

GALERKIN PROJECTION METHODS FOR SOLVING MULTIPLE LINEAR SYSTEMS*

TONY F. CHAN[†] AND MICHAEL K. NG[‡]

Abstract. In this paper, we consider using conjugate gradient (CG) methods for solving multiple linear systems $A^{(i)}x^{(i)} = b^{(i)}$, for $1 \leq i \leq s$, where the coefficient matrices $A^{(i)}$ and the right-hand sides $b^{(i)}$ are different in general. In particular, we focus on the seed projection method which generates a Krylov subspace from a set of direction vectors obtained by solving one of the systems, called the seed system, by the CG method and then projects the residuals of other systems onto the generated Krylov subspace to get the approximate solutions. The whole process is repeated until all the systems are solved. Most papers in the literature [T. F. Chan and W. L. Wan, *SIAM J. Sci. Comput.*, 18 (1997), pp. 1698–1721; B. Parlett *Linear Algebra Appl.*, 29 (1980), pp. 323–346; Y. Saad, *Math. Comp.*, 48 (1987), pp. 651–662; V. Simoncini and E. Gallopoulos, *SIAM J. Sci. Comput.*, 16 (1995), pp. 917–933; C. Smith, A. Peterson, and R. Mitra, *IEEE Trans. Antennas and Propagation*, 37 (1989), pp. 1490–1493] considered only the case where the coefficient matrices $A^{(i)}$ are the same but the right-hand sides are different. We extend and analyze the method to solve multiple linear systems with varying coefficient matrices and right-hand sides. A theoretical error bound is given for the approximation obtained from a projection process onto a Krylov subspace generated from solving a previous linear system. Finally, numerical results for multiple linear systems arising from image restorations and recursive least squares computations are reported to illustrate the effectiveness of the method.

Key words. multiple linear systems, Krylov space, conjugate gradient method, Galerkin projection

AMS subject classifications. 65F10, 65Y20

PII. S1064827598310227

1. Introduction. We want to solve iteratively, using Krylov subspace methods, the following linear systems:

$$(1) \quad A^{(i)}x^{(i)} = b^{(i)}, \quad 1 \leq i \leq s,$$

where $A^{(i)}$ are real symmetric positive definite matrices of order n , and in general $A^{(i)} \neq A^{(j)}$ and $b^{(i)} \neq b^{(j)}$ for $i \neq j$. We note that if the coefficient matrices and the right-hand sides are arbitrary, there is nearly no hope to solve them more efficiently than as s completely unrelated systems. We have to solve the s linear systems independently either by using direct methods or iterative methods. Fortunately, in many practical applications, the coefficient matrices and the right-hand sides are not arbitrary, and often there is information that can be shared among the coefficient matrices and the right-hand sides. Such a situation occurs, for instance, in recursive least squares computations [17], wave scattering problems [11, 2, 6], numerical methods for integral equations [11], and image restorations [10]. In this paper, our aim is to propose a methodology to solve these related multiple linear systems efficiently.

*Received by the editors January 9, 1998; accepted for publication (in revised form) January 15, 1999; published electronically November 4, 1999.

<http://www.siam.org/journals/sisc/21-3/31022.html>

[†]Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90024 (chan@math.ucla.edu). The research of this author was partially supported by Office of Naval Research grants N00014-92-J-1890 and N00014-96-1-0277, the Army Research Office under contract DAAL-03-91-C-0047 (University of Tennessee subcontract ORA4466.04, Amendments 1 and 2), and National Science Foundation grant DMS-9626755.

[‡]Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong (mng@maths.hku.hk). The research of this author was supported by HKU CRCG grants 10201824 and 10201939.

In [21], Smith, Peterson, and Mitra proposed and considered using a *seed* method for solving linear systems of the same coefficient matrix but different right-hand sides, i.e.,

$$AX = [b^{(1)} \ b^{(2)} \ \dots \ b^{(s)}].$$

In the seed method, we select one seed system and solve it by the conjugate gradient (CG) method. Then we perform a Galerkin projection of the residuals onto the Krylov subspace generated by the seed system to obtain approximate solutions for the unsolved ones. The approximate solutions are then refined by the CG method again. In [21], a very effective implementation of the Galerkin projection method was developed which uses direction vectors generated in the CG process to perform the projection. In [5], Chan and Wan observed that the seed method has several nice properties. For instance, the CG method, when applied to the system with the projected solution as the initial guess, converges faster than the usual CG process. Another observation is that if the right-hand sides are closely related, the method automatically exploits this fact and usually only takes a few restarts to solve all the systems. In [5], a theory was developed to explain these phenomena. We remark that the seed method can be viewed as a special implementation of the Galerkin projection method which had been considered and analyzed earlier for solving linear systems with multiple right-hand sides; see for instance, Parlett [16], Saad [18], van der Vorst [23], Papadrakakis and Smerou [15], Simoncini and Gallopoulos [19, 20]. A very different approach based on the Lanczos method with multiple starting vectors was recently proposed by Freund and Malhotra [6]. In [7], Frommer and Maass proposed to use CG methods for solving shifted linear systems simultaneously at the expense of one matrix-vector multiply only.

In this paper, we extend the seed method to solve the multiple linear systems (1), with different coefficient matrices ($A^{(j)} \neq A^{(k)}$) and different right-hand sides ($b^{(j)} \neq b^{(k)}$). We analyze the seed method and extend the theoretical results given in [5]. We will see that the theoretical error bound for the approximation obtained from a projection process depends on the projection of the eigenvector components of the error onto a Krylov subspace generated from the previous seed system and how different the system is from the previous one.

In contrast to [5], in the general case here where the coefficient matrices $A^{(i)}$ can be different, it is not possible to derive very precise error bounds since the $A^{(i)}$'s have different eigenvectors in general. Fortunately, in many applications, even though the $A^{(i)}$'s are indeed different, they may be related to each other in a structured way which allows a more precise error analysis. Such is the case in the two situations that we study in this paper, namely, the eigenvectors of the coefficient matrices are the same and the coefficient matrices differ by rank-1 or rank-2 matrices. These kinds of linear systems arise in image restoration and recursive least squares applications. Numerical examples on these applications are given to illustrate the effectiveness of the projection method. We will see from the numerical results that the eigenvector components of the right-hand sides are effectively reduced after the projection process and the number of iterations required for convergence decreases when we employ the projected solution as an initial guess. Moreover, other examples involving more general coefficient matrices (for instance, those that do not have the same eigenvectors or differ by a low-rank matrix), are also given to test the performance of the projection method. We observe similar behavior in the numerical results as in image restoration and recursive least squares computations. These numerical results demonstrate that the projection method is effective.

The paper is organized as follows. In section 2, we first describe the seed projection algorithm for general multiple linear systems. In section 3, we analyze the Galerkin projection method. In particular, we study multiple linear systems arising from two specific cases. Numerical examples are given in section 4 and concluding remarks are given in section 5.

