NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2001; **8**:297–316 (DOI: 10.1002/nla.241)

A parallel balance scheme for banded linear systems

Gene H. Golub[1,‡], Ahmed H. Sameh[2,§] and Vivek Sarin[3,*,†]

[1]*Department of Computer Science, Stanford University, Stanford, CA 94305-9025, U.S.A.*
[2]*Department of Computer Science, Purdue University, West Lafayette, IN 47907-1398, U.S.A.*
[3]*Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, U.S.A.*

SUMMARY

A parallel algorithm is proposed for the solution of narrow banded non-symmetric linear systems. The linear system is partitioned into blocks of rows with a small number of unknowns common to multiple blocks. Our technique yields a reduced system defined only on these common unknowns which can then be solved by a direct or iterative method. A projection based extension to this approach is also proposed for computing the reduced system implicitly, which gives rise to an inner–outer iteration method. In addition, the product of a vector with the reduced system matrix can be computed efficiently on a multiprocessor by concurrent projections onto subspaces of block rows. Scalable implementations of the algorithm can be devized for hierarchical parallel architectures by exploiting the two-level parallelism inherent in the method. Our experiments indicate that the proposed algorithm is a robust and competitive alternative to existing methods, particularly for difficult problems with strong indefinite symmetric part.

KEY WORDS: sparse linear systems; iterative methods; inner–outer; projections; parallel computation

1. INTRODUCTION

A number of applications in science and engineering give rise to non-symmetric systems of linear equations. Such systems are extremely large and sparse, and require the use of multiprocessors to compute the solution within reasonable time. A variety of iterative methods have been proposed to solve such linear systems (see, e.g., References [1, 2]).

In this paper, we present a new approach to solve a non-symmetric banded linear system that is sparse within the band by reducing it to a smaller system. The rows of the system

*Correspondence to: V. Sarin, Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, U.S.A.
†E-mail: sarin@cs.tamu.edu
‡E-mail: golub@sccm.stanford.edu
§E-mail: sameh@cs.purdue.edu

Contract/grant sponsor: National Science Foundation, research grant CCR; contract/grant number: CCR 9972533
Contract/grant sponsor: research grant ECS; contract/grant number: ECS 9527123
Contract/grant sponsor: infrastructure grant NSF EIA; contract/grant number: NSF EIA 9871053

Received 25 August 1997
Revised 23 November 2000

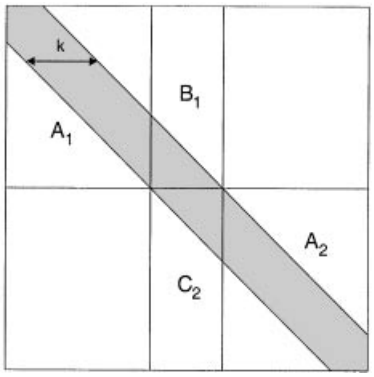


Figure 1. Balance scheme applied to a matrix with two block rows.

matrix are partitioned into p blocks, and a reduced system is obtained which is defined on a small set of unknowns common to multiple blocks. The reduced system may be solved using either a direct or iterative method. An extension of the algorithm is also proposed, in which the reduced system is available only implicitly in the form of a matrix–vector product. Such computation requires projections onto null spaces of the block rows, and can be computed concurrently. We exploit this feature to obtain an efficient parallel formulation of our algorithm.

Several parallel algorithms have been proposed over the years to solve general linear systems [1–6]. Our approach is similar to the one described in Reference [7].

The paper is organized as follows: Section 2 outlines the balance scheme for two blocks followed by a generalization to p blocks. Section 3 provides an upper bound on the condition number of the reduced system and estimates the storage requirements. In Section 4, we present a projection-based extension of the balance scheme which does not require the QR factorization step. Parallel implementation is discussed in Section 5. The experiments presented in Section 6 highlight the advantages of the balance scheme for a wide variety of indefinite banded matrices that are sparse within the band. Conclusions are presented in Section 7 and MATLAB code for the balance scheme is included in Appendix.

2. THE BALANCE SCHEME

We consider the solution of the sparse nonsingular linear system

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

where A is an $n \times n$ banded matrix with bandwidth k. For simplicity, let us partition the rows into two blocks, as shown in Figure 1.

The linear system can be represented as

$$\begin{pmatrix} A_1 & B_1 & \\ & C_2 & A_2 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \xi \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix} \tag{2}$$

where $A_1 \in \mathbf{R}^{m_1 \times n_1}$, $B_1 \in \mathbf{R}^{m_1 \times k}$, $A_2 \in \mathbf{R}^{m_2 \times n_2}$, and $C_2 \in \mathbf{R}^{m_2 \times k}$. Furthermore, ξ denotes the unknowns common to both block rows. These block rows give rise to the following set of underdetermined system of equations

$$(A_1 \quad B_1)\begin{pmatrix} \mathbf{x}_1 \\ \xi \end{pmatrix} = \mathbf{f}_1 \tag{3}$$

$$(C_2 \quad A_2)\begin{pmatrix} \tilde{\xi} \\ \mathbf{x}_2 \end{pmatrix} = \mathbf{f}_2 \tag{4}$$

that can be solved independently. However, the solution of the global system is obtained only when $\xi = \tilde{\xi}$.

Let us denote the submatrices for the block rows as follows:

$$E_1 = (A_1, B_1)$$
$$E_2 = (C_2, A_2)$$

The general form of the solution of the underdetermined systems (3)–(4) is given as

$$\mathbf{x} = \mathbf{p}_i + Q_i \mathbf{y}, \quad i = 1, 2$$

where $\mathbf{p}_i$ is a particular solution, Q_i is a basis for $\mathcal{N}(E_i)$, and $\mathbf{y}$ is an arbitrary vector. Note that $\mathbf{p}_i \in \mathbf{R}^{(n_i+k)}$ and $Q_i \in \mathbf{R}^{(n_i+k)\times(n_i+k-m_i)}$. Thus,

$$\begin{pmatrix} \mathbf{x}_1 \\ \xi \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{1,1} \\ \mathbf{p}_{1,2} \end{pmatrix} + \begin{pmatrix} Q_{1,1} \\ Q_{1,2} \end{pmatrix} \mathbf{y}_1 \tag{5}$$

$$\begin{pmatrix} \tilde{\xi} \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{2,1} \\ \mathbf{p}_{2,2} \end{pmatrix} + \begin{pmatrix} Q_{2,1} \\ Q_{2,2} \end{pmatrix} \mathbf{y}_2 \tag{6}$$

where $\mathbf{y}_1 = \mathbf{R}^{n_1+k-m_1}$ and $\mathbf{y}_2 = \mathbf{R}^{n_2+k-m_2}$. Clearly, the underdetermined linear systems (3)–(4) are consistent under the assumption that the submatrices E_1 and E_2 are full rank. Enforcing the condition for the solution of the global system:

$$\xi(\mathbf{y}_1) = \tilde{\xi}(\mathbf{y}_2)$$

we get

$$\mathbf{p}_{1,2} + Q_{1,2}\mathbf{y}_1 = \mathbf{p}_{2,1} + Q_{2,1}\mathbf{y}_2$$

As a result, we now solve the *reduced* linear system

$$M\mathbf{y} = \mathbf{g}$$

of order k where $M = (Q_{1,2}, -Q_{2,1})$, $\mathbf{y} = (\mathbf{y}_1^{\mathrm{T}}, \mathbf{y}_2^{\mathrm{T}})^{\mathrm{T}}$, and $\mathbf{g} = \mathbf{p}_{2,1} - \mathbf{p}_{1,2}$.

