$$

Now, we are ready to prove that the sequence $\{x_j\}$ converges at least (3,2)-step Q-quadratically to the vector x_*. From Lemma 3.1 and the fact that A is SPD, we have

$$\|e_k\|_2 \leq \alpha_k \|A^{-1}\|_2 \|s_k\|_2 \leq \frac{\lambda_2}{\lambda_1} \|s_k\|_2 \tag{5.17}$$

and

$$\|s_k\|_2 \leq \frac{1}{\alpha_k} \|A\|_2 \|e_k\|_2 \leq \frac{\lambda_2}{\lambda_1} \|e_k\|_2 . \tag{5.18}$$

From (5.17) we obtain

$$\frac{1}{\|e_k\|_2^2} \leq (\frac{\lambda_2}{\lambda_1})^2 \frac{1}{\|s_k\|_2^2} . \tag{5.19}$$

Finally, using (5.17), (5.19), (5.14) and (5.16) we have

$$\frac{\|e_{k+3}\|_2}{\|e_k\|_2^2} \leq (\frac{\lambda_2}{\lambda_1})^3 \frac{\|s_{k+3}\|_2}{\|s_k\|_2^2} \leq (\frac{\lambda_2}{\lambda_1})^3 c_{12} < \infty$$

and

$$\frac{\|e_{k+5}\|_2}{\|e_{k+3}\|_2^2} \leq (\frac{\lambda_2}{\lambda_1})^3 \frac{\|s_{k+5}\|_2}{\|s_{k+3}\|_2^2} \leq (\frac{\lambda_2}{\lambda_1})^3 c_{21} < \infty .$$

Notice that α_{k+5}, as well as α_k, is a good approximation to λ_1, and also that the vectors s_{k+5} and s_k have a similar eigenvector expansion. In fact, the scalar α_{k+5} and the vector s_{k+5} can be written as

$$\alpha_{k+5} = \lambda_1 + \hat{c}\hat{\epsilon} \ ,$$

$$s_{k+5} = \hat{\beta}_1 \hat{\epsilon} y_1 + \hat{\beta}_2 \hat{\epsilon} y_2 \ ,$$

where $\hat{\epsilon} = \epsilon^4$, $\hat{\beta}_1 = c_{18}$, $\hat{\beta}_2 = c_{19}\epsilon$ and $\hat{c} = c_{17}\epsilon^2$. Clearly, $|\hat{\beta}_1| > |\hat{\beta}_2| > 0$ and $\hat{\epsilon} \lll \hat{c}$. Hence, an identical cycle of five iterations starts at the iteration $(k+5)$ with $\hat{\epsilon}$ instead of ϵ.

To prove the last statement, assume $\alpha_k = \lambda_1$, then from (5.9) we obtain that $s_{k+1} = O(\epsilon)y_2$, $\alpha_{k+2} = \lambda_2$ and $s_{k+2} = O(\epsilon)y_2$. Finally,

$$s_{k+3} = \frac{1}{\alpha_{k+3}}(\lambda_2 I - A)O(\epsilon)y_2 = 0 \ .$$

Clearly, $s_{k+3} = 0$ implies that $x_{k+3} = x_*$. □

Notice that the property of the Rayleigh quotient described in Noble and Daniel ([10], page 432) also holds with the choice of $\hat{\alpha}_{k+1} = \frac{s_k^t A^2 s_k}{s_k^t A s_k}$ instead of $\alpha_{k+1} = \frac{s_k^t A s_k}{s_k^t s_k}$. Therefore, the Q-convergence result established in Theorem 5.1 holds for both choices of steplength.

To illustrate Theorem 5.1 we present some randomly generated numerical examples. For each problem, we generate two positive eigenvalues $\lambda_1 < \lambda_2$, a real number α_0 and a real vector $x_0 \in I\!R^2$, and apply Algorithm 2.2 on the quadratic function $f = \frac{1}{2}x^t A x$ where $A = diag(\lambda_1, \lambda_2)$. Numerical results for four such problems are tabulated in Table 5.1. The errors $\|x_k - x_*\|_2$ are given for each problem. It can be seen from the table that the convergence rate of Algorithm 2.2 when the matrix has two distinct eigenvalues is at least (3,2)-step Q-quadratic.

	problem 1	problem 2	problem 3	problem 4
iteration	$\lambda_1 = 5.7812$ $\lambda_2 = 13.8059$	$\lambda_1 = 1.72855$ $\lambda_2 = 3.56595$	$\lambda_1 = 8.9153$ $\lambda_2 = 21.0476$	$\lambda_1 = 11.9857$ $\lambda_2 = 82.6735$
1	.16d+00	.23d+00	.28d+00	.51d+00
2	.83d-01	.98d-01	.11d+00	.32d+00
3	.28d-01	.32d-01	.98d-01	.45d+00
4	.36d-02	.41d-02	.10d-01	.37d-01
5	.44d-02	.36d-02	.77d-02	.15d-02
6	.29d-04	.75d-04	.12d-02	.82d-01
7	.14d-04	.23d-04	.47d-04	.10d-07
8	.69d-05	.11d-04	.26d-04	.10d-07
9	.48d-10	.29d-09	.34d-09	.26d-10
10	.66d-10	.31d-09	.47d-10	.16d-20
11	.76d-19	.15d-14	.55d-10	
12		.13d-20	.98d-22	

Table 5.1 $||e_k||_2$ for 4 randomly generated problems

5.2 The R-rate of Convergence

We now discuss the R-convergence rate of Algorithm 2.2 when the matrix A has only two distinct eigenvalues. For this particular problem, Theorem 5.1 shows at least (3,2)-step Q-quadratic convergence which implies at least 5-step Q-order-four convergence, and this in turn, implies at least a $\sqrt[5]{4}$ R-convergence rate.

Let us define the sequence $\{a_k\}$ of positive numbers by

$$||e_{k+1}||_2 = a_k ||e_{k-4}||_2^4 \qquad k = 4, 5, \ldots \tag{5.20}$$

where $||e_k||$ is the sequence of errors generated by Algorithm 2.2. From the proof of Theorem 5.1 we know that

$$||e_k||_2 = O(\epsilon) \ , \qquad \alpha_k = \lambda_1 + O(\epsilon) \ ,$$

and after five steps

$$||e_{k+5}||_2 = O(\epsilon^4) \quad \mathit{and} \quad \alpha_{k+5} = \lambda_1 + O(\epsilon^6) \ .$$

The extra accuracy in α_{k+5} approximating λ_1 (compared to $\lambda_1 + O(\epsilon^4)$) leads to extra accuracy in $||e_{k+j}||_2$ approaching zero, for $j \geq 8$. The additional accuracy in both the scalars α_k and the errors e_k increases after each five step cycle. In fact, the numbers a_k defined in (5.20) can be written as $a_k = \epsilon^{h(k)}$, where h is a real valued function that depends on k. Therefore, except for pathologies (like the one described in Example 5.1), the sequence $\{a_k\}$ converges to zero. The speed of convergence of the sequence $\{a_k\}$ depends on the function h, which depends on the distribution of the two distinct eigenvalues and the initial data.

Theorem 2.2 of Potra [12] gives the following sufficient condition for a sequence of errors $||e_k||$ that satisfies (5.20) to have an exact R-order of convergence of $\sqrt[5]{4}$

$$\limsup_{k\to\infty} |\log a_k|^{\frac{1}{k}} = 1 \ . \tag{5.21}$$

Computational experience shows that in many cases, condition (5.21) does not hold (i.e., the sequence $\{a_k\}$ converges to zero extremely fast), and so the R-rate of convergence for a particular sequence can be greater than $\sqrt[5]{4}$. In fact, Barzilai and Borwein [3] prove $\sqrt{2}$ R-convergence rate when the method starts from a particular initial guess. However, we now present an example for which condition (5.21) holds and so, the exact R-convergence rate of the algorithm is $\sqrt[5]{4}$, as implied by our (3,2)-step Q-quadratic convergence result.

