Preconditioned Barzilai–Borwein method for the numerical solution of partial differential equations

Brigida Molina* and Marcos Raydan**

Dpto. de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela
E-mail: bmolina; mraydan@conicit.ve

Received 14 November 1994; revised 1 April 1996
Communicated by C. Brezinski

The preconditioned Barzilai–Borwein method is derived and applied to the numerical solution of large, sparse, symmetric and positive definite linear systems that arise in the discretization of partial differential equations. A set of well-known preconditioning techniques are combined with this new method to take advantage of the special features of the Barzilai–Borwein method. Numerical results on some elliptic test problems are presented. These results indicate that the preconditioned Barzilai–Borwein method is competitive and sometimes preferable to the preconditioned conjugate gradient method.

Keywords: Barzilai–Borwein method, conjugate gradient method, incomplete Cholesky, symmetric successive overrelaxation, elliptic equations.

AMS subject classification: 65F10, 65F15, 65L10, 65N20.

1. Introduction

We are interested in the numerical solution of linear systems of equations

$$Ax = b, \tag{1}$$

where $A \in \Re^{n\times n}$ is large, sparse, symmetric and positive definite (SPD). Sparse SPD linear systems arise frequently in areas such as structural analysis, oil reservoir simulation, fluid dynamics, quantum chemistry, and in general, in problems that involve the discretization of partial differential equations (PDE).

* This author was partially supported by the Parallel and Distributed Computing Center at UCV.
** This author was partially supported by BID-CONICIT, project M-51940.

The solution of (1) is the unique global minimizer of the strictly convex quadratic function

$$f(x) = \frac{1}{2}x^tAx - b^tx. \tag{2}$$

Hence, iterative methods that find local minimizers of differentiable functions can be applied to the solution of (1). In particular, we are interested in the use of the Barzilai–Borwein gradient method to solve (1).

In 1988, Barzilai and Borwein [4] present a new choice of steplength for the gradient method that requires less computational work and converges much faster than the well-known steepest descent method for quadratic functions. The method, for the minimization of (2), is defined by

$$x_{k+1} = x_k - \frac{1}{\alpha_k}g_k,$$

where $g_k = Ax_k - b$, and the scalar α_k is given by

$$\alpha_k = \frac{g_{k-1}^tAg_{k-1}}{g_{k-1}^tg_{k-1}}. \tag{3}$$

Notice that the scalar α_k, the inverse of the steplength at every iteration, is the Rayleigh quotient of A at the vector g_{k-1}.

In 1993, Raydan [14] establishes global convergence of the Barzilai–Borwein method for the strictly convex quadratic case. This result has recently been extended for the (not necessarily strictly) convex quadratic case by Friedlander et al. [7], to incorporate the method in a box constrained optimization technique. They report on the succesful behavior of this new method for very large problems. Glunt et al. [8] applied the Barzilai–Borwein method to the problem of molecular conformations in chemistry, and found a relationship with the *shifted* power method that adds understanding to the significant improvement obtained with the choice of steplength given by (3). In particular, they established that the inverse of the steplengths, α_k, are Rayleigh quotient approximations of the eigenvalues of the matrix A. See also Fletcher [6].

The numerical observations given in [7, 8] lead us to derive the preconditioned version of the Barzilai–Borwein (PBB) method in section 2. We also present some theoretical results that help us to decide which preconditioning techniques are suitable for this new method. In section 3 we describe well-known preconditioning techniques that have been used with the preconditioned conjugate gradient (PCG) method to solve problem (1). In particular, we study the use of the symmetric successive overrelaxation (SSOR) and the incomplete Cholesky (IC) techniques with the PBB method. In section 4 we present preliminary numerical results to test the effectiveness of the PBB method as compared with the PCG method. Finally, in section 5 we present some concluding remarks.

2. Preconditioned version

We now derive the preconditioned version of the Barzilai–Borwein method. The basic idea is to transform the problem of minimizing (2), to that of minimizing a related quadratic functional

$$\widetilde{f}(y) = \frac{1}{2} y^t \widetilde{A} y - \widetilde{b}^t y, \tag{4}$$

where $\widetilde{A} = E^{-1}AE^{-t}$, $\widetilde{b} = E^{-1}b$ and $y = E^t x$ for some nonsingular matrix E. The motivation for minimizing $\widetilde{f}(y)$ instead of $f(x)$ is that if E makes the spectral condition number $\kappa(\widetilde{A})$ much smaller than $\kappa(A)$ or if it clusters the spectrum of $\widetilde{A}$, then the convergence of the Barzilai–Borwein method will be faster for $\widetilde{f}(y)$ than for $f(x)$. See [7, 8].

Our derivation is similar to the derivation of the preconditioned conjugate gradient method, see [3]. Let C be a positive definite matrix factored in the form $C = EE^t$, and consider the quadratic functional (4). The matrix $\widetilde{A}$ is clearly symmetric. Moreover, since A is positive definite, $\widetilde{A}$ is also positive definite. The similarity transformation

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

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

Consider the application of the Barzilai–Borwein method to $\widetilde{f}(y)$. The iterations are described by

$$\widetilde{g}_k = \widetilde{A} y_k - \widetilde{b}, \tag{5}$$

$$y_{k+1} = y_k - \frac{1}{\widetilde{\alpha}_k} \widetilde{g}_k, \tag{6}$$

$$\widetilde{\alpha}_k = \frac{\widetilde{g}^t_{k-1} \widetilde{A} \widetilde{g}_{k-1}}{\widetilde{g}^t_{k-1} \widetilde{g}_{k-1}}, \tag{7}$$

where $\widetilde{\alpha}_0 \neq 0$ and y_0 are arbitrarily chosen.