Balance scheme with two blocks for solving the global system (2) consists of the following steps:

1. Solve the underdetermined systems (3)–(4) to obtain $\mathbf{p}_1$, Q_1, $\mathbf{p}_2$, and Q_2.

2. Solve the reduced system $M\mathbf{y} = \mathbf{g}$.
3. Back-substitute $\mathbf{y}$ to determine $\mathbf{x}_1$, $\mathbf{x}_2$, and ξ.

The first step requires the solution of underdetermined systems of equations of the form

$$B\mathbf{z} = \mathbf{d}$$

where B is an $m \times n$ matrix ($m < n$) with full row rank. The unique solution of minimum norm $\bar{z}$, and the general solution $\mathbf{z}$ are given as

$$\begin{aligned} \mathbb{N} &= B^{+}\mathbf{d} \\ \mathbf{z} &= B^{+}\mathbf{d} + P_{\mathcal{N}(B)}\mathbf{u} \end{aligned}$$

where $B^{+} = B^{\mathrm{T}}(BB^{\mathrm{T}})^{-1}$ is the generalized inverse of B, and $P_{\mathcal{N}(B)}$ is the orthogonal projector onto $\mathcal{N}(B)$. Furthermore, the projector $P_{\mathcal{N}(B)}$ can be expressed using B^{+} as follows:

$$P_{\mathcal{N}(B)} = I - B^{+}B$$

There are a number of techniques to obtain the general solution $\mathbf{x}$ (see, e.g., References [8, 9]). Using the sparse QR decomposition approach [10],

$$B^{\mathrm{T}} = (Q_1, Q_2)\begin{pmatrix} R \\ 0 \end{pmatrix}$$

where R is an $m \times m$ matrix and Q_2 is an $n \times (n-m)$ matrix. Since Q_2 is a basis for $\mathcal{N}(B)$, the general solution is given as

$$\begin{aligned} \bar{z} &= Q_1 R^{-\mathrm{T}}\mathbf{d} \\ \mathbf{z} &= \bar{z} + Q_2\mathbf{y} \end{aligned}$$

The reduced system in the second step can be solved using direct or iterative methods. A more detailed discussion is presented in the next section.

2.1. Balance scheme with p blocks

The generalization of the balance scheme for p block rows is quite straightforward. Consider an $n \times n$ banded matrix A whose rows have been partitioned into p blocks (see, e.g., Figure 2). The linear system can be represented in the following block form:

$$\begin{pmatrix} A_1 & B_1 & & & & & & & \\ & C_2 & A_2 & B_2 & & & & & \\ & & & C_3 & A_3 & B_3 & & & \\ & & & & & \ddots & \ddots & & \\ & & & & & & C_{p-1} & A_{p-1} & B_{p-1} \\ & & & & & & & C_p & A_p \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \xi_1 \\ \mathbf{x}_2 \\ \xi_2 \\ \vdots \\ \xi_{p-1} \\ \mathbf{x}_p \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \vdots \\ \mathbf{f}_{p-1} \\ \mathbf{f}_p \end{pmatrix} \tag{7}$$

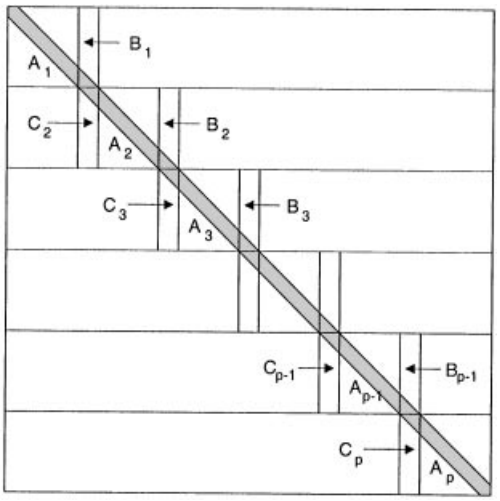


Figure 2. Balance scheme applied to a matrix with p block rows.

With p block rows, the columns of A are divided into $2p-1$ blocks, and the unknowns ξ_i, $i=1,\ldots,p-1$, are common to consecutive blocks of the matrix. Let us assume that $A_i \in \mathbf{R}^{m_i \times n_i}$, $B_i \in \mathbf{R}^{m_i \times k_i}$, and $C_i \in \mathbf{R}^{m_i \times k_{i-1}}$. Since $A \in \mathbf{R}^{n \times n}$, we have

$$n = \sum_{i=1}^{p} m_i = k + \sum_{i=1}^{p} n_i$$

where $k = \sum_{i=1}^{p-1} k_i$. The p block rows give rise to the following set of underdetermined system of equations:

$$(A_1 \quad B_1)\begin{pmatrix} \mathbf{x}_1 \\ \xi_1 \end{pmatrix} = \mathbf{f}_1 \tag{8}$$

$$(C_i \quad A_i \quad B_i)\begin{pmatrix} \tilde{\xi}_{i-1} \\ \mathbf{x}_i \\ \xi_i \end{pmatrix} = \mathbf{f}_i, \quad i=2,\ldots,p-1 \tag{9}$$

$$(C_p \quad A_p)\begin{pmatrix} \tilde{\xi}_{p-1} \\ \mathbf{x}_p \end{pmatrix} = \mathbf{f}_p \tag{10}$$

Denoting the submatrices for the block rows by

$$E_1 = (A_1, B_1)$$
$$E_i = (C_i, A_i, B_i), \quad i=2,\ldots,p-1$$
$$E_p = (C_p, A_p)$$

the general solution of the ith system is given as

$$\mathbf{x} = \mathbf{p}_i + Q_i \mathbf{y}$$

where $\mathbf{p}_i$ is a particular solution, Q_i is a basis for $\mathcal{N}(E_i)$, and $\mathbf{y}$ is an arbitrary vector. The solution of the systems (8)–(10) is

$$\begin{pmatrix} \mathbf{x}_1 \\ \xi_1 \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{1,1} \\ \mathbf{p}_{1,2} \end{pmatrix} + \begin{pmatrix} Q_{1,1} \\ Q_{1,2} \end{pmatrix} \mathbf{y}_1 \tag{11}$$

$$\begin{pmatrix} \tilde{\xi}_{i-1} \\ \mathbf{x}_i \\ \xi_i \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{i,1} \\ \mathbf{p}_{i,2} \\ \mathbf{p}_{i,3} \end{pmatrix} + \begin{pmatrix} Q_{i,1} \\ Q_{i,2} \\ Q_{i,3} \end{pmatrix} \mathbf{y}_i, \quad i = 2, \ldots, p-1 \tag{12}$$

$$\begin{pmatrix} \tilde{\xi}_{p-1} \\ \mathbf{x}_p \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{p,1} \\ \mathbf{p}_{p,2} \end{pmatrix} + \begin{pmatrix} Q_{p,1} \\ Q_{p,2} \end{pmatrix} \mathbf{y}_p \tag{13}$$

where $\mathbf{y}_1 = \mathbf{R}^{n_1+k_1-m_1}$, $\mathbf{y}_i = \mathbf{R}^{n_i+k_{i-1}+k_i-m_i}$, and $\mathbf{y}_p = \mathbf{R}^{n_p+k_{p-1}-m_p}$. Since the underdetermined linear systems (8)–(10) are consistent when the submatrices E_i are full rank, we can enforce the following conditions on the solution of the global system:

$$\xi_i(\mathbf{y}_i) = \tilde{\xi}_{i-1}(\mathbf{y}_{i+1}), \quad i = 1, \ldots, p-1$$