2. Derivation of the algorithm. CG methods can be seen as iterative solution methods to solve a linear system of equations by minimizing an associated quadratic functional. For simplicity, we let

$$f_i(x) = \frac{1}{2}x^T A^{(i)}x - (b^{(i)})^T x$$

be the associated quadratic functional of the linear system $A^{(i)}x^{(i)} = b^{(i)}$. The minimizer of f_j is the solution of the linear system $A^{(i)}x^{(i)} = b^{(i)}$. The idea of the projection method is that for each restart, a seed system $A^{(k)}x^{(k)} = b^{(k)}$ is selected from the unsolved ones which are then solved by the CG method. An approximate solution $\hat{x}^{(j)}$ of the nonseed system $A^{(j)}x^{(j)} = b^{(j)}$ can be obtained by using search direction p_i^k generated from the i th iteration of the seed system. More precisely, given the i th iterate x_i^j of the nonseed system and the direction vector p_i^k , the approximate solution $\hat{x}^{(j)}$ is found by solving the following minimization problem:

$$(2) \quad \min_{\eta} f_j(x_i^j + \eta p_i^k).$$

It is easy to check that the minimizer of (2) is attained at

$$(3) \quad \hat{x}^{(j)} = x_i^j + \eta p_i^k,$$

where

$$(4) \quad \eta = \frac{(p_i^k)^T r_i^j}{(p_i^k)^T A^{(j)} p_i^k} \quad \text{and} \quad r_i^j = b^{(j)} - A^{(j)} x_i^j.$$

After the seed system $A^{(k)}x^{(k)} = b^{(k)}$ is solved to the desired accuracy, a new seed system is selected and the whole procedure is repeated. In the following discussion, we call this method *Projection Method I*. We note from (4) that the matrix-vector multiplication $A^{(j)}p_i^k$ is required for each projection of the nonseed iteration. In general, the matrix-vector multiplication $A^{(j)}p_i^k$ cannot be computed cheaply. The cost of the method will be expensive in the general case where the matrices $A^{(j)}$ and $A^{(k)}$ are different (cf. Table 2). However, in section 3, we will consider two specific applications where the matrices $A^{(k)}$ and $A^{(j)}$ are structurally related. In these applications, the coefficient matrices differ by a parameterized identity matrix or a low-rank matrix. Therefore, the matrix-vector products $A^{(j)}p_i^k$ can be computed cheaply by using the matrix-vector product $A^{(k)}p_i^k$ generated from the seed iteration.

In order to reduce the extra cost in Projection Method I in the general case, we propose using the modified quadratic function \tilde{f}_j ,

$$\tilde{f}_j(x) \equiv \frac{1}{2}x^T A^{(k)}x - (b^{(j)})^T x,$$

to compute the approximate solution of the nonseed system. Note that we have used $A^{(k)}$ instead of $A^{(j)}$ in the above definition. In this case, we determine the next iterate of the nonseed system by solving the following minimization problem:

$$\min_{\alpha} \tilde{f}_j(x_i^j + \alpha p_i^k).$$

TABLE 1

Preconditioned version of Projection Methods I (left) and II (right). The k th system is the seed for the $(k - 1)$ th restart. The first and second superscripts are used to denote the k th restart and the j th system. The subscripts are used to denote the i th step of the CG method. Here $C^{(k)}$ is the preconditioner for $A^{(k)}$.

<pre> for $k=1, \dots, s$ until all the systems are solved Select the kth system as seed for $i=0, 1, 2, \dots, m_{k+1}$ % CG iteration for $j=k, k+1, \dots, s$ % unsolved systems if $j=k$ then perform usual CG steps $\delta_i^{k,k} = (z_i^{k,k})^T r_i^{k,k} / (z_{i-1}^{k,k})^T z_{i-1}^{k,k}$ $p_i^{k,k} = z_i^{k,k} + \delta_i^{k,k} p_{i-1}^{k,k}$ $\sigma_i^{k,k} = (z_i^{k,k})^T r_i^{k,k} / (p_i^{k,k})^T A^{(k)} p_i^{k,k}$ $x_{i+1}^{k,k} = x_i^{k,k} + \sigma_i^{k,k} p_i^{k,k}$ $r_{i+1}^{k,k} = r_i^{k,k} - \sigma_i^{k,k} A^{(k)} p_i^{k,k}$ $z_{i+1}^{k,k} = (C^{(k)})^{-1} r_{i+1}^{k,k}$ % preconditioning else perform Galerkin projection $\eta_i^{k,j} = (z_{i+1}^{k,k})^T r_i^{k,j} / (p_i^{k,k})^T A^{(j)} p_i^{k,k}$ $x_{i+1}^{k,j} = x_i^{k,j} + \eta_i^{k,j} p_i^{k,k}$ $r_{i+1}^{k,j} = r_i^{k,j} - \eta_i^{k,j} A^{(j)} p_i^{k,k}$ end if end for end for end for </pre>	<pre> for $k=1, \dots, s$ until all the systems are solved Select the kth system as seed for $i=0, 1, 2, \dots, m_{k+1}$ % CG iteration for $j=k, k+1, \dots, s$ % unsolved systems if $j=k$ then perform usual CG steps $\delta_i^{k,k} = (z_i^{k,k})^T r_i^{k,k} / (r_{i-1}^{k,k})^T z_{i-1}^{k,k}$ $p_i^{k,k} = z_i^{k,k} + \delta_i^{k,k} p_{i-1}^{k,k}$ $\sigma_i^{k,k} = (z_i^{k,k})^T r_i^{k,k} / (p_i^{k,k})^T A^{(k)} p_i^{k,k}$ $x_{i+1}^{k,k} = x_i^{k,k} + \sigma_i^{k,k} p_i^{k,k}$ $r_{i+1}^{k,k} = r_i^{k,k} - \sigma_i^{k,k} A^{(k)} p_i^{k,k}$ $z_{i+1}^{k,k} = (C^{(k)})^{-1} r_{i+1}^{k,k}$ % preconditioning else perform Galerkin projection $\eta_i^{k,j} = (z_{i+1}^{k,k})^T r_i^{k,j} / (p_i^{k,k})^T A^{(j)} p_i^{k,k}$ $x_{i+1}^{k,j} = x_i^{k,j} + \eta_i^{k,j} p_i^{k,k}$ $r_{i+1}^{k,j} = r_i^{k,j} - \eta_i^{k,j} A^{(j)} p_i^{k,k}$ end if end for end for end for </pre>
--	--

TABLE 2

Number of different operations per iteration of Projection Method I.

Operations	Usual PCG iteration	Projection Method I		
		General matrix $A^{(j)}$	$A^{(j)} = A^{(k)} + \mu I$	$A^{(j)} = A^{(k)} + \mu r r^T$
n -vector DAXPY	3	2	3	3
n -vector DOT PROD.	2	2	2	3
MXV $A^{(k)} x$	1	0	0	0
MXV $A^{(j)} x$	0	1	0	0
MXV $C^{(k)-1} x$	1	0	0	0

The approximate solution $\hat{x}^{(j)}$ of the nonseed system $A^{(j)} x^{(j)} = b^{(j)}$ is given by

$$(5) \quad \hat{x}^{(j)} = x_i^j + \alpha p_i^k,$$

where

$$(6) \quad \alpha = \frac{(p_i^k)^T \tilde{r}_i^j}{(p_i^k)^T A^{(k)} p_i^k} \quad \text{and} \quad \tilde{r}_i^j = b^{(j)} - A^{(k)} x_i^j.$$

Now the projection process does not require the matrix-vector product involving the coefficient matrix $A^{(j)}$ of the nonseed system. Therefore, the method does not increase the dominant cost (matrix-vector multiplies) of each CG iteration. We call this method *Projection Method II*. Of course, unless $A^{(j)}$ is close to $A^{(k)}$ in some sense, we do not expect this method to work well because \tilde{f}_j is then far from the current f_j .

