On the Barzilai-Borwein method

Roger Fletcher (fletcher@maths.dundee.ac.uk)

Department of Mathematics, University of Dundee, Dundee DD1 4HN, Scotland, UK.

Numerical Analysis Report NA/207, October 2001

Abstract

A review is given of the underlying theory and recent developments in regard to the Barzilai-Borwein steepest descent method for large scale unconstrained optimization. One aim is to assess why the method seems to be comparable in practical efficiency to conjugate gradient methods. The importance of using a non-monotone line search is stressed, although some suggestions are made as to why the modification proposed by Raydan [22] often does not usually perform well for an ill-conditioned problem. Extensions for box constraints are discussed. A number of interesting open questions are put forward.

Keywords Barzilai-Borwein method, steepest descent, elliptic systems, unconstrained optimization.

1 Introduction

The context of this paper is the solution of the unconstrained minimization problem

$$\text{minimize} \quad f(\mathbf{x}) \qquad \mathbf{x} \in \mathbb{R}^n \tag{1.1}$$

where the number of variables n is very large, typically 10^6 or so. The case of minimization subject to simple bounds is also considered later in the paper. A related problem is that of the solution of a nonlinear self-adjoint elliptic system of equations

$$\mathbf{g}(\mathbf{x}) = \mathbf{0}, \tag{1.2}$$

in which $\mathbf{g} = \nabla f$ is the gradient of some variational function. The case in which $f(\mathbf{x})$ is a strictly convex quadratic function

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x} \tag{1.3}$$

is also studied, which is equivalent to the solution of the linear system of equations

$$A\mathbf{x} = \mathbf{b} \tag{1.4}$$

in which A is a positive definite symmetric matrix. This is referred to as the *quadratic case* and is important, not only as a model problem to analyse properties of methods, but also in its own right as a problem that becomes difficult to solve for large n, when (sparse) Choleski factorisation is impractical due to lack of time or storage capacity. It is the large scale situation (both for quadratic or non-quadratic problems) that we study in this paper.

The methods that we study are all iterative methods, since in the quadratic case we cannot expect to be able to carry out enough iterations to obtain an exact solution, even if the theory allows this possibility, due to the size of n or the build up of round-off error. Early methods such as relaxation methods or SOR have mostly been supplanted by use of the conjugate gradient method or variations thereof. For the quadratic case the conjugate gradient (CG) method itself (Hestenes and Stiefel, [16]), or some preconditioned conjugate gradient (PCG) method (see for example Golub and Van Loan, [14]), is usually the method of choice, although there are other variants such as the minimum residual (MR) algorithm that are also applicable to the case that A is symmetric and indefinite. A particular feature of these methods is that they terminate in at most n iterations. This is not particularly exciting when n is large, but Reid [23] gives reasons why the methods are effective as iterative methods in that they are able to deliver a reasonably accurate estimate of the solution in substantially fewer than n iterations, particularly if A has a favourable eigenvalue structure. The CG method is particularly attractive because it only requires $4n$ storage locations for its implementation. PCG methods need to store and solve linear systems with some matrix that approximates A and makes tolerable demands on time and storage.

For non-quadratic systems there are various methods of line search type that are based on using the search direction formula of the CG iteration. The simplest methods are those of Fletcher and Reeves [9] ($3n$ locations) and Polak and Ribière [19] ($4n$ locations), the latter being more usually preferred in practice. These can also be preconditioned in a manner similar to the quadratic case. Then there are also methods that use rather more storage, such as CONMIN (Shanno and Phua [20]) ($7n$ locations), the Limited Memory BFGS method (Nocedal [18]), ($9n+$ locations), the Truncated Newton method (Dembo, Eisenstat and Steihaug [7]), and many others.

Amongst all of these, steepest descent methods hardly rate a mention in a modern text-book on optimization, even though the storage requirements are minimal ($3n$ locations). Indeed, the 'classical' steepest descent method with exact line seach (Cauchy, [4]) is known to behave increasingly badly in the quadratic case as the condition number of A deteriorates. Early attempts to modify the method led to the introduction of CG methods, with much superior performance.

In 1988, a paper by Barzilai and Borwein [2] proposed a steepest descent method (the BB method) that uses a different strategy for choosing the step length. The main result of the paper is to show the surprising result that for $n = 2$, the method converges R-superlinearly. Barzilai and Borwein also show that their method is considerably superior to the classical steepest descent method for one instance of a quadratic function with

$n = 4$, but no other numerical results are given. Fletcher ([8], 1990) investigates some connections with the spectrum of A in the quadratic case, and an ingenious proof by Raydan ([21], 1993) demonstrates convergence in the quadratic case. However, neither of these papers gives any numerical results and the method attracted little attention until a seminal paper of Raydan ([22], 1997). This paper introduces a globalization strategy based on the non-monotone line search technique of Grippo, Lampariello and Lucidi [15], which enables global convergence of the BB method to be established for non-quadratic functions. Of equal importance, a wide range of numerical experience is reported on problems of up to 10^4 variables, showing that the method compares reasonably well against the Polak-Ribière and CONMIN techniques. Earlier papers by Glunt, Hayden and Raydan [12], [13], also report promising numerical results on a distance matrix problem. The paper [13] reports on the possibilities for preconditioning the BB method, and this theme is also taken up by Molina and Raydan [17]. Of particular interest is the possibility of applying the BB method to box-constrained optimization problems, and this is considered by Friedlander, Martínez and Raydan [10] (for quadratic functions) and Birgin, Martínez and Raydan [3]. The latter paper considers the BB method in the context of projection on to a convex set. Another recent theoretical development has been the result that the unmodified BB method is R-linearly convergent in the quadratic case (Dai and Liao [6]).

Despite all these advances, there is still much to be learned about the BB method and its modifications. This paper reviews what is known about the method, and advances some reasons that partially explain why the method is competitive with CG based methods. The importance of maintaining the non-monotonicity property of the basic method is stressed. It is argued that the use of the line search technique of Grippo, Lampariello and Lucidi [15] in the manner proposed by Raydan [22] may not be the best way of globalizing the BB method, and some tentative alternatives are suggested. Some other interesting observations about the distribution of the BB steplengths are also made. Many open questions still remain about the BB method and its potential, and these are discussed towards the end of the paper.