This gives rise to the system of equations

$$\mathbf{p}_{1,2} + Q_{1,2}\mathbf{y}_1 = \mathbf{p}_{2,1} + Q_{2,1}\mathbf{y}_2$$

$$\mathbf{p}_{i,3} + Q_{i,3}\mathbf{y}_i = \mathbf{p}_{i+1,1} + Q_{i+1,1}\mathbf{y}_{i+1}, \quad i = 2, \ldots, p-1$$

which yields the reduced linear system

$$M\mathbf{y} = \mathbf{g}$$

Over here, M is a $k \times k$ matrix given by

$$M = \begin{pmatrix} Q_{1,2} & -Q_{2,1} & & & & \\ & Q_{2,3} & -Q_{3,1} & & & \\ & & Q_{3,3} & -Q_{4,1} & & \\ & & & \ddots & \ddots & \\ & & & & Q_{p-1,3} & -Q_{p,1} \end{pmatrix} \tag{14}$$

and the right-hand side is given by

$$\mathbf{g} = \begin{pmatrix} \mathbf{p}_{2,1} - \mathbf{p}_{1,2} \\ \mathbf{p}_{3,1} - \mathbf{p}_{2,3} \\ \vdots \\ \mathbf{p}_{p,1} - \mathbf{p}_{p-1,3} \end{pmatrix}$$

Thus, the banded linear system (7) may be solved by computing the solutions of the underdetermined systems (8)–(10) followed by solving the reduced system shown above. Next, we outline the balance scheme for p blocks.

Algorithm *Balance_Scheme*

1. Solve the underdetermined systems (8)–(10) to obtain $\mathbf{p}_i$ and Q_i, $i = 1, \ldots, p$.
2. Solve the reduced system $M\mathbf{y} = \mathbf{g}$.
3. Back-substitute $\mathbf{y}$ to determine all $\mathbf{x}_i$ and ξ_i.

The general solution of the underdetermined systems in Step 1 has been discussed previously. The reduced system in Step 2 can be solved in a number of ways. The matrix M can be computed explicitly, and the linear system solved using either direct or iterative algorithms. When using a direct method, it is advantageous to exploit the block bidiagonal structure of the reduced system. Iterative methods may be used when storage is at a premium. Instead of explicitly computing M, matrix–vector products with M can be computed at each iteration by obtaining the residual

$$\mathbf{r}(\mathbf{y}) = \mathbf{g} - M\mathbf{y} = \tilde{\xi}(\mathbf{y}) - \xi(\mathbf{y})$$

and computing $M\mathbf{y} = \mathbf{r}(\mathbf{0}) - \mathbf{r}(\mathbf{y})$, where $\mathbf{r}(\mathbf{0}) = \mathbf{g}$. In this case, however, one needs to solve the underdetermined systems at each iteration. An alternate projection based approach for solving the reduced system implicitly via an iterative method is outlined in Section 4.

3. ANALYSIS OF BALANCE SCHEME

3.1. Conditioning of the reduced system

The conditioning of the reduced system is critical to the rapid convergence of the iterative method. Even though experimental evidence in Section 6 suggests that the reduced system M is favourably conditioned in comparison to the original system A, it is useful to obtain an estimate of the condition number of M. The following theorem provides such an estimate of $\kappa(M)$.

Theorem 1. *The reduced system M has a condition number which is at most equal to that of the original system A, i.e.,*

$$\kappa(M) \leqslant \kappa(A)$$

Proof

Consider the QR decomposition $E_i^{\mathrm{T}} = Z_i R_i$, where E_i is the ith block row of A (Z_i has been chosen to represent the orthogonal factor for consistency with the rest of the paper). The proof consists of two parts: first we show that

$$\kappa(M) \leqslant \kappa(R^{-\mathrm{T}} A)$$

where $R = \mathrm{diag}(R_1, R_2, \ldots, R_p)$; and then we prove that $\kappa(R^{-\mathrm{T}} A) \leqslant \kappa(A)$.

For simplicity, we prove the theorem for the case when $p=4$; generalization of the proof to arbitrary p is straightforward. Clearly,

$$\tilde{A}=R^{-\mathrm{T}}A=\begin{pmatrix} Z_{11}^{\mathrm{T}} & Z_{12}^{\mathrm{T}} & & & & & \\ & Z_{21}^{\mathrm{T}} & Z_{22}^{\mathrm{T}} & Z_{23}^{\mathrm{T}} & & & \\ & & & Z_{31}^{\mathrm{T}} & Z_{32}^{\mathrm{T}} & Z_{33}^{\mathrm{T}} & \\ & & & & & Z_{41}^{\mathrm{T}} & Z_{42}^{\mathrm{T}} \end{pmatrix}$$

and

$$\tilde{A}\tilde{A}^{\mathrm{T}}=\begin{pmatrix} I & Z_{12}^{\mathrm{T}}Z_{21} & & \\ Z_{21}^{\mathrm{T}}Z_{12} & I & Z_{23}^{\mathrm{T}}Z_{32} & \\ & Z_{32}^{\mathrm{T}}Z_{23} & I & Z_{33}^{\mathrm{T}}Z_{41} \\ & & Z_{41}^{\mathrm{T}}Z_{33} & I \end{pmatrix}$$

Substituting $H_1=Z_{12}^{\mathrm{T}}Z_{21}$, $H_2=Z_{23}^{\mathrm{T}}Z_{32}$, and $H_3=Z_{33}^{\mathrm{T}}Z_{41}$ in the above matrix, we get

$$\tilde{A}\tilde{A}^{\mathrm{T}}=\begin{pmatrix} I & H_1 & & \\ H_1^{\mathrm{T}} & I & H_2 & \\ & H_2^{\mathrm{T}} & I & H_3 \\ & & H_3^{\mathrm{T}} & I \end{pmatrix}$$

Symmetric odd-even permutation of the blocks yields

$$P\tilde{A}\tilde{A}^{\mathrm{T}}P^{\mathrm{T}}=\begin{pmatrix} I & H \\ H^{\mathrm{T}} & I \end{pmatrix}$$

in which

$$H=\begin{pmatrix} H_1 & \\ H_2^{\mathrm{T}} & H_3 \end{pmatrix}$$

It is easy to show that the eigenvalues of $\tilde{A}\tilde{A}^{\mathrm{T}}$ are given by

$$\lambda=1\pm\eta \tag{15}$$

where η^2 are the eigenvalues of $H^{\mathrm{T}}H$.