To summarize the above methods, Table 1 lists the algorithms of Projection Methods I and II. The major computational costs during Projection Methods I and II are shown as in Tables 2 and 3, respectively.

3. Analysis of projection method. For Projection Methods I and II in the absence of preconditioning, we have the following lemma in exact arithmetic.

TABLE 3
Number of different operations per iteration of Projection Method II.

Operations	Usual PCG iteration	Projection Method II		
		General matrix $A^{(j)}$	$A^{(j)} = A^{(k)} + \mu I$	$A^{(j)} = A^{(k)} + \mu r r^T$
n -vector DAXPY	3	2	2	2
n -vector DOT PROD.	2	2	2	2
MXV $A^{(k)}x$	1	0	0	0
MXV $A^{(j)}x$	0	0	0	0
MXV $C^{(k)-1}x$	1	0	0	0

LEMMA 3.1. *Assume that a seed system $A^{(k)}x^{(k)} = b^{(k)}$ has been selected. Let $x_\ell^{k,j}$ be the ℓ th iterate of the nonseed system and V_i^k be the Lanczos vectors generated by i steps of the Lanczos algorithm.*

(a) *Using Projection Method I, the approximate solution of the nonseed system $A^{(j)}x^{(j)} = b^{(j)}$ at the i th iteration is given by*

$$(7) \quad x_i^{k,j} = x_0^{k,j} + V_i^k [(V_i^k)^T A^{(j)} V_i^k]^{-1} (V_i^k)^T (b^{(j)} - A^{(j)} x_0^{k,j}).$$

(b) *Using Projection Method II, the approximate solution of the nonseed system $A^{(j)}x^{(j)} = b^{(j)}$ at the i th iteration is given by*

$$(8) \quad x_i^{k,j} = x_0^{k,j} + V_i^k [(V_i^k)^T A^{(k)} V_i^k]^{-1} (V_i^k)^T (b^{(j)} - A^{(k)} x_0^{k,j}).$$

Proof. We remark that the proof of part (a) is similar to the proof of Lemma 3.1 in [5], except that V_i^k is not $A^{(j)}$ -orthogonal. (a) It is well-known that V_i^k satisfies the three-term recurrence

$$A^{(k)}V_i^k = V_i^k T_i^k + \beta_{i+1}^k v_{i+1}^k e_i^T,$$

where e_i is the i th column of the identity matrix and β_{i+1}^k is a scalar. From (3), the approximate solution $x_i^{k,j}$ of the nonseed system is computed in the subspace generated by the direction vectors $\{p_\ell^k\}$ generated from the seed iteration which is exactly the subspace spanned by the columns of V_i^k ; see [8]. Moreover, it is easy to check from (4) that

$$(p_\ell^k)^T (b^{(j)} - A^{(j)} x_i^{k,j}) = 0, \quad \ell = 1, 2, \dots, i.$$

It follows that the solution $x_i^{k,j}$ can be obtained by the Galerkin projection onto the Krylov subspace $\mathcal{K}^{(k)}$ generated by the seed system

$$(V_i^k)^T (b^{(j)} - A^{(j)} z), \quad \text{where } z = x_0^{k,j} + y, \text{ and } y \in V_i^{(k)},$$

which gives (7). Part (b) can be proved similarly. \square

To analyze the error bound of the projection method in the absence of preconditioning, without loss of generality, consider only two symmetric positive definite n -by- n linear systems:

$$A^{(1)}x^{(1)} = b^{(1)} \quad \text{and} \quad A^{(2)}x^{(2)} = b^{(2)}.$$

The eigenvalues and normalized eigenvectors of $A^{(i)}$ are denoted by $\lambda_k^{(i)}$ and $q_k^{(i)}$, respectively, and by $0 < \lambda_1^{(i)} \leq \lambda_2^{(i)} \leq \dots \leq \lambda_n^{(i)}$ for $i = 1, 2$. Theorem 3.2 gives

error bounds for solving multiple linear systems with different coefficient matrices and right-hand sides.

THEOREM 3.2. *Suppose the first linear system $A^{(1)}x^{(1)} = b^{(1)}$ is solved to the desired accuracy in m CG steps. Let $x_0^{1,2}$ be the solution of the second system $A^{(2)}x^{(2)} = b^{(2)}$ obtained from the projection onto \mathcal{K}_m generated by the first system, with zero vector as the initial guess of the second system ($x_0^{0,2} = 0$). Let the eigen-decomposition of $x^{(2)} - x_0^{1,2}$ be expressed as $x^{(2)} - x_0^{1,2} = \sum_{k=1}^n c_k q_k^{(2)}$, where $q_k^{(2)}$ is the unit eigenvector. Then we have the following.*

(a) *For Projection Method I, the eigenvector components c_k can be bounded by*

$$(9) \quad |c_k| \leq E_k + F + G, \quad 1 \leq k \leq n,$$

where

$$(10) \quad E_k = \|P_m^\perp x^{(2)}\|_2 |\sin \angle(q_k^{(2)}, \mathcal{K}_m)|,$$

$$F = \|(A^{(2)})^{-1}\|_2 \|(A^{(2)} - A^{(1)})x^{(2)}\|_2, \quad \text{and} \quad G = \|(A^{(1)})^{-1} - (A^{(2)})^{-1}\|_2 \|A^{(1)}x^{(2)}\|_2.$$

Here V_m is an orthonormal basis for \mathcal{K}_m , $P_m^\perp = I - V_m T_m^{-1} V_m^T A^{(1)}$ is the $A^{(1)}$ -orthogonal projection onto \mathcal{K}_m , and $T_m = V_m^T A^{(1)} V_m$ is the matrix representation of the projection of $A^{(1)}$ onto \mathcal{K}_m .

(b) *For Projection Method II, the eigenvector components c_k can be bounded by $|c_k| \leq E_k + \hat{F}$, $1 \leq k \leq n$, where E_k is defined in (10) and $\hat{F} = \|(A^{(1)})^{-1}\|_2 \|(A^{(2)} - A^{(1)})x^{(2)}\|_2$.*

Proof. Using $x_0^{0,2} = 0$ as the initial guess for the second system, by (7), we get $x_0^{1,2} \equiv x_m^{0,2} = V_m (V_m^T A^{(2)} V_m)^{-1} V_m^T b^{(2)}$. Then

$$\begin{aligned} x^{(2)} - x_0^{1,2} &= \left[I - V_m (V_m^T A^{(2)} V_m)^{-1} V_m^T A^{(2)} \right] x^{(2)} \\ &= P_m^\perp x^{(2)} - V_m (V_m^T A^{(2)} V_m)^{-1} V_m^T (A^{(2)} - A^{(1)}) x^{(2)} \\ &\quad + V_m \left[(A^{(1)})^{-1} - (A^{(2)})^{-1} \right] V_m^T A^{(1)} x^{(2)}. \end{aligned}$$