2 The BB Method for Quadratic Functions

The theory and practice of line search methods for minimizing $f(\mathbf{x})$ has been well explored. In such a method, a search direction $\mathbf{s}^{(k)}$ is chosen at the start of iteration k, and a step length θ_k is chosen to (approximately) minimize $f(\mathbf{x}^{(k)} + \theta\mathbf{s}^{(k)})$ with respect to θ. Then $\mathbf{x}^{(k+1} = \mathbf{x}^{(k)} + \theta_k\mathbf{s}^{(k)}$ is set. Usually $\mathbf{s}^{(k)T}\mathbf{g}^{(k)} < 0$ for all k (a descent method) and it is possible to guarantee that the method is *monotonic* in the sense that the sequence $\{f^{(k)}\}$ is strictly monotonically decreasing unless a stationary point is exactly located. The classical steepest descent method belongs to this class, with $\mathbf{s}^{(k)} = -\mathbf{g}^{(k)}$. CG methods have $\mathbf{s}^{(1)} = -\mathbf{g}^{(1)}$ and $\mathbf{s}^{(k)} = -\mathbf{g}^{(k)} + \beta_k\mathbf{s}^{(k-1)}$ for $k > 1$, where $\beta_k = \mathbf{g}^{(k)T}\mathbf{g}^{(k)}/\mathbf{g}^{(k-1)T}\mathbf{g}^{(k-1)}$ in the Fletcher-Reeves (FR) method, and $\beta_k = \mathbf{g}^{(k)T}(\mathbf{g}^{(k)} - \mathbf{g}^{(k-1)})/\mathbf{g}^{(k-1)T}\mathbf{g}^{(k-1)}$ in the (Polak-Ribière) (PR) method. When $f(\mathbf{x})$ is the quadratic function (1.3), the step θ_k

is readily calculated from the expression $\theta_k = -\mathbf{s}^{(k)T}\mathbf{g}^{(k)}/\mathbf{s}^{(k)T}A\mathbf{s}^{(k)}$. For non-quadratic functions it is in general only possible to find an approximate solution of the line search problem, and for CG methods it seems better if the solution is reasonably accurate.

In contrast, the BB method is a fixed step gradient method, which we choose to write in the form

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{d}^{(k)} \qquad \text{where} \qquad \mathbf{d}^{(k)} = -\mathbf{g}^{(k)}/\alpha_k. \tag{2.1}$$

Initially, $\alpha_1 > 0$ is arbitrary, and Barzilai and Borwein give two alternative formulae,

$$\alpha_k = \mathbf{d}^{(k-1)T}\mathbf{y}^{(k-1)}/\mathbf{d}^{(k-1)T}\mathbf{d}^{(k-1)} \tag{2.2}$$

and

$$\alpha_k = \mathbf{y}^{(k-1)T}\mathbf{y}^{(k-1)}/\mathbf{y}^{(k-1)T}\mathbf{d}^{(k-1)}, \tag{2.3}$$

for $k > 1$, where we denote $\mathbf{y}^{(k-1)} = \mathbf{g}^{(k)} - \mathbf{g}^{(k-1)}$. In fact, attention has largely been focussed on (2.2) and it is this formula that is discussed here, although there seems to be some evidence that the properties of (2.3) are not all that dissimilar.

In the rest of this section, we explore the properties of the BB method and other gradient methods for minimizing a strictly convex quadratic function. For the BB method, (2.2) can be expressed in the form

$$\alpha_k = \mathbf{g}^{(k-1)T}A\mathbf{g}^{(k-1)}/\mathbf{g}^{(k-1)T}\mathbf{g}^{(k-1)} \tag{2.4}$$

and can be regarded as a Rayleigh quotient, calculated from the previous gradient vector. This is in contrast to the classical steepest descent method which is equivalent to using a similar formula, but with $\mathbf{g}^{(k-1)}$ replaced by $\mathbf{g}^{(k)}$. Another relevant property, possessed by all gradient methods, and also the conjugate gradient method, is that

$$\mathbf{x}^{(k+1)} - \mathbf{x}^{(1)} \in \text{span}\{\mathbf{g}^{(1)}, A\mathbf{g}^{(1)}, A^2\mathbf{g}^{(1)}, \ldots A^{k-1}\mathbf{g}^{(1)}\}. \tag{2.5}$$

That is to say, the total step lies in the span of the so-called *Krylov sequence*. Also for quadratic functions, the BB method has been shown to converge (Raydan, [21]), and convergence is R-linear (Dai and Liao, [6]). However the sequences $\{f(\mathbf{x}^{(k)})\}$ and $\{\|\mathbf{g}(\mathbf{x}^{(k)})\|_2\}$ are non-monotonic, an explanation of which is given below, and no realistic estimate of the R-linear rate is known. However the case $n = 2$ is special, and it is shown in [2] that the rate of convergence is R-superlinear.

To analyse the convergence of any gradient method for a quadratic function, we can assume without loss of generality that an orthogonal transformation is made that transforms A to a diagonal matrix of eigenvalues $\text{diag}(\lambda_i)$. Moreover, if there are any eigenvalues of multiplicity $m > 1$, then we can choose the corresponding eigenvectors so that $g_i^{(1)} = 0$ for at least $m - 1$ corresponding indices of $\mathbf{g}^{(1)}$. It follows from (2.1)

and the properties of a quadratic function that $\mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} - A\mathbf{g}^{(k)}/\alpha_k$ and hence using $A = \mathrm{diag}(\lambda_i)$ that

$$g_i^{(k+1)} = \left(1 - \frac{\lambda_i}{\alpha_k}\right) g_i^{(k)} \qquad i = 1, 2, \ldots, n. \tag{2.6}$$

It is clear from this recurrence that if $g_i^{(k)} = 0$ for any i and $k = k'$, then this property will persist for all $k > k'$. Thus, without any loss of generality, we can assume that A has distinct eigenvalues

$$0 < \lambda_1 < \lambda_2 < \ldots < \lambda_n, \tag{2.7}$$

and that $g_i^{(1)} \neq 0$ for all $i = 1, 2, \ldots, n$.

Many things can be deduced from these conditions and (2.6). First, if α_k is equal to any eigenvalue λ_i, then $g_i^{(k+1)} = 0$ and this property persists subsequently. If both

$$g_1^{(k-1)} \neq 0 \qquad \text{and} \qquad g_n^{(k-1)} \neq 0 \tag{2.8}$$

then it follows from (2.4) and the extremal properties of the Rayleigh quotient that

$$\lambda_1 < \alpha_k < \lambda_n. \tag{2.9}$$

Thus, for the BB method, and assuming that α_1 is not equal to λ_1 or λ_n, then a simple inductive argument shows that (2.8) and (2.9) hold for all $k > 1$. It follows, for example, that the BB method does not have the property of finite termination.