In order to relate this to the reduced system, we consider the matrix

$$\tilde{A}^{\mathrm{T}}\tilde{A}=\begin{pmatrix} Z_1Z_1^{\mathrm{T}} & & & \\ & Z_2Z_2^{\mathrm{T}} & & \\ & & Z_3Z_3^{\mathrm{T}} & \\ & & & Z_4Z_4^{\mathrm{T}} \end{pmatrix}=\begin{pmatrix} I-\bar{Z}_1\bar{Z}_1^{\mathrm{T}} & & & \\ & I-\bar{Z}_2\bar{Z}_2^{\mathrm{T}} & & \\ & & I-\bar{Z}_3\bar{Z}_3^{\mathrm{T}} & \\ & & & I-\bar{Z}_4\bar{Z}_4^{\mathrm{T}} \end{pmatrix}$$

where $\bar{Z}_i = \mathcal{N}(Z_i^{\mathrm{T}})$ such that $Z_i Z_i^{\mathrm{T}} = I - \bar{Z}_i \bar{Z}_i^{\mathrm{T}}$. The submatrix of block rows and columns of $\tilde{A}^{\mathrm{T}}\tilde{A}$ that corresponds to the overlapped diagonal blocks is given by

$$\begin{aligned} G &= 2I - \begin{pmatrix} \bar{Z}_{12}\bar{Z}_{12}^{\mathrm{T}} + \bar{Z}_{21}\bar{Z}_{21} & \bar{Z}_{21}\bar{Z}_{23}^{\mathrm{T}} & \\ \bar{Z}_{23}\bar{Z}_{21}^{\mathrm{T}} & \bar{Z}_{23}\bar{Z}_{23}^{\mathrm{T}} + \bar{Z}_{31}\bar{Z}_{31}^{\mathrm{T}} & \bar{Z}_{31}\bar{Z}_{33}^{\mathrm{T}} \\ & \bar{Z}_{33}\bar{Z}_{31}^{\mathrm{T}} & \bar{Z}_{33}\bar{Z}_{33}^{\mathrm{T}} + \bar{Z}_{41}\bar{Z}_{41}^{\mathrm{T}} \end{pmatrix} \\ &= 2I - MM^{\mathrm{T}} \end{aligned}$$

The reader should recall that $\bar{Z}_i$ corresponds to Q_i in Equations (8)–(10).

Since the eigenvalues of $\tilde{A}\tilde{A}^{\mathrm{T}}$ and $\tilde{A}^{\mathrm{T}}\tilde{A}$ are identical, and the eigenvalues of G are contained within the spectrum of $\tilde{A}^{\mathrm{T}}\tilde{A}$, it can be inferred from Equation (15) that

$$1 - \eta_1 \leqslant 2 - \lambda(MM^{\mathrm{T}}) \leqslant 1 + \eta_1$$

where η_1^2 is the largest eigenvalue of $H^{\mathrm{T}}H$. Therefore, the eigenvalues of MM^{T} lie within the following bounds:

$$1 - \eta_1 \leqslant \lambda(MM^{\mathrm{T}}) \leqslant 1 + \eta_1$$

and the condition number of MM^{T},

$$\kappa(MM^{\mathrm{T}}) \leqslant \kappa(\tilde{A}\tilde{A}^{\mathrm{T}}) \tag{16}$$

This completes the first part of the proof.

Next, we show that $\kappa(R^{-\mathrm{T}}A) \leqslant \kappa(A)$. Using the singular value decomposition $R = USV^{\mathrm{T}}$, and the relation $AA^{\mathrm{T}} = R\tilde{A}\tilde{A}^{\mathrm{T}}R^{\mathrm{T}}$, we have

$$AA^{\mathrm{T}} = USV^{\mathrm{T}}\tilde{A}\tilde{A}^{\mathrm{T}}VSU^{\mathrm{T}} = US\hat{A}\hat{A}^{\mathrm{T}}SU^{\mathrm{T}}$$

implying $\lambda(AA^{\mathrm{T}}) = \lambda(S\hat{A}\hat{A}^{\mathrm{T}}S)$ and $\lambda(\tilde{A}\tilde{A}^{\mathrm{T}}) = \lambda(\hat{A}\hat{A}^{\mathrm{T}})$. Thus,

$$\begin{aligned} \kappa(AA^{\mathrm{T}}) &= \max_{\mathbf{x},\mathbf{y}} \left[\frac{\mathbf{x}^{\mathrm{T}}S\hat{A}\hat{A}^{\mathrm{T}}S\mathbf{x}}{\mathbf{x}^{\mathrm{T}}\mathbf{x}} \cdot \frac{\mathbf{y}^{\mathrm{T}}\mathbf{y}}{\mathbf{y}^{\mathrm{T}}S\hat{A}\hat{A}^{\mathrm{T}}S\mathbf{y}} \right] \\ &= \max_{\mathbf{u},\mathbf{v}} \left[\frac{\mathbf{u}^{\mathrm{T}}\hat{A}\hat{A}^{\mathrm{T}}\mathbf{u}}{\mathbf{u}^{\mathrm{T}}S^{-2}\mathbf{u}} \cdot \frac{\mathbf{v}^{\mathrm{T}}S^{-2}\mathbf{v}}{\mathbf{v}^{\mathrm{T}}\hat{A}\hat{A}^{\mathrm{T}}\mathbf{v}} \right] \\ &\geqslant \kappa(\hat{A}\hat{A}^{\mathrm{T}}) \left[\frac{\mathbf{v}_n S^{-2}\mathbf{v}_n}{\mathbf{v}_1 S^{-2}\mathbf{v}_1} \right] \end{aligned}$$

where $\mathbf{v}_1$ and $\mathbf{v}_n$ are the eigenvectors for the largest and smallest eigenvalues of $\hat{A}\hat{A}^{\mathrm{T}}$, respectively. Note that $\hat{A}\hat{A}^{\mathrm{T}}$ has a block tridiagonal structure identical to that of $\tilde{A}\tilde{A}^{\mathrm{T}}$. Furthermore, a symmetric odd-even permutation of its blocks yields

$$P\hat{A}\hat{A}^{\mathrm{T}}P^{\mathrm{T}} = \begin{pmatrix} I & \hat{H} \\ \hat{H}^{\mathrm{T}} & I \end{pmatrix}$$

It can be verified that eigenvectors of this matrix take the form $(\hat{H}\mathbf{z}, \pm\eta\mathbf{z})^{\mathrm{T}}$ where $(\eta, \mathbf{z})$ is an eigenpair of $\hat{H}^{\mathrm{T}}\hat{H}$. Therefore,

$$\mathbf{v}_1 = \begin{pmatrix} \hat{H}\mathbf{z}_1 \\ \eta_1\mathbf{z}_1 \end{pmatrix}, \quad \mathbf{v}_n = \begin{pmatrix} \hat{H}\mathbf{z}_1 \\ -\eta_1\mathbf{z}_1 \end{pmatrix}$$

and

$$\frac{\mathbf{v}_n S^{-2}\mathbf{v}_n}{\mathbf{v}_1 S^{-2}\mathbf{v}_1} = 1$$

Therefore, $\kappa(AA^{\mathrm{T}}) \geqslant \kappa(\hat{A}\hat{A}^{\mathrm{T}})$, which further implies that

$$\kappa(\tilde{A}\tilde{A}^{\mathrm{T}}) \leqslant \kappa(AA^{\mathrm{T}}) \tag{17}$$

Combining Equations (16) and (17), we have

$$\kappa(MM^{\mathrm{T}}) \leqslant \kappa(\tilde{A}\tilde{A}^{\mathrm{T}}) \leqslant \kappa(AA^{\mathrm{T}})$$

which proves the theorem. ■

3.2. Storage requirements

The storage requirements of the balance scheme can be computed for a banded system of size $n \times n$ with a lower bandwidth b_{l} and upper bandwidth b_{u}, such that the total bandwidth $b = b_{\mathrm{l}} + b_{\mathrm{u}} + 1$. For simplicity, let us assume that the rows are divided into p blocks of equal size $s = n/p$. The size of the overlap between consecutive blocks is $b - 1 = b_{\mathrm{l}} + b_{\mathrm{u}}$.