Since V_m is the orthogonal vector of \mathcal{K}_m , we have $\|V_m\|_2 \leq 1$. It follows that

$$\begin{aligned} |c_k| &= \left| (q_k^{(2)})^T \left\{ P_m^\perp x^{(2)} - V_m (V_m^T A^{(2)} V_m)^{-1} V_m^T (A^{(2)} - A^{(1)}) x^{(2)} \right. \right. \\ &\quad \left. \left. + V_m \left[(A^{(1)})^{-1} - (A^{(2)})^{-1} \right] V_m^T A^{(1)} x^{(2)} \right\} \right| \\ &\leq \|P_m^\perp x^{(2)}\|_2 \|q_k^{(2)}\|_2 |\cos \angle(q_k^{(2)}, P_m^\perp x^{(2)})| + \|(A^{(2)})^{-1}\|_2 \|(A^{(2)} - A^{(1)})x^{(2)}\|_2 \\ &\quad + \|(A^{(1)})^{-1} - (A^{(2)})^{-1}\|_2 \|A^{(1)}x^{(2)}\|_2. \end{aligned}$$

Knowing that $|\cos \angle(q_k^{(2)}, P_m^\perp x^{(2)})| \leq |\cos \angle(q_k^{(2)}, \mathcal{K}_m^\perp)| = |\sin \angle(q_k^{(2)}, \mathcal{K}_m)|$, we have (9). Part (b) can be proved similarly. \square

Theorem 3.2 basically states that the size of the eigenvector component c_k is bounded by E_k , F (or \hat{F}), and G . If the Krylov subspace \mathcal{K}_m generated by the seed system contains the eigenvectors $q_k^{(2)}$ well, then the projection process will kill off the eigenvector components of the initial error of the nonseed system, i.e., E_k is very small. On the other hand, F and G depend essentially on how different the system $A^{(2)}x^{(2)} = b^{(2)}$ is from the previous one, $A^{(1)}x^{(1)} = b^{(1)}$. In particular, when

$\|A^{(1)} - A^{(2)}\|_2$ and $\|(A^{(1)})^{-1} - (A^{(2)})^{-1}\|_2$ are small, then F (or \hat{F}) and G are also small.

We remark that when $A^{(1)} = A^{(2)}$ and $b^{(1)} \neq b^{(2)}$, the term F becomes zero, and as $q_k^{(1)} = q_k^{(2)}$, the term E_k becomes $\|P_m^\perp x^{(2)}\|_2 |\sin \angle(q_k^{(1)}, \mathcal{K}_m)|$; see Chan and Wan [5]. It is well known that the Krylov subspace \mathcal{K}_m generated by the seed system contains the eigenvectors $q_k^{(1)}$. In particular, Chan and Wan [5] have the following result about the estimate of the bound $\sin \angle(q_k^{(1)}, \mathcal{K}_m)$.

LEMMA 3.3. *Let $\theta_k = \angle(A^{(1)}b^{(1)}, q_k^{(1)})$, $\tau_k = \frac{(\lambda_k^{(1)} - \lambda_{k+1}^{(1)})}{(\lambda_{k+1}^{(1)} - \lambda_n^{(1)})}$, and*

$$\omega_k = \prod_{\nu=1}^{k-1} \left(\frac{\lambda_\nu^{(1)} - \lambda_n^{(1)}}{\lambda_\nu^{(1)} - \lambda_k^{(1)}} \right) / T_{m-k}(1 + 2\tau_k),$$

where $T_j(x)$ is the Chebyshev polynomial of degree j . Then

$$(11) \quad \sin \angle(q_k^{(1)}, \mathcal{K}_m) \leq \omega_k \tan \theta_k.$$

If we assume that the eigenvalues of $A^{(1)}$ are distinct, then $T_{m-k}(1 + 2\tau_k)$ grows exponentially as m increases and therefore the magnitude $\sin \angle(q_k^{(1)}, \mathcal{K}_m)$ is very small for sufficiently large m . It implies that the magnitude E_k is very small when m is sufficiently large. Unfortunately, we cannot have this result in the general case since $q_k^{(1)} \neq q_k^{(2)}$, except in some special cases that will be discussed in the next section.

3.1. Special cases. In this section, we consider using the Galerkin projection method for solving multiple linear systems arising in two particular situations: the coefficient matrices differ by a parameterized identity matrix or a low-rank matrix. We note from Theorem 3.2 that the theoretical error bound of the projection method depends on E_k , F , and G . In general, it is not easy to refine the error bound E_k , F , and G . However, in these cases, the error bound E_k , F , and G can be further investigated.

Case 1. Consider for simplicity two linear systems:

$$(\mu_i I + A)x_i \equiv A^{(i)}x_i = b^{(i)}, \quad i = 1, 2.$$

For instance, these kinds of linear systems arise in image restoration computations that we will discuss in the next section. In this case, we can employ Projection Method I to solve these multiple linear systems as the matrix-vector product $(\mu_2 I + A)p$ in the nonseed iteration can be computed cheaply by adding together $(\mu_1 I + A)p$ generated from the seed iteration and $(\mu_2 - \mu_1)p$ (cf. Table 2). Moreover, we can further refine the error bound of Projection Method II in Theorem 3.2. Now assume that m steps of the CG algorithm have been performed to solve the first system. We note in this case that the eigenvectors of the first and the second linear systems are the same, i.e., $q_k^{(1)} = q_k^{(2)}$. Therefore, we can bound $\sin \angle(q_k^{(1)}, \mathcal{K}_m)$ using Lemma 3.3. We shall prove that if the Krylov subspace of the first linear system contains the extreme eigenvectors well, the bound for the convergence rate is effectively the classical CG bound but with a reduced condition number.

THEOREM 3.4. *Let $x_0^{1,2}$ be the solution of the second system obtained from the projection onto \mathcal{K}_m generated by the first system. Then we have*

(a) for Projection Method I, the bound for the $A^{(2)}$ -norm of the error vector after i steps of the CG process is given by

$$\|x^{(2)} - x_i^{1,2}\|_{A^{(2)}}^2 \leq 4\|x^{(2)} - \bar{x}_0^{1,2}\|_{A^{(2)}}^2 \left(\frac{\sqrt{\kappa_r} - 1}{\sqrt{\kappa_r} + 1}\right)^{2i} + \delta,$$

where $\bar{x}_i^{1,2}$ is the i th iterate of the CG process for $A^{(2)}x^{(2)} = b^{(2)}$ with the projection of $x^{(2)} - x_0^{1,2}$ onto $\text{span}\{q_k^{(1)} : k = 1, 2, \dots, \ell\}^\perp$ as initial guess, $\kappa_r = \lambda_n^{(2)}/\lambda_{\ell+1}^{(2)}$ is the reduced condition number of $A^{(2)}$, and

$$(12) \quad \delta = \sum_{k=1}^{\ell} \lambda_k^{(2)} \left[\left| 1 - \frac{\mu_1}{\mu_2} \right| \|x^{(2)}\|_2 + \left| \frac{\mu_1 - \mu_2}{\mu_1 \mu_2} \right| \|A^{(1)}x^{(2)}\|_2 + \|P_m^\perp x^{(2)}\|_2^2 \omega_k \tan \theta_k \right]^2;$$