Example 5.2

Let $f(x) = \frac{1}{2}x^tAx$ where $A = diag(1,3)$ and let the starting data be $x_0 = (\epsilon, \epsilon^2)^t$ and $\alpha_0 = 1 + \epsilon^{\frac{1}{2}}$, where $0 < \epsilon < 1$. The sequence $\{a_k\}$ converges to zero for this particular function and initial data. However, it has the following property:

$$|a_k| \geq \epsilon^{k^5} \quad for \quad all \quad k.$$

Since

$$\lim_{k\to\infty} |\log \epsilon^{k^5}|^{\frac{1}{k}} = 1 \ ,$$

we conclude that condition (5.21) holds for this particular example. Summing up, we get the following result.

Corollary 5.1 Under the hypothesis of Theorem 5.1, Algorithm 2.2 has the exact R-rate of convergence of $\sqrt[5]{4}$.

To illustrate our discussion about the R-rate of convergence we present some numerical results. We consider the Barzilai and Borwein gradient method for the minimization of $f(x) = \frac{1}{2}x^tAx$ where $A = diag(1,3)$ with two different sets of initial data. The starting data proposed in Example 5.2 that allowed us to prove $\sqrt[5]{4}$ R-rate of convergence, and the initial data proposed by Barzilai and Borwein [3] for which they proved $\sqrt{2}$ R-rate of convergence. For this particular example, the Barzilai and Borwein choice is $x_0 = \epsilon(3, \frac{1}{3})^t$ and $\alpha_0 = (9\epsilon^2 + 3\epsilon)/(9\epsilon^2 + 1)$. Table 5.2 shows $||e_k||_2$ and $|\log ||e_k||_2|^{\frac{1}{k}}$ for both sets of initial data when $\epsilon = 0.4$.

Proposition 1.2 of Potra [12] shows that the exact R-rate of convergence is equal to the $\liminf_{k\to\infty} |\log ||e_k||_2|^{\frac{1}{k}}$. And so, we can observe from Table 5.2 that the R-rate of convergence for this particular example is very close to the theoretical results obtained for both sets of initial data (recall: $\sqrt{2} \approx 1.414$ and $\sqrt[5]{4} \approx 1.319$).

	Example 4 initial data		Barzilai and Borwein initial data	
k	$\|e_k\|_2$	$\|\log \|e_k\|_2\|^{\frac{1}{k}}$	$\|e_k\|_2$	$\|\log \|e_k\|_2\|^{\frac{1}{k}}$
1	.20d+00	1.2591	.19d+00	1.2711
2	.97d-01	1.3246	.29d+00	1.0642
3	.53d-01	1.3081	.82d-02	1.4798
4	.32d-01	1.2799	.41d-02	1.4058
5	.41d-02	1.3287	.26d-02	1.3459
6	.28d-02	1.2873	.71d-07	1.4920
7	.11d-02	1.2700	.14d-06	1.4116
8	.18d-04	1.3043	.40d-12	1.4511
9	.12d-04	1.2747	.65d-18	1.4527
10	.47d-09	1.3215	.43d-18	1.4055
11	.43d-11	1.3126	.96d-34	1.4382
12	.87d-11	1.2827	.87d-40	1.4162
13	.18d-25	1.3385		
14	.12d-25	1.3134		
15	.29d-41	1.3298		

Table 5.2 R-rate of convergence behavior of Algorithm 2.2 with two different sets of initial data

Chapter 6

Numerical Investigation of the R-rate of Convergence in the General Case

We showed in Chapter 5 that the R-rate of convergence of Algorithm 2.2 is exactly $\sqrt[5]{4}$ whenever the Hessian matrix A has only 2 distinct eigenvalues. In this chapter, we present a numerical investigation of the R-rate of convergence of Algorithm 2.2 when A has more than 2 distinct eigenvalues.

All experiments in this chapter were run on a SUN 3/50 workstation in double precision FORTRAN with a machine epsilon of about 2x10^{-16}. We considered $f(x) = \frac{1}{2}x^tAx$ where A is SPD and has already been diagonalized, i.e., $A = diag(d_1, ..., d_n)$. Clearly, f has a unique minimizer at $x_* = (0, ..., 0)^t$. We ran an implementation of Algorithm 2.2 for different choices of the diagonal elements d_i, starting always at $x_0 = (.1, .1, ..., .1)^t$ and $\alpha_0 = 1.5$. We stopped the process whenever $||e_k||_2 \leq 10^{-14}$, and estimated the R-rate of convergence as $\min_k |\log ||e_k||_2|^{\frac{1}{k}}$ over the last 10 iterations. The value obtained by this procedure is an accurate approximation to the exact R-rate of convergence that is defined as the $\liminf_{k\to\infty} |\log ||e_k||_2|^{\frac{1}{k}}$, see [12].

6.1 Description of Experiments and Numerical Results

The first set of experiments is chosen to demonstrate the effect of reducing the width of the spectrum of A on the R-rate of convergence. Consider the following 3 choices of diagonal elements for the matrix A :

1. $d_i = i \qquad i = 1, ..., n$
2. $d_i = (i + 4)/5 \qquad i = 1, ..., n$
3. $d_i = (i + 9)/10 \qquad i = 1, ..., n$

For each of these 3 different cases, the smallest eigenvalue of A is equal to 1. The largest eigenvalue of A ranges from n in the first case to approximately $n/5$ in the second case to $n/10$ in the third case. Therefore, for this particular experiment, a reduction of the width of the spectrum is equivalent to a reduction of $\kappa(A)$, the spectral condition number of A. (Recall: $\kappa(A) = \lambda_{max}/\lambda_{min}$).

Figure 6.1 shows the effect of reducing $\kappa(A)$ on the R-rate of convergence for the 3 different choices of diagonal elements. The convergence is clearly R-superlinear when the Hessian matrix A has very few eigenvalues, but it decreases very fast as the number of distinct eigenvalues increases. In fact, the R-rate of convergence seems to converge asymptotically to 1 from above. It can also be observed that the R-rate of convergence decreases at different speeds for different condition numbers of the matrix A. In fact, the R-rate of convergence decreases faster for $\kappa(A) = n$ than for $\kappa(A) = n/5$, which decreases faster than for $\kappa(A) = n/10$. In other words, the method enjoys R-superlinearity for a larger number of distinct eigenvalues when the problem is better conditioned.

We also study the effect of reducing $\kappa(A)$ on the number of iterations required to terminate the process. Figure 6.2 shows this effect for the 3 different choices of diagonal elements. One may observe that the number of iterations increases as the number of distinct eigenvalues increases. However, it does so at different speeds; the slope is steeper when $\kappa(A)$ is higher. It is also interesting to observe that for n large, the difference in the R-rate of convergence is very small. However, this small difference represents a very large difference in the number of iterations.

The second set of experiments is chosen to demonstrate the effect that clustering the eigenvalues has on the R-rate of convergence. Consider the following 2 choices of diagonal elements for the matrix A :

1. $d_i = 1 + \frac{499}{n-1}(i-1) \qquad i = 1, ..., n$

2. $d_i = 1 + \frac{i-1}{\frac{n}{2}-1} \qquad i = 1, ..., \frac{n}{2}$ and

   $d_i = 499 + \frac{i-\frac{n}{2}-1}{\frac{n}{2}-1} \qquad i = \frac{n}{2}+1, ..., n$

In the first case, all eigenvalues are uniformly distributed in the interval $[1, 500]$. In the second case, there is a cluster of $n/2$ eigenvalues in the interval $[1, 2]$ and another cluster of $n/2$ eigenvalues in the interval $[499, 500]$. For this experiment, the spectral condition number of A is constant. Indeed, $\kappa(A) = 500$ in both cases.