The matrix E_i^{T}, $i = 2, \ldots, p-1$, is a lower triangular banded matrix of size $(s + b - 1) \times s$ with a lower bandwidth $b - 1$. The amount of memory needed to store Q_i is $s(b-1)$ and R_i is $sb - (b-1)^2/2$, respectively. The storage needed for E_1 and E_p is less than this amount. Therefore, the memory requirement for the QR factorization of E_i^{T}, $i = 1, \ldots, p$ is bounded by $sp(b-1) + spb \approx 2nb$.

Next, we estimate the storage needed for the reduced system M. M is a block matrix with $p-1$ block rows of size $b-1$ each, and p block columns of size $b-1$ each except for the first and last block columns that are of sizes b_{u} and b_{l}, respectively. Since M is a block bidiagonal matrix, the memory required for storing the nonzero elements is given by $2(p-2)(b-1)^2 + (b_{\mathrm{l}} + b_{\mathrm{u}})(b-1) = (2p-3)(b-1)^2$, which can be bounded by $2pb^2$. Thus, the total storage needed for balance scheme is $2b(n + pb)$. When Q_i are not computed explicitly, the storage requirements can be reduced drastically. The next section outlines an alternate approach that computes the reduced system implicitly without the QR factorization of the block rows.

4. A PROJECTION BASED APPROACH

In general, it is expensive to compute the QR decomposition of the submatrix E_i. In this section, we propose an extension of the balance scheme that does not require QR decompositions. In this approach, the matrix M is never computed explicitly; in fact, the reduced system is available only implicitly in the form of a matrix–vector product with MM^{T}. As a

result, an iterative method such as the conjugate gradients method (CG) can be used to solve the system

$$MM^{\mathrm{T}}\hat{\mathbf{y}} = \mathbf{g} \tag{18}$$

where $\mathbf{y} = M^{\mathrm{T}}\hat{\mathbf{y}}$, instead of the reduced system in Step 2 of Algorithm *Balance_Scheme*.

Consider the block column form of the matrix M in Equation (14)

$$M = (M_1, M_2, \ldots, M_p)$$

where

$$M_1 = \begin{pmatrix} Q_{1,2} \\ 0 \end{pmatrix} \tag{19}$$

$$M_i = \begin{pmatrix} 0 \\ -Q_{i,1} \\ Q_{i,3} \\ 0 \end{pmatrix}, \quad i = 2, \ldots, p-1 \tag{20}$$

$$M_p = \begin{pmatrix} 0 \\ -Q_{p,1} \end{pmatrix} \tag{21}$$

Therefore,

$$MM^{\mathrm{T}} = \sum_{i=1}^{p} M_i M_i^{\mathrm{T}} \tag{22}$$

In order to compute a matrix–vector product with MM^{T}, we need to show that the matrix $M_i M_i^{\mathrm{T}}$ is actually a section of the projector

$$P_i = Q_i Q_i^{\mathrm{T}} = I - E_i^{+} E_i \tag{23}$$

in which

$$Q_i = \begin{pmatrix} Q_{i,1} \\ Q_{i,2} \\ Q_{i,3} \end{pmatrix} \tag{24}$$

is an orthogonal basis for $\mathcal{N}(E_i)$. From Equations (19)–(21) and (24) it is clear that the non-zero blocks of $M_i M_i^{\mathrm{T}}$, denoted by $\tilde{M}_i \tilde{M}_i^{\mathrm{T}}$, may be obtained from Q_i in the following way

$$\begin{aligned} \tilde{M}_i \tilde{M}_i^{\mathrm{T}} &= \begin{pmatrix} -Q_{i,1} \\ Q_{i,3} \end{pmatrix} (-Q_{i,1}^{\mathrm{T}} \quad Q_{i,3}^{\mathrm{T}}) \\ &= \begin{pmatrix} -I & 0 & 0 \\ 0 & 0 & I \end{pmatrix} Q_i Q_i^{\mathrm{T}} \begin{pmatrix} -I & 0 \\ 0 & 0 \\ 0 & I \end{pmatrix} \end{aligned}$$

In combination with Equation (23), it can be seen that a matrix–vector product with MM^{T} can be computed as the sum of sections of the projectors P_i, $i = 1, \ldots, p$.

Due to the unavailability of M, $\mathbf{y}$ cannot be computed explicitly. Instead, $\mathbf{x}_i$ is updated using the projector P_i and $\hat{\mathbf{y}}$ directly. In particular, Equation (12) requires computation of $Q_{i,2}\mathbf{y}_i$ and $Q_{i,3}\mathbf{y}_i$ in order to update $\mathbf{x}_i$ and ξ, respectively. As shown below, this can also be accomplished using the projectors P_i:

$$\begin{pmatrix} Q_{i,2} \\ Q_{i,3} \end{pmatrix} \mathbf{y}_i = \begin{pmatrix} Q_{i,2} \\ Q_{i,3} \end{pmatrix} (-Q_{i,1}^{\mathrm{T}} \;\; Q_{i,3}^{\mathrm{T}})\hat{\mathbf{y}}$$
$$= \begin{pmatrix} 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} Q_i Q_i^{\mathrm{T}} \begin{pmatrix} -I & 0 \\ 0 & 0 \\ 0 & I \end{pmatrix} \hat{\mathbf{y}}$$

The projection based approach has the advantage of replacing the QR factorization of block rows E_i with projections onto the null spaces of E_i. This can lead to significant savings in storage and computation. The projections can be computed with an inner iterative method, giving rise to an inner–outer iterative scheme. Furthermore, the convergence of outer iterative solver can be improved with suitable preconditioners, especially since the matrix MM^{T} is not as well conditioned as the original reduced system. The experiments in Section 6 are based on this projection based extension of the balance scheme.

5. PARALLELISM

Parallelism in the balance scheme is available at two levels of the computation. At the coarse level, p blocks of the linear system give rise to independent problems that can be solved concurrently. Fine-grain parallelism within each problem allows use of multiple processors as well. This form of hierarchical parallelism removes the implicit limit of p on the number of processors. On hierarchical parallel architectures that support two-level parallelism, the balance scheme can be implemented in a scalable manner.