Let $x_k = E^{-t} y_k$, for all k. Simple calculations based on (5)–(7) and the definitions of $\widetilde{A}$ and $\widetilde{b}$, show that $g_k = E\widetilde{g}_k$ and

$$\widetilde{\alpha}_k = \frac{h^t_{k-1} A h_{k-1}}{g^t_{k-1} h_{k-1}},$$

where $h_{k-1} = C^{-1} g_{k-1}$. Now, using (6) we obtain the recursion formula

$$E^t x_{k+1} = E^t x_k - \frac{1}{\widetilde{\alpha}_k} \widetilde{g}_k,$$

and also

$$x_{k+1} = x_k - \frac{1}{\widetilde{\alpha}_k} h_k. \tag{8}$$

Now, applying A and subtracting b in both sides of (8), we have that

$$g_{k+1} = g_k - \frac{1}{\widetilde{\alpha}_k} A h_k.$$

Therefore, the sequence $\{x_k\}$ is produced by the following algorithm:

Preconditioned Barzilai–Borwein (PBB) algorithm

Given $x_0 \in \Re^n$, α_0 a nonzero real number and C a SPD $n \times n$ matrix. Set $g_0 = Ax_0 - b$. For $k = 0, 1, \ldots$ (until convergence) do

```
    Step 1. Solve $Ch_k = g_k$ for $h_k$;
    Step 2. Set $p_k = Ah_k$;
    Step 3. Set $x_{k+1} = x_k - \frac{1}{\alpha_k} h_k$;
    Step 4. Set $g_{k+1} = g_k - \frac{1}{\alpha_k} p_k$;
    Step 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 $\widetilde{A}$ the *preconditioned matrix*. Notice that every iteration of the PBB algorithm 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. Theorem 1 establishes the convergence of the PBB algorithm when applied to the minimization of (2).

Theorem 1. Let f be the strictly convex quadratic function given by (2). Let $\{x_k\}$ be the sequence generated by the PBB algorithm, $C = EE^t$ an SPD preconditioning matrix, and $x_\star$ the unique minimizer of f. Then, the sequence $\{x_k\}$ converges to $x_\star$.

Proof. Since C and A are SPD matrices, then $\widetilde{A}$ is also an SPD matrix. Therefore, using theorem 1 in [14], we conclude that the sequence $\{y_k\}$ given by (6) and generated by the Barzilai–Borwein method, converges to the unique global minimizer of (4), i.e.,

$$\lim_{k \to \infty} y_k = \widetilde{A}^{-1} \widetilde{b}.$$

Now, using that $x_k = E^{-t} y_k$ for all k, and the definitions of $\widetilde{A}$ and $\widetilde{b}$, we obtain

$$\lim_{k \to \infty} x_k = E^{-t} \lim_{k \to \infty} y_k = E^{-t} \widetilde{A}^{-1} \widetilde{b} = A^{-1} b = x_\star,$$

and the result is established. □

Our next result establishes the Q-linear rate of convergence of the PBB method under a restrictive assumption on the eigenvalues of the matrix $\widetilde{A}$. Nevertheless, we present this result because it will help us to decide which preconditioning strategies are suitable for this new method. First, let us introduce some notation. For a given preconditioned matrix $\widetilde{A}$ we assume that $\{v_1, v_2, \ldots, v_n\}$ are orthonormal eigenvectors of $\widetilde{A}$ associated with the eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ such that

$$0 < \lambda_{\min} = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n = \lambda_{\max}.$$

For a given preconditioning matrix $C = EE^t$, we define the E-norm in $\Re^n$ as follows. For any $z \in \Re^n$,

$$\|z\|_E = \left\|E^t z\right\|_2.$$

Theorem 2. Let f be the strictly convex quadratic function given by (2). Let $\{x_k\}$ be the sequence generated by the PBB algorithm, and $x_\star$ the unique minimizer of f. Let the matrix $C = EE^t$ be chosen such that the eigenvalues of $\widetilde{A}$ satisfy

$$\lambda_{\max} < 2 \times \lambda_{\min}. \tag{9}$$

Then, the sequence $\{x_k\}$ converges Q-linearly to $x_\star$ in the E-norm with convergence factor $\bar{c} = (\lambda_{\max} - \lambda_{\min})/\lambda_{\min}$.

Proof. It suffices to prove that the sequence $\{y_k\}$ given by (6) and generated by the Barzilai–Borwein method converges Q-linearly to $y_\star = \widetilde{A}^{-1}\widetilde{b}$ in the Euclidean norm with convergence factor $\bar{c}$. Indeed, for any integer k

$$\|y_\star - y_k\|_2 = \left\|(x_\star - x_k)\right\|_E.$$

Let us define $e_k = y_\star - y_k$ for all k. Using (5) and (6), we have for any integer k

$$e_{k+1} = \frac{1}{\widetilde{\alpha}_k}\big(\widetilde{\alpha}_k I - \widetilde{A}\big)e_k. \tag{10}$$

Now for any initial error e_0, there exist constants $d_1^0, d_2^0, \ldots, d_n^0$ such that:

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

Using (10) we obtain for any k,

$$e_{k+1} = \sum_{i=1}^{n} d_i^{k+1} v_i, \tag{11}$$

where

$$d_i^{k+1} = \frac{\widetilde{\alpha}_k - \lambda_i}{\widetilde{\alpha}_k} d_i^k. \tag{12}$$

Using (10)–(12) we obtain

$$e_{k+1} = \frac{1}{\widetilde{\alpha}_k}\left(\widetilde{\alpha}_k I - \widetilde{A}\right)\sum_{i=1}^{n} d_i^k v_i = \sum_{i=1}^{n} \frac{\widetilde{\alpha}_k - \lambda_i}{\widetilde{\alpha}_k} d_i^k v_i.$$

By the orthonormality of the eigenvectors we have

$$\|e_{k+1}\|_2^2 = \sum_{i=1}^{n} \left(d_i^k\right)^2 \left(\frac{\widetilde{\alpha}_k - \lambda_i}{\widetilde{\alpha}_k}\right)^2 \leqslant \max_i \left(\frac{\widetilde{\alpha}_k - \lambda_i}{\widetilde{\alpha}_k}\right)^2 \|e_k\|_2^2. \tag{13}$$

Since $\widetilde{\alpha}_k$ is the Rayleigh quotient of $\widetilde{A}$ at the vector $\widetilde{g}_{k-1}$, then it satisfies for all k

$$0 < \lambda_{\text{min}} \leqslant \widetilde{\alpha}_k \leqslant \lambda_{\text{max}}. \tag{14}$$

Therefore,

$$\max_i \left|\frac{\widetilde{\alpha}_k - \lambda_i}{\widetilde{\alpha}_k}\right| \leqslant \frac{\lambda_{\text{max}} - \lambda_{\text{min}}}{\lambda_{\text{min}}}. \tag{15}$$

Finally, combining (13)–(15) gives

$$\|e_{k+1}\|_2 \leqslant \bar{c}\|e_k\|_2 \quad \text{where } \bar{c} = \frac{\lambda_{\text{max}} - \lambda_{\text{min}}}{\lambda_{\text{min}}} < 1,$$

as we wanted to prove. □

3. Suitable preconditioning techniques

Theorem 2 from the previous section states that the PBB method converges fast if λ_{max} is close to λ_{min} (i.e., $\kappa_2(\widetilde{A}) \approx 1$), and λ_{min} is away from zero. On the other hand, Barzilai and Borwein [4] proved R-superlinear convergence if $\widetilde{A}$ possesses two distinct eigenvalues. In practice, this speed of convergence is observed whenever $\widetilde{A}$ possesses few eigenvalues. So, a heuristic version of the Barzilai and Borwein result can be written as follows: the PBB method performs well when the spectrum of $\widetilde{A}$ has been clustered in very few groups. In conclusion, the features that a good preconditioning matrix must have for the PBB method are quite similar to the features that a good preconditioning matrix must have for the preconditioned conjugate gradient (PCG) method. See Golub and Van Loan [9].

The PCG method has been extensively applied to the solution of large sparse SPD linear systems, see for instance [1–3, 10, 12, 13, 15]. There are many equivalent formulations of the PCG algorithm. However, the version that has proved to be the most efficient, and the one used throughout this work, is the following procedure:

Preconditioned conjugate gradient (PCG) algorithm