Figure 6.3 shows the effect that clustering the eigenvalues of the matrix A has on the R-rate of convergence. One may observe that the R-rate of convergence decreases asymptotically to 1 when the eigenvalues are uniformly distributed in the spectrum. One may also observe that the R-rate of convergence seems to converge to a constant that is clearly larger than 1 when the spectrum is divided in two separate clusters. In other words, if the spectrum of A is clustered, then the method seems to enjoy R-superlinearity for a very large number of distinct eigenvalues.

We also study the effect of clustering the eigenvalues on the number of iterations required to terminate the process. Figure 6.4 shows this effect in our experiment. The number of iterations increases as the number of distinct eigenvalues increases when they are uniformly distributed in the interval $[1, 500]$. However, the number of iterations remains constant when the eigenvalues are clustered in the small intervals $[1, 2]$ and $[499, 500]$. Indeed, only 60 iterations are required to terminate the process for any large number of distinct eigenvalues.

In conclusion, our numerical experiments seem to indicate that the Barzilai and Borwein method has a much better performance when the Hessian matrix A has been preconditioned in such a way that either $\kappa(A)$ is reduced or the eigenvalues of A are clustered. However, for large problems that do not have any special distribution of eigenvalues, the method seems to have an R-linear rate of convergence.

The conclusions obtained are consistent with the interpretation of the method that we presented in Chapter 3. As a consequence of the relationship to the shifted power method, the scalars α_k are "chasing" eigenvalues within the spectrum of A and the speed of convergence depends on how fast the coefficients associated with the eigenvectors in the error expansion are driven to zero. Therefore, if the spectrum is reduced or clustered, the "chasing" of eigenvalues will be more effective and convergence will be attained faster.

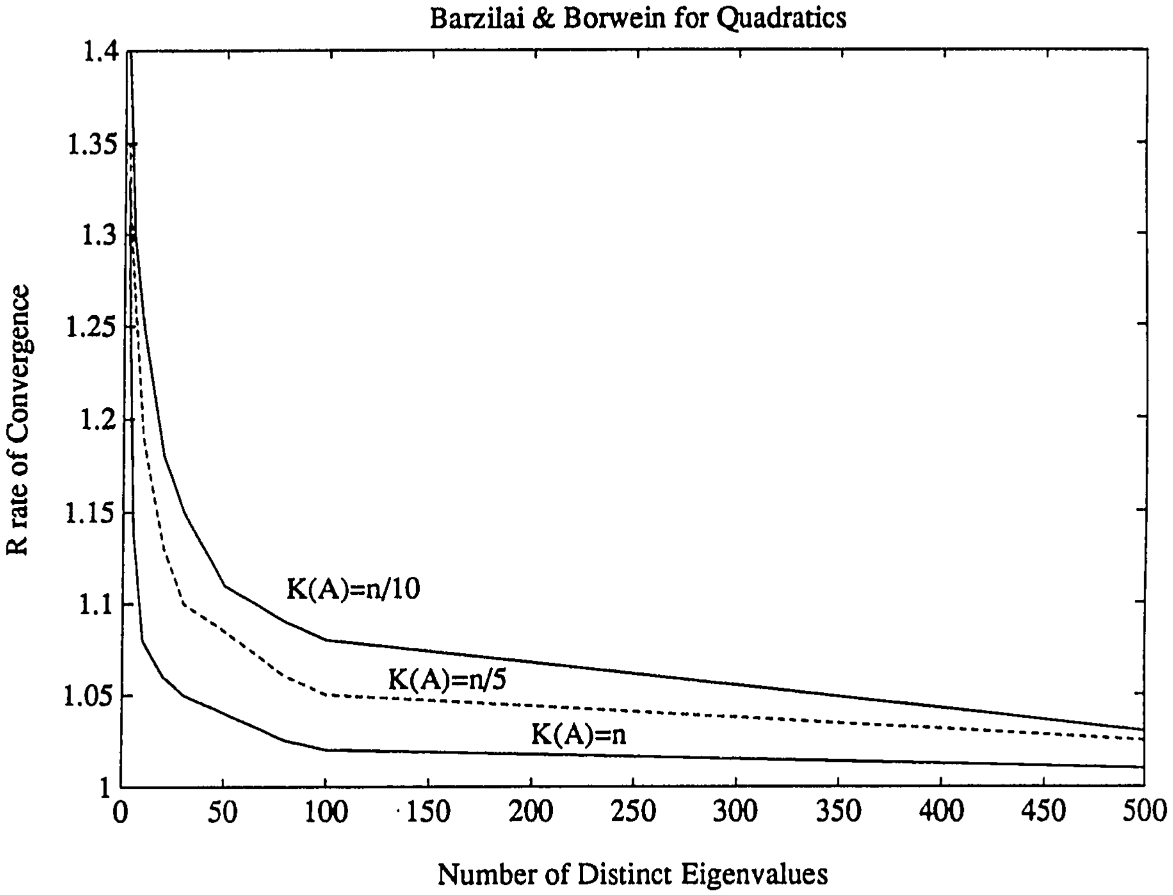

Figure 6.1 Effect of reducing $\kappa(A)$ on the R-rate of convergence.

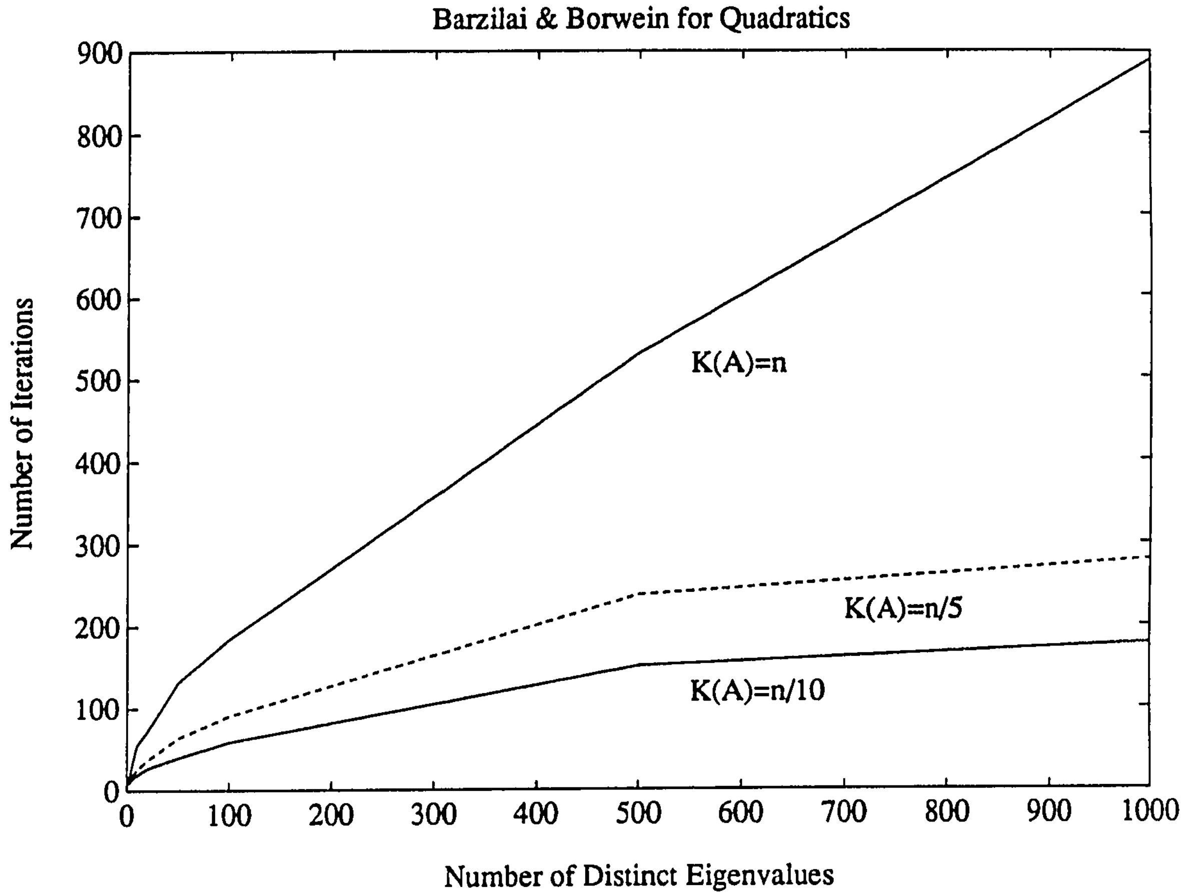

Figure 6.2 Effect of reducing $\kappa(A)$ on the number of iterations required to achieve $\|e_k\|_2 \leq 10^{-14}$.