In Step 1 of the balance scheme, p independent underdetermined systems (8)–(10) are solved in parallel. Once block rows E_i have been allocated to individual processors, computation proceeds in parallel without any communication overhead. The particular solution $\mathbf{p}_i$ and the null space basis Q_i are computed via sparse QR factorization of E_i. Care must be taken to balance the load on each processor by appropriate choice of row blocks. The reduced system in Step 2 may be solved in a number of ways. When the size of M is relatively small, a fast direct algorithm is suitable. Parallel versions of direct solvers for banded systems can also be used. Alternately, an iterative method can be used in which matrix–vector products with the reduced system are computed in parallel. This is particularly useful for the projection-based extension of the balance scheme. Step 3 of the algorithm requires back substitution of $\mathbf{y}_i$ to determine $\mathbf{x}_i$ and ξ_i. The computation consists of matrix–vector products between Q_i and $\mathbf{y}_i$. This is also a perfectly parallel stage that has linear speedup.

The projection based extension discussed in the previous section consists of the following three steps. In Step 1, we solve p underdetermined linear least-squares problems corrresponding to the p blocks. These least-square problems may be solved either via a direct scheme such as orthogonal factorization, or iteratively using the CG algorithm on normal equations with incomplete Cholesky factorization as a preconditioner. In Step 2, we solve the reduced

system via an iterative scheme such as CG on Equation (18). Note that at each iteration of the CG method, we need to compute the product of MM^{T} with a vector. This consists of p projections of the form:

$$c_i = (I - P_i)b, \quad i = 1, \ldots, p$$

This projection–vector product is nothing but the residual of the least-squares problems

$$\min_{c_i} \|b - E_i^{\mathrm{T}} c_i\|_2$$

which can be obtained by using the CG scheme. In other words, solving the reduced system requires solving p independent least-squares problems in each CG iteration. Note that if we store the incomplete Cholesky factorizations as preconditioners we can achieve greater efficiency. In the third step, we again solve p independent least-squares problems to retrieve the solution of the system $Ax = f$.

The projection based approach uses the CG scheme to solve the reduced symmetric positive definite system. It is well known that the parallelism available at each iteration is limited only by the matrix–vector product and a constant number of vector dot products. In contrast, however, iterative methods for non-symmetric systems such as the generalized minimum residual method (GMRES) are less favourable to parallelism. This is due to linear growth in the number of dot products as the number of iterations increases, and often proves to be a bottleneck in the parallel computation.

6. EXPERIMENTS

In this section, we present results of experiments to solve banded linear systems that are sparse within the band. The main purpose of these experiments is to highlight the robustness of the balance scheme. It is also our intention to illustrate the favourable convergence properties of our approach. With this in view, we have chosen non-symmetric linear systems with strong indefinite symmetric components. Such systems prove to be extremely difficult for the commonly used Krylov subspace methods. Furthermore, attempts at preconditioning these systems with conventional incomplete factorizations frequently result in failures.

In each of these experiments, we used the *projection-based extension* of the balance scheme described in Section 4. The unpreconditioned CG method was used to solve the normal form of the reduced system (18). The projections needed for matrix–vector products at each iteration were computed via an inner iterative method. We used the preconditioned CG method for normal equations at the inner level, with a preconditioner based on incomplete Cholesky factors. The amount of fill allowed is indicated separately in each experiment. The outer iterations were terminated when the relative residual norm was reduced below a specified tolerance which has been indicated in the experiments as well. The inner iterations were terminated when the relative residual norm fell below a tenth of the outer iteration tolerance.

6.1. Partial differential equations

The first experiment was designed to highlight the advantages of balance scheme over preconditioned Krylov subspace based methods. We consider the following partial differential

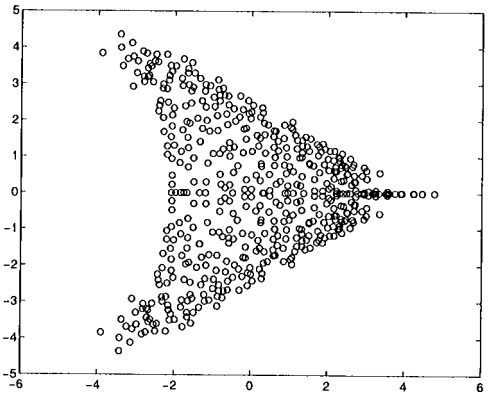


Figure 3. Eigenvalues of sample problem of size 516 with $\alpha = -4000$ and $v = 100$.

equation:

$$\Delta \mathbf{u} + v \cdot \nabla \mathbf{u} + \alpha \mathbf{u} = \mathbf{f}, \quad \text{in } \Omega \tag{25}$$

$$\mathbf{u} = \mathbf{0}, \quad \text{on } \Gamma \tag{26}$$

which is solved on a two-dimensional unit square domain. An unstructured mesh is used to discretize the domain, and the linear system is obtained by the finite-element approximation.

The parameters α and v can be selected to obtain linear systems with desired spectrum. In fact, the degree of difficulty inherent in these problems can be gauged by the spectrum of the linear system shown in Figure 3 for a problem with 516 unknowns in which $\alpha = -4000$ and $v = 100$.

Table I presents a comparison of the balance scheme with preconditioned GMRES. We used a restarted version of GMRES with a Krylov subspace of size 20. GMRES was preconditioned with ILUT, with a fill-in of 15 per row and a drop tolerance threshold of 10^{-4}. The matrix was reordered with reverse Cuthill–McKee algorithm to reduce the bandwidth. The balance scheme partitions the reordered matrix into four equal sized blocks. To provide a fair comparison, a fill-in of 20 was allowed for the incomplete Cholesky factors used by the inner iterative solver, thereby allowing equal amount of storage for both methods. The iterations for both the methods were terminated when the relative residual norm reduced below 10^{-4}. The right-hand side was generated from a known solution. As shown in the table, the balance scheme converges to acceptable tolerance in each of the cases, demonstrating the robustness of our approach. On the other hand, GMRES is able to solve only one instance of the problem which corresponds to a positive definite system.

Since the coarse-grained parallelism available in our approach depends on the number of block rows which can be processed concurrently, it is imperative that the balance scheme performs adequately on large problems with more blocks. Table II presents the results for a linear system of order 3101 for which we are able to use eight equal sized blocks. The rate of convergence of CG for the balance scheme is affected adversely by the increase in the number of blocks as well as the increase in the size of the problem. In spite of that,

Table I. Comparison of balance scheme with four equal sized blocks and preconditioned GMRES for the sample problem with 516 unknowns.

Order	α	v	Balance scheme		GMRES	
			Iter	Rnorm	Iter	Rnorm
516	0.0	0.0	25	0.95E-04	3	0.38E-07
516	−800.0	25.0	78	0.67E-04	—	—
516	−2000.0	50.0	47	0.18E-04	—	—
516	−4000.0	100.0	77	0.57E-05	—	—

Table II. Comparison of balance scheme with eight equal sized blocks and preconditioned GMRES for the sample problem with 3101 unknowns.

Order	α	v	Balance scheme		GMRES	
			Iter	Rnorm	Iter	Rnorm
3101	0.0	100.0	79	0.91E-04	3	0.49E-06
3101	−1500.0	50.0	180	0.22E-03	—	—
3101	−5000.0	100.0	158	0.60E-04	—	—
3101	−10000.0	200.0	129	0.28E-04	—	—

convergence is achieved within acceptable number of iterations, thus demonstrating robustness of the approach.