```
Given $x_0 \in \Re^n$ and $C$ an SPD $n \times n$ matrix.
Set $g_0 = Ax_0 - b$, solve $Ch_0 = g_0$ and set $d_0 = -h_0$.
For $k = 0, 1, \ldots$ (until convergence) do
    Step 1. Set $p_k = Ad_k$;
    Step 2. Set $\alpha_k = \frac{g_k^t h_k}{d_k^t p_k}$;
    Step 3. Set $x_{k+1} = x_k + \alpha_k d_k$;
    Step 4. Set $g_{k+1} = g_k + \alpha_k p_k$;
    Step 5. Solve $Ch_{k+1} = g_{k+1}$ for $h_{k+1}$;
    Step 6. Set $\beta_k = \frac{g_{k+1}^t h_{k+1}}{g_k^t h_k}$;
    Step 7. Set $d_{k+1} = -h_{k+1} + \beta_k d_k$;
End do.
```

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

We now present two of the most popular preconditioning techniques for the PCG method. The first one is based on the SSOR iterative method to solve linear systems of equations [17]. Let the SPD 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 (1) can be formulated as a one-stage algorithm (see [3, 17]):

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

where $C - R = A$ and

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

The matrix C is the SSOR preconditioning matrix, and ω is a real scalar parameter between 0 and 2. The choice of ω has been 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 preconditioning scheme, the optimal value of ω (i.e., the value of ω that minimizes $\kappa_2(\widetilde{A})$) has been shown to be

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

where

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

and

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

Unfortunately, μ and δ are 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 very insensitive to the estimates of μ and δ, see [3]. In our numerical experiments we use the SSOR preconditioning scheme with the PBB method and also with the PCG method to compare their performance. In both cases, we approximate the value ω^* with a simple scheme that is closely related to our model problem.

We also use, in our numerical experiments, another important strategy to preconditioning that is based on incomplete Cholesky factorizations of the matrix A. Given the SPD matrix A, by an incomplete Cholesky (IC) factorization we mean a relation of the form

$$A = \widehat{L}\widehat{L}^t + R,$$

where $\widehat{L}$ is lower triangular, has some preestablished sparsity structure and approximates the exact Cholesky factor of A. The matrix R is not zero and $\|R\|$, hopefully, is small compared to $\|A\|$. In this case, the preconditioning matrix is given by $C = \widehat{L}\widehat{L}^t$. One way to obtain such incomplete factorization is to suppress the fill-in that appears during the factorization process, i.e., if $a_{ij} = 0$ then $\widehat{l}_{ij} = 0$.