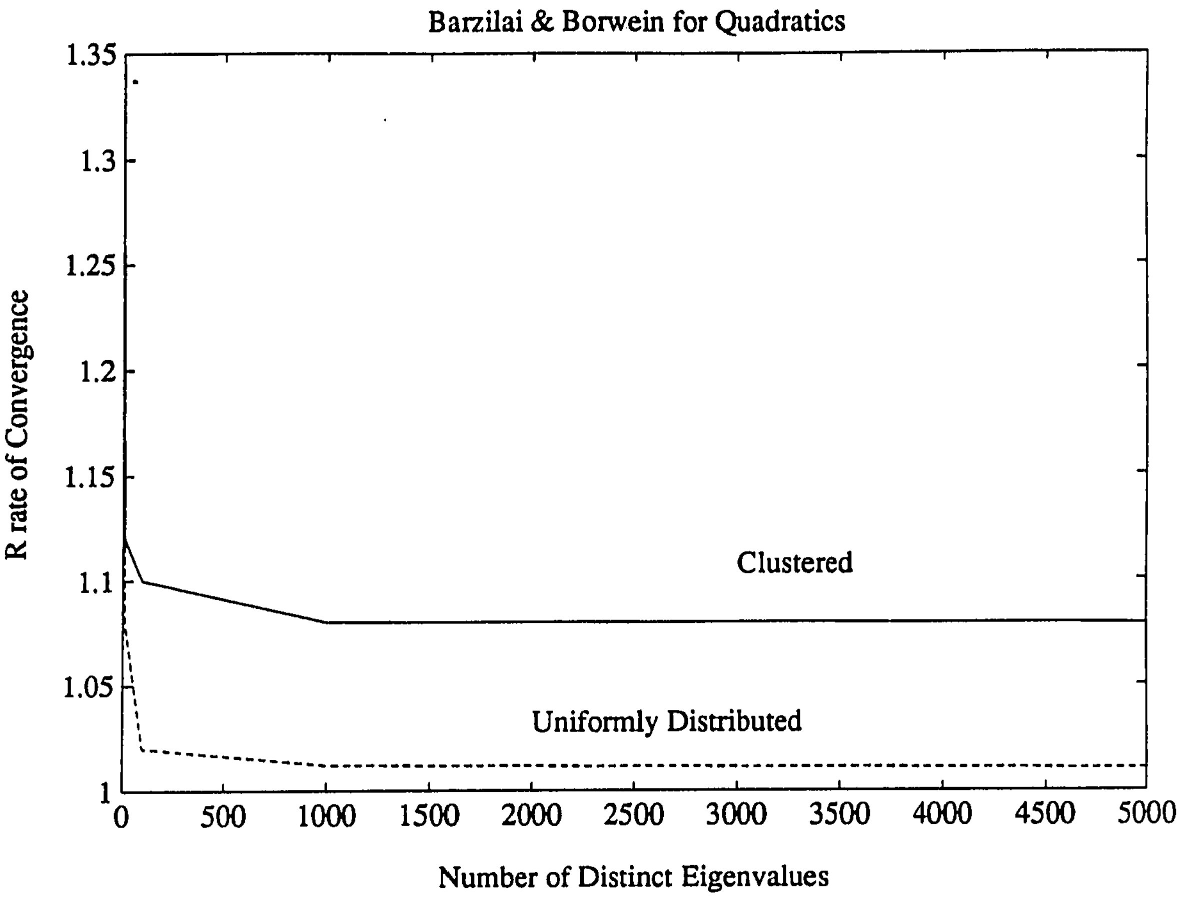

Figure 6.3 Effect of clustering the eigenvalues of A on the R-rate of convergence.

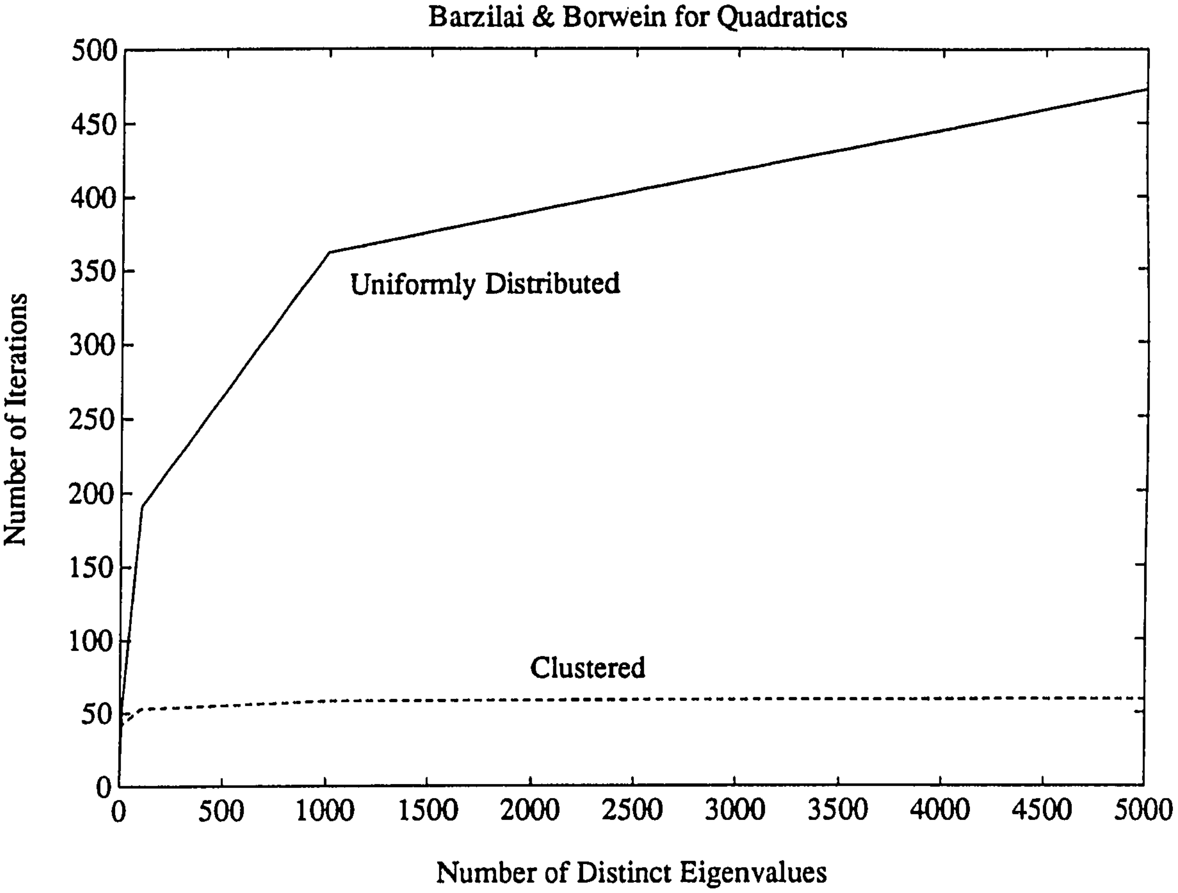

Figure 6.4 Effect of clustering the eigenvalues of A on the number of iterations required to achieve $\|e_k\|_2 \leq 10^{-14}$.