(b) for Projection Method II, the bound for the $A^{(2)}$ -norm of the error vector after i steps of the CG process is given by

$$\|x^{(2)} - x_i^{1,2}\|_{A^{(2)}}^2 \leq 4\|x^{(2)} - \bar{x}_0^{1,2}\|_{A^{(2)}}^2 \left(\frac{\sqrt{\kappa_r} - 1}{\sqrt{\kappa_r} + 1}\right)^{2i} + \hat{\delta},$$

where

$$(13) \quad \hat{\delta} = \sum_{k=1}^{\ell} \lambda_k^{(2)} \left[\left| 1 - \frac{\mu_2}{\mu_1} \right| \|x^{(2)}\|_2 + \|P_m^\perp x^{(2)}\|_2^2 \omega_k \tan \theta_k \right]^2.$$

Proof. We remark that the proof of part (a) is similar to the proof of Lemma 3.2 in [5], except that the quantity δ is different. By using properties of the CG iteration, we have

$$(14) \quad \begin{aligned} \|x^{(2)} - x_i^{1,2}\|_{A^{(2)}}^2 &\leq \|x^{(2)} - \bar{x}_i^{1,2}\|_{A^{(2)}}^2 + \sum_{k=1}^{\ell} \bar{p}_i^2(\lambda_k^{(2)}) c_k^2 \lambda_k^{(2)} \\ &\leq \|x^{(2)} - \bar{x}_0^{1,2}\|_{A^{(2)}}^2 \left(\frac{\sqrt{\kappa_r} - 1}{\sqrt{\kappa_r} + 1}\right)^{2i} + \sum_{k=1}^{\ell} \bar{p}_i^2(\lambda_k^{(2)}) c_k^2 \lambda_k^{(2)}, \end{aligned}$$

where \bar{p}_i is a polynomial of degree at most i and constant term 1; see [5, Equation 8]. Noting that $\|(A^{(2)})^{-1}\|_2 \leq 1/\mu_2$, $A^{(2)} - A^{(1)} = (\mu_2 - \mu_1)I$, $\|(A^{(1)})^{-1} - (A^{(2)})^{-1}\|_2 \leq |\mu_1 - \mu_2|/|\mu_1 \mu_2|$, and $\max_{1 \leq k \leq \ell} \bar{p}_i^2(\lambda_k^{(2)}) \leq 1$, see [5], and using Theorem 3.2 and Lemma 3.3, the result follows by substitution of (11) into (14). Similarly, we can prove the result for Projection Method II. \square

We see that for Projection Method I, the perturbation term δ contains two parts. One depends on the ratio μ_1/μ_2 and the difference $\mu_1 - \mu_2$ of the regularization parameters between two linear systems, and the other depends on how well the Krylov subspace of the seed system contains the extreme eigenvectors. We remark that the regularization parameter μ in practice is always greater than 0 in image restoration applications [3] because of the ill-conditioning of A . In particular, $\mu_1 \neq 0$. If the ratio μ_1/μ_2 is near to 1 and the difference $\mu_1 - \mu_2$ is near to 0, then the magnitude of this term will be near to zero. On the other hand, according to Lemma 3.3, the Galerkin projection will kill off the extreme eigenvector components and therefore the quantity $\omega_k \tan \theta_k$ in (12) will be also small for k close to 1. Hence the perturbation term δ

becomes very small and the CG method, when applied to solve the nonseed system, converges faster than the usual CG process. The above discussion can also be applied to Projection Method II.

We remark that the special case of shifted linear systems with identical right-hand sides has been considered in [7] and the references therein. The CG method can be performed on all systems simultaneously at the expense of one matrix-vector multiply only [7]. However, our projection method can be applied to more general systems, for example, the linear systems $(\mu_i B + A)x_i = b_i$ arising from Tikhonov regularization using the Laplacian as the regularization matrix; see [4].

Case 2. Consider two symmetric positive definite n -by- n linear systems:

$$A^{(1)}x^{(1)} = b^{(1)} \quad \text{and} \quad A^{(2)} = A^{(1)} + \rho r r^T = b^{(2)},$$

where r has unit 2-norm and each component is nonzero. For instance, these kinds of linear systems arise in recursive least squares computations that we will discuss in the next section. In this case, Projection Method I can be used to solve these multiple linear systems as the matrix-vector product in the nonseed iteration can be computed inexpensively. For instance, the matrix-vector product for the new system can be computed by

$$A^{(2)}p := p_1 + r r^T p,$$

where $p_1 := A^{(1)}p$ is generated from the seed iteration. The extra cost is some inner products (cf. Table 2).

By using the eigenvalue-eigenvector decomposition of $A^{(1)}$, we obtain

$$A^{(2)} = Q(\Lambda^{(1)} + \rho z z^T)Q^T,$$

with $Q = [q_1^{(1)} q_2^{(1)} \dots q_n^{(1)}]$, $\Lambda^{(1)}$ a diagonal matrix containing eigenvalues $\lambda_i^{(1)}$ of $A^{(1)}$, and $z = Q^T r$. It has been shown in [9] that if $\lambda_k^{(1)} \neq \lambda_k^{(2)}$ for all k , then the eigenvalues $\lambda_k^{(2)}$ of $A^{(2)}$ can be computed by solving the secular equation

$$\phi(\lambda) = 1 + \rho \sum_{i=1}^n \frac{[(q_i^{(1)})^T r]^2}{(\lambda_k^{(1)} - \lambda)} = 0.$$

Moreover, the eigenvectors $q_k^{(2)}$ of $A^{(2)}$ can be calculated by the formula

$$(15) \quad q_k^{(2)} = \frac{Q(\Lambda^{(1)} - \lambda_k^{(2)}I)^{-1}Q^T r}{\|(\Lambda^{(1)} - \lambda_k^{(2)}I)^{-1}Q^T r\|_2}, \quad 1 \leq k \leq n.$$

THEOREM 3.5. *Suppose the first linear system $A^{(1)}x^{(1)} = b^{(1)}$ is solved to the desired accuracy in m CG steps. Then we have*

(a) *for Projection Method I, the eigenvector components c_k of the second system are bounded by $|c_k| \leq E_k + F + G$, for $1 \leq k \leq n$, where*

$$(16) E_k = \|P_m^\perp x^{(2)}\|_2 \sum_{i=1}^n |\gamma_{i,k}| \left| \sin \angle(q_i^{(1)}, \mathcal{K}_m) \right| \quad \text{and} \quad F = |\rho| \|(A^{(2)})^{-1}\|_2 \|r^T x^{(2)}\|_2,$$

$$G = |\rho| \|(A^{(1)})^{-1}\|_2 \|A^{(1)}x^{(2)}\|_2 \quad \text{and} \quad \gamma_{i,k} = \frac{\frac{(q_i^{(1)})^T r}{(\lambda_i^{(1)} - \lambda_k^{(2)})}}{\sqrt{\sum_{j=1}^n \left[\frac{(q_j^{(1)})^T r}{\lambda_j^{(1)} - \lambda_k^{(2)}} \right]^2}},$$

where $\{q_i^{(1)}\}$ is the orthonormal eigenvector of $A^{(1)}$ and \mathcal{K}_m is the Krylov subspace generated for the first system;

(b) for Projection Method II, the eigenvector components c_k of the second system are bounded by $|c_k| \leq E_k + \hat{F}$, for $1 \leq k \leq n$, where E_k is defined as in (16) and $\hat{F} = |\rho| \|(A^{(1)})^{-1}\|_2 \|r^T x^{(2)}\|_2$.