An interesting modification of the incomplete factorization is obtained by stumping, during the process, the neglected values of the exact Cholesky factor of A to the diagonal, i.e.,

$$\sum_{j=1}^{n} r_{ij} = 0 \quad \text{for } 1 \leqslant i \leqslant n.$$

This so-called modified incomplete Cholesky (MIC), introduced by Gustafsson [11], guarantees that the preconditioning matrix C is an SPD matrix if A is weakly diagonally dominant. In our numerical experiments, the coefficient matrices are weakly diagonally dominant, and so the MIC produces a feasible preconditioning strategy.

Concus et al. [5] show that MIC has a strong effect on the eigenvalue distribution of the preconditioned matrix $\widetilde{A}$. In fact, $\lambda_{\min} \geqslant 1$ if the MIC scheme is used [5, 13]. This property, on the smallest eigenvalue of $\widetilde{A}$, has positive effects on the performance of the PBB method.

4. Numerical experiments

In this section, we present results of numerical experiments that were designed to test the effectiveness of the PBB method as compared with the PCG method. The test problems are large, sparse, SPD linear systems arising from the discretization of elliptic PDE problems. All our experiments were run on a SparcStation II in double precision FORTRAN with a machine epsilon of about 2×10^{-16}.

For our model problem, consider the elliptic partial differential equation

$$-\frac{\partial}{\partial x}\left(a(x,y)\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(a(x,y)\frac{\partial u}{\partial y}\right)+\gamma u=f_1(x,y), \tag{16}$$

where $\gamma \geqslant 0$ is a real scalar parameter, and the positive function $a(x,y)$ and the function $f_1(x,y)$ are given. We pose (16) on the unit square $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$, with homogeneous Dirichlet boundary conditions, i.e., we seek a function u that is continuous on the unit square, satisfies (16) in the interior of the unit square, and equals zero on the boundary.

We discretize (16) using the five-point centered finite difference scheme on a uniform $m \times m$ grid with $h = 1/(m+1)$ and the natural ordering, producing a linear system $Ax = b$ of order $n = m^2$. The matrix A has the following block tridiagonal form

$$A=\begin{pmatrix} A_1 & D_2 & & & \\ D_2 & A_2 & D_3 & & \\ & \ddots & \ddots & \ddots & \\ & & D_{m-1} & A_{m-1} & D_m \\ & & & D_m & A_m \end{pmatrix},$$

where each A_i is a square and symmetric tridiagonal matrix of order m, and each D_i is a diagonal matrix of order m. Since γ is a nonnegative real number and $a(x,y) > 0$ on the unit square, the matrix A is symmetric and positive definite. Notice that for any vector $z \in \Re^n$, the cost of the matrix–vector multiplication Az is approximately $5n$. Finally, notice that for $\gamma = 0$ and $a(x,y) = 1$ our model problem is the classical Dirichlet problem for which A is ill-conditioned.