Chapter 7

Preconditioned Barzilai and Borwein and Numerical Results for a PDE Application

In this chapter we derive the preconditioned Barzilai and Borwein method for the minimization of strictly convex quadratic functions. We present without derivation the preconditioned conjugate gradient method, and the Symmetric Successive Overrelaxation (SSOR) preconditioning technique, see [1],[2],[15]. The test problems are large sparse linear systems arising from the numerical solution of elliptic PDE problems. We present a set of numerical experiments to compare the preconditioned Barzilai and Borwein with the preconditioned conjugate gradient and discuss numerical results.

7.1 Preconditioned Barzilai and Borwein

In this section, we derive the preconditioned Barzilai and Borwein method. The basic idea is to transform the problem of minimizing

$$f(x) = \frac{1}{2}x^tAx - b^tx + c \ ,$$

to that of minimizing a related quadratic functional

$$\tilde{f}(y) = \frac{1}{2}y^t\tilde{A}y - \tilde{b}^ty + \tilde{c}\,, \tag{7.1}$$

where

$$\tilde{A} = E^{-1}AE^{-t}, \qquad \tilde{b} = E^{-1}b, \qquad \tilde{c} = c,$$

for some nonsingular matrix E. The motivation for minimizing $\tilde{f}(y)$ instead of $f(x)$ is that if E makes $\kappa(\tilde{A})$ much smaller than $\kappa(A)$ or if it clusters the spectrum of $\tilde{A}$, then our numerical investigation indicates that the rate of convergence is greater for $\tilde{f}(y)$ than for $f(x)$.

Our derivation is similar to the derivation of the preconditioned conjugate gradient method, see [2]. Let C be a positive definite matrix factored in the form $C = EE^t$, and consider the quadratic functional (7.1). The matrix $\tilde{A}$ is clearly symmetric. Moreover, since A is positive definite, $\tilde{A}$ is also positive definite. The similarity transformation

$$E^{-t}\tilde{A}E^t = E^{-t}E^{-1}A = C^{-1}A$$

reveals that $\tilde{A}$ and $C^{-1}A$ have the same eigenvalues, and so the spectral condition number of $\tilde{A}$ is completely determined by C and A.

Consider the application of Algorithm 2.2 to $\tilde{f}(y)$. The iterations are described by

$$\tilde{g}_k = \tilde{A}y_k - \tilde{b}, \tag{7.2}$$

$$\tilde{s}_k = -\frac{1}{\tilde{\alpha}_k}\tilde{g}_k, \tag{7.3}$$

$$y_{k+1} = y_k + \tilde{s}_k, \tag{7.4}$$

$$\tilde{\alpha}_{k+1} = \frac{\tilde{s}_k^t\tilde{A}\tilde{s}_k}{\tilde{s}_k^t\tilde{s}_k}, \tag{7.5}$$

where $\tilde{\alpha}_0 \neq 0$ and y_0 are arbitrarily chosen. It follows from the convergence analysis in Chapter 4 that

$$\lim_{k\to\infty} y_k = \tilde{y},$$

where $\tilde{y} = \tilde{A}^{-1}\tilde{b}$. It also follows , from our numerical investigation in Chapter 6, that the rate of convergence of the Barzilai and Borwein method applied to (7.1) depends on $\kappa(\tilde{A})$.

Let $x_k = E^{-t}y_k$, for all k. Simple calculations based on (7.2), (7.3),(7.4),(7.5) and the definitions of $\tilde{A}$ and $\tilde{b}$, show that

$$g_k = E\tilde{g}_k\ ,$$

and

$$\tilde{\alpha}_{k+1} = \frac{h_k^tAh_k}{g_k^th_k}\ ,$$

where $h_k = C^{-1}g_k$. Now, by using the recursion formula

$$g_{k+1} = g_k - \frac{1}{\alpha_k}Ah_k \ ,$$

we can see that the sequence $\{x_k\}$ is produced by the following algorithm

Algorithm 7.1 (Preconditioned Barzilai and Borwein Method)

```
Given $x_0 \, \epsilon \, I\!R^n$, $\alpha_0 \, \epsilon \, I\!R$ and $C$ a SPD $nxn$ matrix. Set $g_0 = Ax_0 - b$

For $k = 0, 1, ...$(Until convergence) do
    1. Solve $Ch_k = g_k$   for $h_k$
    2. Set $p_k = Ah_k$
    3. Set $x_{k+1} = x_k - \frac{1}{\alpha_k}h_k$
    4. Set $g_{k+1} = g_k - \frac{1}{\alpha_k}p_k$
    5. Set $\alpha_{k+1} = \frac{h_k^t p_k}{g_k^t h_k}$
End do
```

The matrix C is called the *preconditioning matrix* and $\tilde{A}$ the *preconditioned matrix.* Notice that every iteration of the preconditioned Barzilai and Borwein method involves two inner products, two scalar-vector multiplications, two vector additions, one matrix-vector multiplication and solving a system of linear equations with the preconditioning matrix C.

7.2 Preconditioned Conjugate Gradient

In this section, we present the preconditioned conjugate gradient (CG) method for finding the minimizer of a quadratic functional $f(x) = \frac{1}{2}x^tAx - b^tx + c$ with a SPD Hessian. The CG method has been extensively applied to the solution of large sparse SPD linear systems [1],[8], [9]. For a derivation of the preconditioned version of the CG method, see [2],[9].

There are many equivalent formulations of the preconditioned CG method, but the one that has proved to be the most efficient is the following procedure

Algorithm 7.2 (Preconditioned Conjugate Gradient Method)

```
Given $x_0 \epsilon I\!R^n$ and $C$ a SPD $nxn$ matrix, set $g_0 = Ax_0 - b$, solve $Ch_0 = g_0$
and set $d_0 = -h_0$

For $k = 0, 1, ...$(Until convergence) do

    1. Set $p_k = Ad_k$
    2. Set $\alpha_k = \frac{g_k^t h_k}{d_k^t p_k}$
    3. Set $x_{k+1} = x_k + \alpha_k d_k$
    4. Set $g_{k+1} = g_k + \alpha_k p_k$
    5. Solve $Ch_{k+1} = g_{k+1}$ for $h_{k+1}$
    6. Set $\beta_k = \frac{g_{k+1}^t h_{k+1}}{g_k^t h_k}$
    7. Set $d_{k+1} = -h_{k+1} + \beta_k d_k$

End do
```

Notice that every iteration of the preconditioned CG method involves two inner products, three scalar-vector multiplications, three vector additions, one matrix-vector multiplication and solving a system of linear equations with preconditioning matrix C.

7.3 The SSOR Preconditioning Matrix

One of the most popular preconditioning techniques is based on the SSOR iterative method to solve linear systems of equations, [15]. Let the Hessian matrix A be decomposed as

$$A = D + L + L^t \, ,$$

where D and L are the diagonal and lower triangular parts of A, respectively. The SSOR iterative method to solve $Ax = b$, is the following two-stage algorithm :

$$x_{k+\frac{1}{2}} = (1-\omega)x_k - \omega D^{-1}(Lx_{k+\frac{1}{2}} + L^t x_k - b) \, ,$$

$$x_{k+1} = (1-\omega)x_{k+\frac{1}{2}} - \omega D^{-1}(Lx_{k+\frac{1}{2}} + L^t x_{k+1} - b) ,$$

where ω is a real scalar parameter between 0 and 2.