From (2.6), it follows for any eigenvalue λ_i close to α_k that $|g_i^{(k+1)}| \ll |g_i^{(k)}|$. It also follows that the values $|g_1^{(k)}|$ are monotonically decreasing. However, if on any iteration $\alpha_k < \frac{1}{2}\lambda_n$, then $|g_n^{(k+1)}| > |g_n^{(k)}|$ and if α_k is close to λ_1 then the ratio of $|g_n^{(k+1)}/g_n^{(k)}|$ can approach $\lambda_n/\lambda_1 - 1$. Thus we see the potential for non-monotonic behaviour in the sequences $\{f(\mathbf{x}^{(k)})\}$ and $\{\|\mathbf{g}(\mathbf{x}^{(k)})\|_2\}$, and the extent of the non-monotonicity depends in some way on the size of the condition number of A. On the other hand, if α_k is close to λ_n then all the coefficients g_i decrease in modulus, but the change in g_1 is negligible if the condition number is large. Moreover, small values of α_k tend to diminish the components $|g_i|$ for small i and hence enhance the relative contribution of components for large i. This in turn leads to large values of α_k on a subsequent iteration, if the step is calculated from (2.4). Thus, in the BB method, we see values of α_k being selected from all parts of the interior of the spectrum, with no apparent pattern, with jumps in the values of $f(\mathbf{x}^{(k)})$ and $\|\mathbf{g}(\mathbf{x}^{(k)})\|_2$ occurring when α_k is small.

There are a number of reasons that might lead one to doubt whether the BB method could be effective in practice. Although a nice convergence proof is given by Raydan [21], we have to recognise the fact that although both the CG and BB methods select iterates that satisfy the Krylov sequence property (2.5), it is the CG method that gives the minimum possible value of $f(\mathbf{x}^{(k+1)})$. Likewise the Minimum Residual (MR) method

gives the minimum possible value of $\|\mathbf{g}^{(k+1)}\|_2$. Thus we must accept that the BB method is necessarily inferior in regard to these measures in exact arithmetic, and there is limited scope for the BB method to improve as regards elapsed time, for example. Also the possibility of non-monotonic behaviour of the BB might seem to give further reason to prefer the CG method.

To see just how inferior the BB method is, a large scale test problem is devised, based on the solution of an elliptic system of linear equations arising from a 3D Laplacian on a box, discretized using a standard 7-point finite difference stencil. Thus we define the matrices

$$T = \begin{bmatrix} 6 & -1 & & & \\ -1 & 6 & -1 & & \\ & -1 & 6 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 6 \end{bmatrix}, \qquad W = \begin{bmatrix} T & -I & & & \\ -I & T & -I & & \\ & -I & T & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & T \end{bmatrix}$$

and

$$A = \begin{bmatrix} W & -I & & & \\ -I & W & -I & & \\ & -I & W & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & W \end{bmatrix} \tag{2.10}$$

where T is $l \times l$, W is block $m \times m$ and A is block $n \times n$ where l, m, n are the number of interior nodes in each coordinate direction, and are specified by the user. The interval length is taken to be $h = 1/(l+1)$ and is the same in each direction. Hence the dimensions of the box are $1 \times Y \times Z$ where $Y = (m+1)h$ and $Z = (n+1)h$. We fix the solution to the problem to be function

$$u(x,y,z) = x(x-1)y(y-Y)z(z-Z)\exp(-\tfrac{1}{2}\sigma^2((x-\alpha)^2 + (y-\beta)^2 + (z-\gamma)^2), \tag{2.11}$$

evaluated at the nodal points. This function is a Gaussian centered on the point (α, β, γ), multiplied by quadratics $x(x-1)$ etc., that give $u = 0$ on the boundary. The parameter σ controls the rate of decay of the Gaussian. The problem has lmn variables, and we denote $\mathbf{u}^*$ to be the solution derived from (2.11) and calculate the right hand side $\mathbf{b} = A\mathbf{u}^*$. In our tests we choose $l = m = n = 100$ giving a problem with 10^6 variables, in which case the condition number of A is $\lambda_n/\lambda_1 = 4133.6 = 10^{3.61}$. We also choose $\mathbf{u}^{(1)} = \mathbf{0}$. We denote the resulting test problem to be `Laplace1` and choose the parameters in two different ways, that is

(a) $\sigma = 20, \quad \alpha = \beta = \gamma = 0.5$ (b) $\sigma = 50, \ \alpha = 0.4, \ \beta = 0.7, \ \gamma = 0.5.$

The problem `Laplace1(a)` has the centre of the Gaussian in the centre of the box, giving the problem a high degree of symmetry. Also the smaller value of σ gives a smoother solution. Hence this problem is more easy to solve than `Laplace1(b)`.

The results for this problem are given in Table 1 below. The CG method is coded as recommended by Reid [23]. Times are given in seconds and double precision Fortran is used on a SUN Ultra 10 at 440 MHz. The iteration is terminated when $\|\mathbf{g}^{(k)}\|_2$ is less than 10^{-6} of its initial value. We see from the table that there is little to choose between

	BB		CG		MR	
Problem	Time	Iterations	Time	Iterations	Time	Iterations
Laplace1(a)	543	859	162	178	179	171
Laplace1(b)	640	1009	285	306	302	290

Table 1: Double length comparison (quadratic case)

the CG and MR methods, and the elapsed time improves on the BB method by a factor of over 3 for Laplace1(a) and a factor of over 2 for Laplace1(b). For comparsion purposes, the classical steepest descent method was manually terminated after 2000 iterations (1355 seconds), by which time it had only reduced the initial gradient norm by a factor of 0.18, so that not even one significant figure improvement had been obtained. Thus we see that, although the performance of the BB does not quite match that of the CG method, it is able to solve the problem in reasonable time, and significantly improves on the classical steepest descent method.

Nonetheless, in view of the above, we might ask if there are any circumstances under which the BB method might be worth considering as an alternative to the CG method. The answer lies in the fact that the success of the CG iteration depends very much on the search direction calculation $\mathbf{s}^{(k)} = -\mathbf{g}^{(k)} + \beta_k \mathbf{s}^{(k-1)}$ being consistent with data arising from a quadratic model. Any deviation from the quadratic model can seriously degrade performance. To illustrate that relatively small perturbations can cause this to happen, we repeat the calculations of Table 1 using single precision arithmetic. The results are displayed in Table 2.