Proof. (a) We just note from Theorem 3.2 that

$$|c_k| \leq |(P_m^\perp x^{(2)})^T \cdot q_k^{(2)}| + |\rho| \|(A^{(2)})^{-1}\|_2 \|r^T x^{(2)}\|_2 + |\rho| \|(A^{(1)})^{-1}\|_2 \|A^{(2)} x^{(2)}\|_2.$$

By using (15), Theorem 3.2, and Lemma 3.3, we can analyze the term $|(P_m^\perp x^{(2)})^T \cdot q_k^{(2)}|$ and obtain

$$\left| (P_m^\perp x^{(2)})^T \cdot q_k^{(2)} \right| = \sum_{i=1}^n |\gamma_{i,k}| \left| \cos(q_i^{(1)}, P_m^\perp x^{(2)}) \right| \leq \|P_m^\perp x^{(2)}\|_2 \sum_{i=1}^n |\gamma_{i,k}| \left| \sin \angle(q_i^{(1)}, \mathcal{K}_m) \right|.$$

(b) We note that the term G is equal to zero. \square

Since $|\gamma_{i,k}|$ and $|\sin \angle(q_i^{(1)}, \mathcal{K}_m)|$ are less than 1, we have

$$(17) \quad E_k \leq \|P_m^\perp x^{(2)}\|_2 \left[\sum_{\text{small and large } i} \left| \sin \angle(q_i^{(1)}, \mathcal{K}_m) \right| + \sum_{\text{remaining } i} |\gamma_{i,k}| \right].$$

From Lemma 3.3, for i close to 1 or n , $|\sin \angle(q_i^{(1)}, \mathcal{K}_m)|$ is sufficiently small when m is large. Moreover, we note that if $\rho > 0$, then

$$\lambda_k^{(1)} \leq \lambda_k^{(2)} \leq \lambda_{k+1}^{(1)}, \quad k = 1, 2, \dots, n-1, \quad \text{and} \quad \lambda_n^{(1)} \leq \lambda_n^{(2)} \leq \lambda_n^{(1)} + \rho.$$

If $\rho < 0$, then

$$\lambda_1^{(1)} - \rho \leq \lambda_1^{(2)} \leq \lambda_1^{(1)}, \quad \text{and} \quad \lambda_{k-1}^{(1)} \leq \lambda_k^{(2)} \leq \lambda_k^{(1)}, \quad k = 2, 3, \dots, n;$$

see [8]. Therefore, if the values $(q_i^{(1)})^T r$ are about the same magnitude for each eigenvector $q_i^{(1)}$, then the maximum value of $|\gamma_{i,k}|$ is attained at either $i = k$ or $i = k + 1$. We may expect that the second term of the inequality (17) is small when k is close to 1 or n . By combining these facts, we can deduce that E_k is also small when k is close to 1 or n . On the other hand, if the scalar ρ is small (i.e., the 2-norm of rank-1 matrix is small), then F and G are also small.

4. Numerical results. In this section, we provide some experimental results of using Projection Methods I and II to solve multiple linear systems (1). More numerical examples can be found in [4]. All the experiments are performed in MATLAB with machine precision 10^{-16} . The stopping criterion is $\|r_i^{k,j}\|_2 < tol \times \|b^{(j)}\|_2$, where tol is the tolerance we used. The first and the second examples are Tikhonov regularization in image restoration and the recursive least squares estimation, as discussed in [3] and [17]. The coefficient matrices $A^{(i)}$'s have the same eigenvectors in Example 1. In Example 2, the coefficient matrices $A^{(i)}$'s differ by rank-1 or rank-2 matrices. We will see that the extremal eigenvector components of the right-hand sides are effectively reduced after the projection process. Moreover, the number of iterations required for convergence when we employ the projected solution as initial guess is less than that required in the usual CG process. In Example 3, we consider more general

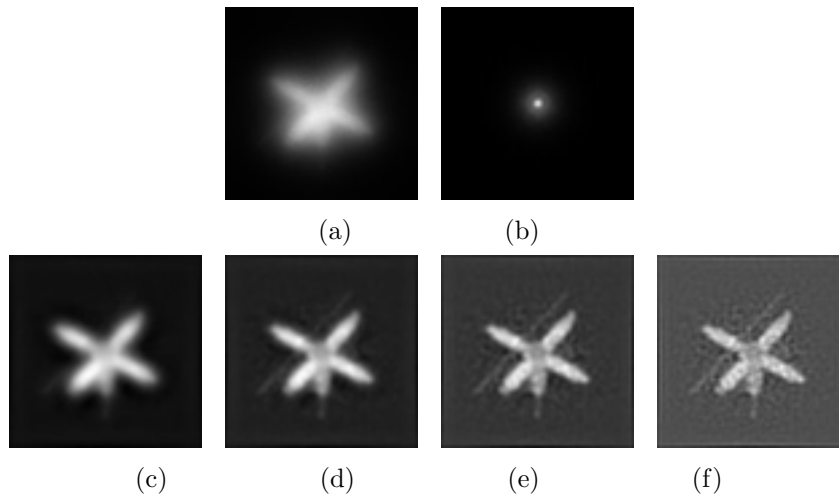


FIG. 1. (Example 1) Observed image (a), guide star image (b), restored images using regularization parameter $\mu =$ (c) 0.072, (d) 0.036, (e) 0.018, and (f) 0.009.

coefficient matrices, i.e., the consecutive linear systems do not differ by the scaled identity matrix and rank-1 or rank-2 matrices. In this example, the matrix-vector products for the nonseed iteration may not be computed cheaply; therefore we apply Projection Method II only to solve the multiple linear system. However, the same phenomenon as in Examples 1 and 2 is observed in this example as well.

Example 1 ($tol = 10^{-4}$). We consider a two-dimensional deconvolution problem arising in ground-based atmospheric imaging and try to remove the blurring in an image (see Figure 1(a)) resulting from the effects of atmospheric turbulence. The problem consists of a 256-by-256 image of an ocean reconnaissance satellite observed by a simulated ground-based imaging system together with a 256-by-256 image of a guide star (Figure 1(b)) observed under similar circumstances. The data are provided by the Phillips Air Force Laboratory at Kirkland AFB, NM, through Professor Bob Plemmons at Wake Forest University, Winston-Salem, NC. We restore the image using the identity matrix as the regularization operator suggested in [3] and therefore solve the linear systems

$$(\mu_i I + A)x_i = b_i$$

with different regularization parameters μ_i ; see [4]. The regularization parameter μ controls the degree of smoothness of the solution. Choosing μ is not a trivial problem. In some cases a priori information about the signal and the degree of perturbations in the right-hand side can be used to choose μ [1]; if no a priori information is known, then it may be necessary to solve the linear systems for several values of μ .