This method can be formulated as a one-stage algorithm (see [2],[15]) :

$$Cx_{k+1} = Rx_k + b ,$$

where $C - R = A$ and

$$C = \frac{1}{2-\omega}(\frac{1}{\omega}D + L)(\frac{1}{\omega}D)^{-1}(\frac{1}{\omega}D + L)^t .$$

The matrix C is the SSOR preconditioning matrix. It involves the choice of the parameter ω that has been extensively studied for the iterative solution of large sparse linear systems of equations. An excellent overview of the development of this topic can be found in Young [16].

For the SSOR preconditioned CG method, the optimal value of ω (i.e., the value of ω that minimizes $\kappa(\tilde{A})$) has been shown to be

$$\omega^* = 2/[1 + (2/\sqrt{\mu})\sqrt{\frac{1}{2} + \delta}\ ] ,$$

where

$$\mu = \max_{x \neq 0} \frac{x^t D x}{x^t A x} ,$$

and

$$\delta = \max_{x \neq 0} \frac{x^t(LD^{-1}L^t - \frac{1}{4}D)x}{x^t A x} .$$

Unfortunately, μ and δ are extremely difficult to obtain in practice and only rough estimates are available. However, it has also been established that the rate of convergence of the CG method with SSOR preconditioning is remarkably insensitive to the estimates of μ and δ, see [2].

In our numerical experiments we use the SSOR preconditioning with the CG method and also with the Barzilai and Borwein method. We approximate the value ω^* with a simple scheme that is closely related to our model problem and is based on numerical experience.

7.4 Model Problem

Consider the elliptic partial differential equation

$$-\left(u_{xx}+u_{yy}\right)+\alpha u=f \tag{7.6}$$

where $\alpha \geq 0$ is a real scalar parameter, and the function f is given.

For the model problem, we pose (7.6) on the unit square $0 \leq x \leq 1$, $0 \leq y \leq 1$, with homogeneous Dirichlet boundary conditions. We seek a function u that is continuous on the unit square, satisfies (7.6) in the interior of the unit square, and equals zero on the boundary.

We discretize (7.6) using the five-point centered finite difference scheme [14] on a uniform mxm grid with $h = 1/(m+1)$ and the natural ordering, producing a linear system

$$Ax = b \tag{7.7}$$

of order $n = m^2$. The matrix A has the following block tridiagonal form

$$A = \begin{pmatrix} D & -I & & & \\ -I & D & -I & & \\ & . & . & . & \\ & & -I & D & -I \\ & & & -I & D \end{pmatrix},$$

where D is a square and symmetric tridiagonal matrix of order m, and there are exactly m blocks on the diagonal of A.

The diagonal elements of D have the value $(4+\alpha)$ and the subdiagonal elements have the value -1. Since α is a nonnegative real number, the matrix A is symmetric and positive definite. Notice that for any vector $z \epsilon \mathbb{R}^n$, the cost of the matrix-vector multiplication Az is approximately $5n$.

The i^{th} position of the vector b in (7.7) equals $h^2 f_i$ where f_i is the evaluation of the function f at the i^{th} node of the grid in the natural ordering. Finally, notice that for $\alpha = 0$ our model problem is the classical Dirichlet problem.

7.5 Description of Experiments and Numerical Results

In this section we describe results of numerical experiments that were designed to test the effectiveness of the preconditioned Barzilai and Borwein method as compared with the preconditioned CG method.

All experiments in this section were run on a CONVEX C120 in single precision FORTRAN with the vectorization option, and the optimized BLAS (Basic Linear Algebra Subroutines [4]).

Consider the model problem described in Section 7.4. We ran an implementation of Algorithms 7.1 and 7.2 for different values of the parameter α and different values of the dimension n (i.e., different step sizes h). We used the SSOR preconditioning matrix C defined in Section 7.3 for both algorithms. The parameter ω associated with the matrix C was chosen in the following way:

$$\omega = \begin{cases} 2/(1+0.6\alpha+2.6h) & if \quad 0 \leq \alpha \leq 1 \\ 1+(3\alpha)^{-1} & if \quad \alpha > 1\,. \end{cases}$$

This choice of ω has been discussed by Axelsson and Barker [2] for $\alpha = 0$. We extended the choice of ω for $\alpha > 0$ based on numerical experience.

The function f in (7.6) is defined to have the constant value $1/h^2$, so that the vector b in (7.7) has the value 1 at every position.

We started the process in both algorithms at $x_0 = (0,0,\ldots 0)^t$ and we set $\alpha_0 = 2$ in Algorithm 7.1. For this particular choice of x_0, the initial gradient g_0 equals b and hence $\|g_0\|_2 = 1$. We stopped the process whenever $\|g_k\|_2 \leq 10^{-8}$.

In our first experiment we fix the parameter $\alpha = 0$ and study the number of iterations and the computational work required for both methods and different values of n. Figure 7.1 shows the number of iterations required for the preconditioned Barzilai and Borwein and the preconditioned CG for different values of n when they are applied to the classical Dirichlet problem (i.e., $\alpha = 0$). We can see that for both algorithms the number of iterations increases as n increases. However, the number of iterations required using the preconditioned CG is approximately 30% less than the number required using the preconditioned Barzilai and Borwein. This is because

$\kappa(A)$ in 7.7 increases rapidly as n increases. Moreover, the smallest eigenvalue of A is extremely close to zero for large n, and this represents a negative effect on the stability of the Barzilai and Borwein method. In fact, we have observed for this particular problem that the convergence deteriorates when the scalar α_k approaches the smallest eigenvalue.

The second experiment was chosen to demonstrate the effect of increasing the parameter α on the effectiveness of the preconditioned Barzilai and Borwein method as compared with the preconditioned CG method. Figure 7.2 shows the number of iterations required for both methods when they are applied to the model problem with the dimension fixed at $n = 10^6$. It can be seen that the number of iterations required by the preconditioned CG is smaller than the number required by the preconditoned Barzilai and Borwein for very small values of α. However, when $\alpha \geq 0.5$ the two methods require exactly the same number of iterations. It may also be observed that the number of iterations decreases in both methods as α increases. This is a consequence of the decrease in the spectral condition number of A and the fact that the smallest eigenvalue of A is bounded away from zero.

We also study the effect of increasing the parameter α on the computational work required by the two methods. Figure 7.3 shows the number of multiplications required when the two methods are applied to the model problem with different values of α and $n = 10^6$. It may be observed that for α small the preconditioned CG requires less computational work than the preconditioned Barzilai and Borwein. However, for $\alpha \geq 0.35$ we observe the opposite result. In that case, the preconditioned Barzilai and Borwein requires less computational work than the preconditioned CG method. The difference in the required work is approximately 10% when $\alpha \geq 0.4$, which is not in general a significant improvement. However, in this particular case, it should be a surprise because the Barzilai and Borwein method is the gradient method with a different choice of steplength, and the classical gradient method is never expected to be competitive with the CG method.

In conclusion, our preliminary numerical results seem to indicate that the preconditioned Barzilai and Borwein method is effective as compared to the preconditioned

CG method for the numerical solution of large sparse SPD linear systems of equations $Ax = b$, whenever A is not too ill-conditioned and the smallest eigenvalue of A is not too close to zero. However, when the matrix A is ill-conditioned, the preconditioned CG method is clearly a better option.