	BB		CG		MR	
Problem	Time	Iterations	Time	Iterations	Time	Iterations
Laplace1(a)	462	964	254	387	340	448
Laplace1(b)	310	645	290	443	397	523

Table 2: Single length comparison (quadratic case)

We see that the CG and MR methods now take more than twice as many iterations for Laplace1(a), with a similar, but not quite as bad, outcome for Laplace1(b). The comparison in time is less marked, presumably because of the cost savings associated with using single rather than double precision. For the BB method a different picture emerges. For Laplace1(a), somewhat more iterations are required, whereas for Laplace1(b), considerably fewer iterations are required. Again the time comparison is improved by using single precision, to such an extent that there is now little difference between the performance

of the BB and CG methods on the Laplace1(b) problem. My interpretation of this is that the BB method is affected in a much more random way by round off errors, and small departures of $\mathbf{g}^{(k)}$ and α_k from the values arising from a quadratic problem are not necessarily detrimental.

This has implications for the likely success of the BB method in other contexts. For example, if $f(\mathbf{x})$ is made up of a quadratic function plus a small non-quadratic term, we might expect the BB method to still converge, and show improved performance relative to the CG method. Another situation is in the minimization of a quadratic function subject to simple bounds by an active set or projection type of method. If the number of active constraints changes, as is often the case, then it is usually not possible to continue to use the standard CG formula for the search direction and yet preserve the termination and optimality properties. To do this it is necessary to restart using the steepest descent direction when a new active set is obtained. Thus it is more attractive to use the BB method in some way in this situation.

3 The BB Method for Non-quadratic Functions

If the deviation of $f(\mathbf{x})$ from a quadratic function is small then it may still be possible to use the unmodified BB method successfully. However, in general it is possible for the method to diverge. This is illustrated by using the test problem referred to as `Strictly Convex 2` by Raydan [22], in which $f(\mathbf{x})$ is defined by

$$f(\mathbf{x}) = \sum_{i=1}^{n} \tfrac{1}{10} i \left(e^{x_i} - x_i\right). \tag{3.1}$$

The initial point is $\mathbf{x}^{(1)} = (1,\, 1,\, \ldots, 1)^T$ and the solution is $\mathbf{x}^* = \mathbf{0}$. The Hessian matrix at $\mathbf{x}^*$ is $\frac{1}{10}\,\mathrm{diag}((1,\, 2,\, \ldots, n)$ so that the condition number is n. It is readily verified that the unmodified BB method converges for $n = 20$ but diverges for $n = 30$, even though (3.1) is a strictly convex function and has a positive definite Hessian matrix for all $\mathbf{x}$.

It is therefore necessary to modify the BB method if it is to be used as a general purpose solver for non-quadratic problems. An important contribution is that of Raydan [22] who suggests using the non-monotonic line search of Grippo, Lampariello and Lucidi [15]. In Raydan's method (the BB-Raydan method, say) the initial estimate of the line search step is $\theta = \alpha_k^{-1}$, with some adjustment if α_k^{-1} turns out to be unreasonably large or small (and θ is required to be positive). An Armijo-type line search on θ is then carried out until the acceptance condition

$$f(\mathbf{x}^{(k)} + \mathbf{d}) \leq \max_{\max(k-M,1)\leq j\leq k} f^{(j)} - \gamma \mathbf{g}^{(k)T}\mathbf{d} \tag{3.2}$$

is met, where $\mathbf{d} = -\theta\mathbf{g}^{(k)}$ is the displacement along the steepest descent direction. This allows any point to be accepted if it improves sufficiently on the largest of the $M+1$ (or k if $k \leq M$) most recent function values. As usual $\gamma > 0$ is a small preset constant, and the

integer M controls the amount of monotonicity that is allowed. Raydan recommends the value $M = 10$ and presents a lot of encouraging numerical evidence on test problems with up to $n = 10^4$ variables. His results are competitive with CG methods but he observes some poorer results on ill-conditioned problems.

To obtain more insight, a non-quadratic test problem of 10^6 variables is derived, based on a 3D Laplacian, in which the objective function is

$$\tfrac{1}{2}\mathbf{u}^T A\mathbf{u} - \mathbf{b}^T\mathbf{u} + \tfrac{1}{4}h^2 \sum_{ijk} u_{ijk}^4, \tag{3.3}$$

which is not untypical of what might arise from a nonlinear partial differential equation. This problem is referred to as `Laplace2`. The matrix A is that defined in (2.10), and the vector $\mathbf{b}$ is chosen so that the minimizer $\mathbf{u}^*$ of (3.3) is the function $u(x, y, z)$ in (2.11), evaluated at the discretization points. The non-quadratic term in (3.3) includes a factor h^2, and $0 \le u_{ijk} < 1$ which also makes the u_{ijk}^4 term small. Thus the relative contribution of the non-quadratic term is small, and as we shall see, the unmodified BB method is able to solve instances of this problem.

The progress of various methods for solving Laplace2(b) is given in Table 3. The

	5 figures				6 figures			
Problem	Time	#ls	#f	#g	Time	#ls	#f	#g
Polak-Ribière CG	20.6	445	697	684	∞			
BB-Raydan M=10	29.0	274	1140	866	40.4	394	1595	1201
Unmodified BB	14.4	-	487	487	16.7	-	572	572
Limited memory BFGS	35.4	315	711	669	∞			
BB method ($\mathbf{g}$ only)	8.8	-	-	487	10.3	-	-	572

Table 3: Time (minutes) and numbers of evaluations to solve Laplace2(b)

columns headed `#ls`, `#f` and `#g` give the numbers of line searches, function evaluations and gradient evaluations required to solve the problem. The calculations are carried out in double precision Fortran and the non-BB methods both use the same line search based on standard strong Wolfe conditions with a relative slope tolerance of 0.1. For the BB-Raydan method the column `#ls` gives the number of occasions on which the Armijo line search is used. In the limited memory method, 3 back pairs of vectors are stored. Initially an accuracy tolerance of better than 10^{-6} of the initial gradient norm was required (the column headed '6 figures' in the table) but only the BB methods were able to find the solution to this accuracy. Therefore the comparison is carried out on the basis of 5 figure accuracy (10^{-5} of the initial gradient norm required).