Table 4 shows the number of matrix-vector multiplies required for the convergence of all the systems. Using the projection method, we save on the number of matrix-vector multiplies in the iterative process with or without preconditioning. From Table 4, we also see that the performance of Projection Method I is better than that of Projection Method II. For comparison, we present the restorations of the images when the regularization parameters are 0.072, 0.036, 0.018, and 0.009 in Figure 1. We see that when the value of μ is large, the restored image is very smooth, while when

TABLE 4

(Example 1) Number of matrix-vector multiplies required for convergence of all the systems. Regularization parameter $\mu = (1) 0.072, (2) 0.036, (3) 0.018, \text{ and } (4) 0.009$.

Linear systems	(1)	(2)	(3)	(4)	Total
Starting with Projection Method I	36	40	43	49	168
Starting with Projection Method II	36	48	55	76	205
Starting with previous solution	36	54	66	87	243

TABLE 5

(Example 1) Number of flops required for solving the second linear system using different methods.

	Projection	Conjugate gradient		Total
		Matrix-vector Products	Others	
Starting with Projection Method I	8.5×10^4	1.8×10^9	1.0×10^5	1.8002×10^9
Starting with Projection Method II	6.4×10^4	2.45×10^9	1.2×10^5	2.4502×10^9
Starting with previous solution	0	2.75×10^9	1.38×10^5	2.7501×10^9

the value of μ is small, the noise is amplified in the restored image. By solving these multiple linear systems successively by the projection method, we can select Figure 1(e) which presents the restored image better than the others.

To further show the effectiveness of the projection method, we see from Table 5 that the dominant cost is matrix-vector multiplications per iteration (cf. Tables 2 and 3). According to these results, projection methods can reduce the cost of solving multiple linear systems in this example.

Example 2 ($tol = 10^{-8}$). In this example, we test the performance of Projection Methods I and II in the block (sliding window and exponentially-weighted) recursive least squares (RLS) computations. We illustrate the convergence rate of the method by using the adaptive finite impulse response (FIR) system identification model; see [12]. The second-order autoregressive process $x_t + 0.8x_{t-1} + 0.1x_{t-2} = v_t$, where $\{v_t\}$ is a white noise process with variance being 1, is used to construct the data matrix. The reference (unknown) system is an n th order FIR filter. The Gaussian white noise measurement error with variance 0.025 is added into the desired response. In the tests, the forgetting factor β is 0.99 and the order n of filter is 100.

In the case of the exponentially-weighted RLS computations, the consecutive systems differ by a rank-1 positive semidefinite matrix,

$$A^{(1)}x^{(1)} = b^{(1)} \quad \text{and} \quad A^{(2)}x^{(2)} = [A^{(1)} + \beta^{-1}x(t)x(t)^T]x^{(2)} = b^{(2)},$$

whereas in the case of the sliding window computations, the consecutive systems differ by the sum of a rank-1 positive definite matrix and a rank-1 negative definite matrix,

$$A^{(1)}x^{(1)} = b^{(1)} \quad \text{and} \quad A^{(2)}x^{(2)} = [A^{(1)} + x(t)x(t)^T - x(t-k)x(t-k)^T]x^{(2)} = b^{(2)}.$$

Here $x(t) = [x_t, x_{t-1}, \dots, x_{t-n+1}]^T$ and k is the length of sliding window. For details, we refer the reader to [4, 12]. Table 6 lists the number of matrix-vector multiplies required for the convergence of all the systems arising from exponentially weighted and sliding window RLS computations. We observe that the performance of Projection

TABLE 6

(Example 2) Number of matrix-vector multiplies required for convergence of all the systems.
 (a) Exponentially-weighted RLS computations and (b) sliding window RLS computations.

Linear systems	(1)	(2)	(3)	(4)	(5)	Total
Starting with Projection Method I	45	31	28	25	24	153
Starting with Projection Method II	45	37	32	31	29	174
Starting with previous solution	45	43	44	42	40	214

(a)

Linear systems	(1)	(2)	(3)	(4)	(5)	Total
Starting with Projection Method I	68	51	45	36	30	165
Starting with Projection Method II	68	55	49	42	35	249
Starting with previous solution	68	61	59	56	54	308

(b)

TABLE 7

(Example 3) Number of matrix-vector multiplies required for convergence of all the systems with $c_k = 0.1551 \times 0.9524^k$ and $d_k = 7.7566 \times 0.9524^k$.

Linear systems	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	Total
Starting with Projection Method II	83	82	80	75	62	53	47	21	26	24	553
Starting with previous solution	83	83	83	83	83	83	83	83	84	83	831

Method I is better than that of Projection Method II. The projection method requires fewer matrix-vector multiplies than using the previous solution as an initial guess. We find that matrix-vector multiplies always count for 99% or more of the total flop count. Projection methods can reduce the cost of solving multiple linear systems. Moreover, we find that the eigenvector components of $b^{(2)}$ are effectively reduced after projection in both cases of exponentially weighted and sliding window RLS computations. We see that the decreases of eigenvector components when using Projection Method I are indeed greater than those obtained when using Projection Method II; see the figures in [4].

Example 3 ($tol = 10^{-7}$). The matrices for the final set of experiments arise from the three-point centered discretization of the operator $-\frac{d}{dx}(a(x)\frac{du}{dx})$ in $[0, 1]$, where the function $a(x)$ is given by $a(x) = c + dx$, where c and d are two parameters. The discretization is performed using a grid size of $h = 1/65$, yielding matrices of size 64 with different values of c and d . The right-hand sides of these systems are generated randomly with their 2-norms being 1. We remark that the consecutive linear systems do not differ by low-rank or small norm matrices in this example.

Table 7 shows the number of iterations required for convergence of all the systems using Projection Method II and using a previous solution as an initial guess having the same residual norm. We observe from Figure 2 that the eigenvector components are effectively reduced after the projection.

5. Concluding remarks. In this paper, we developed Galerkin projection methods for solving multiple linear systems. Experimental results show that the method is an efficient method. We end with concluding remarks about the extensions of the Galerkin projection method.