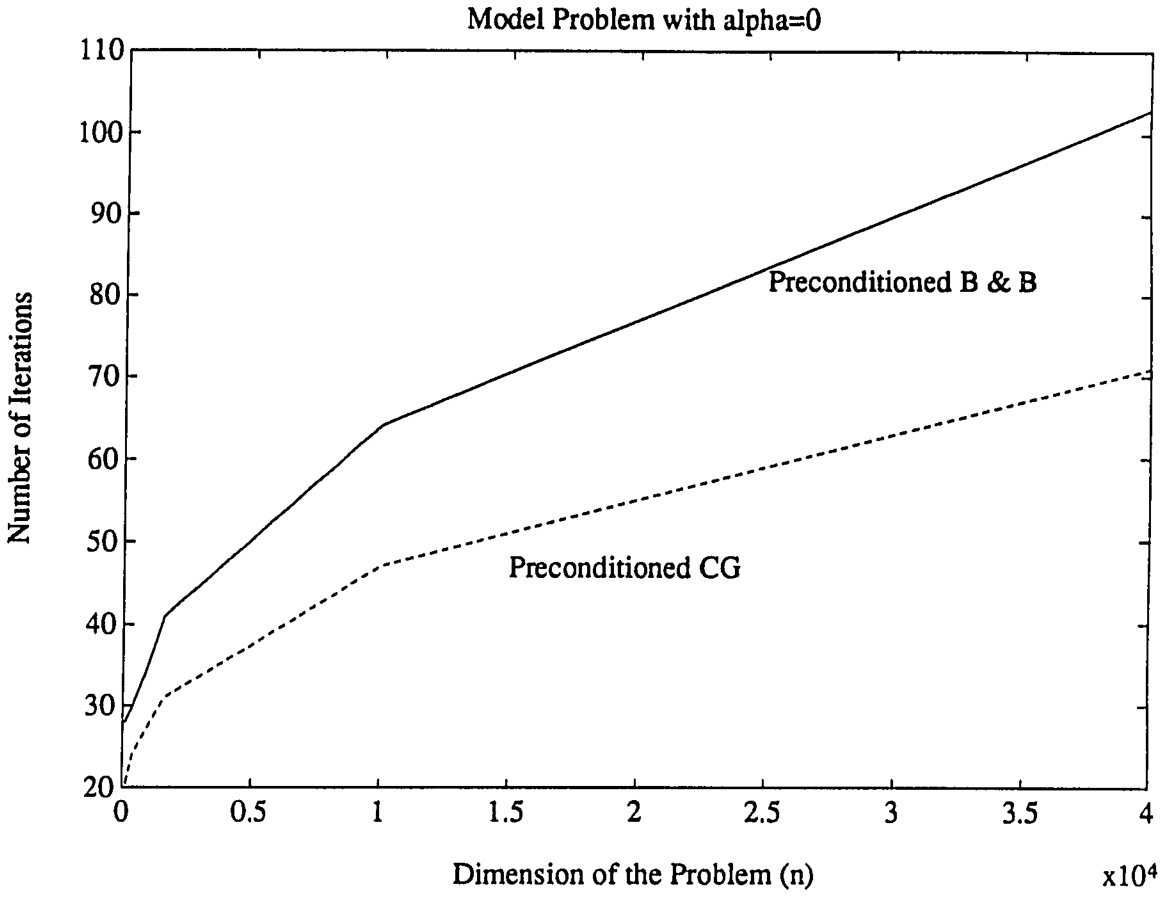

Figure 7.1 Number of iterations required for the preconditioned Barzilai and Borwein and the preconditioned Conjugate Gradient to achieve $\|g_k\|_2 \leq 10^{-8}$ when $\alpha = 0$.

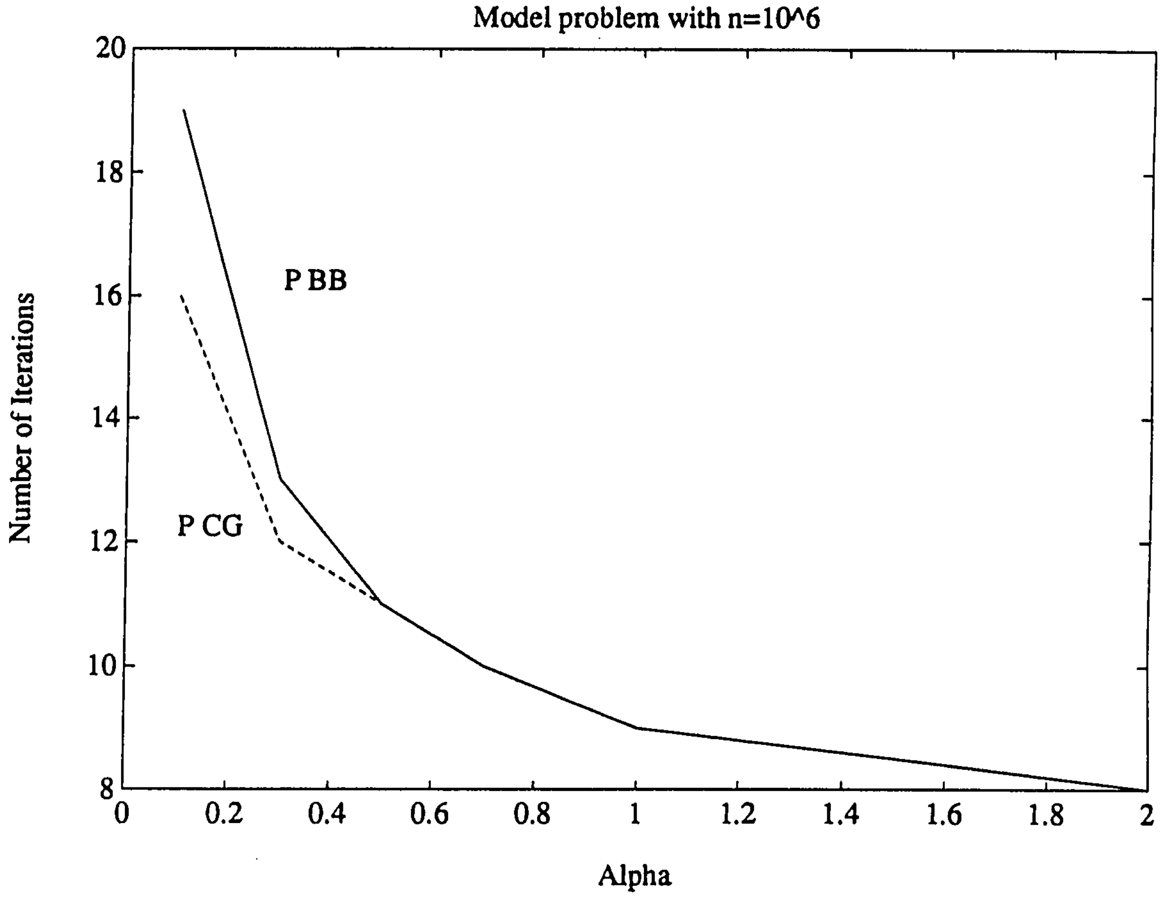

Figure 7.2 Number of iterations required for the preconditioned Barzilai and Borwein and the preconditioned CG to achieve $\|g_k\|_2 \leq 10^{-8}$ for different values of α and $n = 10^6$.