It can be seen that here the unmodified BB method actually improves on the PR-CG method, but that this improvement is not maintained for the BB-Raydan method. The limited memory BFGS method shows up worst in the tests. The line search for the non-BB methods is seen to be reasonably efficient with only about two function and gradients calls per line search. The BB method ($\mathbf{g}$ only), to be described below, gives the best

performance. One reason for the improvement of the unmodified BB method over the PR-CG method might be the effect of non-quadratic terms degrading the performance of the CG method. Another possibility is that the CG line search now requires additional evaluations of the function and gradient to attain the required accuracy in the line search. One would not like draw any firm conclusions on the basis of just one set of results, but these results do reinforce Raydan's conclusion that the BB method, suitably modified, can match or even improve on the PR-CG method.

Probably the most interesting outcome to emerge is the difference in performance of the unmodified BB method and the BB-Raydan method. The reasons for this are readily seen by examining the performance of the unmodified method shown in Figure 1. Here the difference $f^{(k)} - f^*$ is plotted on a log scale against the number of iterations. A noticeable feature is the four occasions on which a huge jump is seen in $f^{(k)} - f^*$

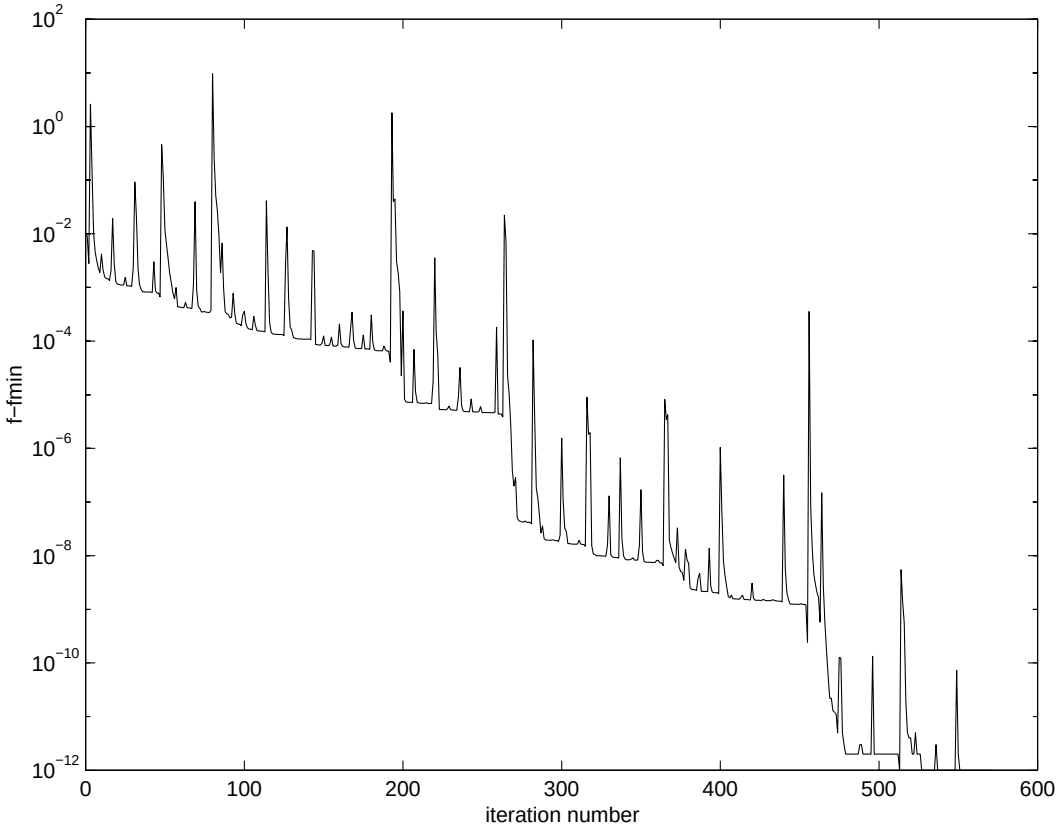

Figure 1: Performance of unmodified BB on Laplace2(b)

above the slowly varying part of the graph. In particular the jump around iteration 460 is over 10^5 in magnitude. Although somewhat disconcerting, these jumps are actually very beneficial in that they are soon followed by a considerable overall improvement in $f^{(k)} - f^*$. For example, before the spike at around iterations 260-270, $f^{(k)} - f^*$ is slowly

varying at around $10^{-5.5}$, whereas afterwards it is around $10^{-7.5}$, an improvement of two orders of magnitude. Similar improvements can be seen either side of the other large spikes. I think the explanation for this is as follows. Consider the quadratic case of the previous section and the effect of 'small' components of the gradient (by which I mean components $g_i^{(k)}$ for small i) in the case that the condition number λ_n/λ_1 is large. In this case, large values of α_k have very little effect on the small components, but they diminish significantly the size of the large components, by virtue of (2.6). Subsequently therefore, an iteration occurs on which the small components dominate $\mathbf{g}^{(k-1)}$, and this gives rise to a small Rayleigh quotient for α_k (see (2.4)), which in turn causes a large increase in the large components of the gradient. The effect over this, possibly over two or three iterations, is to cause the huge spike in the graph of $f^{(k)} - f^*$. (A similar effect is observed if $\|\mathbf{g}^{(k)}\|_2$ is graphed.) However the effect of these iterations is to significantly reduce the relative contribution of the small components in the gradient, *and it is only by allowing large increases in the large components that these small components can be efficiently removed.* Gradient methods which do not permit non-monotonic steps, or which limit their effect, are only able to remove the small components slowly, and hence suffer from slow convergence.

Looking at the spike around iteration 460 in Figure 1, this value of $f^{(k)}$ could only be accepted by Raydan's modification for a value of about $M = 200$, corresponding to the position of the previous higher spike. Therefore we see that the value of $M = 10$ used by Raydan severely restricts the amount of non-monotonicity that can occur. Moreover, the test (3.2) does not allow values of $f^{(k)}$ that are larger than $f^{(1)}$ to be accepted. For Laplace2(b), the value of $f^{(1)} - f^*$ is about 0.94×10^{-2}. Thus we see from Figure 1 that many of the early spike values would not be acceptable, and it is only after iteration 270 or thereabouts that there are no spike values for which $f^{(k)}$ is greater than $f^{(1)}$. Therefore this is another feature of Raydan's modification that restricts the amount of non-monotonicity. The above interpretation also explains why the numerical results obtained by Raydan are poor for very ill-conditioned problems. This is because, from (2.6), the non-monotonicity effects are most noticeable if the condition number is very large. We have seen that the value of $M = 10$ fails to allow the very large spikes to be accepted, which, as we argue above, is important for avoiding slow convergence in a gradient method.