We ran an implementation of algorithms PBB and PCG to solve problem (16) for different functions a, different values of the parameter γ and different values of the dimension n (i.e., different step sizes h). We used the SSOR, the IC, and the MIC preconditioning strategies described in section 3 for both algorithms. The parameter ω associated with the SSOR technique was chosen in the following way:

$$\omega = \frac{2}{1+0.6\gamma+2.6h} \quad \text{for } 0 \leqslant \gamma \leqslant 1.$$

This choice of ω has been discussed by Axelsson and Barker [3] for $\gamma = 0$. For this case, they proved that $\kappa_2(C^{-1}A) = \mathrm{O}(1/h)$. We extended the choice of ω for $\gamma > 0$ based on numerical experience.

In all our experiments, the function f_1 in (16) is defined to have the constant value 1. We start the process in both algorithms at $x_0 = (0,0,\dots,0)^t$ and we set $\alpha_0 = 1$ in algorithm PBB. We stop the process, in both algorithms, whenever $\|g_k\|_2 \leqslant 10^{-8} \times \|g_0\|_2$.

In our first experiment we fix the parameter $\gamma = 0$ and the function $a(x,y) = 1$, and we study the number of iterations required for both methods and the three preconditioning techniques, for different values of n. Figures 1 and 2 show the number

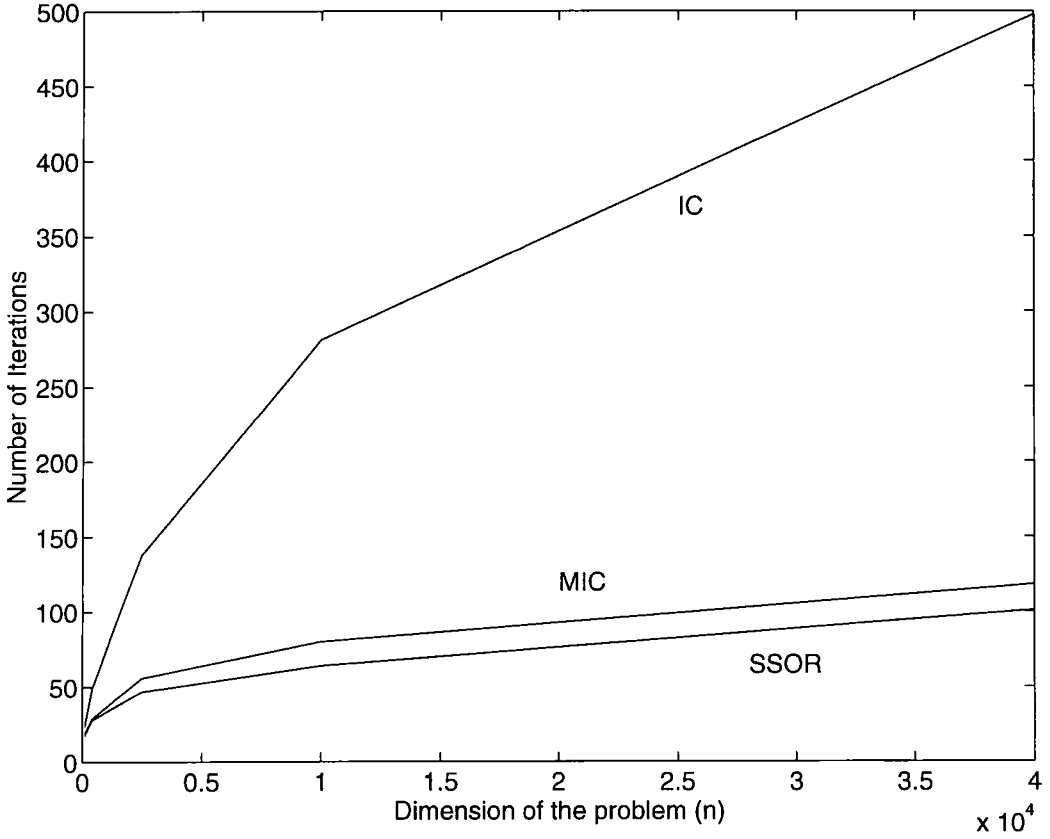

Figure 1. Number of iterations required for the PBB method when $\gamma = 0$ and $a(x,y) = 1$.

of iterations required for the PBB method and the PCG method, for different values of n, respectively. We can see that, in all cases, the number of iterations increases as n increases. However, in general, the PCG method requires fewer iterations than the PBB method. For SSOR and MIC the difference oscillates between 5 and 10%, and for IC it oscillates between 30 and 50%. It may also be observed that the number of iterations required by the IC scheme is significantly higher than the number of iterations needed by the SSOR and the MIC techniques. Indeed, the smallest eigenvalue of $\widetilde{A}$, in the IC case, is very close to zero for large n, and this represents a negative effect for both methods. In particular, for the PBB method the convergence deteriorates, as we discussed in section 3, when the scalar α_k approaches the smallest eigenvalue, and this eigenvalue is very close to zero. On the other hand, in the MIC case, $\lambda_{\min} \geqslant 1$, which is a nice feature for the PBB method. Moreover, for SSOR and MIC, it has been established that $\kappa_2(\widetilde{A}) = O(1/h)$, see [3, 11]. Consequently, from now on, we will only report on the behavior of the two methods with the MIC and SSOR preconditioning techniques.