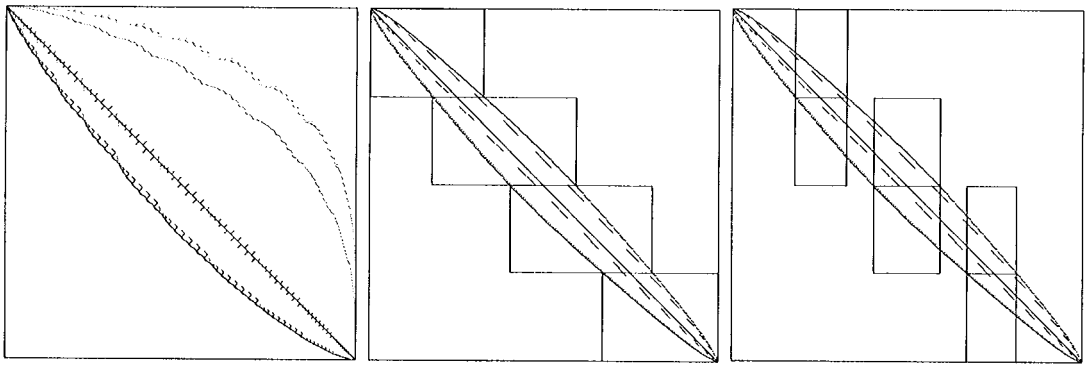
6.2. *Examples from Harwell–Boeing matrices*

In general, our technique is applicable to banded systems with a small bandwidth relative to the size of the system. Such systems often arise in two-dimensional problems from thin domains. In addition, the bandwidth of a matrix can sometimes be reduced by reordering the rows and columns. The balance scheme is applicable to these reordered matrices as well.

For the sake of illustration, we consider the matrices from the GRENOBLE set in the Harwell–Boeing collection. The matrix GRE1107, which arises from simulation studies in computer systems, serves as a typical example of systems that can be reordered to minimize bandwidth. The sparsity structure of GRE1107 is shown in Figure 4(a). Reordering with reverse Cuthill–McKee algorithm results in the structure shown in Figure 4(b). The rows are partitioned into four equal sized blocks. The common unknowns ξ which are used to define the reduced system are shown in Figure 4(c).

The spectrum of GRE1107 is shown in Figure 5. Table III presents a comparison of balance scheme with GMRES preconditioned with ILUT. A Krylov subspace of size 20 was used for GMRES, a fill-in of 15 per row was allowed for ILUT, and the drop tolerance was set to 10^{-4}. For the balance scheme, we allowed a fill-in of 20 for the incomplete Cholesky factors used by the inner iterative solver, thereby allowing equal amount of storage for both methods. Convergence was assumed when the relative residual norm reduced below 10^{-4}. The right-hand side was generated from a known solution.

Table III presents the iterations needed by the solvers. These results indicate quite clearly that the balance scheme is a robust algorithm. Furthermore, the iteration count also suggests



(a) (b) (c)

Figure 4. (a) Original sparsity structure of GRE1107, (b) Equal sized block rows of GRE1107 after reordering to minimize the bandwidth, and (c) Reduced system defined over the unknowns common to consecutive block rows.

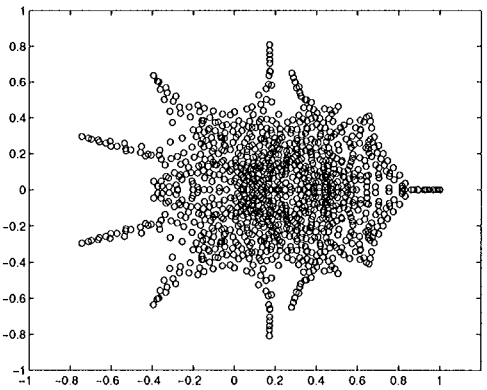


Figure 5. Eigenvalues of GRE1107.

that our approach is impervious to the indefiniteness of the system. In contrast, the strong indefinite symmetric component of the matrices was mainly responsible for the failure of the preconditioned GMRES method.

6.3. Comparison with ScaLAPACK

The last experiment demonstrates the parallel performance of the balance scheme. The linear systems were chosen to be banded Toeplitz matrices with nonzero elements restricted to the diagonals at a distance $-b_l, -1, 1$, and b_u from the main diagonal, where a negative value indicates subdiagonals. The values assigned to these diagonals were $-1, 1, 1$, and 1, respectively. The main diagonal was set to zero to assure indefiniteness of the system.

The spectrum of such a matrix of size 512 with lower and upper bandwidths of 16 each is shown in Figure 6. The eigenvalues are always confined to a square of side 4 units centred at the origin even when the size or bandwidth of the matrix is varied. For larger bandwidth, the

Table III. Convergence of balance scheme with four equal sized blocks for the GRENOBLE set of matrices in the Harwell–Boeing collection. Preconditioned GMRES failed to converge for any of these problems.

Problem	Order	Iter	Rnorm
GRE216A	216	57	0.14E-04
GRE216B	216	61	0.39E-06
GRE343	343	66	0.88E-05
GRE512	512	80	0.56E-05
GRE1107	1107	200	0.72E-04

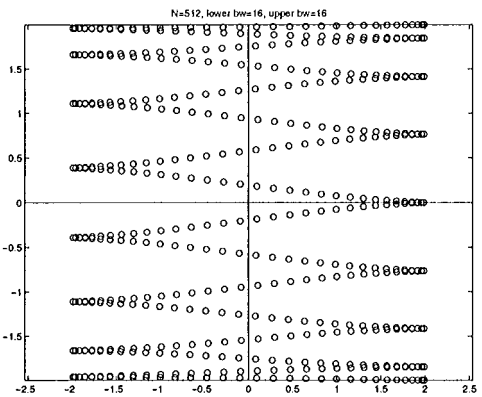


Figure 6. Eigenvalues of sparse, banded Toeplitz matrix of order 512 with lower and upper bandwidths of 16.

number of horizontal chains increases while the distance between them decreases, effectively increasing the condition number of the system. For larger systems with constant bandwidth, the density of the eigenvalues in the chains increases without altering the layout of the chains.

Table IV illustrates the parallel performance of the balance scheme with sixteen equal sized blocks on two instances of the above banded Toeplitz matrices. In each case, the bandwidth is restricted to be approximately 0.8 per cent of the size of the matrix. The last column gives the ratio of computation time for balance scheme on p processors and ScaLAPACK on 2 processors (in these experiments, the best performance of ScaLAPACK was observed on two processors).

The balance scheme required 38 iterations to converge to the solution of the system of order 16,384, and 49 iterations to solve the system of order 32,768. The iterations were stopped when the relative residual was reduced below 10^{-4} which corresponded to 3-digit accuracy in each element of the solution vector. The number of iterations would increase for a stricter tolerance. It should be emphasized, however, that the balance scheme has the ability to exploit sparsity within the band, making it a competitive alternative to dense banded solvers.

These experiments were performed on the SGI Origin 2000 at the National Computational Science Alliance at the University of Illinois. Only the projections at each iteration of the balance scheme were computed concurrently. These projections were computed via an inner

Table IV. Comparison of parallel balance scheme with ScaLAPACK on banded Toeplitz matrices that are sparse within the band.