Obvious suggestions to improve the performance of Raydan's modification are first to choose much larger values of M, especially if the problem is likely to be ill-conditioned. Another suggestion is to allow increases in $f^{(k)}$ up to a user supplied value $\bar{f} > f^{(1)}$ on early iterations. This is readily implemented by defining $f^{(k)} = \bar{f}$ for $k < 1$ and changing (3.2) so that the range of indices j is $k - M \leq j \leq k$. These changes make it more likely that the non-monotonic steps observed in Figure 1 are able to be accepted. On the other hand, although the convergence proof presented by Raydan would still hold, this extra freedom to accept 'bad' points might cause difficulties for very non-quadratic problems, and further research on how best to devise a non-monotone line search is needed.

Another idea to speed up Raydan's method is based on the observation that the

unmodified BB method does not need to refer to values of the objective function. These are only needed when the non-monotone line search based on (3.2) is used. Therefore it is suggested that a non-monotone line search based on $\|\mathbf{g}\|$ is used. As in (3.2), an Armijo-type search is used along the line $\mathbf{x}^{(k)} + \mathbf{d}$ where $\mathbf{d} = -\theta\mathbf{g}^{(k)}$, using a sequence of values such as $\theta = 1$, $\frac{1}{10}$, $\frac{1}{100}$, ... An acceptance test such as

$$\|\mathbf{g}\|_2 \leq \max_{k-M\leq j\leq k} \|\mathbf{g}^{(j)}\|_2(1 - \gamma\theta\alpha) \tag{3.4}$$

would be used, where $\gamma \in (0, 1)$ is a small constant and we denote $\mathbf{g} = \mathbf{g}(\mathbf{x}^{(k)} + \mathbf{d})$ and $\alpha = \mathbf{d}^T(\mathbf{g} - \mathbf{g}^{(k)})/\mathbf{d}^T\mathbf{d}$, which is the prospective value of α_{k+1} (see (2.2)) if the step is accepted. Also we denote $\|\mathbf{g}^{(k)}\|_2 = \bar{g}$ for $k < 1$, where $\bar{g}$ is a user supplied upper limit on $\|\mathbf{g}^{(k)}\|_2$. The calculation shown in the last line of Table 3 is obtained by choosing M and $\bar{g}$ sufficiently large so that $\theta_k = 1$ for all k, and shows the benefit to be gained by not evaluating $f(\mathbf{x})$. To prove convergence it is necessary to show that $\mathbf{x}^{(k)} + \mathbf{d}$ would always be accepted for sufficiently small θ in the Armijo sequence. This is readily proved if $f(\mathbf{x})$ is a strictly convex function, as follows. Using the identity

$$\mathbf{g}^T\mathbf{g} = \mathbf{g}^{(k)T}\mathbf{g}^{(k)} + 2\mathbf{g}^{(k)T}(\mathbf{g} - \mathbf{g}^{(k)}) + (\mathbf{g} - \mathbf{g}^{(k)})^T(\mathbf{g} - \mathbf{g}^{(k)})$$

and a Taylor series for $\mathbf{g}(\mathbf{x}^{(k)} + \mathbf{d}))$ about $\mathbf{x}^{(k)}$ we may obtain

$$\mathbf{g}^T\mathbf{g} = \mathbf{g}^{(k)T}\mathbf{g}^{(k)} + 2\mathbf{g}^{(k)T}(\mathbf{g} - \mathbf{g}^{(k)}) + o(\theta)$$

and hence using the binomial theorem that

$$\|\mathbf{g}\|_2 = \|\mathbf{g}^{(k)}\|_2(1 - \theta\alpha) + o(\theta),$$

where α is defined above. It follows from the Taylor series and the strict convexity of $f(\mathbf{x})$ that there exists a constant $\lambda > 0$ such that $\alpha \geq \lambda$, and consequently that

$$\|\mathbf{g}\|_2 \leq \|\mathbf{g}^{(k)}\|_2(1 - \theta\gamma\alpha)$$

if θ is sufficiently small. Thus we can improve on the most recent value $\|\mathbf{g}^{(k)}\|_2$ in this case, and hence (3.4) holds *a fortiori*.

4 Discussion

One thing that I think emerges from this review is just how little we understand about the BB method. In the non-quadratic case, all the proofs of convergence use standard ideas for convergence of the steepest descent method with a line search, so do not tell us much about the BB method itself, so we shall restrict this discussion to the quadratic case. Here we have Raydan's ingenious proof of convergence [21], but this is a proof by contradiction and does not explain for example why the method is significantly better

than the classical steepest descent method. For the latter method we have the much more telling result of Akaike [1] that the asymptotic rate of convergence is linear and the rate constant is $(\lambda_n - \lambda_1)/(\lambda_n + \lambda_1)$ and this exactly matches what is observed in practice. This result is obtained by defining the vector $\mathbf{p}^{(k)}$ by $p_i^{(k)} = (g_i^{(k)})^2/\mathbf{g}^{(k)T}\mathbf{g}^{(k)}$ in the notation of Section 2. This vector acts like a probability distribution for $\mathbf{g}^{(k)}$ over the eigenvectors of A, insofar as it satisfies the conditions $\mathbf{p}^{(k)} \geq \mathbf{0}$ and $\mathbf{e}^T\mathbf{p}^{(k)} = 1$. Akaike shows that the components of $\mathbf{p}^{(k)}$ satisfy the recurrence relation

$$p_i^{(k+1)} = \frac{p_i^{(k)}(\lambda_i - \boldsymbol{\lambda}^T\mathbf{p}^{(k)})}{\sum_i p_i^{(k)}(\lambda_i - \boldsymbol{\lambda}^T\mathbf{p}^{(k)})} \tag{4.1}$$

and that the sequence $\{\mathbf{p}^{(k)}\}$ in general oscillates between two accumulation points $\mathbf{e}_1$ and $\mathbf{e}_n$. Here the scalar product $\boldsymbol{\lambda}^T\mathbf{p}^{(k)}$ is just the Rayleigh quotient calculated from $\mathbf{g}^{(k)}$ (like (2.4) but using $\mathbf{g}^{(k)}$ on the right hand side). A similar analysis is possible for the BB method, in which a superficially similar two term recurrence

$$p_i^{(k+1)} = \frac{p_i^{(k)}(\lambda_i - \boldsymbol{\lambda}^T\mathbf{p}^{(k-1)})}{\sum_i p_i^{(k)}(\lambda_i - \boldsymbol{\lambda}^T\mathbf{p}^{(k-1)})} \tag{4.2}$$

is obtained. However the resulting sequence $\{\mathbf{p}^{(k)}\}$ shows no obvious pattern, and although it must have accumulation points, it is not obvious what they are (probably they include $\mathbf{e}_1$ and $\mathbf{e}_n$, but there may well be others), and the oscillatory behaviour of classical steepest descent is certainly not seen.