The second experiment was chosen to demonstrate the effect of increasing the parameter γ on the effectiveness of the PBB method as compared with the PCG method. Figures 3 and 4 show the number of iterations and the computational work,

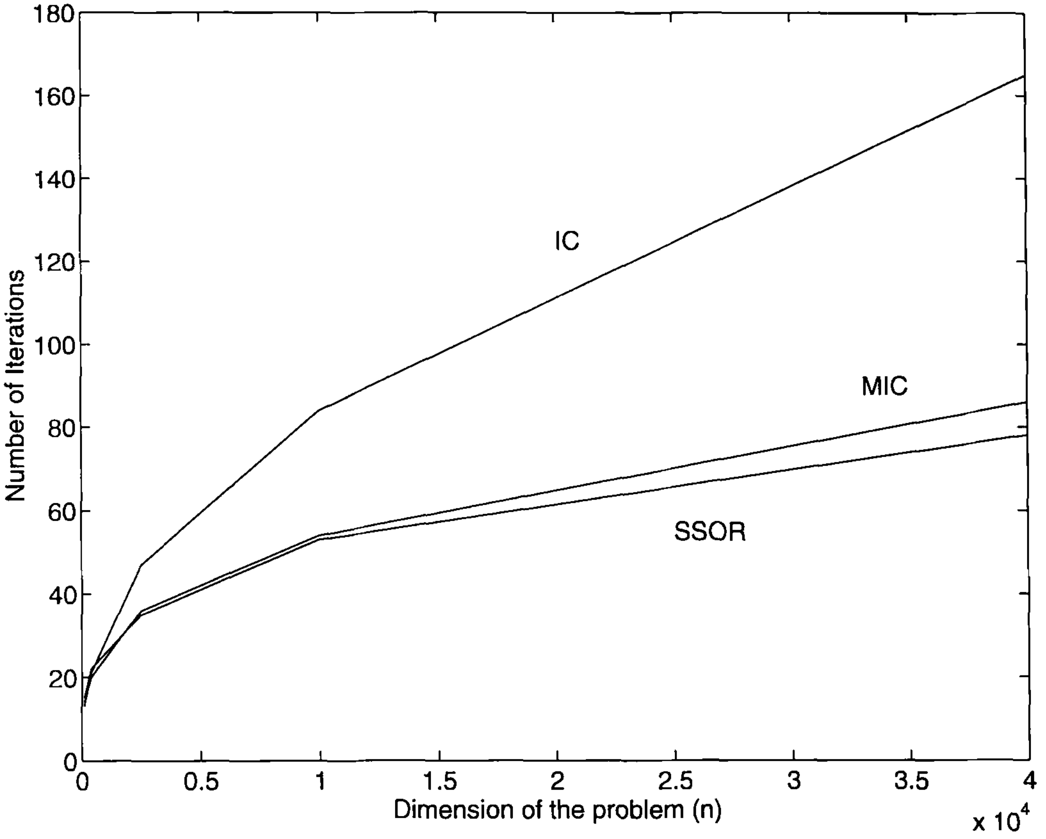

Figure 2. Number of iterations required for the PCG method when $\gamma = 0$ and $a(x, y) = 1$.

respectively, required for both methods and both preconditioning schemes when they are applied to the model problem (16) with the dimension fixed at $n = 40 \times 10^3$. It can be seen that the number of iterations required by the PCG is smaller than the number required by the PBB method, although the difference is not significant. Furthermore, when γ increases the two methods tend to require the same number of iterations, and this number of iterations decreases. This is a consequence of the decrease in the spectral condition number of $\tilde{A}$ and the fact that the smallest eigenvalue of $\tilde{A}$ is bounded away from zero. Finally, it is worth mentioning that MIC performs better than SSOR for both methods.

We also study the effect of increasing the parameter γ on the computational work required by the PBB and the PCG methods. Figure 4 shows the number of floating point operations required when the two methods are applied to the model problem when $a(x, y) = 1$, with different values of γ and $n = 40 \times 10^3$. It may be observed that, in all cases, the PBB method requires less computational work than the PCG method. The difference in the required work is approximately 10% and tends to increase when γ increases, which is not in general a significant improvement.

In our third experiment we fix the parameter $\gamma = 0$ and we define the function

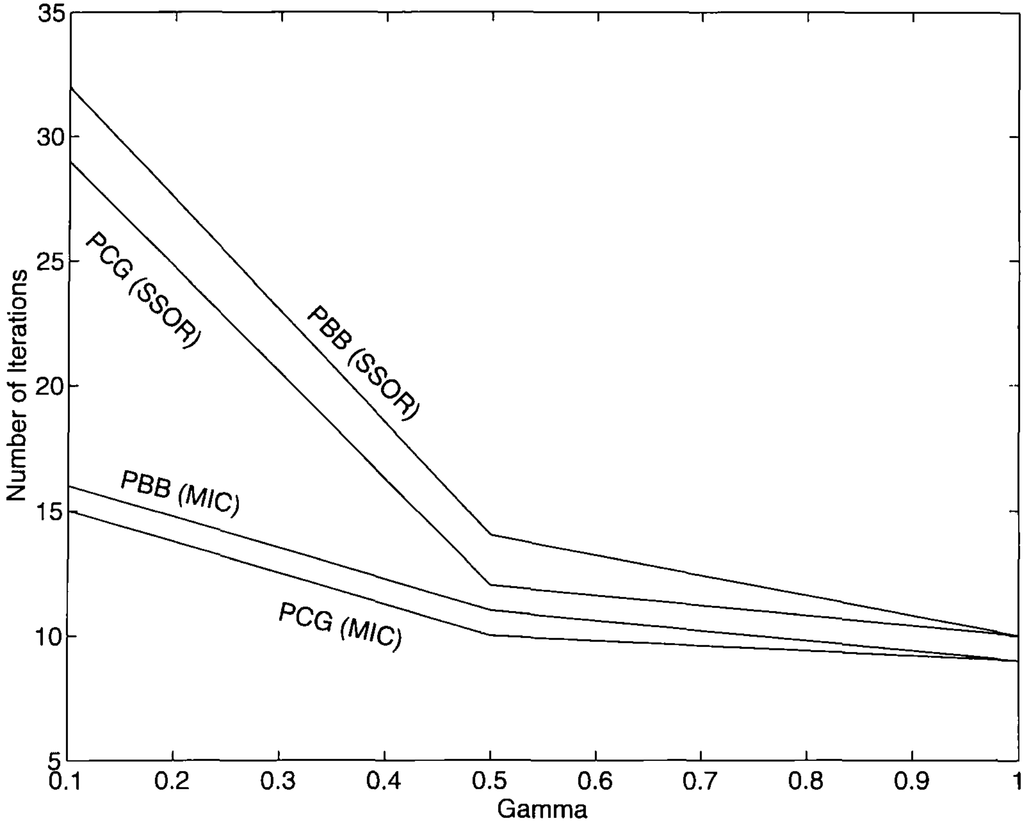

Figure 3. Number of iterations required for the PBB and the PCG methods for different values of γ when $a(x, y) = 1$ and $n = 40 \times 10^3$.

$a(x, y)$ as follows:

$$a(x,y) = \begin{cases} 10000 & \text{for } (x,y) \in [0, 0.75] \times [0, 0.75], \\ 1 & \text{for } (x,y) \in [0.75, 1] \times [0, 0.75] \cup [0, 0.75] \times [0.75, 1], \\ 0.1 & \text{for } (x,y) \in [0.75, 1] \times [0.75, 1]. \end{cases} \quad (17)$$