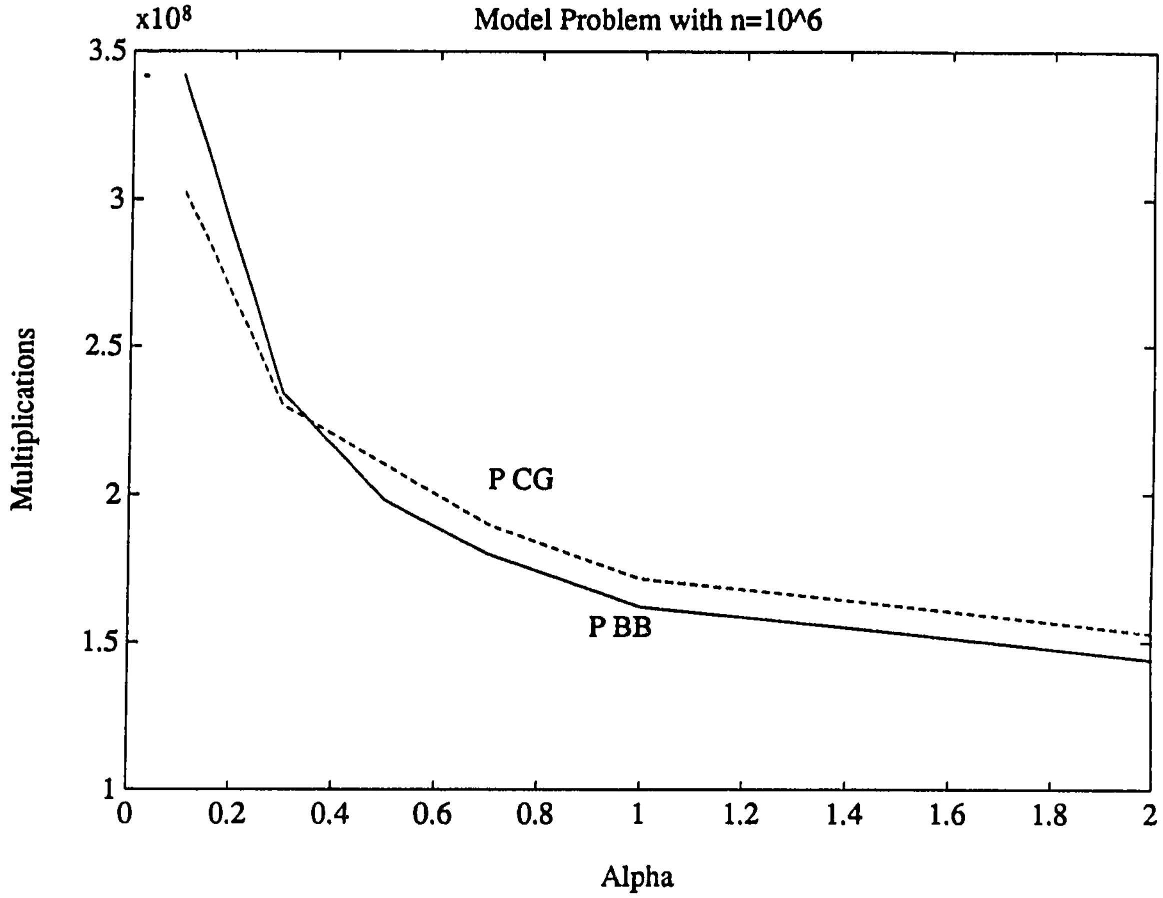

Figure 7.3 Computational work (Number of multiplications) required for the preconditioned Barzilài and Borwein and the preconditioned CG for different values of α and $n = 10^6$.

Chapter 8

Concluding Remarks

By establishing a relationship with the shifted power method, we have added significant understanding to the convergence of the Barzilai and Borwein method applied to the minimization of a quadratic function with a SPD Hessian. We have demonstrated the convergence of the method, in this case, by considering two ingredients: the resemblance to the shifted power method and the property that the Rayleigh quotient approximates eigenvalues within the spectrum of the Hessian.

For the 2-dimensional case, Barzilai and Borwein [3] have indicated a $\sqrt{2}$ R-convergence rate when the method starts in a particular manner. On the other hand, our analysis shows at least (3,2)-step Q-quadratic convergence and exactly $\sqrt[5]{4}$ R-convergence rate. Both results are obtained from a "worst case" analysis. We stress that the R-rate of convergence was established as a consequence of the recent results of Potra [12] on this topic. Previous results on R-rate of convergence, in the literature, would not allow the sequence $\{a_k\}$ defined in (5.20) to approach zero. Thus, we would not have been able to prove the R-rate of convergence. Numerical experiments have been included to illustrate the interesting Q and R-convergence behavior of the method for this particular case.

We have studied the applicability of the Barzilai and Borwein method to the iterative solution of large sparse SPD linear systems, that arise from the numerical solution of PDE. In particular, we have presented preliminary numerical results for an elliptic model problem that are quite promising. These results indicate that the Barzilai and Borwein method is competitive with the Conjugate Gradient method when the Hessian matrix has been preconditioned in such a way, that either the condition number is reasonable or the eigenvalues are clustered. We stress that in our experiments we choose the parameters involved in the preconditioning techniques to be suitable for the Conjugate Gradient method. We would like to study the possibility of finding suitable preconditioning techniques for the Barzilai and Borwein method, and also the possibility of using this new method as a preconditioning technique. In a recent paper, Fletcher [7] considers incorporating the Barzilai and Borwein method into a new approach for the large scale unconstrained minimization problem. This approach can be viewed as a preconditioning technique for the Barzilai and Borwein method. He also indicates a number of potentially interesting ideas that deserve further investigation.

We would also like to extend our analysis to establish local convergence of the method when applied to the general smooth unconstrained minimization problem. None of the Conjugate Gradient methods (Fletcher-Reeves, Polak-Ribiere,etc., see Fletcher [6]) has proved to be completely succesful for this problem. In fact, the finite termination of Conjugate Gradient is lost, and the convergence is not expected to be better than Q-linear (Powell [13]). On the other hand, the Barzilai and Borwein method requires no line search and so, near the solution, it requires considerably less computational effort than any of the Conjugate Gradient methods.

The question of how to embed the Barzilai and Borwein method in a globalization framework now arises. The goal is to find a globalization scheme that preserves the "inexpensive" fast local convergence of the method, and only requires storage of first order information during the process. This is an interesting challenge that deserves further numerical and theoretical investigation.

Bibliography

[1] O. AXELSSON. A survey of preconditioned iterative methods for linear systems of algebraic equations. *Bit*, 25:166–187, 1985.

[2] O. AXELSSON and V.A. BARKER. *Finite Element Solution of Boundary Value Problems, Theory and Computation.* Academic Press, New York, 1984.

[3] J. BARZILAI and J.M. BORWEIN. Two point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[4] D.R. KINCAID C.L. LAWSON, R.J. HANSON and F.T. KROGH. Basic linear algebra subprograms for Fortran usage. *ACMTOMS*, 5:308–323, 1979.

[5] J.E. DENNIS Jr. and R.B. SCHNABEL. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, NJ, 1983.

[6] R. FLETCHER. *Practical Methods of Optimization.* Wiley, New York, 1987.

[7] R. FLETCHER. Low storage methods for unconstrained optimization. *Lectures in Applied Mathematics (AMS)*, 26:165–179, 1990.

[8] G.H. GOLUB and D.P. O'LEARY. Some history of the conjugate gradient and Lanczos methods. *SIAM Review*, 31:50–102, 1989.

[9] L.A.HAGEMAN and D.M. YOUNG. *Applied Iterative Methods.* Academic Press, New York, 1981.

[10] B. NOBLE and J. DANIEL. *Applied Linear Algebra.* Prentice-Hall, Englewood Cliffs, NJ, 1977. Second edition.

[11] J.M. ORTEGA and W.C. RHEINBOLDT. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, 1970.

[12] F. POTRA. On Q-Order and R-Order of convergence. *J. Optim. Theory Appl.*, 63:415–431, 1989.

[13] M.J.D. POWELL. Some convergence properties of the conjugate gradient method. *Mathematical Programming*, 11:42–49, 1976.

[14] G.D. SMITH. *Numerical Solution of Partial Differential Equations: Finite Difference Methods.* Oxford University Press, New York, 1978.

[15] D.M. YOUNG. *Iterative Solution of Large Linear Systems.* Academic Press, New York, 1971.

[16] D.M. YOUNG. A historical overview of iterative methods. *Computer Physics Communications*, 53:1–17, 89.