In an attempt to obtain further insight into the behaviour of the BB method, the distribution of the 1009 values of α_k obtained in Table 1 is graphed in Figure 2. It can be seen that a very characteristic pattern is obtained, with most of the α_k values being generated in the vicinity of λ_n. The range of values is consistent with a condition number of $\lambda_n/\lambda_1 = 10^{3.61}$. It is also seen that there are very few values close to λ_1, and it is values at this end of the spectrum that give rise to the large non-monotonic spikes such as are seen in Figure 1. It is easily shown from (2.6) that any $\alpha_k \in (\frac{1}{2}\lambda_n, \lambda_n)$ guarantees to reduce *all* components $|g_i|$, so we see that the great majority of iterations cause an improvement in f, and only relatively few iterations give rise to an increase in f. I have observed the pattern of behaviour shown in Figure 2 on a number of ill-conditioned problems, although N. Gould (private communication) indicates that he has generated problems for which the distribution of the α_k does not show this pattern.

What we would like, and what we do not have, is a comprehensive theory that explains these phenomena, and gives a realistic estimate of the rate of convergence averaged over a large number of steps. A useful piece of information at any $\mathbf{x}^{(k)}$ would be a realistic bound on the number of iterations needed to obtain a sufficient improvement on the best value of $f(\mathbf{x})$ that has currently been obtained. This for example could be used in a watchdog-type test for non-quadratic functions, returning to the best previous iterate if the required improvement were not obtained in the said number of iterations. It would also be useful to have a theory that relates to how the eigenvalues λ_i are distributed

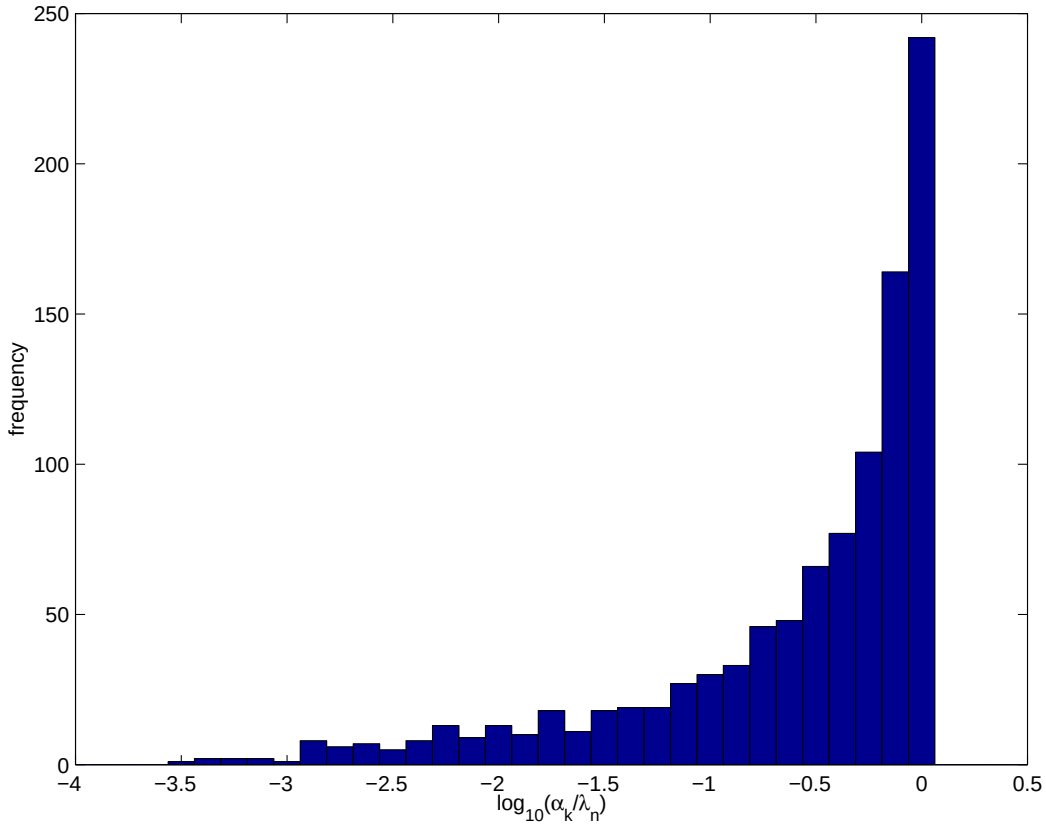

Figure 2: Distribution of α_k values for Laplace1(b)

within the spectrum. For example, if the eigenvalues are distributed in two clusters close to λ_1 and λ_n, then we would expect to be able to show rapid linear convergence, by virtue of the R-superlinear result for $n = 2$.

Then there is the possibility of alternative choices of the step in a steepest descent method. A range of possibilities have been suggested by Friedlander, Martinéz, Molina and Raydan [11], amongst which is the repeated use of a group of iterations in which a classical steepest descent step with $\alpha_k = \mathbf{g}^{(k)T}A\mathbf{g}^{(k)}/\mathbf{g}^{(k)T}\mathbf{g}^{(k)}$ is followed by the use of $\alpha_j = \alpha_k$ for the subsequent m iterations, $j = k+1, \ldots, k+m$. For $m > 1$, this method can considerably increase the non-monotonic behaviour observed in the sequences $\{f(\mathbf{x}^{(k)})\}$ and $\{\|\mathbf{g}(\mathbf{x}^{(k)})\|_2\}$. This is shown from (2.6) because term $(1 - \lambda_i/\alpha_k)$ is repeated m times, so that the effect of a value of α_k close to λ_1 is to increase large components $|g_i|$ by a factor close to $(\lambda_n/\lambda_1)^m$ over the m iterations. Nonetheless, it seems overall that the effect of this modification is beneficial, and values up to say $m = 4$ can work well (Y-H. Dai, private communication). Clearly further study of these possibilities is called for.