Notice that $\tilde{A}$, for this particular choice of $a(x, y)$, is very ill-conditioned. We study the behavior of both methods and both preconditioning techniques, for different values of n. Figure 5 shows the number of iterations required for the PBB method and the PCG method to achieve convergence. We can see that, in all cases, the number of iterations increases as n increases. However, in general, the PCG method requires fewer iterations than the PBB method. For MIC the difference oscillates between 5 and 20%, and for SSOR the difference oscillates between 5 and 15%. It is worth mentioning that, in this very ill-conditioned case, the SSOR technique performs better than the MIC technique.

Figure 6 shows the number of floating point operations required when the two methods are applied to the model problem when $a(x, y)$ is given by (17) with different values of n. It may be observed that, in all cases, the PCG method requires

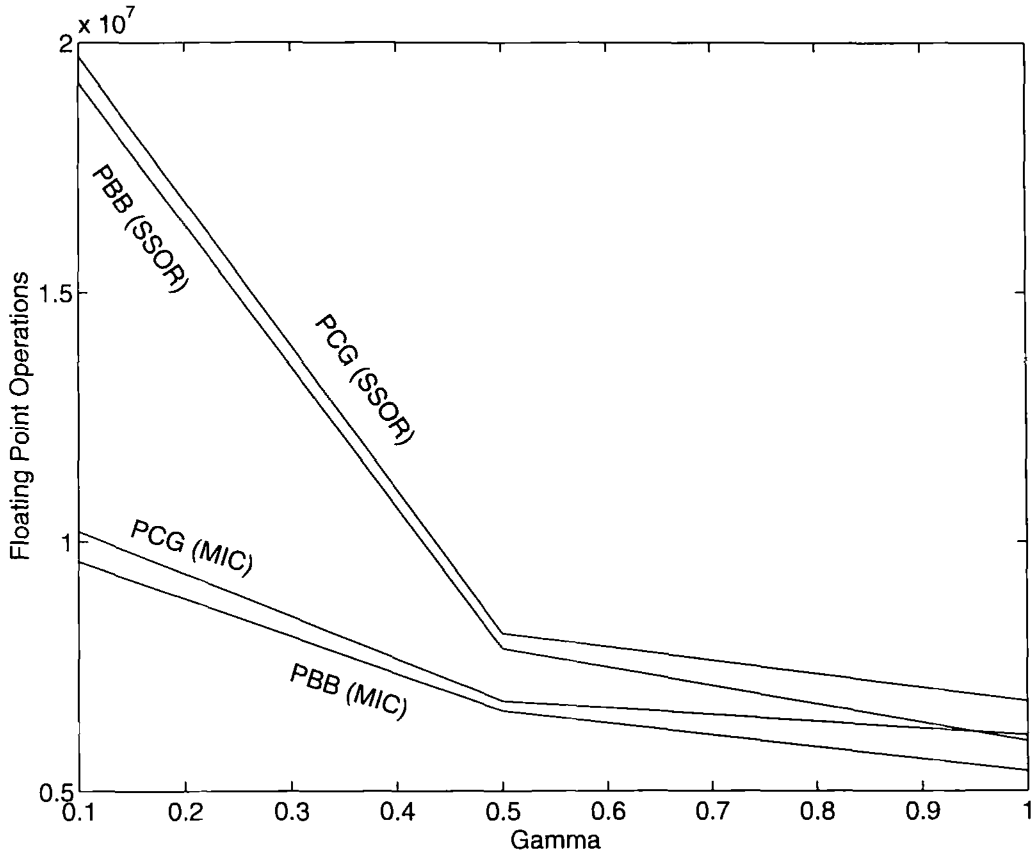

Figure 4. Floating point operations required for the PBB and the PCG methods for different values of γ when $a(x, y) = 1$ and $n = 40 \times 10^3$.

less computational work than the PBB method. The difference in the required work oscillates between 5 and 10%, which is not a significant improvement.

5. Concluding remarks

We have introduced the preconditioned Barzilai and Borwein method and compared it with the preconditioned conjugate gradient method when they are both applied to the large sparse SPD linear systems that arise from the discretization of elliptic differential equations. Our preliminary numerical results indicate that the PBB method is comparable and sometimes preferable to the PCG method, whenever $\tilde{A}$ is not extremely ill-conditioned and the smallest eigenvalue of $\tilde{A}$ is not too close to zero. This conclusion 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 conjugate gradient method for large sparse SPD linear systems. However, when the matrix $\tilde{A}$ is extremely ill-conditioned, the PCG method is still a better option.

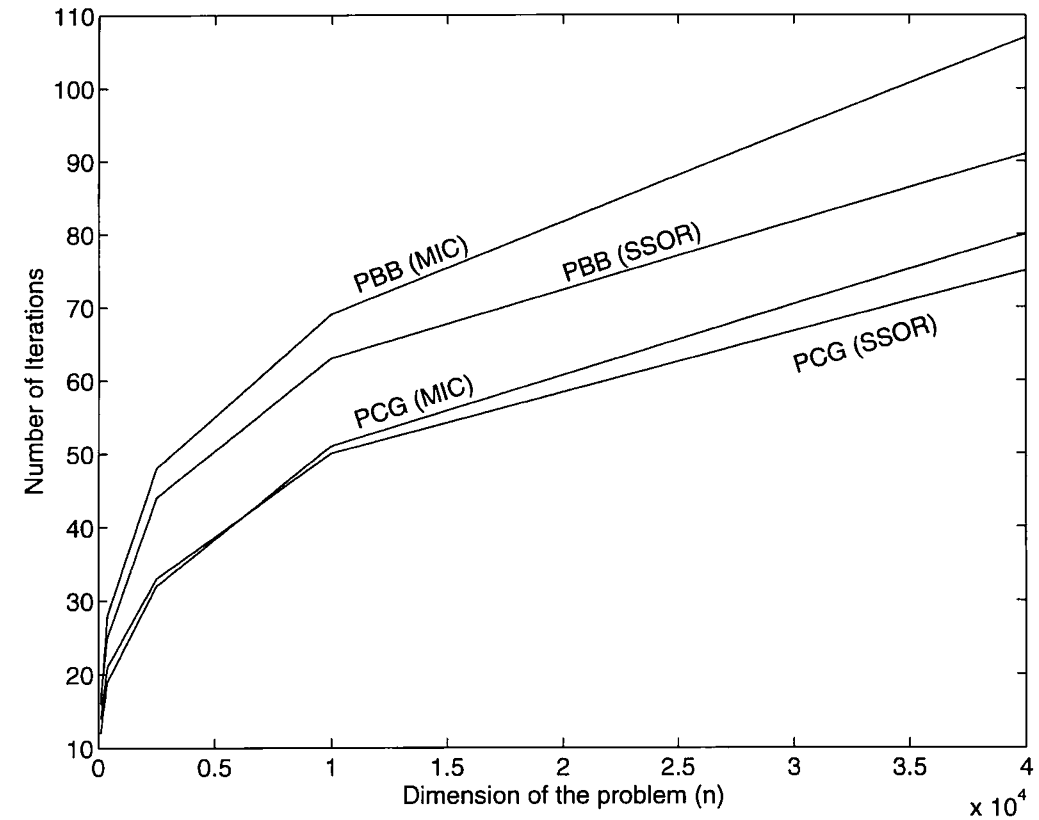

Figure 5. Number of iterations required for the PBB and the PCG methods for different values of n when $a(x, y)$ is given by (17).

It is important to observe that the PCG method requires the storage of one additional vector when compared to the PBB method. In fact, if the storage of the preconditioning matrix C is not considered, the PBB method requires 3 storage vectors and the PCG method needs 4 storage vectors. The PCG method also requires one additional scalar–vector multiplication and one additional vector summation per iteration. These additional requirements are not negligible for very large problems.

From a theoretical point of view, the rate of convergence of the Barzilai and Borwein method remains as an open question. See [4, 6, 8] for some comments on this matter. Based on a restrictive assumption, we have presented a rate of convergence result (theorem 2) whose sole purpose is to motivate the choice of preconditioning strategies with the PBB method. In fact, as pointed out by one of the referees, the PBB method performs better in practice than what one would expect from the conclusion of theorem 2. This gap between theory and practice is an interesting topic that deserves further investigation.