Problem	p	Balance scheme	Speed improvement* over ScaLAPACK on 2 processors
$N = 16\,384$	2	3.23 s	0.90
$b_l = 64$	4	1.68 s	1.73
$b_u = 64$	8	0.94 s	3.09
	16	0.55 s	5.27
$N = 32\,768$	2	8.69 s	2.33
$b_l = 128$	4	4.67 s	4.33
$b_u = 128$	8	2.67 s	7.57
	16	1.74 s	11.62

*Speed improvement of the balance scheme was computed over the best parallel performance of ScaLAPACK.

preconditioned CG method. We allowed a fill-in of 4 for the incomplete Cholesky factors used as preconditioners at the inner level. As expected, the storage requirements of the balance scheme were much less than that of the ScaLAPACK banded solver.

The balance scheme was parallelized via compiler directives, whereas the ScaLAPACK banded solver routine used MPI [4]. Although ScaLAPACK can be very effective in solving dense matrices with extremely narrow bandwidth, the advantage is quickly lost when solving sparse matrices with larger bandwidth. Over here, a bandwidth fraction of $1/128$ was sufficient to illustrate this feature of ScaLAPACK.

7. CONCLUDING REMARKS

A new approach has been proposed for the solution of banded linear systems that are sparse within the band. The balance scheme is a competitive alternative to preconditioned Krylov subspace methods, especially for matrices with strongly indefinite symmetric part. It is well known that the convergence of Krylov subspace methods for such systems is quite unsatisfactory. In contrast, our approach leads to a robust algorithm with convergence properties that are relatively impervious to the indefiniteness of the linear system.

The proposed algorithm solves a reduced system using direct or iterative methods. Direct methods are fast but memory intensive since they require explicit computation as well as factorization of the reduced system. On the other hand, iterative methods overcome the problem of storage at the cost of possible increase in solution time. A projection-based extension to the balance scheme has also been proposed in which the reduced system is never computed explicitly. Instead, matrix–vector products with the normal form of the reduced system are computed using projectors onto the null spaces of blocks of rows of the matrix.

The balance scheme is a scalable parallel algorithm on hierarchical parallel architectures. The main computation consists of projections using block rows of the matrix which can be efficiently parallelized. This feature extends easily to the projection based variant proposed for problems where the reduced system is available only implicitly. Comparison with parallel dense banded solvers have also shown the favorable characteristics of the proposed scheme.

APPENDIX

In this section, we present a simple MATLAB program for the balance scheme. Although the code is computationally inefficient, we believe that it can provide a starting point for developing efficient code. We have not included the code for the projection-based extension of the balance scheme since it is substantially more complicated.

```
function x = balance_scheme (A,f)
n = size (A,1);                        % Order of system
[I,J] = find(A);                       % Compute upper, lower,
bl = max(I-J); bu=max(J-I);            % total bandwidths
bw=bl+bu+1;
ptr=1+ceil(linspace(0,n,p+1))';        % Block row pointer
k=(bw-1) * (p-1); M=zeros(k);          % Initialize reduced
g=zeros (k,1);                         % system: M,g
for i=1:p,
  r=ptr(i):ptr(i+1)-1;                 % Block row, column indices
  c=max(ptr(i)-bl,1):min(ptr(i+1)-1+bu,n);
  Ei=A(r,c); fi=f(r);
  [mi,ni]=size(Ei);
  [Q,R]=qr(Ei');                       % QR factorization
  Q1=Q(:,1:mi); Q2=Q(:,mi+1:ni); R=R(1:mi,1:mi);
  pi=Q1 * (R'\fi);
  rr=(bw-1) * (i-1)+1:(bw-1) * i;      % Block row index
  rr2=rr-(bw-1);                       % Previous block row index
  if (i==1),                           % Compute M-blocks
    cc=1:bu;                           % and g-blocks
    M(rr,cc)=Q2(ni-bw+2:ni,:);
    g(rr)=g(rr)-pi(ni-bw+2:ni);
  elseif (i<p),
    cc=bu+(bw-1) * (i-2)+1:bu+(bw-1) * (i-1);
    M(rr,cc)=Q2(ni-bw+2:ni,:);
    M(rr2,cc)=-Q2(1:bw-1,:);
    g(rr)=g(rr)-pi(ni-bw+2:ni);
    g(rr2)=g(rr2)+pi(1:bw-1);
  else
    cc=k-bl+1:k;
    M(rr2,cc)=-Q2(1:bw-1,:);
    g(rr2)=g(rr2)+pi(1:bw-1);
  end;
end;
```

```
Y=M\g;                    % Solved reduced system
x=zeros(n,1);             % Initialize solution x
for i=1:p,                % Compute x-blocks
   r=ptr(i):ptr(i+1)-1;   % Block row, col indices
   c=max(ptr(i)-bl,1):min(ptr(i+1)-1+bu,n);
```



```
  Ei=A(r,c); fi=f(r);   % Block row, block rhs
  [mi,ni]=size(Ei);
  [Q,R]=qr(Ei');        % QR factorization
  Q1=Q(:,1:mi); Q2=Q(:,mi+1:ni); R=R(1:mi,1:mi);
  pi=Q1 * (R'\fi);      % Particular solution
  if (i==1), cc=1:bu;   % y-index
  elseif (i<p),
   cc=bu+(bw-1) * (i-2)+1:bu+(bw- 1) * (i-1);
  else cc=k-bl+1:k; end;
  x(c)=pi+Q2 * y(cc);   % Update x-block
end;
return;
```

ACKNOWLEDGEMENTS

This research has been supported in part by the National Science Foundation under the research grants CCR 9972533 and ECS 9527123 and the infrastructure grant NSF EIA 9871053.

REFERENCES

1. Axelsson O. *Iterative Solution Methods.* Cambridge University Press: Cambridge, 1994.
2. Saad Y, *Iterative Methods for Sparse Linear Systems.* PWS Publishing Co: Boston, 1996.
3. Arbenz P, Gander W. A survey of direct parallel algorithms for banded linear systems. *Technical Report* 221, Department Informatik, ETH Zürich, 1994.
4. Bramley R, Sameh A. Row projection methods for large nonsymmetric linear systems. *SIAM Journal on Science Statistics and Computers* 1992; **13**:168–193.
5. Dongarra J, Sameh A. On some parallel banded system solvers. *Parallel Computing* 1984; **1**:223–235.
6. Sameh A, Kuck D. On Stable parallel linear system solvers. *Journal of Acoustics and Computational Methods* 1978; **25**:81–91.
7. Lou F, Sameh A. A parallel splitting method for solving linear systems. *Technical Report* 92-125, AHPCRC, University of Minnesota, Minneapolis, 1992.
8. Björck A. *Numerical Methods for Least Squares Problems.* SIAM Publications: Philadelphia, 1996.
9. Golub G, Loan CV. *Matrix Computations.* Johns Hopkins University Press: Baltimore, 1996.
10. Robey T, Sulsky D. Row ordering for a sparse QR decomposition. *SIAM Journal on Matrix Analysis and Applications* 1994; **15**:1208–1225.
11. Blackford LS, Choi J, Cleary A, D'Azevedo E, Demmel J, Dhillon I, Dongarra J, Hammarling S, Henry G, Petitet A, Stanley K, Walker D, Whaley RC. *ScaLAPACK Users Guide.* SIAM: Philadelphia, 1997.