The success of the BB and related methods for unconstrained optimization leads us

to consider how it might be used for constrained optimization. This has already been considered for optimization subject to box constraints, and we review current progress in the next section. An important advance would be to find an effective BB-like method for large-scale linear systems involving the KKT matrix

$$\begin{bmatrix} A & B \\ B^T & O \end{bmatrix}.$$

Such an advance would be an important step in developing methods suitable for large scale quadratic programming, and this could lead to the development of methods for large scale nonlinear programming.

5 Optimization with Box Constraints

Many methods have been suggested for solving optimization problems in which the constraints are just the simple bounds

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \tag{5.1}$$

(see the references in [10] and [3] for a comprehensive review). Use of the BB methodology is considered in two recent papers. That of Friedlander, Martinéz and Raydan [10] is applicable only to quadratic functions and uses an active set type strategy in which the iterates only leave the current face if the norm of reduced gradient is sufficiently small. No numerical results are given, and to me it seems preferable to be able to leave the current face at any time if the components of the gradient vector have the appropriate sign. Such an approach is allowed in the BB-like projected gradient method of Birgin, Martinéz and Raydan [3]. This method is applicable to the minimization of a non-quadratic function on any closed convex set, although here we just consider the case of box constraints for which the required projections are readily calculated.

Birgin, Martinéz and Raydan give two methods, both of which use an Armijo-type search on a parameter θ. Both methods use an acceptance test similar to (3.2) which only require sufficient improvement on the largest of the $M+1$ most recent function values in the iteration. In Method 1, the projection

$$\mathbf{x}^+ = P(\mathbf{x}^{(k)} - \theta \mathbf{g}^{(k)}/\alpha_k)$$

is calculated, where α_k is the BB quotient given in (2.2). Then an Armijo search on θ is carried out until an acceptable point is obtained. In Method 2 the point

$$\mathbf{y} = P(\mathbf{x}^{(k)} - \mathbf{g}^{(k)}/\alpha_k)$$

is calculated, and an Armijo search is carried out along the line $\mathbf{x} = \mathbf{x}^{(k)} + \theta(\mathbf{y} - \mathbf{x}^{(k)})$. Both methods are proved to be globally convergent, by using a sufficient reduction property related to the projected gradient. Numerical results on a wide variety of CUTE test

problems of dimension up to about 10^4 are described. These suggest that there is little to choose in practice between Methods 1 and 2, and that the performance is comparable with that of the LANCELOT method of Conn, Gould and Toint [5].

There are a number of aspects in which improvements to Methods 1 and 2 might be sought. For a quadratic function we no longer have the assurance that the unmodified BB method converges (in contrast to Raydan's proof in [21]), so that the methods rely on the Armijo search, and so are open to the criticisms described in Section 3. It would be nice therefore if a convergence theory for some BB-type projected gradient algorithm for the box constrained QP problem could be developed that does not rely on an Armijo search. Similar remarks hold in the non-quadratic case, and any developments for box constrained QP problems can be expected to have implications for the non-quadratic case. However, it will certainly be necessary to have modifications to allow for non-quadratic effects. Any developments for unconstrained optimization, of the sort referred to in Section 3, may well be relevant here. This could include for example the use of a watchdog-type algorithm that requires sufficient improvement over a fixed number of steps. Thus there are many challenging research topics in regard to BB-like methods that suggest themselves, and we can look forward to interesting developments in the future.

References

[1] H. Akaike, *On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method*, Ann. Inst. Statist. Math. Tokyo, 11, (1959), pp. 1-17.

[2] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal., 8, (1988), pp. 141-148.

[3] E. G. Birgin, J. M. Martínez, and M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM J. Optim. 10, (2000), pp. 1196-1211.

[4] A. Cauchy, *Méthode générale pour la résolution des systèms d'équations simultanées,* Comp. Rend. Sci. Paris, 25, (1847), pp. 536-538.

[5] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimizaation with simple bounds*, SIAM J. Numer. Anal., 25, (1988), pp. 433-460, and 26, (1989), pp. 764-767.

[6] Y. H. Dai and L.-Z. Liao, *R-linear convergence of the Barzilai and Borwein gradient method*, Research report, 1999 (accepted by IMA J. Numer. Anal.).

[7] R. S. Dembo, S. C. Eisenstat and T. Steihaug, *Inexact Newton Methods*, SIAM J. Numer. Anal., 19, (1982), pp. 400-408.

[8] R. Fletcher, *Low storage methods for unconstrained optimization,* Lectures in Applied Mathematics (AMS) 26, (1990), pp. 165-179.

[9] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, Comput. J. 7, (1964), pp. 149-154.

[10] A. Friedlander, J. M. Martínez and M. Raydan, *A new method for large-scale box constrained convex quadratic minimization problems*, Optimization Methods and Software, 5, (1995), pp. 57-74.

[11] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan, *Gradient method with retards and generalizations*, SIAM J. Numer. Anal., 36, (1999), pp. 275-289.

[12] W. Glunt, T. L. Hayden, and M. Raydan, *Molecular conformations from distance matrices*, J. Comput. Chem., 14, (1993), pp. 114-120.

[13] W. Glunt, T. L. Hayden, and M. Raydan, *Preconditioners for Distance Matrix Algorithms*, J. Comput. Chem., 15, (1994), pp. 227-232.

[14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd Edition, The Johns Hopkins Press, Baltimore, (1996).

[15] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23, (1986), pp. 707-716.

[16] M. R. Hestenes and E. L. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards Sect. 5:49, (1952), pp. 409-436.

[17] B. Molina and M. Raydan, *Preconditioned Barzilai-Borwein method for the numerical solution of partial differential equations*, Numerical Algorithms, 13, (1996), pp. 45-60.

[18] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Math. of Comp., 35, (1980), pp. 773-782.

[19] E. Polak, *Computational Methods in Optimization: A Unified Approach*, Academic Press, New York, (1971).

[20] D. F. Shanno and K. H. Phua, *Remark on Algorithm 500: Minimization of unconstrained multivariate functions*, ACM Trans. Math. Software, 6, (1980), pp. 618-622.

[21] M. Raydan, *On the Barzilai and Borwein choice of steplength for the gradient method*, IMA J. Numer. Anal., 13, (1993), pp. 321-326.

[22] M. Raydan, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM J. Optim., 7, (1997), pp. 26-33.

[23] J. K. Reid, *Large Sparse Sets of Linear Equations*, Academic Press, London, (1971), Chapter 11, pp. 231-254.