(i) A block generalization of the Galerkin projection method can be employed

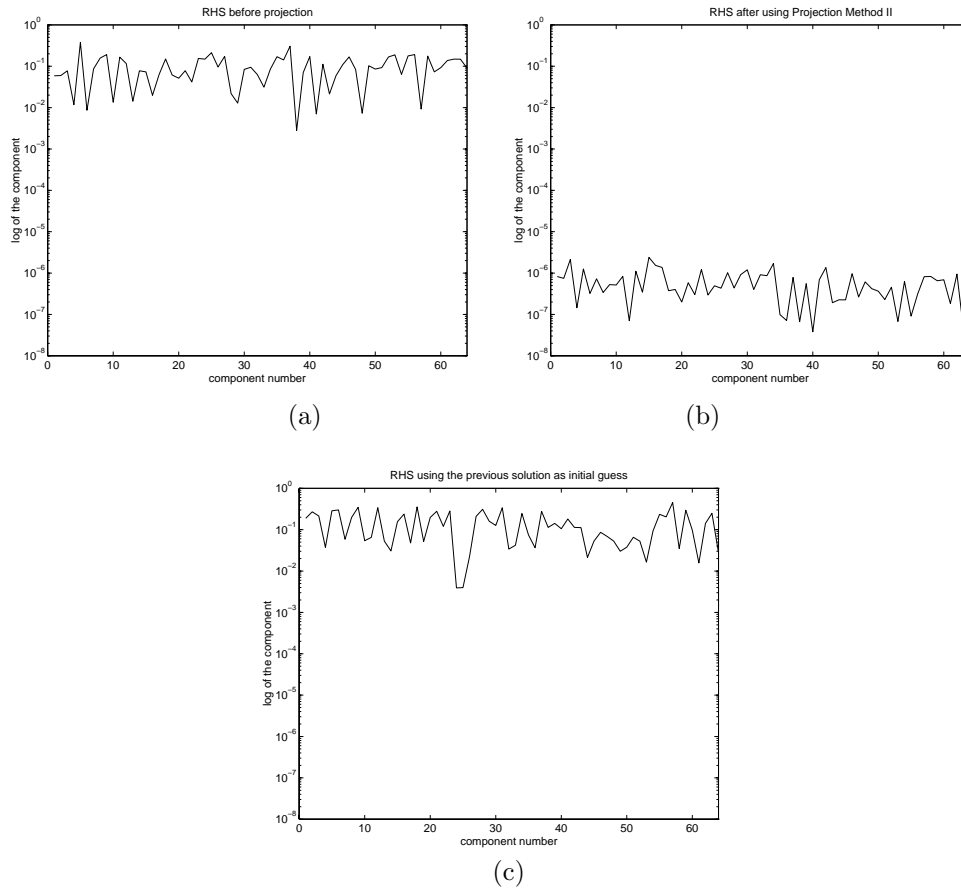


FIG. 2. (Example 3) Size distribution of the components of (a) the original right-hand side $b^{(7)}$, (b) $b^{(7)}$ after Galerkin projection and (c) $b^{(7)} - A^{(7)}x^{(6)}$ (using the previous solution as an initial guess).

in many applications. The method is to select more than one system as seed so that the Krylov subspace generated by the seed is larger and the initial guess obtained from the Galerkin projection onto this subspace is expected to be better. One drawback of the block method is that it may break down when singularity of the matrices arising from the CG process is encountered. For details about block Galerkin projection methods, we refer to Chan and Wan [5].

- (ii) In the literature, there are some methods for solving nonsymmetric systems with multiple right-hand sides. Three methods that have been proposed are block generalizations of solvers for nonsymmetric systems: the block biconjugate gradient algorithm [14, 13], block GMRES [22], block QMR [2, 6]. Recently, Simoncini and Gallopoulos [20] proposed a hybrid method by combining the Galerkin projection process and Richardson acceleration technique to speed up the convergence rate of the CG process. In the same spirit, we can modify the above Galerkin projection algorithms to solve nonsymmetric systems with multiple coefficient matrices and right-hand sides.

REFERENCES

- [1] J. ABBISS AND P. EARWICKER, *Compact operator equations, regularization and super-resolution*, in *Mathematics in Signal Processing*, Clarendon Press, Oxford, 1987.
- [2] W. E. BOYSE AND A. A. SEIDL, *A Block-QMR method for computing multiple simultaneous solutions to complex symmetric systems*, *SIAM J. Sci. Comput.*, 17 (1996), pp. 263–274.
- [3] R. CHAN, M. NG, AND R. PLEMMONS, *Generalization of Strang’s preconditioner with applications to Toeplitz least squares problems*, *Numer. Linear Algebra Appl.*, 3 (1996), pp. 45–64.
- [4] T. CHAN AND M. NG, *Galerkin Projection Methods for Solving Multiple Linear Systems*, CAM Rep. 96-31, Department of Math., University of California, Los Angeles, CA, 1996.
- [5] T. F. CHAN AND W. L. WAN, *Analysis of projection methods for solving linear systems with multiple right-hand sides*, *SIAM J. Sci. Comput.*, 18 (1997), pp. 1698–1721.
- [6] R. FREUND AND M. MALHOTRA, *A Block-QMR Algorithm for Non-Hermitian Linear Systems with Multiple Right-Hand Sides*, Manuscript SCCM-96-01, Scientific Computing and Computational Mathematics Program, Stanford University, Stanford, CA, 1996.
- [7] A. FROMMER AND P. MAASS, *Fast CG-based methods for Tikhonov-Phillips regularization*, *SIAM J. Sci. Comput.*, to appear.
- [8] G. GOLUB AND C. LOAN, *Matrix Computations*, 2nd ed., Johns Hopkins Press, Baltimore, MD., 1989.
- [9] G. GOLUB, *Some modified matrix eigenvalue problems*, *SIAM Rev.*, 15 (1973), pp. 318–334.
- [10] A. JAIN, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [11] R. KRESS, *Linear Integral Equations*, Springer-Verlag, New York, 1989.
- [12] M. NG AND R. PLEMMONS, *Fast RLS adaptive filtering by FFT-based conjugate gradient iterations*, *SIAM J. Sci. Comput.*, 17 (1996), pp. 154–170.
- [13] A. NIKISHIN AND A. YEREMIN, *Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers, I: General iterative scheme*, *SIAM J. Matrix Anal. Appl.*, 16 (1995), pp. 1831–1850.
- [14] D. O’LEARY, *The block conjugate gradient algorithm and related methods*, *Linear Algebra Appl.*, 29 (1980), pp. 293–322.
- [15] M. PAPADRAKAKIS AND S. SMEROU, *A new implementation of the Lanczos method in linear problems*, *Internat. J. Numer. Methods Engrg.*, 29 (1990), pp. 141–159.
- [16] B. PARLETT, *A new look at the Lanczos algorithm for solving symmetric systems of linear equations*, *Linear Algebra Appl.*, 29 (1980), pp. 323–346.
- [17] R. PLEMMONS, *FFT-based RLS in Signal Processing*, *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing 3, 1993*, Springer, New York, pp. 571–574.
- [18] Y. SAAD, *On the Lanczos method for solving symmetric linear systems with several right-hand sides*, *Math. Comp.*, 48 (1987), pp. 651–662.
- [19] V. SIMONCINI AND E. GALLOPOULOS, *A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides*, *J. Comput. Appl. Math.*, 66 (1996), pp. 457–469.
- [20] V. SIMONCINI AND E. GALLOPOULOS, *An iterative method for nonsymmetric systems with multiple right-hand sides*, *SIAM J. Sci. Comput.*, 16 (1995), pp. 917–933.
- [21] C. SMITH, A. PETERSON, AND R. MITTRA, *A conjugate gradient algorithm for the treatment of multiple incident electromagnetic fields*, *IEEE Trans. Antennas and Propagation*, 37 (1989), pp. 1490–1493.
- [22] B. VITAL, *Etude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocesseur*, Ph.D. thesis, Université de Rennes, Rennes, France, 1990.
- [23] H. VAN DER VORST, *An iteration solution method for solving $f(A) = b$, using Krylov subspace information obtained for the symmetric positive definite matrix A* , *J. Comput. Appl. Math.*, 18 (1987), pp. 249–263.

Copyright of SIAM Journal on Scientific Computing is the property of Society for Industrial and Applied Mathematics and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.