Finally, we want to stress that in our experiments we have chosen preconditioning techniques that have been designed for the PCG method. In the near future, we would like to study the possibility of finding suitable preconditioning techniques especially designed for the PBB method.

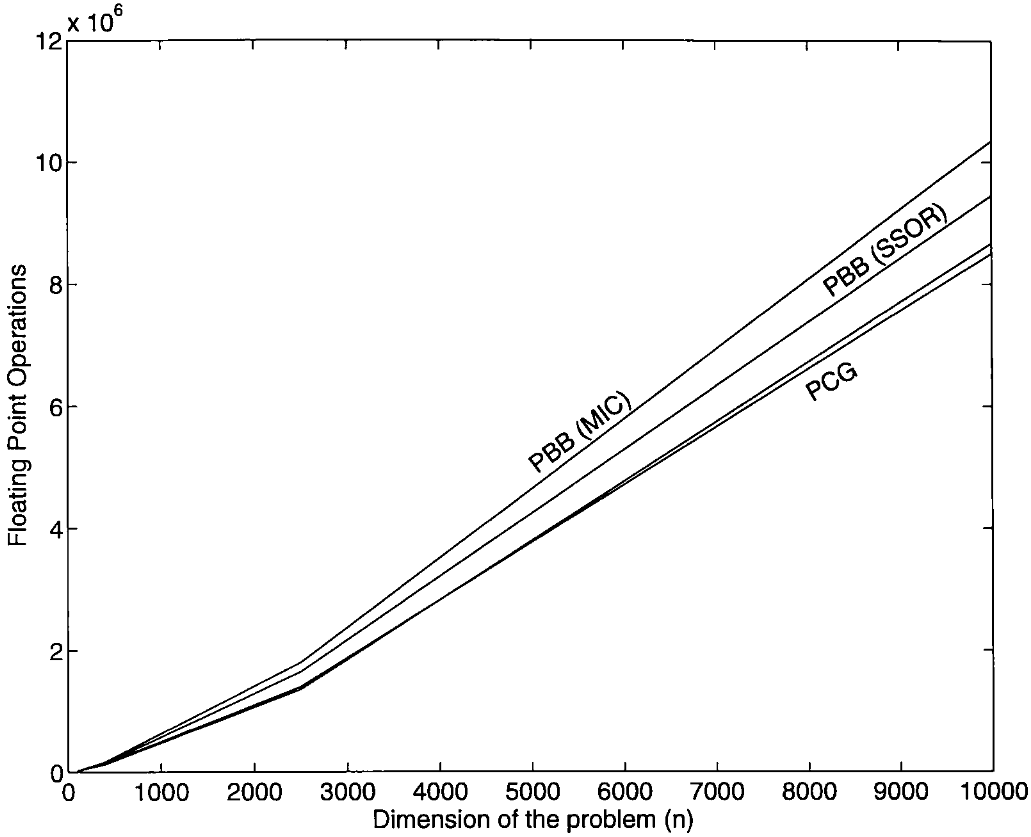

Figure 6. Floating point operations required for the PBB and the PCG methods for different values of n when $a(x, y)$ is given by (17).

Acknowledgements

We would like to thank two anonymous referees for valuable comments.

References

[1] O. Axelsson, *Iterative Solution Methods* (Cambridge University Press, New York, 1994).

[2] O. Axelsson, A survey of preconditioned iterative methods for linear systems of algebraic equations, BIT 25 (1985) 166–187.

[3] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems, Theory and Computation* (Academic Press, New York, 1984).

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

[5] P. Concus, G. H. Golub and G. Meurant, Block preconditioning for the conjugate gradient method, SIAM J. Sci. Stat. Comp. 6 (1985) 220–252.

[6] R. Fletcher, Low storage methods for unconstrained optimization, Lectures in Applied Mathematics (AMS) 26 (1990) 165–179.

[7] A. Friedlander, J. M. Martinez and M. Raydan, A new method for large-scale box constrained convex quadratic minimization problems, Opt. Methods and Software 5 (1995) 57–74.

[8] W. Glunt, T. L. Hayden and M. Raydan, Molecular conformations from distance matrices, J. Comp. Chem. 14 (1993) 114–120.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, 1989).

[10] G. H. Golub and D. P. O'Leary, Some history of the conjugate gradient and Lanczos methods, SIAM Review 31 (1989) 50–102.

[11] I. Gustafsson, A class of first order factorization methods, BIT 18 (1978) 142–156.

[12] L. A. Hageman and D. M. Young, *Applied Iterative Methods* (Academic Press, New York, 1981).

[13] J. M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems* (Plenum Press, New York, 1988).

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

[15] H. A. Van der Vorst and K. Dekker, Conjugate gradient type methods and preconditioning, J. Comp. Appl. Math. 24 (1988) 73–87.

[16] D. M. Young, A historical overview of iterative methods, Computer Physics Communications 53 (1989) 1–17.

[17] D. M. Young, *Iterative Solution of Large Linear Systems* (Academic Press, New York, 1971).